SCHEME: K

Name : ______________________________

Roll No. : ______________ Year : 20___ 20___

Exam Seat No. : ________________________

LABORATORY MANUAL FOR

DATABASE MANAGEMENT SYSTEM

(313302)

COMPUTER ENGINEERING GROUP

MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION, MUMBAI (Autonomous) (ISO 9001: 2015) (ISO/IEC 27001:2013)

VISION

To ensure that the Diploma level Technical Education constantly matches the latest requirements of technology industry and includes the all-round personal development of students including social concerns and to become globally competitive, technology led organization.

MISSION

To provide high quality technical and managerial manpower, information and consultancy services to the industry and community to enable the industry and community to face the changing technological & environmental challenges.

QUALITY POLICY

We, at MSBTE, are committed to offer the best-in-class academic services to the students and institutes to enhance the delight of industry and society. This will be achieved through continual improvement in management practices adopted in the process of curriculum design, development, Implementation, evaluation, and monitoring system along with adequate faculty development programmes.

CORE VALUES

MSBTE believes in the followings:

- Education industry produces live products,
- Market requirements do not wait for curriculum changes.
- Question paper is the reflector of academic standards of educational organization.
- Well-designed curriculum needs effective implementation too.
- Competency based curriculum is the backbone of need-based program.
- Technical skills do need support of life skills,
- Best teachers are the national assets.
- Effective teaching learning process is impossible without learning resources.

A Laboratory Manual

for

Database Management System (313302)

Semester-III

(AI/ AN/ BD/ CM/ CO/ CW/ DS/HA/ IF/ IH/ TE)

Maharashtra State
Board of Technical Education, Mumbai
(Autonomous) (ISO-9001-2008)
(ISO/IEC 27001:2013)

MAHARASHTRA STATE

BOARD OF TECHNICAL EDUCATION

Certificate

This is to certify that Mr. / Ms. ……………………………………………………………………….

Roll No………………………. of Third Semester of Diploma in…………………………..

………………………………………………….………of Institute ……………………………

……………………………………………………………………..(Code …………….)

has completed the term work satisfactorily in course **Database Management System (313302)** for the academic year 20…….to 20…..... as prescribed in the curriculum.

Place: ……………….   Enrollment No.: ……………………

Date: …......................   Exam Seat No.: …...……………….....

Subject Teacher   **Head of the Department**   **Principal**

Seal of the
Institute

Preface

The primary focus of any engineering laboratory/field work in the technical education system is to develop the much-needed industry relevant competencies and skills. With this in view, MSBTE embarked on this innovative 'K' Scheme curricula for engineering Diploma programmes with outcome-based education as the focus and accordingly, relatively large amountof time is allotted for the practical work. This displays the great importance of laboratory work making each teacher, instructor and student realize that every minute of the laboratory time needs to be effectively utilized to develop these outcomes, rather than doing other mundane activities. Therefore, for the successful implementation of this outcome-based curriculum, every practical has been designed to serve as a *'vehicle'* to develop this industry identified competency in every student. The practical skills are difficult to develop through 'chalk and duster' activity in the classroom situation. Accordingly, the 'K' scheme laboratory manual development team designed the practicals to *focus* on *outcomes*, rather than the traditional age-old practice of conducting practical's to 'verify the theory' (which may become a byproduct along the way).

This laboratory manual is designed to help all stakeholders, especially the students, teachers and instructors to develop in the student the pre-determined outcomes. It is expected from each student that at least a day in advance, they must thoroughly read the concerned practical procedure that they will do the next day and understand minimum theoretical backgroundassociated with the practical. Every practical in this manual begins by identifying the competency, industry relevant skills, course outcomes and practical outcomes which serve as a key focal point for doing the practical. Students will then become aware of the skills they willachieve through procedure shown there and necessary precautions to be taken, which will help them to apply in solving real-world problems in their professional life.

This manual also provides guidelines to teachers and instructors to effectively facilitate student-centered lab activities through each practical exercise by arranging and managing necessary resources in order that the students follow the procedures and precautions systematically ensuring the achievement of outcomes in the students.

A database management system is needed to organize, store, and manage large amounts of data efficiently. It ensures data is easily accessible, secure, and can be quickly updated or retrieved when needed. This helps businesses and organizations make better decisions and operate smoothly.

The lab manual development team wishes to thank MSBTE who took initiative in the development of curriculum re-design project and implementation and acknowledge the contribution of individual course experts who have been involved in laboratory manual as well ascurriculum development (K scheme) directly or indirectly.

Although all care has been taken to check for mistakes in this laboratory manual, it is impossible to claim perfection, especially as this is the first edition. Any such errors and suggestions for improvement can be brought to our notice and are highly welcome.

Programme Outcomes (POs) to be achieved through Practicals of this Course

The following programme outcomes are expected to be achieved significantly out of the ten programme outcomes and Computer Engineering and Information Technology programme specific outcomes through the practicals of the course on **Database Management System.**

PO 1. Basic and Discipline specific knowledge: Apply knowledge of basic mathematics,science and engineering fundamentals and engineering specialization to solve the engineering problems.

PO 2. Problem analysis: Identify and analyse well-defined engineering problems using codifiedstandard methods.

PO 3. Design/ development of solutions: Design solutions for well-defined technical problems and assist with the design of systems components or processes to meet specified needs.

PO 4. Engineering Tools, Experimentation and Testing: Apply modern engineering tools andappropriate technique to conduct standard tests and measurements.

PO 5. Engineering practices for society, sustainability and environment: Apply appropriate technology in context of society, sustainability, environment and ethical practices.

PO 6. Project Management: Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.

PO 7. Life-long learning: Ability to analyse individual needs and engage in updating in the context of technological changes.

Program Specific Outcomes (PSOs):

PSO1. Modern Information Technology: Use latest technology for operation and application of information.

PSO2. Information Technology Process: Maintain the information process using modern information and communication technologies

Practical - Course Outcome Matrix

Course Outcomes (COs)

CO1 - Explain concept of database management system.
CO2 - Design the database for given problem.
CO3 - Manage database using SQL.
CO4 - Implement PL/SQL codes for given application.
CO5 - Apply security and backup methods on database

Sr. No.	**Title of the Practical**	**CO1**	**CO2**	**CO3**	**CO4**	**CO5**
1	* Install the provided database software	√				
2	* Create Database schema for given application	√				
3	* Execute DDL commands to manage Database using SQL			√		
4	* Execute DML Commands to manipulate data using SQL.			√		
5	* Execute DCL commands to control the access to data using SQL			√		
6	* Execute TCL Commands to control transactions on data using SQL			√		
7	Write Queries using Arithmetic operators.			√		
8	Apply built-in Logical operators on given data			√		

9	Implement Relational operators to apply various conditions in query.			√		
10	* Use Set operators to perform different operations			√		
11	Execute queries using string functions			√		
12	Execute queries using Arithmetic functions			√		
13	Implement queries using Date and Time functions			√		
14	Implement queries using Aggregate functions			√		
15	* Execute queries for Ordering and Grouping data.			√		
16	* Implement SQL queries for Inner and Outer Join			√		
17	* Create and manage Views for faster access on relations.			√		
18	* Implement PL/SQL program using Conditional Statements				√	
19	* Implement PL/SQL program using Iterative Statements				√	
20	Implement PL/SQL program using Sequential Control				√	
21	* Create Implicit and Explicit Cursors.				√	
22	* Implement PL/SQL program based on Exception Handling (Pre-defined exceptions) application				√	
23	* Implement PL/SQL program based on Exception Handling (User-defined exceptions)				√	
24	* Create Procedures and stored procedures for modularity				√	
25	* Create functions for given database				√	
26	* Implement triggers for given database				√	

List of Industry Relevant Skills

The following industry relevant skills of the competency "To design database and use any RDBMS package as a backend for developing database applications" are expected to be developed in you by performing practicals of this laboratory manual.

1. Explain the concept of a database management system.
2. Design and manage databases using SQL and PL/SQL.
3. Apply security and backup methods on databases.

Brief Guidelines to Teachers

Hints regarding strategies to be used

1. The teacher shall explain prior concepts to the students before starting each experiment.
2. For practical's requiring tools to be used, teacher should provide the demonstration of the practical emphasizing the skills, which the student should achieve.
3. Involve students in the activities during the conduct of each experiment.
4. Teachers should give opportunity to students for hands-on after the demonstration.
5. Assess the skill achievement of the students and COs of each unit.
6. The teacher is expected to share the skills and competencies to be developed in the students.
7. Teacher should ensure that the respective skills and competencies are developed in the students after the completion of the practical exercise.
8. Teacher may provide additional knowledge and skills to the students even though that may not be covered in the manual but are expected from the students by the industries.
9. Teacher may suggest the students to refer additional related literature of the reference books/websites/seminar proceedings etc.
10. During assessment, the teacher is expected to ask questions to the students to tap their knowledge and skill related to that practical.

Instructions for Students

Students shall read the points given below to understand the theoretical concepts andpractical applications.

1. Students shall listen carefully to the lecture given by the teacher about importance of the subject,learning structure, course outcomes.
2. Students shall organize the work in a group of two or three members and make arecord of all observations.
3. Students shall understand the purpose of the experiment and its practical implementation.
4. Students shall write the answers to the questions during practical.
5. Students should feel free to discuss any difficulty faced during the conduct of practical.
6. Students shall develop knowledge of Database Management System fundamentals and manipulation skills as expected by the industries.
7. Students shall attempt to develop related hands-on skills and gain confidence.
8. Students shall refer technical magazines; websites related to the scope of the subjects and update their knowledge and skills.
9. Students shall develop self-learning techniques.
10. Students should develop the habit of submitting the write-ups on the scheduled dates and time.

Content Page

List of Practical and Progressive Assessment Sheet

S. No.	Title of the practical	Page No.	Date of performance	Date of Submission	Assessment Marks (50)	Dated Sign of Teacher	Remarks (if any)
12	Execute queries using Arithmetic functions	54					
13	Implement queries using Date and Time functions	58					
14	Implement queries using Aggregate functions	62					
15	* Execute queries for Ordering and Grouping data.	66					
16	* Implement SQL queries for Inner and Outer Join	71					
17	* Create and manage Views for faster access on relations.	82					
18	* Implement PL/SQL program using Conditional Statements	87					
19	* Implement PL/SQL program using Iterative Statements	92					
20	Implement PL/SQL program using Sequential Control	100					
21	* Create Implicit and Explicit Cursors.	106					
22	* Implement PL/SQL program based on Exception Handling (Pre-defined exceptions) application	111					
23	* Implement PL/SQL program based on Exception Handling (User-defined exceptions)	115					
24	* Create Procedures and stored procedures for modularity	119					
25	* Create functions for given database	123					
26	* Implement triggers for given database	128					
				Total Marks			

To be transferred to Proforma of CIAAN-2023.

Experiment No. 1: Install the provided Database Software

I. Practical Significance

A database management system is used to store, retrieve data for an organization or for an individual. DBMS is used in the fields like Banking, Ticket reservation system, Finance company, Retail industries, Education etc. Students should be able to understand the basic functioning of DBMS, install the DBMS software and should be able to identify the Entities and relationship among them. The proper ER diagram should be transformed into proper tables using Normalization concepts to eliminate redundant data, minimize data modification errors and simplify the query process. This practical is useful for students to understand the Normalization and Create ER diagram for any given application.

II. Industry/Employer Expected Program Outcomes (POs)

The aim of this course is to design databases and use any RDBMS package as a backend for developing database applications

III. Course level learning outcomes

This practical is expected to develop the following skills in you:

1. Understand what is Database management system software.
2. Install the relevant database system software.

IV. Laboratory Learning Outcome(s)

Install the Database management system software.

V. Relevant Affective domain related Outcome(s)

1. Follow ethical practices.
2. Install appropriate DBMS software.

VI. Relevant Theoretical Background

Oracle database express Edition (XE) is a free, small foot-print edition of oracle database**.** It includes all advanced database features like multitenant, In-memory, partitioning, advanced analytics and advanced security.

Step1:

Open any browser and navigate to oracle database 21c XE download.

Step 2:

Download Oracle database 21c express edition for windows depending on your windows configuration. Choose the location as per the requirements and download the compressed zip file.

Step 3:

Follow the Installation procedure and provide the proper Username and password.

Step 4:

Open the SQL PLUS and provide the proper credentials to open it. The Oracle express edition is now ready to use.

VII. Required Resources with specifications:

Sr. No	Instrument/ Object	Specification
1	Operating system	Windows 10 or higher
2	Disk space	8.5 gigabytes min for oracle software, plus 2 gigabytes or more for temporary storage
3	RAM	2 Gigabytes RAM minimum

VIII. Precautions to be followed

1. Use of appropriate syntax
2. Select required relations and use relevant conditions.
3. Provide proper Username and password.

IX. References/ Suggestions for further Reading

1. https://blogs.oracle.com/sql/post/how-to-use-create-table-alter-table-and-drop-table-in-oracle-database

2. https://support.microsoft.com/en-us/office/database-design-basics-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5

X. Assessment Scheme

Performance indicators		Weightage
Sr. No	**Process related (10 Marks)**	**30%**
1.	Tool Selection Ability	20%
2.	Follow Ethical Practices	10%
	Product related (15 Marks)	**70%**
3.	Verifying System Requirement for Installation	20%
4.	Correctness in Use of Command Tools	15%
5.	Use of Attribute/s Options	15%
6.	Timely Submission	10%
7.	Answering Sample Questions	10%
	Total (50 Marks)	**100%**

List of Students /Team Members

1..

2.……………………………………..

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (35)**	**Total (50)**	

Experiment No 02: Create Database schema for given application

I. Practical Significance: A database is an organized collection of data. Databases are used to store, manage, and access all kinds of data. Database keeps information about people, places, or things, making it easy to look at and analyze. A Table in a database consists of columns and rows. Giving a table a primary key means choosing one special piece of information, like an enrolment number, to make sure every record on the table is unique and easy to find. This practical allows students to create database schema for a given application.

II. INDUSTRY / EMPLOYER EXPECTED OUTCOME:

To design database and use any RDBMS package as a backend for developing database applications.

III. COURSE LEVEL LEARNING OUTCOMES (COS):CO1 - Explain concept of database management system.

IV. LABORATORY LEARNING OUTCOME:

Create Database schema for given application.

V. Relevant Affective Domain related outcome(s)

1. Follow precautionary measures.
2. Follow installation steps.
3. Follow ethical practices.

VI. Relevant Theoretical Background

1. Create Database for Given Application:

- What is a Database?

A database is a collection of organized data that can be easily accessed, managed, and updated. Databases are created to store and manage data for an application, like keeping track of us information for a website.

Syntax:

```
CREATE DATABASE database_name;
```

Example:

```
create database employee;
```

2.Create Tables for the Given Application:

- What is a Table?

A table is a structure within a database that organizes data into rows and columns. It is used to store specific types of data, like user details or product information, in a structured way.

Syntax:

A basic command is

```
CREATE TABLE table_name (
  column1 datatype,
  column2 datatype,
  ...
);
```

Example :

```
create table emp (
  empno  number(10),
  ename varchar2(50),
  sal number(6,2)
);
```

3. Assign Primary Key for Created Table:

- What is a Primary Key?

A primary key is a unique identifier for each record in a table. It ensures that each record can be uniquely identified and improves search speed.

Syntax:

```
CREATE TABLE table_name (
 column1 datatype PRIMARY KEY,
  column2 datatype,
  ...
);
```

Example :

```
create table emp (
  empno  number(10) primary key,
  ename varchar2(50),
  sal number(6,2)
);
```

4. Modify the Table as per the Application Needs:

- Why Modify a Table?

To add, remove, or change columns based on new requirements or data types.

Syntax:

```
ALTER TABLE table_name ADD column_name datatype;
or removing a column:
ALTER TABLE table_name DROP COLUMN column_name;
```

Example:

```
alter table emp add mgr number (20);
or removing a column:
ALTER TABLE emp DROP COLUMN mgr;
```

VII. Required Resources/apparatus/equipment with specifications

Sr. No	**Equipment Name with Broad Specifications**	**e Relevant LLO Number**
1	Computer system with all necessary components like; motherboard, random access memory (RAM), read-only memory (ROM), internal hard disk drives, Mouse, Keyboard, and RDBMS applications such as Oracle Express Edition,MySql,SQLite,Oracle Live SQL etc.	All

VIII. Procedure

1. Create Database for given application
2. Create tables for the given application
3. Assign Primary key for created table
4. Modify the table as per the application needs.

IX. Result(s)

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

X. Practical related questions (Provide space for answers)

Note: Below are a few sample questions for reference. Teacher must design more suchquestions so as to ensure the achievement of identified CO.

