ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY & SCIENCES
Autonomous status accorded by UGC and Andhra University
Approved by AICTE, Permanently Affiliated to Andhra University
Accredited by NBA (IT,CSE,EEE,ECE, and Mech) & accredited by NAAC with "A" Grade
Sangivalasa - 531162, Bheemunipatnam (Mandal), Visakhapatnam (Dist.)
Phone: 08933 - 225083, 225084, 226131, Fax: 08933-226395
Email: principal@anits.edu.in
COLLEGE CODE - ANIL

DEPARTMENT OF COMPUTER SCIENCES & ENGINEERING

A LABORATORY MANUAL FOR

DATABASE MANAGEMENT SYSTEMS LAB

VISION OF DEPARTMENT

Our vision is to emerge as a world class Computer Science and Engineering department through excellent teaching and strong research environment that responds swiftly to the challenges of changing computer science technology and addresses technological needs of the stakeholders.

MISSION OF DEPARTMENT

To enable our students to master the fundamental principles of computing and to develop in them the skills needed to solve practical problems using contemporary computer-based technologies and practices to cultivate a community of professionals who will serve the public as resources on state-of-the-art computing science and information technology.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO-1	Employability	Work as Competent Computer Engineer either globally or locally by engaging in professional practice in a variety of roles with ability to serve as a team or individual.
PEO-2	Higher studies	Prepared to pursue masters or research programmes in computer science or other disciplines.
PEO-3	Entrepreneurship	Become successful Entrepreneurs who demonstrate strong technical and leadership skills to bring out innovative designs/products that also addresses social issues.
PEO-4	Lifelong learning and ethics	Adapt to rapidly changing technology in engineering domains through continuous learning and practice code of ethics.

PROGRAM SPECIFIC OUTCOMES (PSOs)

1	Programming and software Development skills: Ability to acquire programming efficiency to analyze, design and develop optimal solutions, apply standard practices in software project development to deliver quality software product.
2	Computer Science Specific Skills: Ability to formulate, simulate and use knowledge in various domains like data engineering, image processing and information and network security, artificial intelligence etc., and provide solutions to new ideas and innovations

PROGRAM OUTCOMES (POs)

Graduate Attribute1:	Engineering Knowledge
PO-1	Apply the knowledge of basic engineering sciences, humanities, core engineering and computing concept in modelling and designing computer based systems.
Graduate Attribute2:	Problem Analysis
PO-2	Identify, analyze the problems in different domains and define the requirements appropriate to the solution.
Graduate Attribute3:	Design/Development of Solution
PO-3	Design, implement & test a computer based system, component or process that meet functional constraints such as public health and safety, cultural, societal and environmental considerations.
Graduate Attribute4:	Conduct Investigations of Complex Problems
PO-4	Apply computing knowledge to conduct experiments and solve complex problems, to analyze and interpret the results obtained within specified timeframe and financial constraints consistently.
Graduate Attribute5:	Modern Tool Usage
PO-5	Apply or create modern techniques and tools to solve engineering problems that demonstrate cognition of limitations involved in design choices.
Graduate Attribute6:	The Engineer and Society
PO-6	Apply contextual reason and assess the local and global impact of professional engineering practices on individuals, organizations and society.
Graduate Attribute7:	Environment and Sustainability
PO-7	Assess the impact of engineering practices on societal and environmental sustainability.
Graduate Attribute8:	Ethics
PO-8	Apply professional ethical practices and transform into good responsible citizens with social concern.
Graduate	Individual and Team Work

Attribute9:	
PO-9	Acquire capacity to understand and solve problems pertaining to various fields of engineering and be able to function effectively as an individual and as a member or leader in a team.
Graduate Attribute10:	Communication
PO-10	Communicate effectively with range of audiences in both oral and written forms through technical papers, seminars, presentations, assignments, project reports etc.
Graduate Attribute11:	Project Management and Finance
PO-11	Apply the knowledge of engineering, management and financial principles to develop and critically assess projects and their outcomes in multidisciplinary areas.
Graduate Attribute12:	Life-long Learning
PO-12	Recognize the need and prepare oneself for lifelong self learning to be abreast with rapidly changing technology.