1. Define Database and Database Management System
2. Describe primary key.
3. Draw E-R diagram for Library Management System.
4. Normalize the following table of EMP to 3NF

   EMP(empno,ename,mgr,job,deptno,loc,dname)

(Space for answer)

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

XI. Exercise

1. How would you create a new database named EMP?
2. Write the SQL commands to create the EMP table with the following structure:
    - empno as a number datatype with up to 4 digits
    - ename as a variable character datatype up to 10 characters
    - job as a variable character up to 9 characters
    - mgr as a number datatype with up to 4 digits
    - hiredate as a date
    - sal as a number with up to 7 digits, including 2 decimal places
    - comm as a number with up to 7 digits, including 2 decimal places
    - deptno as a number with up to 2 digits
3. How would you alter the EMP table to assign the empno column as the primary key?
4. Write the SQL commands to create the DEPT table with the following structure:
    - deptno as a number with up to 2 digits
    - dname as a variable character datatype up to 10 characters
    - loc as a variable character up to 20 characters

XII. References/Suggestions for further reading: include websites/links

1. http://vlabs.iitkgp.ernet.in/se/4/theory/
2. https://www.youtube.com/watch?v=yGU4YfSSjdM
3. https://blogs.oracle.com/sql/post/how-to-use-create-table-alter-table-and-drop-table-in-oracle-database

4. https://support.microsoft.com/en-us/office/database-design-basics-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5

XIII. Assessment Scheme

Performance indicators		**Weightage**
Sr. No	**Process related (10 Marks)**	**30%**
1.	Tool Selection Ability	20%
2.	Follow Ethical Practices	10%
	Product related (15 Marks)	**70%**
3.	Verifying System Requirement for Installation	20%
4.	Correctness in Use of Command Tools	15%
5.	Use of Attribute/s Options	15%
6.	Timely Submission	10%
7.	Answering Sample Questions	10%
	Total (50 Marks)	**100%**

List of student Team Members

1……………………………………

2…………………………………....

3……………………………………

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (35)**	**Total (50)**	

Experiment No. 3: Execute DDL commands to manage Database using SQL

I. Practical Significance

DDL commands are used to define database structure or Schema. DDL deals with description of database schema. It is used to create and modify the structure of database objects. DDL provides a set of definition to specify storage structure and access methods used by the database system. Design database by applying the constraints and modify the structure of table. This practical will help students to create and modify the database.

II. Industry/Employer Expected Program Outcomes (POs)

The aim of this practical is to design databases and modify the database structure as per the database applications.

III. Course level learning outcomes

This practical is expected to develop the following skills in you:

1. Understand how to design the Database system based on the requirements
2. Modify the structure of the database and apply the constraints.

IV. Laboratory Learning Outcome(s)

1. Write and execute SQL queries for creating database.
2. Write queries to Modifying the database.
3. Write queries to apply suitable constraints.

V. Relevant Affective domain related Outcome(s)

1. Follow ethical practices.
2. Use appropriate DBMS software.
3. Demonstrate analytical and logical knowledge as an Individual.
4. Participate in team problem solving activities.
5. Prioritizes time effectively to meet the needs of the team and self.

VI. Minimum Theoretical Background

Basic SQL Datatypes:

1. **CHAR**
2. **VARCHAR/ VARCHAR2(SIZE)**
3. **NUMBERS (P, S)**
4. **DATE**

5. **LONG**
6. **RAW/LONG RAW**
7. **MISCELLANUOUS (clob, blob, xml, Json)**

DDL Commands:

It is a set of SQL commands used to create, modify and delete database structure but not the data. These commands normally used by Database administrator (DBA).

1. CREATE
2. ALTER
3. DROP
4. RENAME
5. TRUNCATE
6. DESCRIBE

Create Tables for the Given Application:

Syntax:

CREATE TABLE table _name (column1 name datatype(size), column2 name datatype(size)…...);

Modify the Table as per the Application:

Syntax:

ALTER TABLE table_name ADD column_name datatype;

or removing a column:

ALTER TABLE table_name DROP COLUMN column_name;

Procedure:

1. Create Database for given application
2. Create tables for the given application
3. Assign Primary key for created table
4. Modify the table as per the application needs.

VII. Additional Software required

…………………………………………………………………………………………………
…………………………………………………………………………………………………
……………………

VIII. Precautions

1) Use of appropriate syntax
2) Select required relations and use relevant conditions.

IX. Additional Resources used

…………………………………………………………………………………………
…………………………………………………………………………………………
…………………………………………………………………………………………
…………………………………………………………………………………………
……………………………………………………………………

X. Result (Output of the procedure)

…………………………………………………………………………………………
…………………………………………………………………………………………
…………………………………………………………………………………………
…………………………………………………………………………………………
……………………………………………………………………

XI. Practical Related Questions

Note: Below are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

(**Note:** Use Point VII and XIII to XV for all relevant practical exercise use blank pages provided or attach more pages if needed.)

1. Create a table EMPLOYEE with following schema:

   Emp (EMP_no as primary key, E_name, Dept_no, Dept_name, Job_id, salary)

2. Create tables EMPLOYEE and DEPARTMENT with following schema by applying Primary and Foreign key:

   Emp(empno as primary key, empname, salary, phoneno)

   Dept(deptno primary key, empno foreign key, deptname, location).

Theory related Questions

1. List DDL commands with its syntax
2. List different SQL Binary datatypes
3. Write difference between Drop and Truncate command.
4. Write the use of Describe command.

(Space for answers)

…………………………………………………………………………………………
…………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

XII. Exercise

Attempt following and teacher shall design and allot more questions to attain desired outcome: (Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Create table for stud using attributes Rollno, Studname, Percentage. Apply primary key for rollno and check constraint on percentage that the percentage should not be greater than 100.
2. Change the stud table structure by adding column City.
3. Increase the size by 10 of studentname column.
4. Write the output of the following:

Create table Passenger_details(passenger_name varchar2(30), train_details varchar2(30),travelling_date date, birthdate date);	Output:

Alter table stud add phone_no number;	Output:

(Space for answers)

……………………………………………………………………………………
……………………………………………………………………………………
……………………………………………………………………………………
……………………………………………………………………………………
……………………………………………………………………………………
……………………………………………………………………………………
……………………………………………………………………………………
……………………………………………………………………………………
……………………………………………………………………………………
……………………………………………………………………………………
……………………………………………………………………………………
……………………………………………………………………………………
……………………………………………………………………………………
……………………………………………………………………………………
……………………………………………………………………………………
……………………………………………………………………………………
……………………………………………………………………………………
……………………………………………………………………………………
……………………………………………………………………………………
……………………………………………………………………………………
……………………………………………………………………………………
……………………………………………………………………………………
…………………………………………………………………

XIII. Precautions to be followed

1. Use of appropriate syntax
2. Select required relations and use relevant conditions.

XIV. References/ Suggestions for further Reading

1. https://blogs.oracle.com/sql/post/how-to-use-create-table-alter-table-and-drop-table-in-oracle-database
2. https://support.microsoft.com/en-us/office/database-design-basics-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5
3. https://www.oracle.com/
4. www.w3school.com
5. https://www.youtube.com/watch?v=vUj-kUEC_oA

XV. Assessment Scheme

Performance indicators		**Weightage**
Sr. No	**Process related (10 Marks)**	**30%**
1.	Tool Selection Ability	20%
2.	Follow Ethical Practices	10%
Product related (15 Marks)		**70%**
3.	Verifying System Requirement for Installation	20%
4.	Correctness in Use of Command Tools	15%
5.	Use of Attribute/s Options	15%
6.	Timely Submission	10%
7.	Answering Sample Questions	10%
	Total (50 Marks)	**100%**

List of Students /Team Members

1. ....

2.…………………………………………

3.…………………………………………

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (35)**	**Total (50)**	

Experiment No 04: Execute DML Commands to manipulate data using SQL

I. **Practical Significance**: A data manipulation language (DML) is used for adding (inserting), deleting, and modifying (updating) data in a database. This practical allows students to deal with modifying data in the database schema for a given application.

II. **INDUSTRY / EMPLOYER EXPECTED OUTCOME:**

To execute DML Commands to manipulate data using SQL on the given database using any RDBMS package.

III. **COURSE LEVEL LEARNING OUTCOMES (COS)**:CO3
- Manage database using SQL.

IV. **LABORATORY LEARNING OUTCOME:**

Execute DML Commands to manipulate data using SQL.

V. **Relevant Affective Domain related outcome(s)**

a. Follow precautionary measures.

b. Follow installation steps.

c. Follow ethical practices.

VI. **Relevant Theoretical Background**

Data Manipulation Language (DML) Commands are used to manage and manipulate data in a database. These commands include INSERT, UPDATE, DELETE, and SELECT, which help you add, modify, remove, and retrieve data from tables, respectively.

1. INSERT

Explanation: The INSERT command is used to add new rows of data to a table.

Syntax:

1)INSERT INTO table_name (column1, column2, column3, ...)VALUES (value1, value2, value3, ...);

2) INSERT INTO table_name VALUES (value1, value2, value3, ...);

3) INSERT INTO table_name (column2, column3, ...)VALUES (value2, value3 , ...);

Example:

1) INSERT INTO emp (empno, name, age) VALUES (1, 'Aditya Shinde', 20);
2) INSERT INTO emp VALUES (1, 'Aditya Shinde', 20);
3) INSERT INTO emp (empno , Age) VALUES (1, 20);

2. UPDATE

Explanation: The UPDATE command is used to modify existing data in a table.

Syntax:

UPDATE table_name

SET column1 = value1, column2 = value2, ...

WHERE condition;

Example:

UPDATE emp

SET age = 21

WHERE empno = 1;

3. DELETE

Explanation: The DELETE command is used to remove existing rows from a table.

Syntax:

DELETE FROM table_name

```
WHERE condition;
```

Example:

```
DELETE FROM emp
WHERE empno = 1;
```

4. SELECT

Explanation: The SELECT command is used to retrieve data from a table.

The following syntax is used to display data of specific columns from the table.

Syntax:

```
    1)SELECT column1, column2, ...
FROM table_name;
```

Example:

```
SELECT Name, Age
FROM emp;
```

The following syntax is used to display data of the entire table

Syntax:

```
    2)SELECT  *  FROM table_name;
```

Example:

```
SELECT  *    FROM emp;
```

Summary Table:

Command	**Description**	**Example**
INSERT	Add new rows to a table	INSERT INTO Students (ID, Name, Age) VALUES (...);
UPDATE	Modify existing rows in a table	UPDATE Students SET Age = 21 WHERE ID = 1;

DELETE	Remove rows from a table	DELETE FROM Students WHERE ID = 1;

VII. Required Resources/apparatus/equipment with specifications

Sr. No	Equipment Name with Broad Specifications	Relevant LLO Number
1	Computer system with all necessary components like; motherboard, random access memory (RAM), read-only memory (ROM), internal hard disk drives, Mouse, Keyboard, and RDBMS applications such as Oracle Express Edition, MySql, SQLite,Oracle Live SQL etc.	All

VIII. Procedure

1. Create Database for given application
2. Follow syntax to modify the table according to question.

IX. Result(s)

……………………………………………………………………………………………………
…………………………………………………………………………………………………
……………………………………………………………………………………………………

X. Practical related questions (Provide space for answers)

Note: Below are a few sample questions for reference. Teacher must design more suchquestions to ensure the achievement of identified CO.

1. Describe check constraint.
2. Describe referential integrity constraint.
3. Explain the differences between the DELETE, DROP, and TRUNCATE commands in SQL.

(Space for answer)

……………………………………………………………………………………………………
…………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
…………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
…………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
…………………………………………………………………………………………………
……………………………………………………………………………………………………

…………………………………………………………………………………………………
……………………………………………………………………………………….
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………

XI. Exercise

1. Using various syntax of insert command insert the following rows of data in the EMP table.

EMPNO	ENAME	DNAME	JOB	HIREDATE	LOC
7876	ADAMS	RESEARCH	CLERK	23-MAY-87	DALLAS
7499	ALLEN	SALES	SALESMAN	20-FEB-81	CHICAGO
7698	SMITH	SALES	MANAGER	01-MAY-81	CHICAGO
7782	CLARK	ACCOUNTING	MANAGER	09-JUN-81	NEW YORK

2. Insert the multiple records in the EMP table using single insert command.

EMPNO	ENAME	DNAME	JOB	HIREDATE	LOC
7902	FORD	RESEARCH	ANALYST	03-DEC-81	DALLAS
7900	JAMES	SALES	CLERK	03-DEC-81	CHICAGO
7566	JONES	RESEARCH	MANAGER	02-APR-81	DALLAS
7839	KING	ACCOUNTING	PRESIDENT	17-NOV-81	NEW YORK

3. Delete record of SMITH from the above table
4. Change the job of ADAMS to MANAGER
5. Display contents of empno and sal

XII. References/Suggestions for further reading: include websites/links

1. https://www.youtube.com/watch?v=yGU4YfSSjdM
2. https://www.javatpoint.com/dml-commands-in-sql
3. https://www.tutorialspoint.com/what-are-the-dml-commands-in-dbms

XIII. Assessment Scheme

Performance indicators		Weightage
Sr. No	**Process related (10 Marks)**	**30%**
1.	Tool Selection Ability	20%
2.	Follow Ethical Practices	10%
Product related (15 Marks)		**70%**
3.	Verifying System Requirement for Installation	20%
4.	Correctness in Use of Command Tools	15%
5.	Use of Attribute/s Options	15%
6.	Timely Submission	10%
7.	Answering Sample Questions	10%
Total (50 Marks)		**100%**

List of student Team Members

1…………………………………

2……………………………….…

3…………………………………

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (35)**	**Total (50)**	

Experiment No. 5: Execute DCL commands to control the access to data using SQL

I. Practical Significance

DCL stands for Data Control Language. This command is used to provide different user access to the stored data. It enables the data administrator to Grant and Revoke the required access to the database. The DCL commands are easier to implement with its simple syntax.

II. Industry/Employer Expected Program Outcomes (POs)

The aim of this practical is to create user and give access to users using Grant command and deny access using revoke command.

III. Course level learning outcomes

This practical is expected to develop the following skills in you:

1. Understand how to design the Database system based on the requirements.
2. Create user/ multiple users.
3. Provide access to Database.
4. Deny access to Database

IV. Laboratory Learning Outcome(s)

1. Write and execute SQL queries for creating Users.
2. Write queries for providing access to User to database.
3. Write queries to deny access to User.

V. Relevant Affective domain related Outcome(s)

1. Follow ethical practices.
2. Use appropriate DBMS software.
3. Demonstrate analytical and logical knowledge as an Individual.
4. Participate in team problem solving activities.
5. Prioritizes time effectively to meet the needs of the team and self.

VI. Minimum Theoretical Background

DDL Commands:

The Database Administrator has authority to create as many users as needed. The user is created using the CREATE USER command. Privileges is a right to execute the SQL statement or to access object. There are two types of privileges.

SYSTEM privileges: It is generally granted by DBA to users. Example: Create table, create user etc.…

OBJECT privileges: This allows access to objects or privileges on objects, that is tables, table columns, Tables, Views etc. It includes Alter, delete, insert, select, update commands. The DBA user the GRANT statement to allocate system privileges to another user. The REVOKE command is used to remove privileges granted to users.

VII. Additional Software required

……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
………………………………………………………………………………

VIII. Precautions

1) Use of appropriate syntax
2) Select required relations and use relevant conditions.

IX. Additional Resources used

……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………

X. Result (Output of the procedure)

……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
………………………………………………………………………………

XI. Practical Related Questions

Note: Below are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

(**Note:** Use Point VII and XIII to XV for all relevant practical exercise use blank pages provided or attach more pages if needed.)

1. State the use of 'with grant options' clause in grant command.
2. Consider table EMPLOYEE and DEPARTMENT with following schema:
   EMP (empno, empname, salary, phno) Dept (deptno, empno, deptname, location, jobtype)
   Write the output of the following queries:

- Create user Jay identifies by any admin;
- Grant create table, create view to Jay;
- Grant select, insert, update on Emp to Jay;
- Grant select, update (deptno, empno) on Dept to Jay;
- Alter user Jay identified by admin;
- Revoke create table, create views from Jay;
- Revoke select, insert, update on Emp from Jay;
- Create role emp_pvr;
- Grant create table, create views to emp_pvr;
- Grant emp_pvr to Jay, John;

(Space for answers)

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

……………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

XII. Exercise

Attempt following and teacher shall design and allot more questions to attain desired outcome: (Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Create the user Jay and implement the following commands on table EMP and Dept.
2. Write a query to grant select, insert, delete privileges on Emp and Dept table.
3. Write a query to grant update privilege on columns of empno and salary on Emp table.
4. Write a query to revoke all above privileges from Emp and Dept table.
5. Write a query to create role dept_pvr.
6. Write a query to assign system privileges- create table, create view to role dept_pvr;
7. Write a query to assign above system privileges to users Jay and John.
8. Write a query to assign object privileges- select, insert, delete to role dept_pvr.
9. Write a query to assign above object privileges to users' jay and john.

(Space for answers)

……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………

Precautions to be followed

1. Use of appropriate syntax
2. Select required relations and use relevant conditions.

XIV. References/ Suggestions for further Reading

1. https://blogs.oracle.com/sql/post/how-to-use-create-table-alter-table-and-drop-table-in-oracle-database
2. https://support.microsoft.com/en-us/office/database-design-basics-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5
3. https://www.oracle.com/
4. www.w3school.com

XV. Assessment Scheme

Performance indicators		Weightage
Sr. No	**Process related (10 Marks)**	**30%**
1.	Tool Selection Ability	20%
2.	Follow Ethical Practices	10%
	Product related (15 Marks)	**70%**
3.	Verifying System Requirement for Installation	20%
4.	Correctness in Use of Command Tools	15%
5.	Use of Attribute/s Options	15%
6.	Timely Submission	10%
7.	Answering Sample Questions	10%
	Total (50 Marks)	**100%**

List of Students /Team Members

1. ....

2. ……………………………………

3. ……………………………………

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (35)**	**Total (50)**	

Experiment No 06: Execute TCL Commands to control transactions on data using SQL

I. **Practical Significance**: A transaction is set of Read/Write (DML) operations in a database. Transaction Control Language (TCL) Commands are used to manage transactions in a database. They help ensure data integrity by allowing you to save, undo, or partially undo changes made during a transaction. This practical allows students to execute TCL commands to control transactions on data.

II. **INDUSTRY / EMPLOYER EXPECTED OUTCOME:**
To execute TCL Commands to control transactions on data using SQL

III. **COURSE LEVEL LEARNING OUTCOMES (COS)**:
CO3 - Manage database using SQL

IV. **LABORATORY LEARNING OUTCOME:**
Execute TCL Commands to control transactions on data using SQL.

V. **Relevant Affective Domain related outcome(s)**

a. Follow precautionary measures.
b. Follow installation steps.
c. Follow ethical practices.

VI. **Relevant Theoretical Background**

Transaction Control Language (TCL) Commands manage the changes made by DML statements in a database. These commands include COMMIT, ROLLBACK, and SAVEPOINT, which help in saving, undoing, and setting intermediate points within transactions, respectively.

1. COMMIT

Explanation: The COMMIT command is used to save all changes made during the current transaction to the database permanently.

Syntax:

```
COMMIT;
```

Example:

```
update emp
set sal = sal + 5000
where empno = 101;
```

After executing update command execute following TCL command :

```
commit;
```

The salary increases for the employee with empno = 101 is *saved permanently* in the database.

2. ROLLBACK

Explanation: The ROLLBACK command is used to undo changes made during the current transaction, reverting the database to its previous state.

Syntax:

```
ROLLBACK;
```

Example:

```
update emp
set sal = sal+ 5000
where empno = 101;
```

After executing update command execute following TCL command :

```
rollback;
```

The salary increases for the employee with empno = 101 is undone, and the database *returns to its state as it was before the execution of update command*.

3. SAVEPOINT

Explanation: The SAVEPOINT command is used to set a savepoint within a transaction, allowing for partial rollbacks to specific points within the transaction.

Syntax:

```
SAVEPOINT savepoint_name;
```

Example:

```
SAVEPOINT BeforeUpdate;

update emp
set sal = sal + 5000
where empno = 101;

rollback to BeforeUpdate;
```

The salary increases for the employee with EmployeeID = 101 is undone, and the database returns to the state at the SAVEPOINT BeforeUpdate.

Summary Table

Command	**Description**	**Syntax**	**Effects of TCL command**
COMMIT	Save all changes made during the current transaction permanently	COMMIT;	Changes are saved permanently.
ROLLBACK	Undo changes made during the current transaction	ROLLBACK;	Changes are undone, reverting to the previous state.
SAVEPOINT	Set a savepoint within a transaction for partial rollbacks	SAVEPOINT savepoint_name;	A savepoint is created, allowing for partial rollbacks.

VII. Required Resources/apparatus/equipment with specifications

Sr.No	**Equipment Name with Broad Specifications**	**Relevant LLO Number**
1	Computer system with all necessary components like; motherboard, random access memory (RAM), read-only memory (ROM), internal hard disk drives, Mouse, Keyboard, and RDBMS applications such as Oracle Express Edition, MySql,SQLite,Oracle Live SQL etc.	All

VIII. Procedure

1. Create Database for given application
2. Create tables for the given application
3. Execute TCL Commands after DML Commands

IX. Result(s)

……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………

X. Practical related questions (Provide space for answers)

Note: Below are a few sample questions for reference. Teacher must design more suchquestions so as to ensure the achievement of identified CO.

1. Explain the need of TCL command in SQL
2. Differentiate between COMMIT and ROLLBACK commands in SQL

3. Describe savepoint
4. Describe states of transaction with neat diagram
5. Describe ACID properties of transaction

(Space for answer)

…………………………………………………………………………………………………
…………………………………………………………………………………………………
……………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
……………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
……………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
……………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
……………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
……………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
……………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
……………………………………………………………

XI. Exercise

1. Write TCL command to save all the changes made so far in the EMP
2. Delete any one record in the EMP table created earlier and undo the deletion operation
3. You are in the middle of a transaction and want to set a savepoint named BeforeSalaryUpdate. Write the SQL command to set this savepoint.

XII. References/Suggestions for further reading: include websites/links

1. https://www.youtube.com/watch?v=LSB4eceRsw8
2. https://www.youtube.com/watch?v=yGU4YfSSjdM
3. https://www.javatpoint.com/tcl-commands-in-sql
4. https://www.programiz.com/sql/tcl-commands

XIII. Assessment Scheme

Performance indicators		Weightage
Sr. No	**Process related (10 Marks)**	**30%**
1.	Tool Selection Ability	20%
2.	Follow Ethical Practices	10%
	Product related (15 Marks)	**70%**
3.	Verifying System Requirement for Installation	20%
4.	Correctness in Use of Command Tools	15%
5.	Use of Attribute/s Options	15%
6.	Timely Submission	10%
7.	Answering Sample Questions	10%
	Total (50 Marks)	**100%**

List of student Team Members

1…………………………………

2……………………………….....

3…………………………………

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (35)**	**Total (50)**	

Experiment No 07: Write Queries using Arithmetic operators.

I. Practical Significance:

The arithmetic operators are used on the data stored in the tables. We can use these operators with the SELECT statement in SQL. We can also use WHERE clause in the SELECT statement for performing operations on particular rows. The arithmetic operators are used between two numerical operands for performing addition, subtraction, multiplication and division operations. This practical allows students to write SQL queries using arithmetic operators and performs mathematical operations on the data stored in the tables.

II. INDUSTRY / EMPLOYER EXPECTED OUTCOME:

To implement arithmetic operators to apply various conditions in query on the given database application

III. COURSE LEVEL LEARNING OUTCOMES (COS):CO3 -

Manage database using SQL.

IV. LABORATORY LEARNING OUTCOME:

Implement arithmetic operators to data stored in the table.

V. Relevant Affective Domain related outcome(s)

a. Follow precautionary measures.
b. Follow installation steps.
c. Follow ethical practices.

VI. Relevant Theoretical Background

Serial Number	Operator Name	Explanation	Syntax
1	ADDITION	The addition plus **(+)** operator is used	SELECT <Expression>[arithmetic

		to add two or more expressions or numbers.	operator]<expression>...
			FROM [table_name]
			WHERE [expression];

2	SUBTRACTION	The subtraction minus **(-)** operator is used to subtract one expression or number from another expression or number.	SELECT <Expression>[arithmetic operator]<expression>...
			FROM [table_name]
			WHERE [expression];

3	MULTIPLICATION	The multiply operators (*) is used to multiply two or more expressions or numbers.	SELECT <Expression>[arithmetic operator]<expression>...
			FROM [table_name]
			WHERE [expression];

4	DIVISION	The division operators (/) is used to divide one expression or number by another.	SELECT <Expression>[arithmetic operator]<expression>...
			FROM [table_name]
			WHERE [expression];

5	MODULO	The modulo operator (%) returns the remainder (integer) of the division.	SELECT <Expression>[arithmetic operator]<expression>...
			FROM [table_name]
			WHERE [expression];

Compound Operators:

Operator	**Operator Name**	**Explanation**
+=	+= (Add Assignment) (Transact-SQL)	Adds some amount to the original value and sets the original value to the result.
-=	-= (Subtract Assignment) (Transact-SQL)	Subtracts some amount from the original value and sets the original value to the result.
*=	*= (Multiply Assignment) (Transact-SQL)	Multiplies by an amount and sets the original value to the result.
/=	(Divide Assignment) (Transact-SQL)	Divides by an amount and sets the original value to the result.
%=	Modulus Assignment (Transact-SQL)	Divides by an amount and sets the original value to the modulo.

&=	&= (Bitwise AND Assignment) (Transact-SQL)	Performs a bitwise AND and sets the original value to the result.
^=	^= (Bitwise Exclusive OR Assignment) (Transact-SQL)	Performs a bitwise exclusive OR and sets the original value to the result.
|=	|= (Bitwise OR Assignment) (Transact-SQL)	Performs a bitwise OR and sets the original value to the result.

VII. Required Resources/apparatus/equipment with specifications

Sr.No	Equipment Name with Broad Specifications	Relevant LLO Number
1	Computer system with all necessary components like; motherboard, random access memory (RAM), read-only memory (ROM), internal hard disk drives, Mouse, Keyboard, and RDBMS applications such as Oracle Express Edition, MySQL, SQLite, Oracle Live SQL etc.	All

VIII. Procedure

1. Create tables for the given application
2. Apply Arithmetic operators on the given application

IX. Result(s)

…………………………………………………………………………………………………
…………………………………………………………………………………………………
……………………………………………………………………………………………

X. Practical related questions (Provide space for answers)

Note: Below are a few sample questions for reference. Teacher must design more suchquestions so as to ensure the achievement of identified CO.

1. What is precedence/ SQL arithmetic order in arithmetic operators?
2. Explain the SQL exponentiation operator.
3. List few compound operators.

(Space for answer)

…………………………………………………………………………………………………
…………………………………………………………………………………………………
……………………………………………………………………………………………

…………………………………………………………………………………………………
…………………………………………………………………………………………………

……………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

……………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

……………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

……………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

……………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

……………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………………

………………………………………………………………………………………………………

……………………………………………………………………………………………

…………………………………………………………………………………………………

XI. Exercise

1. Consider the following schema
   Orders(cust_id, order_id, items, amount)
   Write queries for the following:
   i. Display new column named total_amount which is 200 added to the amount field.
   ii. Display new column named offer_price which is 100 subtracted from the amount field.
   iii. Display new column named revised_amount which is multiplied by 5 times the amount field.
   iv. Display new column named half_amount which is divided by 2 to the amount field.

XII. References/Suggestions for further reading: include websites/links

1. https://www.youtube.com/watch?v=lF_JcTfVS4w
2. https://www.tutorialspoint.com/sql/sql-logical-operators.htm
3. https://www.javatpoint.com/sql-logical-operators
4. https://www.youtube.com/watch?v=NJZ206_iJ0I
5. Compound Operators (Transact-SQL) - SQL Server | Microsoft Learn

XIII. Assessment Scheme

Performance indicators		Weightage
Sr. No	**Process related (10 Marks)**	**30%**
1.	Tool Selection Ability	20%
2.	Follow Ethical Practices	10%
	Product related (15 Marks)	**70%**
3.	Verifying System Requirement for Installation	20%
4.	Correctness in Use of Command Tools	15%
5.	Use of Attribute/s Options	15%
6.	Timely Submission	10%
7.	Answering Sample Questions	10%
	Total (50 Marks)	**100%**

List of student Team Members

1……………………………………

2………………………………….....

3……………………………………

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (35)**	**Total (50)**	

Experiment No 08: Apply built-in Logical operators on given data

I. **Practical Significance**: Logical operators in SQL are used to combining conditions. They find specific data by checking multiple rules at once. For example, one can find employees who work in a certain department and were hired this year. This practical allows students to write SQL queries using logical operators and retrieve data from the database.

II. **INDUSTRY / EMPLOYER EXPECTED OUTCOME:**
To implement Logical operators to apply various conditions in query on the given database application

III. **COURSE LEVEL LEARNING OUTCOMES (COS)**:CO3
- Manage database using SQL.

IV. **LABORATORY LEARNING OUTCOME:**
Implement Logical operators to apply various conditions in query.

V. **Relevant Affective Domain related outcome(s)**

a. Follow precautionary measures.

b. Follow installation steps.

c. Follow ethical practices.

VI. **Relevant Theoretical Background**

Sr No.	Operator Name	Explanation	Syntax	SQL Query Example
1	AND	Returns true if both conditions are true	Select * from tablename where condition1 AND condition2;	SELECT * FROM employees WHERE age > 30 AND salary > 50000;
2	OR	Returns true if either condition is true	Select * from tablename where condition1	SELECT * FROM employees WHERE department = 'IT' OR department = 'HR';

			OR condition2;	
3	NOT	Returns true if the condition is false	NOT condition	SELECT * FROM employees WHERE NOT department = 'Finance';
4	BETWEEN	Returns true if a value is within a range	Select * from tablename where BETWEEN low AND high;	SELECT * FROM employees WHERE age BETWEEN 25 AND 35;

5	IN	Returns true if a value matches any in a list	Select * from tablename where columnname IN (list);	SELECT * FROM employees WHERE department
				IN ('IT', 'Finance');

6	LIKE	Returns data based on a specified pattern, using '%' and '_' as wildcard characters for pattern matching. • %: Matches zero or more characters. • _: Matches any single character.	value LIKE pattern	SELECT * FROM employees WHERE name LIKE 'P%'; **Note:** • P%: Matches strings that start with "P" followed by any characters. • %P: Matches strings that end with "P", with any characters preceding it. • %P%: Matches strings that contain "P" anywhere, surrounded by any characters.

VII. Required Resources/apparatus/equipment with specifications

Sr.No	Equipment Name with Broad Specifications	Relevant LLO Number
1	Computer system with all necessary components like; motherboard, random access memory (RAM), read-only memory (ROM), internal hard disk drives, Mouse, Keyboard, and RDBMS applications such as Oracle Express Edition,MySql,SQLite,Oracle Live SQL etc.	All

VIII. Procedure

1. Create tables for the given application
2. Apply logical operators on the given application

IX. Result(s)

………………………………………………………………………………………………

………………………………………………………………………………………………

X. Practical related questions (Provide space for answers)

Note: Below are a few sample questions for reference. Teacher must design more suchquestions so as to ensure the achievement of identified CO.

1. Explain the need of logical operators in SQL
2. List three logical operators and describe them with an example.

(Space for answer)

………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………

XI. Exercise

1. Consider the following schema
   Emp(empno,ename,job,mgr,hiredate,sal,comm,deptno)
   Write queries for the following:

i. Display employees whose city is ‘Mumbai’ and earns more than 50000
ii. Display employees who job is Clerk or commission is 500
iii. Display details of employees whose salary is between 20000 and 50000.
iv. Display details of employees who stays at Mumbai, Pune, Nashik or Nagpur

XII. References/Suggestions for further reading: include websites/links

1. https://www.youtube.com/watch?v=yGU4YfSSjdM
2. https://www.tutorialspoint.com/sql/sql-logical-operators.htm
3. https://www.javatpoint.com/sql-logical-operators

XIII. Assessment Scheme

Performance indicators		**Weightage**
Sr. No	**Process related (10 Marks)**	**30%**
1.	Tool Selection Ability	20%
2.	Follow Ethical Practices	10%
Product related (15 Marks)		**70%**
3.	Verifying System Requirement for Installation	20%
4.	Correctness in Use of Command Tools	15%
5.	Use of Attribute/s Options	15%
6.	Timely Submission	10%
7.	Answering Sample Questions	10%
Total (50 Marks)		**100%**

List of student Team Members

1…………………………………

2……………………………….…

3…………………………………

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (35)**	**Total (50)**	

Experiment No 09: Implement Relational operators to apply various conditions in query.

I. **Practical Significance**:

The Relational operators in SQL is used to compare two expressions or values and return a Boolean result. The Relational operators are used on the data stored in the tables. We can use these operators with the WHERE clause in the SQL queries.