Course outcomes:

CO-1:	Write and Compile basic sql queries
CO-2:	Write and Compile complex queries like nested queries and joins.
CO-3:	Construct triggers, views and stored procedures.
CO-4:	Design and implement a database schema for a given problem domain
CO-5:	Normalize a database and formulate integrity constraints accordingly

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES

A Laboratory Manual
For

DATABASE MANAGEMENT SYSTEMS LAB- CSE 316

Semester- III/IV BTECH SEM I

PRAGNANAM BRAHMA

ANITS

PREPARED BY

1. **G.GOWRI PUSHPA**
2. **A.YASASWI**
3. **K.SURESH**

DEPARTMENT OF COMPUTER SCIENCE &ENGINEERING

SI.No	List of Experiments	CO
1	SQL DDL,DML	1
2	SQL Constraints	1
3	Inbuilt functions in RDBMS.	1
4	Aggregate functions	1
5	Nested Queries & JoinQueries	2
6	Creation and dropping of Views.	3
7	Creating Triggers	3
8	Stored Procedures.	3

Sample Applications:

1. Development of an Online Course Portal for acampus
2. Book Bank ManagementSystem
3. Car Rental ManagementSystem
4. Exam/academic system for College Management
5. Real estate Managementsystem
6. University ManagementSystem
7. Database manager for a Magazine agency or a news paper agency
8. Ticket booking forperformances
9. Inventory ControlSystem
10. Students managementSystem

GUIDELINES TO TEACHERS

- Faculty must verify the observations and records before assigning the system.
- Faculty must verify Students Id cards before entering into Laboratory
- Faculty must take the attendance at the starting and ending of the lab time period.

Sessional marks : 50 marks

1. Daily Evaluation (Includes Record, Observation & regular performance) – 30 marks
2. Attendance – 5 marks
3. Internal Exam – 10 marks
4. Viva Voce – 5 marks

External Exam (50 marks)

- Viva voce – 10 marks
- Write up + Execution – 40 marks

Write up + Execution (40 marks)

- Every student is given 4 questions in the external exam out of which the difficulty level of 2 questions is easy / medium and 2 questions of difficulty level is high
- Each easy / medium level question carries 30% of marks and difficulty level question carries 20% of marks.

INSTRUCTIONS TO STUDENTS:

- Students should use computer related components smoothly
- Students should not carry other items into lab.
- Students must wear the dress code and ID cards.
- Every student is given 4 questions in the external exam out of which the difficulty level of 2 questions is easy / medium and 2 questions of difficulty level is high
- Each easy / medium level question carries 30% of marks and difficulty level question carries 20% of marks.

GUIDELINES TO LAB PROGRAMMERS:

- Lab Programmers must Choose a database engine suitable or compatible with the systems in the lab instructed by the course coordinator and install it in every computer available and make sure the software is working,example softwares are oracle 12c,mysql etc
- Lab Programmers must verify All the Systems whether they are working properly or not.
- Lab Programmers must verify All the other equipments (devices like ACs).

LAB RUBRICS

Name of the Programme: B.TECH (CSE),UG

Semester: III/IV-I semester

Session Duration: 3 HOURS

SNO	Key Performance Criteria(KPC)	4-Very Good	3-Good	2-Fair	1-Need to improve
1	**Understanding the Scenario/Project/Problem/Query**	Understand the problem/scenario/project/query exceptionally well with feasibility study.	Understand the problem/scenario/project/query Properly	Understand the problem/scenario/project/query at basic level.	Understand the problem/scenario/project/query partially.
2	**Analyze the Scenario/Project/Problem/Query and design**	Proper analysis and marking of all the notations,relations in the design with a neat diagrammatic representation	Proper analysis and marking of all the notations,relations in the design with Understandable Diagrammatic representation	Improper analysis and Ambigious marking of all the notations,relations in the design	Incomplete analysis without proper diagrammatic representation.
3	**Correctness of the solution**	The solution produces appropriate results for all the inputs tested along with test cases	The solution produces correct results for most of the inputs	The solution results the correct answers for some inputs and results wrong answers for some cases	Solution does not produce the appropriate results for the given inputs
4	**viva/contents of observation/record/Documentation**	Able to answer more than 75 % of the questions	able to answer 51 % to 75% of the questions	Able to answer 50% of the questions	Unable to answer any question related to the given problem