II. **INDUSTRY / EMPLOYER EXPECTED OUTCOME:**

To implement Relational operators to apply various conditions in query on the given database application

III. **COURSE LEVEL LEARNING OUTCOMES (COS)**:

CO3 - Manage database using SQL.

IV. **LABORATORY LEARNING OUTCOME:**

Implement Relational operators to data stored in the table.

V. **Relevant Affective Domain related outcome(s)**

a. Follow precautionary measures.

b. Follow installation steps.

c. Follow ethical practices.

VI. **Relevant Theoretical Background**

Serial Number	Operator Name	Explanation	Syntax/example
1	SQL Equal Operator (=)	This operator is highly used in SQL queries. The **Equal Operator** in SQL shows only data that matches the specified value in the query.	SELECT * FROM Employee WHERE Gender = ‘Male’;
2	SQL Equal Not Operator (!=)	The **Equal Not Operator** in SQL shows only those data that do not match the query's specified value.	SELECT * FROM Employee_details WHERE Emp_Salary! = 45000;
3	SQL Greater Than	The **Greater Than Operator** in SQL shows only those data which	SELECT * FROM Employee_details WHERE Emp_Id > 202;

	Operator (>)	are greater than the value of the right-hand operand	
4	SQL Greater Than Equals to Operator (>=)	The **Greater Than Equals to Operator** in SQL shows those data from the table which are greater than and equal to the value of the right-hand operand.	SELECT * FROM Employe e_details WHERE Emp_Id >= 202;
5	SQL Less Than Operator (<)	The **Less Than Operator** in SQL shows only those data from the database tables which are less than the value of the right-side operand.	SELECT * FROM Employe e_details WHERE Emp_Id < 204;
6	SQL Less Than Equals to Operator (<=)	The **Less Than Equals to Operator** in SQL shows those data from the table which are lesser and equal to the value of the right-side operand.	SELECT * FROM Employe e_details WHERE Emp_Id <= 203;

VII. Required Resources/apparatus/equipment with specifications

Sr.No	Equipment Name with Broad Specifications	Relevant LLO Number
1	Computer system with all necessary components like; motherboard, random access memory (RAM), read-only memory (ROM), internal hard disk drives, Mouse, Keyboard, and RDBMS applications such as Oracle Express Edition, MySQL, SQLite, Oracle Live SQL etc.	All

VIII. Procedure

1. Create tables for the given application
2. Apply Relational operators on the given application

IX. Result(s)

……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
…………………………………………………………………………………………………

X. Practical related questions (Provide space for answers)

Note: Below are a few sample questions for reference. Teacher must design more suchquestions so as to ensure the achievement of identified CO.

1. What is the use of relational operators?
2. List all Relational operators of SQL.

(Space for answer)

……………………………………………………………………………………………
……………………………………………………………………………………………
……………………………………………………………………………………………
……………………………………………………………………………………………
……………………………………………………………………………………………
……………………………………………………………………………………………
……………………………………………………………………………………………
……………………………………………………………………………………………
……………………………………………………………………………………………
……………………………………………………………………………………………
……………………………………………………………………………………………
……………………………………………………………………………………………
……………………………………………………………………………………………
……………………………………………………………………………………………
……………………………………………………………………………………………
……………………………………………………………………………………………
……………………………………………………………………………………………
……………………………………………………………………………………………

XI. Exercise

1. Consider the following schema
Student (stu_name, course_id, Roll_no, percentage)
Write queries for the following:

i. Select stu_name, course_id, from Student WHERE percentage is >=60 and <=100;
ii. Select details of students whose Roll numbers are above 15;
iii. Select stu_id, Roll_no from Student WHERE course_id! =121;

XII. References/Suggestions for further reading: include websites/links

1. https://www.youtube.com/watch?v=yGU4YfSSjdM
2. https://www.tutorialspoint.com/sql/sql-logical-operators.htm
3. https://www.javatpoint.com/sql-logical-operators

XIII. Assessment Scheme

Performance indicators		Weightage
Sr. No	Process related (10 Marks)	30%
1.	Tool Selection Ability	20%
2.	Follow Ethical Practices	10%
	Product related (15 Marks)	70%
3.	Verifying System Requirement for Installation	20%
4.	Correctness in Use of Command Tools	15%
5.	Use of Attribute/s Options	15%
6.	Timely Submission	10%
7.	Answering Sample Questions	10%
	Total (50 Marks)	100%

List of student Team Members

1…………………………………

2………………………………....

3…………………………………

Marks Obtained			Dated signature of Teacher
Process Related (15)	Product Related (35)	Total (50)	

Experiment No 10: Use Set operators to perform different operations

I. **Practical Significance**: Set operators in SQL help to combine and compare the results of two or more queries. You can use set operators to find common elements, unique elements, or differences between tables. This makes it easier to analyze and understand your data. Students will learn to manipulate and analyze data by combining and comparing multiple datasets using set operators in SQL.

II. INDUSTRY / EMPLOYER EXPECTED OUTCOME:

To write SQL queries to implement SET operators using SQL

III. COURSE LEVEL LEARNING OUTCOMES(COS):CO3

- Manage database using SQL.

IV. LABORATORY LEARNING OUTCOME:

Write Queries to implement SET operations using SQL.

V. Relevant Affective Domain related outcome(s)

a. Follow precautionary measures.

b. Follow installation steps.

c. Follow ethical practices.

VI. Relevant Theoretical Background

There are certain rules which must be followed to perform operations using *set operators* in SQL.

Rules are as follows:

1. The *number and order of columns* must be the *same.*
2. *Data types* must be *compatible*

Serial Number	Operator	Description	Syntax	SQL Query Example
1	UNION	Combines two	SELECT * FROM	SELECT * FROM emp

		tables, *removes duplicates*	tablename1 UNION SELECT * FROM tablename2;	UNION SELECT * FROM dept;

2	UNION ALL	Combines two tables, *keeps duplicates*	SELECT * FROM tablename1
			UNION ALL
			SELECT * FROM tablename2;
				SELECT * FROM emp
				UNION ALL
				SELECT * FROM dept;

3	INTERSECT	Returns *common rows* between two tables	SELECT * FROM tablename1
			INTERSECT
			SELECT * FROM tablename2;
				SELECT * FROM emp
				INTERSECT SELECT * FROM dept;

4	MINUS	(*table1 minus table2*)
		Returns rows from first table not in second
			SELECT * FROM tablename1
			MINUS
			SELECT * FROM tablename2;
				SELECT * FROM emp
				MINUS
				SELECT * FROM dept;

VII. Required Resources/apparatus/equipment with specifications

Sr.No	Equipment Name with Broad Specifications	Relevant LLO Number
1	Computer system with all necessary components like; motherboard, random access memory (RAM), read-only memory (ROM), internal hard disk drives, Mouse, Keyboard, and RDBMS applications such as Oracle Express Edition,MySql,SQLite,Oracle Live SQL etc.	All

VIII. Procedure

1. Create tables for the given application
2. Apply set operators on the given tables

IX. Result(s)

………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………

X. Practical related questions (Provide space for answers)

Note: Below given are few sample questions for reference. Teacher must design more suchquestions so as to ensure the achievement of identified CO.

1. Explain the need of set operators in SQL
2. Describe various set operators in SQL

(Space for answer)

……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………

XI. Exercise

1. Consider following Schema :
emp1(empno,ename,deptno)
emp2(empno,ename,deptno)
Write SQL commands for the following statements.

1. Display the names of employees including duplicate employee names.
2. Display the names of employees excluding duplicate employee names.
3. Display the common employee names from both the tables.
4. List employees who are not assigned to any department?

XII. References/Suggestions for further reading: include websites/links

1. https://www.youtube.com/watch?v=yGU4YfSSjdM
2. https://www.w3schools.com/sql/sql_union.asp
3. https://www.javatpoint.com/set-operators-in-sql

XIII. Assessment Scheme

Performance indicators		Weightage
Sr. No	**Process related (10 Marks)**	**30%**
1.	Tool Selection Ability	20%
2.	Follow Ethical Practices	10%
	Product related (15 Marks)	**70%**
3.	Verifying System Requirement for Installation	20%
4.	Correctness in Use of Command Tools	15%
5.	Use of Attribute/s Options	15%
6.	Timely Submission	10%
7.	Answering Sample Questions	10%
	Total (50 Marks)	**100%**

List of student Team Members

1…………………………………

2……………………………….....

3…………………………………

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (35)**	**Total (50)**	

Experiment No 11: Execute queries using string functions

I. Practical Significance:

The string is a collection of characters used to store multiple characters. Most of the time we need to modify and access the strings. SQL has built-in string functions. The SQL built-in string functions take an input string and return an output string. String functions are used to perform various character manipulations.

II. INDUSTRY / EMPLOYER EXPECTED OUTCOME:

To execute queries using string functions to perform advance calculations

III. COURSE LEVEL LEARNING OUTCOMES (COS):

CO3 - Manage database using SQL.

IV. LABORATORY LEARNING OUTCOME:

Execute queries using String functions.

V. Relevant Affective Domain related outcome(s)

a. Follow precautionary measures.

b. Follow installation steps.

c. Follow ethical practices.

VI. Relevant Theoretical Background

String Functions are used to perform operations on characters and return characters.

Some of the string functions are as follows:

Function	Explanation	Example	Output
ASCII	Converts a single character string to its corresponding ASCII code, between 0 and 255.	SELECT ASCII('t');	116
CHAR_LENGTH ()	It gives you the number of characters in the string.	SELECT CHAR_LENGTH('world!');	6

CHARACTER_LENGTH ()	It gives you the number of characters in a given set of strings.	SELECT CHARACTER_LENGTH (‘HELLO WORLD');	11
CONCAT ()	It appends two strings to create the new single string	SELECT CONCAT ('SQL', ' is', ' very', ' interesting, ' subject.');	SQL is very interesting subject
LCASE String Function .	This string function allows users to convert the specified string into lower case letters	SELECT LCASE (‘The CAPITAL of INDIA is NEW DELHI');	the capital of india is new delhi
LEFT/RIGHT String Function .	This string function shows the leftmost characters from the given string. It reads the characters to the given index position	SELECT LEFT (‘The CAPITAL of INDIA is NEW DELHI', 11);	The CAPITAL
LOCATE String Function	This string function shows the index value of the first occurrence of the word in the given string.	SELECT LOCATE('INDIA','The CAPITAL of INDIA is NEW DELHI ', 1);	16
LPAD /RPAD String Function	This string function adds the given symbol to the left of the given string.	SELECT LPAD('NEW', 6, '#');	###NEW
LTRIM/ RTRIM String Function	This string function cuts the given character or string from the left of the given original string. It also removes the space from the left of the specified string.	SELECT LTRIM('NEW DELHI IS THE CAPITAL OF INDIA', 'NEW DELHI');	IS THE CAPITAL OF INDIA

VII. Required Resources/apparatus/equipment with specifications

Sr.No	Equipment Name with Broad Specifications	Relevant LLO Number
1	Computer system with all necessary components like; motherboard, random access memory (RAM), read-only memory (ROM), internal hard disk drives, Mouse, Keyboard, and RDBMS applications such as Oracle Express Edition,MySql,SQLite, Oracle Live SQL etc.	All

VIII. Procedure

Execute SQL queries using String functions

IX. Result(s)

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

X. Practical related questions (Provide space for answers)

Note: Below are a few sample questions for reference. Teacher must design more suchquestions so as to ensure the achievement of identified CO.

1. What is REPLACE string functions?
2. What is the function of SPACE string function?
3. What output you get when you compare two strings using STRCMP built in string function?

(Space for answer)

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

XI. Exercise:

1. Write output of the following queries.
2. Select concat (‘Jay’ ‘IITB’) from Dual;
3. Select ltrim (‘Shreya’,’s’) from Dual;
4. Select upper(‘raj’) from Dual;
5. Select rpad (“HR’, 10, ‘*’) from Dual;

XII. References/Suggestions for further reading: include websites/links

1. https://www.oreilly.com/library/view/oracle-plsql-programming/0596003811/ch09s03.html
2. https://beginner-sql-tutorial.com/oracle-functions.htm
3. https://www.javatpoint.com

XIII. Assessment Scheme

Performance indicators		Weightage
Sr. No	**Process related (10 Marks)**	**30%**
1.	Tool Selection Ability	20%
2.	Follow Ethical Practices	10%
	Product related (15 Marks)	**70%**
3.	Verifying System Requirement for Installation	20%
4.	Correctness in Use of Command Tools	15%
5.	Use of Attribute/s Options	15%
6.	Timely Submission	10%
7.	Answering Sample Questions	10%
	Total (50 Marks)	**100%**

List of student Team Members

1…………………………………

2………………………………....

3…………………………………

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (35)**	**Total (50)**	

Experiment No 12: Execute queries using Arithmetic functions

I. **Practical Significance**: Arithmetic functions in SQL are needed to perform mathematical operations on data. They enable to analyze and manipulate numerical data directly within the database. This practical allows students to write SQL queries using built-in arithmetic functions given application.

II. **INDUSTRY / EMPLOYER EXPECTED OUTCOME:**

To execute queries using arithmetic functions to perform advance calculations

III. **COURSE LEVEL LEARNING OUTCOMES (COS)**:CO3

- Manage database using SQL.

IV. **LABORATORY LEARNING OUTCOME:**

Execute queries using Arithmetic functions.

V. **Relevant Affective Domain related outcome(s)**

a. Follow precautionary measures.

b. Follow installation steps.

c. Follow ethical practices.

VI. **Relevant Theoretical Background**

Numeric Functions are used to perform operations on numbers and return numbers

Some of the arithmetic functions are as follows:

Function	Explanation	Example	Output
ABS()	It returns the absolute value of a number.	SELECT ABS(-363.4) from dual;	363.4

CEIL()	It returns the smallest integer value that is greater than or equal to a number.	SELECT CEIL (2.83) from dual;	3
FLOOR()	Integer value that is Less than or equal to the number 'x'	SELECT FLOOR (2.83) from dual;	2
MOD()	It returns the remainder of n divided by m.	SELECT MOD(21, 4) from dual;	1
POWER()	POWER(m, n): It returns m raised to the nth power.	SELECT POWER(5, 2) from dual;	25
ROUND()	It returns a number rounded to a certain number of decimal places.	SELECT ROUND(7.553) from dual;	8
SQRT()	It returns the square root of a number.	SELECT SQRT(49) from dual;	7
TRUNC()	It returns numeric value truncated to n places right of the decimal point.	SELECT TRUNC(56.53635, 2) from dual;	56.53

VII. Required Resources/apparatus/equipment with specifications

Sr.No	Equipment Name with Broad Specifications	Relevant LLO Number
1	Computer system with all necessary components like; motherboard, random access memory (RAM), read-only memory (ROM), internal hard disk drives, Mouse, Keyboard, and RDBMS applications such as Oracle Express Edition,MySql,SQLite, Oracle Live SQL etc.	All

VIII. Procedure

Execute SQL queries using arithmetic functions

IX. Result(s)

……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………

X. Practical related questions (Provide space for answers)

Note: Below are a few sample questions for reference. Teacher must design more suchquestions so as to ensure the achievement of identified CO.

1. Differentiate between round() and trunc()
2. Differentiate between floor() and ceil()
3. List arithmetic functions which are not listed here.

(Space for answer)

……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………

XI. Exercise

1. Calculate cube of following numbers: 12,14,16
2. Display details of salary of employees with roundup value.

XII. References/Suggestions for further reading: include websites/links

1. https://www.youtube.com/watch?v=yGU4YfSSjdM
2. https://www.oreilly.com/library/view/oracle-plsql-programming/0596003811/ch09s03.html
3. https://beginner-sql-tutorial.com/oracle-functions.html

XIII. Assessment Scheme

Performance indicators		Weightage
Sr. No	**Process related (10 Marks)**	**30%**
1.	Tool Selection Ability	20%
2.	Follow Ethical Practices	10%
	Product related (15 Marks)	**70%**
3.	Verifying System Requirement for Installation	20%
4.	Correctness in Use of Command Tools	15%
5.	Use of Attribute/s Options	15%
6.	Timely Submission	10%
7.	Answering Sample Questions	10%
	Total (50 Marks)	**100%**

List of student Team Members

1…………………………………

2………………………………....

3…………………………………

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (35)**	**Total (50)**	

Experiment No 13: Implement queries using Date and Time functions

I. Practical Significance:

SQL Date and Time operations are used for operations on date and time provided by users. As long as your data contains only the date portion, your queries will work as expected. However, if a time portion is involved, it gets more complicated. SQL supports various date and time data types and formats. The specific date format depends on the database system you are using.

II. INDUSTRY / EMPLOYER EXPECTED OUTCOME:

To execute queries using Date and Time functions to perform advance calculations

III. COURSE LEVEL LEARNING OUTCOMES (COS):CO3

- Manage database using SQL.

IV. LABORATORY LEARNING OUTCOME:

Execute queries using Date and Time functions.

V. Relevant Affective Domain related outcome(s)

a. Follow precautionary measures.
b. Follow installation steps.
c. Follow ethical practices.

VI. Relevant Theoretical Background

SQL also provides multiple functions to handle date and time values.

Functions are as follows:

Function	Explanation	Syntax
CURDATE() Function	To get the current date, we use the CURDATE() function in MySQL. The format of the resultant date will be 'YYYY-MM-DD' (string) or YYYYMMDD (numeric).	SELECT CURDATE();

NOW() Function	The MySQL NOW() function will retrieve the current date and time value as a timestamp based on the context and, the value returned will be in either of the two formats: 'YYYY-MM-DD hh:mm:ss' and 'YYYYMMDDhhmmss'.	NOW()
CURRENT_TIMESTAMP() Function	The MySQL CURRENT_TIMESTAMP() function is used to get the current timestamp. The value returned will be in 'YYYY-MM-DD hh:mm:ss' (string) or YYYYMMDDhhmmss (numeric) format. This function is a synonym for NOW().	SELECT CURRENT_TIMESTAMP();

VII. Required Resources/apparatus/equipment with specifications

Sr.No	Equipment Name with Broad Specifications	Relevant LLO Number
1	Computer system with all necessary components like; motherboard, random access memory (RAM), read-only memory (ROM), internal hard disk drives, Mouse, Keyboard, and RDBMS applications such as Oracle Express Edition,MySql,SQLite, Oracle Live SQL etc.	All

VIII. Procedure

Execute SQL queries using date and Time functions.

IX. Result(s)

……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………

X. Practical related questions (Provide space for answers)

Note: Below are a few sample questions for reference. Teacher must design more suchquestions so as to ensure the achievement of identified CO.

1. Write the use of format () function with syntax and example.
2. Write the use of months_between function with syntax and example.
3. Write the syntax for displaying the local time stamp.

(Space for answer)

....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....

XI. Exercise:

1. Write output of the following queries.

a. Select sysdate from Dual;
b. Select last_day() from Dual;
c. Select dayofweek() from Dual;
d. Select last_day(sysdate) from Dual;
e. Select monthnow() from Dual;

XII. References/Suggestions for further reading: include websites/links

1. https://www.oreilly.com/library/view/oracle-plsql-programming/0596003811/ch09s03.html
2. https://beginner-sql-tutorial.com/oracle-functions.htm

3. https://www.javatpoint.com
4. www.tutorialsteacher.com
5. www.w3schools.com

XIII. Assessment Scheme

Performance indicators		Weightage
Sr. No	**Process related (10 Marks)**	**30%**
1.	Tool Selection Ability	20%
2.	Follow Ethical Practices	10%
Product related (15 Marks)		**70%**
3.	Verifying System Requirement for Installation	20%
4.	Correctness in Use of Command Tools	15%
5.	Use of Attribute/s Options	15%
6.	Timely Submission	10%
7.	Answering Sample Questions	10%
Total (50 Marks)		**100%**

List of student Team Members

1…………………………………

2………………………………....

3…………………………………

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (35)**	**Total (50)**	

Experiment No 14: Implement queries using Aggregate functions

I. **Practical Significance**: Aggregate functions or group functions in SQL are needed to calculate summary values from multiple rows of data. They help you add up numeric values (SUM), find the average (AVG), get the smallest (MIN) and largest (MAX) values, and count rows (COUNT) of the entire column. These functions make it easier to understand and analyze large amounts of data. This practical allows students to write queries using SQL aggregate functions.

II. **INDUSTRY / EMPLOYER EXPECTED OUTCOME:**

To find totals, averages, minimums, maximums, and counts using aggregate functions.

III. **COURSE LEVEL LEARNING OUTCOMES (COS)**:CO3 - Manage database using SQL.

IV. **LABORATORY LEARNING OUTCOME:**

Implement queries using Aggregate functions.

V. **Relevant Affective Domain related outcome(s)**

a. Follow precautionary measures.

b. Follow installation steps.

c. Follow ethical practices.

VI. **Relevant Theoretical Background**

Serial Number	**Group Function Name**	**Explanation of Group Function**	**Example**
1	AVG()	Returns average value of specified column	SELECT AVG(salary) FROM emp;
2	SUM()	Returns summation of specified column	SELECT SUM(salary) FROM emp;
3	MIN()	Returns lowest value of specified column	SELECT MIN(salary) FROM emp;
4	MAX()	Returns highest value of specified column	SELECT MAX(salary) FROM emp;
5	COUNT (*)	Counts all rows including duplicates and nulls	SELECT COUNT(*) FROM emp;
6	COUNT (column_name)	Counts number of values in specified column excluding	SELECT COUNT(comm) FROM emp;

		nulls	
7	COUNT (DISTINCT column_name)	Counts distinct non-null values in specified column	SELECT COUNT(DISTINCT deptno) FROM emp;

VII. Required Resources/apparatus/equipment with specifications

Sr.No	Equipment Name with Broad Specifications	Relevant LLO Number
1	Computer system with all necessary components like; motherboard, random access memory (RAM), read-only memory (ROM), internal hard disk drives, Mouse, Keyboard, and RDBMS applications such as Oracle Express Edition,MySql,SQLite,Oracle Live SQL etc.	All

VIII. Procedure

1. Create tables for the given application
2. Write SQL queries using aggregate functions

IX. Result(s)

……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………

X. Practical related questions (Provide space for answers)

Note: Below are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. Describe group functions or aggregate functions in SQL
2. Differentiate between count(*) and count(columname)

(Space for answer)

……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

XI. Exercise

Consider the following schema

Emp(empno,ename,job,mgr,hiredate,sal,comm,deptno)

1. Display the minimum, maximum, sum and average salary of all employees. Label the columns Maximum, Minimum, Sum and Average respectively.
2. Determine the number of managers without listing them. Label the column number of managers
3. Write a query that will display the difference between the highest and lowest salaries. Label the column DIFFERENCE.
4. Display the number of employees in department 10 who earns a commission

XII. References/Suggestions for further reading: include websites/links

1. https://www.youtube.com/watch?v=yGU4YfSSjdM
2. https://beginner-sql-tutorial.com/sql-group-functions.htm
3. https://www.tutorialspoint.com/sql_certificate/using_the_group_functions.htm

XIII. Assessment Scheme

Performance indicators		Weightage
Sr. No	**Process related (10 Marks)**	**30%**
1.	Tool Selection Ability	20%
2.	Follow Ethical Practices	10%
Product related (15 Marks)		**70%**
3.	Verifying System Requirement for Installation	20%
4.	Correctness in Use of Command Tools	15%
5.	Use of Attribute/s Options	15%
6.	Timely Submission	10%
7.	Answering Sample Questions	10%
Total (50 Marks)		**100%**

List of student Team Members

1…………………………………

2………………………………....

3…………………………………

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (35)**	**Total (50)**	

Experiment No 15: Execute queries for Ordering and Grouping data.

I. Practical Significance:

This practical will help students to understand different clauses used in SQL. The clauses are used to retrieve the information from the table. SQL clause helps us to retrieve a set or bundles of records from the table. SQL clause helps us to specify a condition on the columns or the records of a table. Different clauses available in the Structured Query Language are as follows: WHERE CLAUSE, GROUP BY CLAUSE, HAVING CLAUSE, ORDER BY CLAUSE.

II. INDUSTRY / EMPLOYER EXPECTED OUTCOME:

To execute queries using different clauses to perform advance calculations.

III. COURSE LEVEL LEARNING OUTCOMES (COS):

CO3 - Manage database using SQL.

IV. LABORATORY LEARNING OUTCOME:

Execute queries using different clauses.

V. Relevant Affective Domain related outcome(s)

a. Follow precautionary measures.

b. Follow installation steps.

c. Follow ethical practices.

VI. Relevant Theoretical Background

Clauses	Explanation	Syntax

WHERE	The WHERE clause is used to filter records. It is used to extract only those records that fulfill a specified condition.	SELECT column1, column2, ...
		FROM table_name
		WHERE condition;

HAVING	The HAVING clause was added to SQL because the WHERE keyword cannot be used with aggregate functions.	SELECT column_name(s)
		FROM table_name
		WHERE condition
		GROUP BY column_name(s)
		HAVING condition
		ORDER BY column_name(s);

GROUP BY	The GROUP BY statement groups rows that have the same values into summary rows. The GROUP BY statement is often used with aggregate functions (COUNT(), MAX(), MIN(), SUM(), AVG()) to group the result-set by one or more columns.	SELECT column_name(s)
		FROM table_name
		WHERE condition
		GROUP BY column_name(s)
		ORDER BY column_name(s);

ORDER BY	The ORDER BY keyword is used to sort the result-set in ascending or descending order.	SELECT column1, column2, ...
		FROM table_name
		ORDER BY column1, column2, ... ASC|DESC;

VII. Required Resources/apparatus/equipment with specifications

Sr. No	Equipment Name with Broad Specifications	Relevant LLO Number
1	Computer system with all necessary components like; motherboard, random access memory (RAM), read-only memory (ROM), internal hard disk drives, Mouse, Keyboard, and RDBMS applications such as Oracle Express Edition, MySQL, SQLite, Oracle Live SQL etc.	All

VIII. Procedure

Execute SQL queries using different WHERE, HAVING, GROUPBY, ORDER BY clause.

IX. Result(s)

…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………

X. Practical related questions (Provide space for answers)

Note: Below are a few sample questions for reference. Teacher must design more suchquestions so as to ensure the achievement of identified CO.

Create Two Tables Emp and Dept, Consider the following schema

Emp (Emp_no as primary key, E_name, Dept_no, Dept_name, name, Job_id, Salary) and Dept (Dept_no as primary key, emp_no foreign key, deptname, location)

1. Display the information of the tables using SELECT command.
2. Execute the SQL queries using SELECT, WHERE, GROUP BY clause.
3. Execute the SQL queries using SELECT, WHERE, GROUP BY and HAVING clause.
4. Execute the SQL queries using SELECT, WHERE, ORDER BY clause.
5. Write the advantages of using the Order By clause.

(Space for answer)

…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………

……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……

XI. Exercise:

1. Write output of the following queries.

a. Display minimum salary of employee from every department;
b. Display total salary of every department.
c. Display the department having total employees more than 5.
d. Display details of employees with employee name in ascending order.
a. Display emp_no, dept_no from dept Group By deptname.

XII. References/Suggestions for further reading: include websites/links

1. https://www.oreilly.com/library/view/oracle-plsql-programming/0596003811/ch09s03.html
2. https://beginner-sql-tutorial.com/oracle-functions.htm

3. https://www.javatpoint.com
4. www.tutorialsteacher.com
5. www.w3schools.com

XIII. Assessment Scheme

Performance indicators		Weightage
Sr. No	**Process related (10 Marks)**	**30%**
1.	Tool Selection Ability	20%
2.	Follow Ethical Practices	10%
Product related (15 Marks)		**70%**
3.	Verifying System Requirement for Installation	20%
4.	Correctness in Use of Command Tools	15%
5.	Use of Attribute/s Options	15%
6.	Timely Submission	10%
7.	Answering Sample Questions	10%
Total (50 Marks)		**100%**

List of student Team Members

1…………………………………

2………………………………....

3…………………………………

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (35)**	**Total (50)**	

Experiment No 16: Implement SQL queries for Inner and Outer Join

I. Practical Significance:

Join is used to combine the data spread across tables. A join is performed by the 'where' clause which combines the specified rows of tables. This practical allows students to join two or more tables.

II. INDUSTRY / EMPLOYER EXPECTED OUTCOME:

To efficiently join data from multiple tables

III. COURSE LEVEL LEARNIN OUTCOMES (COS):

CO3 - Manage database using SQL.

IV. LABORATORY LEARNING OUTCOME:

Execute the queries based on Inner & Outer join

V. Relevant Affective Domain related outcome(s)

a. Follow precautionary measures.
b. Follow installation steps.
c. Follow ethical practices.

VI. Relevant Theoretical Background

Joins :

Serial No	Join Type	Explanation	Syntax	Example	Output

1	Equi-Join	A join based on equalities using the = operator. It retrieves rows with matching	SELECT columns FROM table1, table2
			WHERE table1.column1 = table2.column2;
				SELECT emp. empno, emp.ename, emp.deptno, dept.deptno, dept.loc FROM emp, dept	Rows with matching deptno values from both emp and dept tables.

		values in both tables.		WHERE emp.deptno = dept.deptno;	

2	Non Equi-Join	A join using relational operators other than = (e.g., <, >, <=, >=, !=).	SELECT columns FROM table1, table2 WHERE table1.column1 < table2.column2;	SELECT e. ename, e. salary, s. grade FROM
				emp e, salgrade s WHERE
				e.salary
				BETWEEN s.losal AND s.hisal;
					Rows from emp and salgrade where emp.salary falls between salgrade.losal and salgrade.hisal.

3	Self Join	A join where a table is joined with itself. It compares rows within the same table.	SELECT a.column1, b.column2
			FROM table a, table b WHERE a.column = b.column;
				SELECT worker.ename "employee", manager.ename "manager" FROM emp worker,
				emp manager WHERE worker.mgr = manager.empno;
					Rows where each worker's manager is also listed in the emp table.

4	Left Outer Join	Returns all records from the left table and matched records from the right table. Returns NULL for non-matching rows from the right table.	SELECT columns FROM
			table1
			LEFT OUTER JOIN
			table2
			ON table1.column = table2.column;
				SELECT EMP.emp_id, EMP.name, DEPT.dept_name FROM EMP
				LEFT OUTER JOIN
				DEPT ON EMP.dept_id = DEPT.dept_id;
					All records from EMP table and matching records from DEPT. Non-matching DEPT rows are NULL.

5	Right Outer	Returns all records from	SELECT columns FROM	SELECT EMP.emp_id,	All records from DEPT

	Join	the right table and matched records from the left table. Returns NULL for non-matching rows from the left table.	table1 RIGHT OUTER JOIN table2 ON table1.column = table2.column;	EMP.name, DEPT.dept_name FROM EMP RIGHT OUTER JOIN DEPT ON EMP.dept_id = DEPT.dept_id;	table and matching records from EMP. Non-matching EMP rows are NULL.

Note:

Guidelines for Equi-Join:

1. Rows in one table can be joined to rows in another table according to the common values existing in corresponding columns, that are usually primary and foreign key columns.
2. When writing a select statement that joins tables, precede the column name with the table name. (e.g. dept.deptno)
3. If the same column name appears in more than one table, the column name must be prefixed to the table name. (e.g. dept.deptno, emp.deptno)
4. If the column names are unique, then we need not prefix it with the table name.

Table aliases

Table aliases are used to make multiple tables queries shorter and more readable. As a result, we give an alias name or short name to the table in the 'from' clause. The alias can be used instead of the table name throughout the query.

OR

Following are the types of joins:

- Equi - join
- Non equi-join
- Self-Join
- Outer Join

Equi -join

A join, which is based on equalities, is called equi-join. In equi-join comparison operator equal to (=) is used to perform a join. It retrieves rows from tables having a common column. It is also called simple join.

Syntax: select table1.column, table1.column,table2.column,….,

```
        from table1, table2
        where table1.column1 = table2.column2;
```

Example:

```
Select emp.empno,emp. ename,emp.deptno,
        dept.deptno,dept.loc
from emp, dept
where  emp.deptno = dept. deptno;
```

Guidelines:

1. Rows in one table can be joined to rows in another table according to the common values existing in corresponding columns, that are usually primary and foreign key columns.
2. When writing a select statement that joins tables, precede the column name with the table name. (e.g. dept.deptno)
3. If the same column name appears in more than one table, the column name must be prefixed to the table name. (e.g. dept.deptno, emp.deptno)
4. If the column names are unique, then we need not prefix it with the table name.

```
        select e. empno, e.ename, e.deptno,
d.deptno, d.loc
from emp e, dept d
where e.deptno = d.deptno;
```

Note: - The above example is same as example of equi join but uses table aliases where 'e' refers to emp table and 'd' refers to dept table.

Non equi-join

A join that specifies the relationship between columns belonging to different tables by making use of the relational operators (<, >, <=, >=, ! =) other than '=' operator is called as non equi-join.

To use non equi-join create the following table.