5	**Preparation/Submission of Documentation/ ppt/record**	Submitted on time, presented / communicated diagrams, tables, writing style etc. excellently	Submitted on time, presented / communicated neatly the diagrams, tables, writing style etc.	Submitted on time, presented / communicated diagrams, tables, writing style, etc. not so neatly but in a readable manner	Submitted , presented / communicated diagrams, tables, writing style, etc. with ambiguity/not in a readable manner.
6	**Courtesy , Ethics, team work (based on physical observation)**	While performing the project/experiment, the student is in proper dress code, always respectful of others, mindful of safety, and leaves the area clean.	While performing the project/experiment, the student is in proper dress code, many times respectful of others, many times mindful of safety, and leaves the area clean only after being reminded.	While performing the project/experiment , the student is in partial dress code, sometimes respectful of others, sometimes mindful of safety, and leaves the area clean only after being reminded.	While performing the project/experiment , the student is not in proper dress code, not respectful of others, not mindful of safety, and leaves the area messy even after being reminded.
7	**Presentation of Case Study/Mini Project**	flow of presentation,visible representaion of the case study/mini project with excellent communication	flow of presentation,visible representaion of the case study/mini project with good communication	flow of presentation,visible representaion of the case study/mini project communicated moderately	flow of presentation,visible representaion of the case study/mini project communicated Not upto the mark

PRACTICAL 1: SQL DDL And DML

1.PRACTICAL SIGNIFICANCE :

In this experiment we can learn how to create a database and perform some operations using inbuilt dbms commands like create,alter,drop,insert,select etc,here when we are using the commands and executing we will learn how to manipulate or modify the data we want to store or insert in a databse and maintain it .

2. RELEVANT PROGRAM OUTCOMES : PO1,PO2,PO3,PO5,PO12

3. COMPETENCY AND PRACTICAL SKILLS :

The practical is expected to develop the following skills :

1.while working with databases students will learn how to create a table in the real time enviroment for example:suppose to insert a customer for a bank in the bank's database by creating a separate database table.

4. PREREQUISITES:

1. Having knowledge about what fields to be present for a particular table created for customers,employees etc and also know how to connect to database.

2. Hands on practice with ER diagrams and datatypes which are going to be used for the columns to be inserted

5. RESOURCES REQUIRED :

1.systems should be installed with a licensed database software such as oracle,mysql etc

6. PRECAUTIONS:

1. Check Whether the computer is getting proper power or not.

2. Ensure the keyboard, mouse and monitor are properly working.

3. Ensure that there are no power fluctuations while executing the program.

4. Safe working conditions help prevent injury to people and damage to computer equipment.

5. A safe work space is clean, organized, and properly lighted. Everyone must understand and follow safety procedures.

6. Follow electrical safety guidelines to prevent electrical fires, injuries, and fatalities in the home and the workplace. Power supplies and CRT monitors contain high voltage.

7. DESCRIPTION - ALGORITHM/CIRCUIT/DIAGRAM/DESCRIPTION:

DDL COMMANDS

1. **The Create Table Command:** - it defines each column of the table uniquely. Each column has minimum of three attributes, a name , data type and size.

Syntax:

Create table <table name> (<col1> <datatype>(<size>),<col2> <datatype><size>));

Ex:create table emp(empno number(4) primary key, ename char(10));

2. **Modifying the structure of tables.**

a) Add new columns

Syntax:

Alter table <tablename> add(<new col><datatype(size),<new col>datatype(size));

Ex:alter table emp add(sal number(7,2));

3. **Dropping a column from a table.**

Syntax:

Alter table <tablename> drop column <col>;

Ex:alter table emp drop column sal;

4. **Modifying existing columns**.

Syntax:

Alter table <tablename> modify(<col><newdatatype>(<newsize>));

Ex:alter table emp modify(ename varchar2(15));

5. Renaming the tables

Syntax:

Rename <oldtable> to <new table>;

Ex:rename emp to emp1;

6. truncating the tables.