Table - Salgrade

Column-name	Datatype
Grade	Number (4)
Losal	Number (8)
Hisal	Number (8)

[**Note** - Students will insert the following records in a given table.]

GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

To relate emp and salgrade tables using non-equi join.

Example:

```
    select e.ename, e.salary, s.grade
    from emp e, salgrade s
where e.salary
    between s.losal and s.hisal;
```

<u>OR</u>

```
    select e.ename, e.salary, s.grade
from emp e, salgrade s
    where e.salary >= s.losal and
            e.salary <= s.hisal;
```

Self-join

Joining a table to itself is known as self-join. i.e. it joins one row in a table to another. It can compare each row of the table to itself and with other rows of the same table.

Example: To find the name of each employee's manager you need to join EMP table to itself.

```
        select worker.ename "employee", manager.ename "manager"
                from emp worker, emp manager
    where worker.mgr = manager.empno;
```

Note:

The above example joins the emp table to itself. To stimulate two tables in the FROM clause, there are two aliases, namely WORKER and MANAGER, for the same table, EMP.

Outer Join

An outer join returns all the rows returned by simple join or equi join as well as those rows from one table that *do not match any row from the other table*.

Consider following tables for outer join:

EMP Table

emp_id	**ename**	**dept_id**
1	Riya	1
2	Pranav	2
3	Mansi	NULL

DEPT Table

dept_id	**dept_name**
1	HR
2	IT
3	Finance

1. LEFT OUTER JOIN

Description: The LEFT OUTER JOIN returns all records from the left table (EMP), and the matched records from the right table (DEPT). The *result is NULL* from the right side, *if there is no match.*

Syntax:

```
SELECT columns
FROM EMP
LEFT OUTER JOIN DEPT
ON EMP.dept_id = DEPT.dept_id.
```

Example:

```
SELECT EMP.emp_id, EMP.name, DEPT.dept_name
FROM EMP
LEFT OUTER JOIN DEPT
ON EMP.dept_id = DEPT.dept_id;
```

Output:

emp_id	**name**	**dept_name**
1	Riya	HR
2	Pranav	IT
3	Mansi	NULL

2. RIGHT OUTER JOIN

Description: The RIGHT OUTER JOIN returns all records from the right table (DEPT), and the matched records from the left table (EMP). The result is NULL from the left side, when there is no match.

Syntax:

```
SELECT columns
FROM EMP
RIGHT OUTER JOIN DEPT
ON EMP.dept_id = DEPT.dept_id;
```

Example:

```
SELECT EMP.emp_id, EMP.name, DEPT.dept_name
FROM EMP
RIGHT OUTER JOIN DEPT
ON EMP.dept_id = DEPT.dept_id;
```

Output:

emp_id	**name**	**dept_name**
1	Riya	HR
2	Pranav	IT
NULL	NULL	Finance

3. FULL OUTER JOIN

Description: The FULL OUTER JOIN returns all records when there is a match in either left (EMP) or right (DEPT) table records. It *returns NULL for records* that *do not have a match in the other table.*

Syntax:

```
SELECT columns
```

```
FROM EMP
FULL OUTER JOIN DEPT
ON EMP.dept_id = DEPT.dept_id;
```

Example:

```
SELECT EMP.emp_id, EMP.name, DEPT.dept_name
FROM EMP
FULL OUTER JOIN DEPT
ON EMP.dept_id = DEPT.dept_id;
```

Output:

emp_id	**name**	**dept_name**
1	Riya	HR
2	Pranav	IT
3	Mansi	NULL
NULL	NULL	Finance

VII. Required Resources/apparatus/equipment with specifications

Sr.No	**Equipment Name with Broad Specifications**	**Relevant LLO Number**
1	Computer system with all necessary components like; motherboard, random access memory (RAM), read-only memory (ROM), internal hard disk drives, Mouse, Keyboard, and RDBMS applications such as Oracle Express Edition,MySql,SQLite,Oracle Live SQL etc.	All

VIII. Procedure

1. Create tables for given application
2. Assign Primary key for created table
3. Join two or more tables

IX. Result(s)

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

X. Practical related questions (Provide space for answers)

Note: Below are a few sample questions for reference. Teacher must design more such questions to ensure the achievement of identified CO.

1. Define Join. List types of Joins.
2. Describe outer join.

(Space for answer)

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

XI. Exercise

1. Display employee Nikhil’s employee number, name, department number, and department location.

2. Display the list of employees who work in the sales department.
3. Display the list of employees who do not work in the sales department.
4. Display the employee names and salary of all employees who report to Sumit Patil.

XII. References/Suggestions for further reading: include websites/links

1. https://www.youtube.com/watch?v=yGU4YfSSjdM
2. https://www.w3schools.com/sql/sql_join_full.asp
3. https://www.javatpoint.com/sql-server-joins

XIII. Assessment Scheme

Performance indicators		Weightage
Sr. No	**Process related (10 Marks)**	**30%**
1.	Tool Selection Ability	20%
2.	Follow Ethical Practices	10%
	Product related (15 Marks)	**70%**
3.	Verifying System Requirement for Installation	20%
4.	Correctness in Use of Command Tools	15%
5.	Use of Attribute/s Options	15%
6.	Timely Submission	10%
7.	Answering Sample Questions	10%
	Total (50 Marks)	**100%**

List of student Team Members

1…………………………………
2………………………………....
3…………………………………

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (35)**	**Total (50)**	

Experiment No 17: Create and manage Views for faster access on relations.

I. Practical Significance:

This practical will help students to understand Views, Sequences and Indexes used in SQL. A view is a virtual table based on the result-set of an SQL statement. A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database. You can add SQL statements and functions to a view and present the data as if the data were coming from one single table. A sequence is a user-defined schema-bound object that generates a series of numeric values. Indexes are used to retrieve data from the database more quickly than otherwise. The users cannot see the indexes, they are just used to speed up searches/queries.

II. INDUSTRY / EMPLOYER EXPECTED OUTCOME:

To execute queries using Views, Sequences, Indexes to perform advance calculations.

III. COURSE LEVEL LEARNING OUTCOMES (COS):CO3

- Manage database using SQL.

IV. LABORATORY LEARNING OUTCOME:

i. Create, Update, drop/ Delete views.

ii. Create and execute Sequences.

iii. Create and execute indexes.

V. Relevant Affective Domain related outcome(s)

a. Follow precautionary measures.

b. Follow installation steps.

c. Follow ethical practices.

VI. Relevant Theoretical Background

a. Views:

A view is created with the CREATE VIEW statement. View can hide complexity and can be used as a security mechanism. View can be a virtual table which is derived from one or more than one table. View is created using tables of same database or different database. It is used for security purposes because they provide encapsulation of the name of the table. View has several benefits:

1. Complexity: Views help to reduce complexity. Different views can be created on the

same base table for different users.

2. Security: It increases security by excluding the sensitive information from the view.
3. Query Simplicity: It helps to simplify commands from the user. A view can draw data from several different tables and present it as a single table.
4. Consistency: A view can present a consistent, unchanged image of the structure of the database. Views can be used to rename the columns without affecting the base table.
5. Data Integrity: If data is accessed and entered through a view, the DBMS can automatically check the data to ensure that it meets the specified integrity constraints.
6. Storage Capacity: Views take very little space to store the data.
7. Logical Data Independence: View can make the application and database tables to a certain extent independent.

Syntax

```
                    CREATE VIEW view_name AS
                     SELECT column1, column2, ...
                     FROM table_name
WHERE condition;
```

b. SEQUENCE

The sequence of numeric values is generated in an ascending or descending order at defined intervals and can be configured to restart when it exceeds maximum value. A sequence is a set of integers that are generated in order of demand. Sequences are commonly used in databases because many applications require each row in a table to contain a unique value and sequence provides an easy way to generate them.

SYNTAX:

```
         CREATE SEQUENCE sequence_name
          START WITH initial_value
          INCREMENT BY increment_value MINVALUE minimum value
          MAXVALUE maximum value
        CYCLE|NOCYCLE;
```

c. INDEX

The Index in SQL is a special table used to speed up the searching of the data in the database tables. Index provides a fast access path to column that is indexed. Indexes are stored independently from actual data. It is mostly useful on large tables and on columns that frequently appear in WHERE clause. When the table is dropped, index will also automatically drop. More than one index is allowed in one table.

SYNTAX:

CREATE INDEX Index_Name ON Table_Name (Column_Name);

VII. Required Resources/apparatus/equipment with specifications

Sr. No	Equipment Name with Broad Specifications	Relevant LLO Number
1	Computer system with all necessary components like; motherboard, random access memory (RAM), read-only memory (ROM), internal hard disk drives, Mouse, Keyboard, and RDBMS applications such as Oracle Express Edition, MySQL, SQLite, Oracle Live SQL etc.	All

VIII. Procedure

a. Write and execute query for view.

b. Write and execute query to insert, modify and delete records through views

c. Write and execute query to delete view.

d. Write and execute query for creating altering and dropping sequence.

e. Write and execute query for simple and composite index.

IX. Result(s)

…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………

X. Practical related questions (Provide space for answers)

Note: Below are a few sample questions for reference. Teacher must design more suchquestions so as to ensure the achievement of identified CO.

a. What are synonyms, write its syntax and advantages.

b. What is the difference between simple and composite index?

c. What are the disadvantages of views?

d. Write the syntax to delete view.

(Space for answer)

…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………

…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
………………………………………………………………………………………………

…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………

XI. Exercise:

1. Write output of the following queries.

i. Create view emp_view as select emp_no, enema, salary from emp;
ii. Update emp_view set e_name='Jay' where emp_no=101;
iii. Delete from emp_view where emp_no= 105;
iv. Drop view emp_view;
v. Modify location of dept_no of dept_view;
vi. Write output of following queries.
vii. Create simple index dept_simple-index on dept table.
viii. Create composite index dept_composite_index on dept table.
ix. Drop index dept_simple_index and dept_composite_index.
x. Create index raj on emp (empno, ename).

2 Write output of following queries.

a. Create sequence emp_sequence
Incremented by 2
Start with 1
Nomaxvalue
Nocycle
Cache 10;

b. Alter sequence emp_sequence
Incremented by 15
Max value 1000
Cycle
Cache20;

c. Drop sequence emp_sequence;

XII. References/Suggestions for further reading: include websites/links

1. https://www.oreilly.com/library/view/oracle-plsql-programming/0596003811/ch09s03.html
2. https://beginner-sql-tutorial.com/oracle-functions.htm
3. https://www.javatpoint.com
4. www.tutorialsteacher.com

XIII. Assessment Scheme

Performance indicators		Weightage
Sr. No	**Process related (10 Marks)**	**30%**
1.	Tool Selection Ability	20%
2.	Follow Ethical Practices	10%
	Product related (15 Marks)	**70%**
3.	Verifying System Requirement for Installation	20%
4.	Correctness in Use of Command Tools	15%
5.	Use of Attribute/s Options	15%
6.	Timely Submission	10%
7.	Answering Sample Questions	10%
	Total (50 Marks)	**100%**

List of student Team Members

1…………………………………

2………………………………....

3…………………………………

Marks Obtained		
Process Related (15)	**Product Related (35)**	**Total (50)**

Experiment No 18: Implement PL/SQL program using Conditional Statements

I. **Practical Significance**: Conditional statements in PL/SQL are like decision-makers in PL/SQL code. They help to make choices based on different conditions. For example, if something is true, do one thing, if not, do something else. This flexibility makes PL/SQL code more powerful and capable of handling different situations. This practical allows students to write a PL/SQL program using Conditional Statements- if, if then else, nested if, if elsif else.

II. **INDUSTRY / EMPLOYER EXPECTED OUTCOME:**

To implement conditional statements in PL/SQL to make better decisions and to improve problem-solving skills.

III. **COURSE LEVEL LEARNING OUTCOMES (COS)**:CO4
- Implement PL/SQL codes for given application.

IV. **LABORATORY LEARNING OUTCOME:**

Implement PL/SQL program using Conditional Statements.

V. **Relevant Affective Domain related outcome(s)**

a. Follow precautionary measures.

b. Follow installation steps.

c. Follow ethical practices.

VI. **Relevant Theoretical Background**

Sr. No.	Title	Explanation	Syntax	Example Program

1	If	Checks a condition and executes a block of code if it's true.	IF condition THEN statement; END IF;	DECLARE
				x NUMBER := 10;
				BEGIN IF x > 5 THEN DBMS_OUTPUT.PUT_LINE ('x is greater than 5');
				END IF;
				END;

2	If-Then-Else	Executes one block of code if a condition	IF condition THEN	DECLARE x NUMBER := 3;

		is true and another if it's false.	statement; ELSE statement2; END IF;	BEGIN
				IF x > 5 THEN DBMS_OUTPUT.PUT_LINE('x is greater than 5');
				ELSE DBMS_OUTPUT.PUT_LINE('x is not greater than 5'); END IF;
				END;

3	If-Elsif	Allows checking multiple conditions in sequence and executing corresponding blocks of code.	IF condition1 THEN statement1; ELSIF condition2 THEN statement2; ELSE statement3; END IF;	DECLARE
				x NUMBER:= 3;
				BEGIN
				IF x > 5 THEN DBMS_OUTPUT.PUT_LINE('x is greater than 5');
				ELSIF x = 5 THEN DBMS_OUTPUT.PUT_LINE('x is equal to 5');
				ELSE DBMS_OUTPUT.PUT_LINE('x is less than 5');
				END IF;
				END;

4	Nested If	Uses one or more if statements inside another if statement.	IF condition1 THEN IF condition2 THEN statement;
			END IF; END IF;
				DECLARE
				x NUMBER := 3;
				y NUMBER := 10;
				BEGIN IF x > 5 THEN
				IF y > 5 THEN DBMS_OUTPUT.PUT_LINE('Both x and y are greater than 5');
				END IF;

				END IF;
				END;

VII. Required Resources/apparatus/equipment with specifications

Sr.No	Equipment Name with Broad Specifications	Relevant LLO Number
1	Computer system with all necessary components like; motherboard, random access memory (RAM), read-only memory (ROM), internal hard disk drives, Mouse, Keyboard, and RDBMS applications such as Oracle Express Edition,MySql,SQLite,Oracle Apex etc.	All

VIII. Procedure

Implement PL/SQL program based on the given problem

IX. Result(s)

……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……

Practical related questions (Provide space for answers)

Note: Below are a few sample questions for reference. Teacher must design more suchquestions to ensure the achievement of identified CO.

1. List conditional statement in PL/SQL.
2. Describe any three conditional statements in PL/SQL

(Space for answer)

……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

X. Exercise

1. Write a PL/SQL program that checks if a given number is positive, and if it is, prints "Number is positive".
2. Write a PL/SQL program that asks the user for their age and then prints “You can vote" if they are over 18, and "You cannot vote" otherwise.
3. Write a PL/SQL program that asks the user for percentage and then assigns grades based on the following conditions:

   Distinction (>=75%)

   First Class (>=60 and <75)

   Second Class (>=45 and <60)

   Pass Class (>=40 and <45)

   Fail (<40)

XI. References/Suggestions for further reading: include websites/links

1. https://www.youtube.com/watch?v=yGU4YfSSjdM

2. https://www.tutorialspoint.com/plsql/plsql_conditional_control.htm
3. https://www.oreilly.com/library/view/oracle-plsql-programming/9781449324445/ch04.html

XII. Assessment Scheme

Performance indicators		Weightage
Sr. No	**Process related (10 Marks)**	**30%**
1.	Tool Selection Ability	20%
2.	Follow Ethical Practices	10%
	Product related (15 Marks)	**70%**
3.	Verifying System Requirement for Installation	20%
4.	Correctness in Use of Command Tools	15%
5.	Use of Attribute/s Options	15%
6.	Timely Submission	10%
7.	Answering Sample Questions	10%
	Total (50 Marks)	**100%**

List of student Team Members

1……………………………………

2…………………………………....

3……………………………………

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (35)**	**Total (50)**	

Experiment No 19: Implement PL/SQL program using Iterative Statements

I. Practical Significance:

The PL/SQL loops are used to repeat the execution of one or more statements for a specified number of times. These are also known as iterative control statements. The iterative statement can be embedded in a PL/SQL procedure, function, or anonymous block statement. In PL/SQL we have three different loop options to choose from when we want to execute a statement repeatedly in our code block. They are:

a. Basic loop.

b. For loop

c. While loop

II. INDUSTRY / EMPLOYER EXPECTED OUTCOME:

To implement conditional statements in PL/SQL to make better decisions and to improve problem-solving skills.

III. COURSE LEVEL LEARNING OUTCOMES (COS):CO4

- Implement PL/SQL codes for given application.

IV. LABORATORY LEARNING OUTCOME:

Implement PL/SQL program using Iterative Statements.