Syntax:

Truncate table <tablename>;

Ex:trunc table emp1;

7. Destroying tables.

Syntax:

Drop table <tablename>;

Ex:drop table emp;

CREATION OF TABLE:

SYNTAX:

create table<tablename>(column1 datatype,column2 datatype...);

EXAMPLE:

```
SQL>create table std(sno number(5),sname varchar(20),age number(5),sdob date,sm1
number(4,2),sm2 number(4,2),sm3 number(4,4));
Table created.
SQL>insert into std values(101,’AAA’,16,’03-jul-88’,80,90,98);
1 row created.
SQL>insert into std values(102,’BBB’,18,’04-aug-89’,88,98,90);
1 row created.
```

OUTPUT:

```
Select * from std;
SNO   SNAME   AGE SDOB      SM1   SM2   SM3
101   AAA      16   03-jul-88   80    90    98
102   BBB      18   04-aug-89  88    98    90
```

ALTER TABLE WITH ADD:

```
SQL>create table student(id number(5),name varchar(10),game varchar(20));
Table created.
SQL>insert into student values(1,’mercy’,’cricket’);
1 row created.
```

SYNTAX:

alter table<tablename>add(col1 datatype,col2 datatype..);

EXAMPLE:

```
SQL>alter table student add(age number(4));

SQL>insert into student values(2,’sharmi’,’tennis’,19);
```

OUTPUT:

```
ALTER: select * from student;

ID NAME  GAME

1    Mercy  Cricket


ADD: select * from student;

ID NAME GAME AGE

1    Mercy  cricket

2    Sharmi Tennis 19
```

<u>ALTER TABLE WITH MODIFY:</u>

SYNTAX:

```
Alter table<tablename>modify(col1 datatype,col2 datatype..);
```

EXAMPLE:

```
SQL>alter table student modify(id number(6),game varchar(25));
```

OUTPUT:

```
MODIFY

desc student;

NAME NULL? TYPE

Id        Number(6)

Name Varchar(20)

Game Varchar(25)

Age     Number(4)
```

DROP:

SYNTAX: drop table<tablename>;

EXAMPLE:

```
SQL>drop table student;

SQL>Table dropped.
```

TRUNCATE TABLE

SYNTAX: TRUNCATE TABLE <TABLE NAME>;

Example: Truncate table stud;

DESC

Example: desc emp;

```
Name Null? Type

------------------------------ ------

EmpNo NOT NULL number(5)

EName VarChar(15)

Job NOT NULL Char(10)

DeptNo NOT NULL number(3)

PHONE_NO number (10)
```

DML COMMANDS

DML commands are the most frequently used SQL commands and is used to query and manipulate the existing database objects. Some of the commands are Insert, Select, Update, Delete.

Insert Command This is used to add one or more rows to a table. The values are separated by commas and the data types char and date are enclosed in apostrophes. The values must be entered in the same order as they are defined.

Select Commands It is used to retrieve information from the table. It is generally referred to as querying the table. We can either display all columns in a table or only specify column from the table.

Update Command It is used to alter the column values in a table. A single column may be updated or more than one column could be updated.

Delete command After inserting row in a table we can also delete them if required. The delete command consists of a from clause followed by an optional where clause.

Q1: Insert a single record into dept table.

```
Ans: SQL> insert into dept values (1,'IT','Tholudur');

1 row created.
```

Q2: Insert more than a record into emp table using a single insert command.

```
Ans: SQL> insert into emp values(&empno,'&ename','&job',&deptno,&sal);

Enter value for empno: 1

Enter value for ename: Mathi

Enter value for job: AP

Enter value for deptno: 1

Enter value for sal: 10000

old 1: insert into emp values(&empno,'&ename','&job',&deptno,&sal)
```

```
new 1: insert into emp values(1,'Mathi','AP',1,10000)

1 row created.

SQL> / Enter value for empno: 2

Enter value for ename: Arjun

Enter value for job: ASP

Enter value for deptno: 2

Enter value for sal: 12000

old 1: insert into emp values(&empno,'&ename','&job',&deptno,&sal)

new 1: insert into emp values(2,'Arjun','ASP',2,12000)

1 row created.


SQL> Enter value for empno: 3

Enter value for ename: Gugan

Enter value for job: ASP

Enter value for deptno: 1

Enter value for sal: 12000

old 1: insert into emp values(&empno,'&ename','&job',&deptno,&sal)

new 1: insert into emp values(3,'Gugan','ASP',1,12000)

1 row created.
```