V. Relevant Affective Domain related outcome(s)

a. Follow precautionary measures.

b. Follow installation steps.

c. Follow ethical practices.

VI. Relevant Theoretical Background

Title	Explanation	Syntax	Example Program

Basic loop	Basic loop or simple loop is preferred in PL/SQL code when there is no surety about how many times the block of code is to be repeated.	Loop
		sequence of statements
		end loop;
			set serveroutput on;
			DECLARE i int;
			BEGIN i := 1;
			LOOP
			if i>10 then
			exit;
			end if;
			dbms_output.put_line(i);
			i := i+1;
			END LOOP;
			END;

For	This loop is used when some statements in PL/SQL code block are to be repeated for a fixed number of times.	FOR counter_variable
		IN start_value..end_value
		LOOP
		statement to be executed
		END LOOP;
			set serveroutput on;
			DECLARE
			i number(2);
			BEGIN
			FOR i IN 1..10 LOOP
			dbms_output.put_line(i);
			END LOOP;
			END;

while	It is an entry-controlled loop which means that before entering in a while loop first the condition is tested,	WHILE <test_condition>
		LOOP
		<action>
			set serveroutput on;
			DECLARE num int: =1;
			BEGIN

	if the condition is TRUE the statement or a group of statements get executed and if the condition is FALSE the control will move out of the while loop.	END LOOP;	while (num <= 10) LOOP
			dbms_output.put_ line(''|| no);
			num: = num+2;
			END LOOP;
			END;

VII. Required Resources/apparatus/equipment with specifications

Sr.No	Equipment Name with Broad Specifications	Relevant LLO Number
1	Computer system with all necessary components like; motherboard, random access memory (RAM), read-only memory (ROM), internal hard disk drives, Mouse, Keyboard, and RDBMS applications such as Oracle Express Edition,MySql,SQLite,Oracle Apex etc.	All

VIII. Procedure

Implement PL/SQL program based on the given problem

IX. Result(s)

…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………

X. Practical related questions (Provide space for answers)

Note: Below are a few sample questions for reference. Teacher must design more suchquestions to ensure the achievement of identified CO.

1. List Iterative statement in PL/SQL.
2. Write PL/SQL program using any one Iterative statement.

(Space for answer)

…………………………………………………………………………………………………
…………………………………………………………………………………………………

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

XI. Exercise

a. Write a PL/SQL program to display multiplication table of 5 using FOR loop.

b. Write a PL/SQL program to calculate factorial of 10 by using PL/SQL WHILE LOOP statement.

c. Write a PL/SQL program to calculate the prime numbers between 1 to 50.

XII. References/Suggestions for further reading: include websites/links

1. https://www.youtube.com/watch?v=yGU4YfSSjdM

2. https://www.tutorialspoint.com/plsql/plsql
3. https://www.oreilly.com/library/view/oracle-plsql-programming/9781449324445/ch04.html
4. PL/SQL WHILE Loop Tutorial (plsqltutorial.com)
5. https://www.ibm.com/pl/sql

XIII. Assessment Scheme

Performance indicators		Weightage
Sr. No	**Process related (10 Marks)**	**30%**
1.	Tool Selection Ability	20%
2.	Follow Ethical Practices	10%
Product related (15 Marks)		**70%**
3.	Verifying System Requirement for Installation	20%
4.	Correctness in Use of Command Tools	15%
5.	Use of Attribute/s Options	15%
6.	Timely Submission	10%
7.	Answering Sample Questions	10%
Total (50 Marks)		**100%**

List of student Team Members

1……………………………………

2……………………………….....

3……………………………………

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (35)**	**Total (50)**	

Experiment No 20: Implement PL/SQL program using Sequential Control

I. **Practical Significance**: Sequential statements are used in PL/SQL help to control the order of execution in PL/SQL code. The Case statement checks different conditions and executes specific code based on those conditions. Goto jumps to a specific part of PL/SQL program, continue skips to the next iteration of a loop. This practical allows students to implement PL/SQL program using sequential control.

II. INDUSTRY / EMPLOYER EXPECTED OUTCOME:

To implement PL/SQL program with simple code structure using Sequential Control

III. COURSE LEVEL LEARNING OUTCOMES (COS):

CO4 - Implement PL/SQL codes for given application.

IV. LABORATORY LEARNING OUTCOME:

Implement PL/SQL program using Sequential Control.

V. Relevant Affective Domain related outcome(s)

a. Follow precautionary measures.

b. Follow installation steps.

c. Follow ethical practices.

VI. Relevant Theoretical Background

a. Case Statement

The PL/SQL CASE statement runs a series of actions depending on a selector, which can be a variable, function, or expression. It functions similarly to an IF statement but uses the keyword WHEN instead. The CASE statement checks conditions one by one from top to bottom, executing the corresponding action when it finds a true condition, then moving to the END CASE clause**.**

Syntax:

```
CASE selector
WHEN selector_value_1 THEN
   statements_1
WHEN selector_value_1 THEN
   statement_2
...
ELSE
```

```
    else statements
          END; CASE;
```

Example:

```
DECLARE
   grade char(1) := 'A';
BEGIN
   CASE grade
      when 'A' then dbms_output.put_line('Excellent');
      when 'B' then dbms_output.put_line('Very good');
      when 'C' then dbms_output.put_line('Good');
      when 'D' then dbms_output.put_line('Average');
      when 'F' then dbms_output.put_line('Passed with Grace');
      else dbms_output.put_line('Failed');
   END CASE;
END;
```

2. **CONTINUE**

Explanation:

In PL/SQL, the CONTINUE statement allows you to skip the rest of the current iteration of a loop and start the next iteration

Syntax:

```
CONTINUE;
```

Example:

```
DECLARE
    counter NUMBER := 1;
BEGIN
    WHILE counter <= 5 LOOP
        IF counter = 3 THEN
            CONTINUE; -- Skip the rest of the loop for counter = 3
        END IF;
        DBMS_OUTPUT.PUT_LINE('Counter: ' || counter);
        counter := counter + 1;
```

```
   END LOOP;
END;
```

In this example, the loop will print numbers from 1 to 5, but it will skip printing the number 3 due to the CONTINUE statement.

3. **GOTO**

In PL/SQL, the GOTO statement transfers control to a labeled statement within the same block or subprogram.

Syntax:

```
GOTO label_name;
```

Example:

```
DECLARE
   x NUMBER := 1;
BEGIN
   <<start>>
   DBMS_OUTPUT.PUT_LINE('Start');
   GOTO end;
   <<middle>>
   DBMS_OUTPUT.PUT_LINE('Middle'); -- This line will be skipped
   <<end>>
   DBMS_OUTPUT.PUT_LINE('End');
END;
```

In this example, the program *jumps directly from the start label to the end label*, skipping the middle label and its associated statement.

VII. Required Resources/apparatus/equipment with specifications

Sr.No	Equipment Name with Broad Specifications	Relevant LLO Number
1	Computer system with all necessary components like; motherboard, random access memory (RAM), read-only memory (ROM), internal hard disk drives, Mouse, Keyboard, and RDBMS applications such as Oracle Express Edition,MySql,SQLite,Oracle Apex etc.	All

VIII. Procedure

1.Define the PL/SQL block structure

2.Write the control statements

3.Implement the logic for the given problem

IX. Result(s)

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

X. Practical related questions (Provide space for answers)

Note: Below are a few sample questions for reference. Teacher must design more such questions to ensure the achievement of identified CO.

1. List sequential statements in PL/SQL
2. Describe any two sequential statements in PL/SQL

(Space for answer)

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

XI. Exercise

1. Write a PL/SQL program that calculates the sum of all even numbers between 1 and 10, skipping odd numbers [Use CONTINUE statement]
2. Write a PL/SQL program that prints numbers from 1 to 10, but skips printing the number 5 [Use GOTO statement]
3. Write a PL/SQL program that asks the user to input a number (1 for Monday, 2 for Tuesday, ..., 7 for Sunday) and prints the corresponding day of the week.

XII. References/Suggestions for further reading: include websites/links

1. https://www.youtube.com/watch?v=yGU4YfSSjdM
2. https://www.plsql.co/plsql-sequential-control.html
3. https://www.oreilly.com/library/view/oracle-plsql-programming/0596009771/ch04.htm

XIII. Assessment Scheme

Performance indicators		Weightage
Sr. No	**Process related (10 Marks)**	**30%**
1.	Tool Selection Ability	20%
2.	Follow Ethical Practices	10%
Product related (15 Marks)		**70%**
3.	Verifying System Requirement for Installation	20%
4.	Correctness in Use of Command Tools	15%
5.	Use of Attribute/s Options	15%
6.	Timely Submission	10%
7.	Answering Sample Questions	10%
Total (50 Marks)		**100%**

List of student Team Members

1…………………………………

2……………………………….....

3…………………………………

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (35)**	**Total (50)**	

Experiment No 21: Create Implicit and Explicit Cursors.

I. Practical Significance:

A cursor is a pointer to this context area. PL/SQL controls the context area through a cursor. A cursor holds the rows (one or more) returned by a SQL statement. The set of rows the cursor holds is referred to as the active set. You can name a cursor so that it could be referred to in a program to fetch and process the rows returned by the SQL statement, one at a time. There are two types of cursors namely: Implicit cursors, Explicit cursors.

II. INDUSTRY / EMPLOYER EXPECTED OUTCOME:

To implement PL/SQL programs to implement cursors.

III. COURSE LEVEL LEARNING OUTCOMES (COS):

CO4 - Implement PL/SQL codes for given application.

IV. LABORATORY LEARNING OUTCOME:

Implement PL/SQL program based on cursors.

V. Relevant Affective Domain related outcome(s)

a. Follow precautionary measures.

b. Follow installation steps.

c. Follow ethical practices.

VI. Relevant Theoretical Background:

Cursors provide a way to iterate through the rows returned by a query, one at a time, and perform various operations on the data. With cursors, we can access records individually, control them as needed, or display them on the console accordingly.

The PL/SQL programming language supports two kinds of cursors.

Implicit Cursor

Explicit Cursor

I. **Implicit Cursors**

An implicit cursor is generated by Oracle when DML statements such as INSERT, UPDATE, and DELETE are executed. Oracle offers some attributes through which programmers can perform operations on these cursors. Implicit cursors help to handle database transactions without requiring to manually create cursors every time. The attributes offered by Oracle are as follows:

a. **%FOUND:** This function returns a Boolean value of True,if INSERT, UPDATE, and DELETE, affect single or multiple rows. Similarly, it returns True if any SELECT statement also returns single or multiple rows. If neither of these conditions are met, the function will return False.

b. **%NOTFOUND:** This attribute operates in reverse to its counterpart, %FOUND. If a DML statement has not affected any rows or a SELECT statement does not return any results, then the %NOTFOUND attribute will be evaluated as True. Alternatively, it will return False when rows have been impacted by a DML statement or when the SELECT DML statement returns at least one result.

c. **%ISOPEN:** Regarding Implicit cursors, Oracle shut off the cursor immediately following the SQL statement's execution, resulting in a False return value.

d. **%ROWCOUNT:** The function returns the rows affected by DML statements such as INSERT, UPDATE, and DELETE. It can also provide the count of rows returned from SELECT INTO statements in PL/SQL code.

II. **Explicit Cursor**

An explicit cursor is declared in a program's declaration block and are particularly useful when working with SQL statements that produce multiple rows of results. To use an explicit cursor, many steps need to be implemented, including careful definition and initialization of the cursor within the program. If used correctly, explicit cursors can improve program performance and data accuracy.

Explicit Cursor Syntax

Cursor Cursor-Name

IS

Select-Statement;

Steps to Utilize Explicit Cursors

Step 1: Cursor Declaration

Syntax: CURSOR cursor_name IS SELECT statement;

Step 2: Open Cursor

Syntax: OPEN cursor_name;

Step 3: Fetching Cursor

Syntax: FETCH cursor_name INTO variable;

Step 4: Close Cursor

Syntax: CLOSE cursor_name;

VII. Required Resources/apparatus/equipment with specifications

Sr.No	Equipment Name with Broad Specifications	Relevant LLO Number
1	Computer system with all necessary components like; motherboard, random access memory (RAM), read-only memory (ROM), internal hard disk drives, Mouse, Keyboard, and RDBMS applications such as Oracle Express Edition,MySql,SQLite,Oracle Apex etc.	All

VIII. Procedure

1.Write program in notepad, save the program with .sql extension.
2. Take the program to SQL command prompt.
3. Execute and check the program output.

IX. Result(s)

……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………

X. Practical related questions (Provide space for answers)

Note: Below are a few sample questions for reference. Teacher must design more such questions to ensure the achievement of identified CO.

1. Distinguish between Implicit and Explicit cursors in oracle.
2. List advantages of using cursors in SQL programming.

(Space for answer)

……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……………………………………………………………………………………………………

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XI. Exercise

a. Write a PL/SQL program for displaying details of students studying in computer department using cursors.

b. Write a PL/SQL program to print even number of records stored in the student table.

c. Write a PL/SQL program to display the number of items having price more than 10000 in store table using cursors.

XII. References/Suggestions for further reading: include websites/links

1. CURSORS IN PL/SQL WITH EXAMPLES | PL/SQL TUTORIAL - YouTube
2. https://www.oracletutorial.com/plsql-tutorial/plsql-cursors/

3. https://www.javatpoint.com/pl-sql-cursors
4. https://docs.oracle.com/en/database/other-databases/timesten/22.1/plsql-developer/use-cursors-pl-sql-programs.html
5. https://traininginchennai.in/plsql_cursors.html

XIII. Assessment Scheme

Performance indicators		**Weightage**
Sr. No	**Process related (10 Marks)**	**30%**
1.	Tool Selection Ability	20%
2.	Follow Ethical Practices	10%
	Product related (15 Marks)	**70%**
3.	Verifying System Requirement for Installation	20%
4.	Correctness in Use of Command Tools	15%
5.	Use of Attribute/s Options	15%
6.	Timely Submission	10%
7.	Answering Sample Questions	10%
	Total (50 Marks)	**100%**

List of student Team Members

1……………………………………

2………………………………....

3……………………………………

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (35)**	**Total (50)**	

Experiment No 22: Implement PL/SQL program based on Exception Handling (Pre-defined exceptions) application

I. **Practical Significance**: Creating a PL/SQL program with exception handling using pre-defined exceptions allows PL/SQL code to deal with unexpected errors or problems that might happen while it runs. This makes PL/SQL program more reliable and prevents it from crashing or giving wrong results when things go wrong unexpectedly. This practical allows students to implement PL/SQL program based on pre-defined exceptions.

II. INDUSTRY / EMPLOYER EXPECTED OUTCOME:

To implement PL/SQL programs with exception handling using pre-defined exceptions that leads to more reliable code.

III. COURSE LEVEL LEARNING OUTCOMES (COS):

CO4 - Implement PL/SQL codes for given application.

IV. LABORATORY LEARNING OUTCOME:

Implement PL/SQL program based on Exception Handling (Pre-defined exceptions).

V. Relevant Affective Domain related outcome(s)

a. Follow precautionary measures.

b. Follow installation steps.

c. Follow ethical practices.

VI. Relevant Theoretical Background

An *exception* in PL/SQL is a problem or error that happens during the execution of a program, which disrupts the normal flow of the code. The exception block in PL/SQL is important because it helps manage unexpected errors during program execution. It allows you to handle these errors gracefully, preventing program crashes and ensuring a smoother user experience.

Exception Name	**Description**
NO_DATA_FOUND	Raised when a SELECT INTO statement returns no rows.
TOO_MANY_ROWS	Raised when a SELECT INTO statement returns more than one row.

ZERO_DIVIDE	Raised when a division by zero occurs.
INVALID_NUMBER	Raised when trying to convert a character string to a number fails.
VALUE_ERROR	Raised when an arithmetic, conversion, truncation, or constraint error occurs.

Example:

```
DECLARE
temp number;
BEGIN
SELECT p_name into temp from student where p_name='Ameya';
dbms_output.put_line('the p_name is '||temp);
EXCEPTION
WHEN value_error THEN
dbms_output.put_line('Error');
dbms_output.put_line('Change data type of temp to varchar(20)');
END;
```

VALUE_ERROR exception is raised WHEN a statement is executed that resulted in an arithmetic, numeric, string, conversion, or constraint error. This error mainly results from programmer error or invalid data input. Here, it is raised because temp is declared with the datatype number.*temp* datatype should have been varchar2

VII. Required Resources/apparatus/equipment with specifications

Sr.No	**Equipment Name with Broad Specifications**	**Relevant LLO Number**
1	Computer system with all necessary components like; motherboard, random access memory (RAM), read-only memory (ROM), internal hard disk drives, Mouse, Keyboard, and RDBMS applications such as Oracle Express Edition,MySql,SQLite,Oracle Apex etc.	All

VIII. Procedure

1.Define the PL/SQL block structure

2.Implement the logic for the given problem

IX. Result(s)

……………………………………………………………………………………………………

……………………………………………………………………………………………………

…………………………………………………………………………………………………

X. Practical related questions (Provide space for answers)

Note: Below are a few sample questions for reference. Teacher must design more such questions to ensure the achievement of identified CO.

1. Distinguish between user defined exception and predefined exception
2. Explain pre-defined exception with an example.

(Space for answer)

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

XI. Exercise

1. Write a PL/SQL program that asks the user to input two numbers and divide the first number by the second. Handle the predefined exception for division by zero and display an appropriate message if it occurs.
2. Write a PL/SQL program that retrieves the salary of an employee based on their

employee ID (emp_id). If the employee ID does not exist in the database, handle the NO_DATA_FOUND exception and print a message saying, "Employee ID not found."

XII. References/Suggestions for further reading: include websites/links

1. https://www.youtube.com/watch?v=yGU4YfSSjdM
2. https://www.oracletutorial.com/plsql-tutorial/plsql-exception/
3. https://www.javatpoint.com/pl-sql-exception

XIII. Assessment Scheme

	Performance indicators	**Weightage**
Sr. No	**Process related (10 Marks)**	**30%**
1.	Tool Selection Ability	20%
2.	Follow Ethical Practices	10%
	Product related (15 Marks)	**70%**
3.	Verifying System Requirement for Installation	20%
4.	Correctness in Use of Command Tools	15%
5.	Use of Attribute/s Options	15%
6.	Timely Submission	10%
7.	Answering Sample Questions	10%
	Total (50 Marks)	**100%**

List of student Team Members

1……………………………………

2………………………………....

3……………………………………

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (35)**	**Total (50)**	

Experiment No 23: Implement PL/SQL program based on Exception Handling (User-defined exceptions)

I. **Practical Significance**: PL/SQL allows you to define your own exceptions according to the need of your program. A user-defined exception must be declared and then raised explicitly, using either a RAISE statement or the procedure DBMS_STANDARD.RAISE_APPLICATION_ERROR.

II. INDUSTRY / EMPLOYER EXPECTED OUTCOME:

To implement PL/SQL programs with exception handling using user-defined exceptions that leads to more reliable code.

III. COURSE LEVEL LEARNING OUTCOMES (COS):

CO4 - Implement PL/SQL codes for given application.

IV. LABORATORY LEARNING OUTCOME:

Implement PL/SQL program based on Exception Handling (User-defined exceptions).

V. Relevant Affective Domain related outcome(s)

a. Follow precautionary measures.

b. Follow installation steps.

c. Follow ethical practices.

VI. Relevant Theoretical Background

User defined exceptions: This type of users can create their own exceptions according to the need and to raise these exceptions explicitly raise command is used. Example: Divide non-negative integer x by y such that the result is greater than or equal to 1. From the given question we can conclude that there exist two exceptions Division by zero. If result is greater than or equal to 1 means y is less than or equal to x.

```
DECLARE
 x int:=&x; /*taking value at run time*/
 y int:=&y;
 div_r float;
```

```
   exp1 EXCEPTION;
   exp2 EXCEPTION;
   BEGIN
   IF y=0 then
   raise exp1;
  ELSEIF y > x then
  raise exp2;
 ELSE
div_r:= x / y;
   dbms_output.put_line('the result is '||div_r);
   END IF;
   EXCEPTION
  WHEN exp1 THEN
  dbms_output.put_line('Error');
   dbms_output.put_line('division by zero not allowed');
   WHEN exp2 THEN
   dbms_output.put_line('Error');
    dbms_output.put_line('y is greater than x please check the input');
     END;
```

VII. Required Resources/apparatus/equipment with specifications

Sr.No	Equipment Name with Broad Specifications	Relevant LLO Number
1	Computer system with all necessary components like; motherboard, random access memory (RAM), read-only memory (ROM), internal hard disk drives, Mouse, Keyboard, and RDBMS applications such as Oracle Express Edition, MySql,SQLite,Oracle Apex etc.	All

VIII. Procedure

1.Follow the rules while raising the exception.

2.Implement the logic for the given problem

IX. Result(s)

X. Practical related questions (Provide space for answers)

Note: Below are a few sample questions for reference. Teacher must design more such questions to ensure the achievement of identified CO.

1. How to define the exception?
2. Create the Procedure with the help of user defined exception.

(Space for answer)

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

XI. Exercise

1. Write a PL/SQL program by using the user defined exception.
2. Write a PL/SQL program that asks for customer Id, when user enters invalid Id, the exception **Invalid-Id** is raised.

XII. References/Suggestions for further reading: include websites/links

1. https://www.oracletutorial.com/plsql-tutorial/plsql-exception/
2. https://www.javatpoint.com/pl-sql-exception
3. https://www.tutorialspoint.com/plsql/plsql_exceptions.html
4. https://way2tutorial.com/plsql/plsql-user-defined-exception.php

XIII. Assessment Scheme

	Performance indicators	**Weightage**
Sr. No	**Process related (10 Marks)**	**30%**
1.	Tool Selection Ability	20%
2.	Follow Ethical Practices	10%
	Product related (15 Marks)	**70%**
3.	Verifying System Requirement for Installation	20%
4.	Correctness in Use of Command Tools	15%
5.	Use of Attribute/s Options	15%
6.	Timely Submission	10%
7.	Answering Sample Questions	10%
	Total (50 Marks)	**100%**

List of student Team Members

1…………………………………

2………………………………....

3…………………………………

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (35)**	**Total (50)**	

Experiment No 24: Create Procedures and stored procedures for modularity

I. Practical Significance: Procedures in PL/SQL is a named block of code that can be stored in a database. Procedures accept input parameters, perform a specific task. They provide a way to modularize code, enhance reusability, and promote efficient development practices. This practical allows students to implement PL/SQL procedures and call them to perform tasks.

II. INDUSTRY / EMPLOYER EXPECTED OUTCOME:

To implement PL/SQL procedure for modularity and to optimize performance, enhance security

III. COURSE LEVEL LEARNING OUTCOMES (COS):

CO4 - Implement PL/SQL codes for given application.

IV. LABORATORY LEARNING OUTCOME:

Create Procedures and stored procedures for modularity.

V. Relevant Affective Domain related outcome(s)

a. Follow precautionary measures.

b. Follow installation steps.

c. Follow ethical practices.

VI. Relevant Theoretical Background

PL/SQL procedures, also known as stored procedures, are subprograms that can be called to perform a specific action in a database. They can contain a series of SQL statements and take parameters. PL/SQL procedures can be stored in a database and called by name from an application.

Syntax for creating procedure:

```
CREATE [OR REPLACE] PROCEDURE procedure_name
   [ (parameter [, parameter]) ]
IS
   [declaration_section]
BEGIN
   executable_section
[EXCEPTION
   exception_section]
```

```
END [procedure_name];
```

Example:

```
CREATE OR REPLACE PROCEDURE greetings
AS
BEGIN
   dbms_output.put_line('Hello World!');
END;
```

Calling a procedure:

```
BEGIN
   greetings;
END;
```

In the above example, procedure is called by its name *greetings*

VII. Required Resources/apparatus/equipment with specifications

Sr.No	Equipment Name with Broad Specifications	Relevant LLO Number
1	Computer system with all necessary components like; motherboard, random access memory (RAM), read-only memory (ROM), internal hard disk drives, Mouse, Keyboard, and RDBMS applications such as Oracle Express Edition,MySql,SQLite,Oracle Apex etc.	All

VIII. Procedure

1.Define the PL/SQL block structure for procedure

2.Implement the logic for the given problem

3. Call the stored procedure

IX. Result(s)

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

X. Practical related questions (Provide space for answers)

Note: Below are a few sample questions for reference. Teacher must design more such questions to ensure the achievement of identified CO.

1. Write syntax for creating PL/SQL Procedure.
2. Write steps to call a procedure in PL/SQL
3. List types of parameters in Procedure and explain them.

(Space for answer)

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

XI. Exercise

1. Write a procedure emp_count () to count number of employees in department, use dept_no as input parameter
2. Create a stored procedure to accept name and greet user with name.

3. Write a PL/SQL procedure to insert any three records in EMP table.

XII. References/Suggestions for further reading: include websites/links

1. https://www.youtube.com/watch?v=yGU4YfSSjdM
2. https://www.tutorialspoint.com/plsql/plsql_procedures.htm
3. https://www.javatpoint.com/pl-sql-procedure

XIII. Assessment Scheme

Performance indicators		Weightage
Sr. No	**Process related (10 Marks)**	**30%**
1.	Tool Selection Ability	20%
2.	Follow Ethical Practices	10%
Product related (15 Marks)		**70%**
3.	Verifying System Requirement for Installation	20%
4.	Correctness in Use of Command Tools	15%
5.	Use of Attribute/s Options	15%
6.	Timely Submission	10%
7.	Answering Sample Questions	10%
Total (50 Marks)		**100%**

List of student Team Members

1…………………………………

2………………………………....

3…………………………………

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (35)**	**Total (50)**	

Experiment No 25: Create functions for given database

I. **Practical Significance**: PL/SQL function is a named, self-contained block of code that performs a specific task and returns a value. Functions are designed to encapsulate logical operations, promoting code modularity and reusability. Developers can build a more maintainable and efficient PL/SQL codebase by isolating specific tasks within functions. This practical allows students to implement PL/SQL Functions for the given database.

II. INDUSTRY / EMPLOYER EXPECTED OUTCOME:

To implement PL/SQL function on database.

III. COURSE LEVEL LEARNING OUTCOMES (COS):

CO4 - Implement PL/SQL codes for given application.

IV. LABORATORY LEARNING OUTCOME:

Implement Function for given database.

V. Relevant Affective Domain related outcome(s)

a. Follow precautionary measures.
b. Follow installation steps.
c. Follow ethical practices.

VI. Relevant Theoretical Background

A typical PL/SQL function consists of the following components:

Function Declaration

The function declaration serves as the initial step in defining a PL/SQL function. It includes the function’s name, input parameters (if any), and the return type. Parameters can be either input-only or input-output, allowing for dynamic and flexible function behavior.

Local Declarations

Within the function’s block, local declarations allocate memory for variables that are used to store intermediate values or perform calculations. These variables are confined to the scope of the function, preventing them from interfering with other parts of the codebase.

Function Body

The function body contains the actual logic of the function. It consists of a series of statements that manipulate the input parameters and local variables to achieve the desired outcome. This is where the magic happens, as developers can employ conditional statements, loops, and other programming constructs to implement complex operations.

Return Statement

The return statement marks the end of the function and specifies the value that will be returned to the caller. It is essential to ensure that the return type matches the one declared in the function header.

SYNTAX:

```
FUNCTION function_name (parameter1 [IN | OUT | IN OUT] datatype, parameter2 [IN | OUT
| IN OUT] datatype, ...)
  RETURN return_datatype
IS
  -- Declaration section (optional)
  variable1 datatype;
  variable2 datatype;
  ...
BEGIN
  -- Function body
  -- SQL and PL/SQL statements
  RETURN return_value;
EXCEPTION
  -- Exception handling (optional)
  WHEN exception_name1 THEN
    -- Handle exception 1
  WHEN exception_name2 THEN
     -- Handle exception 2
  ...
END;
```

VII. Required Resources/apparatus/equipment with specifications

Sr.No	Equipment Name with Broad Specifications	Relevant LLO Number
1	Computer system with all necessary components like; motherboard, random access memory (RAM), read-only memory (ROM), internal hard disk drives, Mouse, Keyboard, and RDBMS applications such as Oracle Express Edition,MySql,SQLite,Oracle Apex etc.	All

VIII. Procedure

1.Define the PL/SQL block structure

2.Implement the logic for the given problem

IX. Result(s)

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

X. Practical related questions (Provide space for answers)

Note: Below are a few sample questions for reference. Teacher must design more such questions to ensure the achievement of identified CO.

1. What is the difference between Function and Procedure.
2. Write syntax for creating and replacing function.

(Space for answer)

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

XI. Exercise

1. Write PL/SQL function which will compute and return the maximum of two values.
2. Write PL/SQL function to calculate the factorial of given no.

XII. References/Suggestions for further reading: include websites/links

1. https://www.youtube.com/watch?v=yGU4YfSSjdM
2. https://www.javatpoint.com/pl-sql-Functions.
3. https://www.tutorialspoint.com/plsql/plsql_functions.html
4. PL/SQL Developer - Allround Automations

XIII. Assessment Scheme

Performance indicators		Weightage
Sr. No	**Process related (10 Marks)**	**30%**
1.	Tool Selection Ability	20%
2.	Follow Ethical Practices	10%
	Product related (15 Marks)	**70%**
3.	Verifying System Requirement for Installation	20%
4.	Correctness in Use of Command Tools	15%
5.	Use of Attribute/s Options	15%
6.	Timely Submission	10%
7.	Answering Sample Questions	10%
	Total (50 Marks)	**100%**

List of student Team Members

1……………………………………

2………………………………....

3……………………………………

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (35)**	**Total (50)**	

Experiment No 26: Implement triggers for given database

I. **Practical Significance**: PL/SQL triggers are named database objects stored in a database that can be called automatically on the occurrence of a particular event. Triggers can be used to automate database processes, enforce data integrity, and improve database performance. This practical allows students to implement PL/SQL triggers for given database.

II. INDUSTRY / EMPLOYER EXPECTED OUTCOME:

To implement PL/SQL triggers for applying integrity, consistency and error handling in a database

III. COURSE LEVEL LEARNING OUTCOMES (COS):

CO4 - Implement PL/SQL codes for given application.

IV. LABORATORY LEARNING OUTCOME:

Implement triggers for given database.

V. Relevant Affective Domain related outcome(s)

a. Follow precautionary measures.

b. Follow installation steps.

c. Follow ethical practices.

VI. Relevant Theoretical Background

A PL/SQL trigger is a database object that *automatically executes* a specific action such as inserting, updating, or deleting data in response to events like a row being modified or a table being dropped. Triggers are often used to enforce business rules, maintain data integrity, and automate tasks within the database.

Syntax:

```
CREATE [OR REPLACE ] TRIGGER trigger_name
{BEFORE | AFTER | INSTEAD OF }
{INSERT [OR] | UPDATE [OR] | DELETE}
```

```
[OF col_name]
ON table_name
[REFERENCING OLD AS o NEW AS n]
[FOR EACH ROW]
WHEN (condition)
DECLARE
  Declaration-statements
BEGIN
  Executable-statements
EXCEPTION
  Exception-handling-statements
END;
```

VII. Required Resources/apparatus/equipment with specifications

Sr.No	Equipment Name with Broad Specifications	Relevant LLO Number
1	Computer system with all necessary components like; motherboard, random access memory (RAM), read-only memory (ROM), internal hard disk drives, Mouse, Keyboard, and RDBMS applications such as Oracle Express Edition,MySql,SQLite,Oracle Apex etc.	All

VIII. Procedure

1.Define the PL/SQL block structure

2.Implement the logic for the given problem

IX. Result(s)

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

X. Practical related questions (Provide space for answers)

Note: Below are a few sample questions for reference. Teacher must design more such questions to ensure the achievement of identified CO.

1. Differentiate between row level trigger and statement level trigger
2. What is the purpose of using WHEN clause in a trigger

(Space for answer)

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

XI. Exercise

1. Create a trigger on EMP table which is invoked when salary is below 5000 .
2. Create a trigger which invokes on updation of record in Department table

XII. References/Suggestions for further reading: include websites/links

1. https://www.youtube.com/watch?v=yGU4YfSSjdM
2. https://www.javatpoint.com/pl-sql-trigger
3. https://www.tutorialspoint.com/plsql/plsql_triggers.htm

XIII. Assessment Scheme

Performance indicators		Weightage
Sr. No	**Process related (10 Marks)**	**30%**
1.	Tool Selection Ability	20%
2.	Follow Ethical Practices	10%
	Product related (15 Marks)	**70%**
3.	Verifying System Requirement for Installation	20%
4.	Correctness in Use of Command Tools	15%
5.	Use of Attribute/s Options	15%
6.	Timely Submission	10%
7.	Answering Sample Questions	10%
	Total (50 Marks)	**100%**

List of student Team Members

1…………………………………

2……………………………….…

3…………………………………

Marks Obtained			**Dated signature of Teacher**
Process Related (15)	**Product Related (35)**	**Total (50)**	