Q3: Update the emp table to set the salary of all employees to Rs15000/- who are working as ASP

Ans: SQL> select * from emp;

```
EMPNO ENAME JOB DEPTNO SAL

 - - - - - - ------------------------- - - - - - - - - - - - - - - - - - - - - - - -


1 Mathi AP 1 10000

 2 Arjun ASP 2 12000

3 Gugan ASP 1 12000
```

```
SQL> update emp set sal=15000 where job='ASP'; 2 rows updated.

SQL> select * from emp;

EMPNO ENAME JOB DEPTNO SAL

------------------------------------------------------------

1 Mathi AP 1 10000

2 Arjun ASP 2 15000

3 Gugan ASP 1 15000
```

Q4: Create a pseudo table employee with the same structure as the table emp and insert rows into the table using select clauses.

```
Ans: SQL> create table employee as select * from emp;

Table created.

SQL> desc employee;

Name Null? Type

--------------------------------------------------------------------

EMPNO NUMBER(6)

ENAME NOT NULL VARCHAR2(20)

JOB NOT NULL VARCHAR2(13)

DEPTNO NUMBER(3)

SAL NUMBER(7,2)
```

Q5: select employee name, job from the emp table

```
Ans: SQL> select ename, job from emp;
ENAME JOB
------------ ---------

Mathi  AP
Arjun  ASP
Gugan  ASP
Karthik Prof
Akalya  AP
suresh  lect
6 rows selected.
```

Q6: Delete only those who are working as lecturer

```
Ans: SQL> select * from emp;
EMPNO ENAME JOB DEPTNO SAL
- - - - - - ------------------------ - - - - - - - - - - - - - - ------------

1 Mathi AP 1 10000
2 Arjun ASP 2 15000
3 Gugan ASP 1 15000
4 Karthik Prof 2 30000
5 Akalya AP 1 10000
6 suresh lect 1 8000
7 rows selected.

SQL> delete from emp where job='lect';
1 row deleted.
SQL> select * from emp;
EMPNO ENAME JOB DEPTNO SAL
-                  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1 Mathi AP 1 10000
2 Arjun ASP 2 15000
3 Gugan ASP 1 15000
4 Karthik Prof 2 30000
5 Akalya AP 1 10000
```

8.TEST CASES

From the above constructed table write the queries for following

1 List the records in the emp table orderby salary in ascending order

2 List the records in the emp table orderby salary in descending order.

3 Display only those employees whose deptno is 30.

4 Display deptno from the table employee avoiding the duplicated values.

9.SAMPLE OUTPUT

1. List the records in the emp table orderby salary in descending order.
   Ans: SQL> select * from emp order by sal desc;

```
EMPNO ENAME JOB DEPTNO SAL
- - - - - - ------------------------ - - - - - - - - - - - - - - ------------

4 Karthik Prof 2 30000
2 Arjun ASP 2 15000
3 Gugan ASP 1 15000
1 Mathi AP 1 10000
5 Akalya AP 1 10000
```

2. Display only those employees whose deptno is 30.
   Solution: Use SELECT FROM WHERE syntax.

```
Ans: SQL> select * from emp where deptno=1;

EMPNO ENAME JOB DEPTNO SAL
------------------------------------------------------------

1 Mathi AP 1 10000
3 Gugan ASP 1 15000
4 Akalya AP 1 10000
```

10. PRACTICAL RELATED QUESTION

1. What are DML commands and DDL commands
2. What is the difference between Drop, Delete and Truncate statements
3. What are tables in SQL?

11.EXCERCISE RELATED QUESTIONS

1. Create a table Universities DB
2. Projects have a project number, a sponsor name ,a starting date, an ending date, and a budget
3. Graduate students have an SSN, a name, an age, and a degree program (e.g., M.S. or Ph.D.)

PRACTICAL 2: SQL CONSTRAINTS

1.PRACTICAL SIGNIFICANCE :

In this experiment we can learn how to create a database and perform some operations using inbuilt dbms commands like create,alter,drop,insert,select etc,here when we are using the commands and executing we will learn how to manipulate or modify the data we want to store or insert in a databse and maintain it .

2. RELEVANT PROGRAM OUTCOMES : PO1,PO2,PO3,PO5,PO12

3. COMPETENCY AND PRACTICAL SKILLS :

The practical is expected to develop the following skills :

1.while working with databases students will learn how to create a table in the real time enviroment for example:suppose to insert a customer for a bank in the bank's database by creating a separate database table.

4. PREREQUISITES:

1. Having knowledge about what fields to be present for a particular table created for customers,employees etc and also know how to connect to database.

2. Hands on practice with ER diagrams and datatypes which are going to be used for the columns to be inserted

5. RESOURCES REQUIRED :

1.systems should be installed with a licensed database software such as oracle,mysql etc

6. PRECAUTIONS:

1. Check Whether the computer is getting proper power or not.

2. Ensure the keyboard, mouse and monitor are properly working.

3. Ensure that there are no power fluctuations while executing the program.

4. Safe working conditions help prevent injury to people and damage to computer equipment.

5. A safe work space is clean, organized, and properly lighted. Everyone must understand and follow safety procedures.

6. Follow electrical safety guidelines to prevent electrical fires, injuries, and fatalities in the home and the workplace. Power supplies and CRT monitors contain high voltage.

7. Description - Algorithm/circuit/Diagram/Description:

CONSTRAINTS:

SQL constraints are used to specify rules for the data in a table.
Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted.
Constraints can be column level or table level. Column level constraints apply to a column, and table level constraints apply to the whole table.
The following constraints are commonly used in SQL:
NOT NULL - Ensures that a column cannot have a NULL value
UNIQUE - Ensures that all values in a column are different
PRIMARY KEY - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
FOREIGN KEY - Prevents actions that would destroy links between tables
CHECK - Ensures that the values in a column satisfies a specific condition
DEFAULT - Sets a default value for a column if no value is specified
CREATE INDEX - Used to create and retrieve data from the database very quickly

Create table tablename (column_name1 data_ type constraints, column_name2 data_ type constraints …)

Example:

Create table Emp (EmpNo number(5), EName VarChar(15), Job Char(10) constraint un unique, DeptNo number(3) CONSTRAINT FKey2 REFERENCES DEPT(DeptNo));

Create table stud (sname varchar2(20) not null, rollno number(10) not null,dob date not null);

DOMAIN INTEGRITY

Example: Create table cust(custid number(6) not null, name char(10));

Alter table cust modify (name not null);

CHECK CONSTRAINT

Example: Create table student (regno number (6), mark number (3) constraint b check (mark >=0 and mark <=100)); Alter table student add constraint b2 check (length(regno<=4));

ENTITY INTEGRITY

a) Unique key constraint

Ensures that all values in a column are different.
Example: Create table cust(custid number(6) constraint unique, name char(10)); Alter table cust add(constraint c unique(custid));

b) Primary Key Constraint

Uniquely identifies each row/record in a database table.
Example: Create table stud(regno number(6) constraint primary key, name char(20));

Queries:

Q1. Create a table called EMP with the following structure.
Name Type

EMPNO NUMBER(6)
ENAME VARCHAR2(20)
JOB VARCHAR2(10)
DEPTNO NUMBER(3)
SAL NUMBER(7,2)
Allow NULL for all columns except ename and job.

Solution:

1. Understand create table syntax.
2. Use the create table syntax to create the said tables.
3. Create primary key constraint for each table as understand from logical table structure.

Ans:

```
SQL> create table emp(empno number(6),ename varchar2(20)not null,job varchar2(10) not null, deptno number(3),sal number(7,2));
Table created.
```

Q2: Add a column experience to the emp table.

experience numeric null allowed.

Solution:

1. Learn alter table syntax.
2. Define the new column and its data type.
3. Use the alter table syntax.

Ans: SQL> alter table emp add(experience number(2));

Table altered.

Q3: Modify the column width of the job field of emp table.

Solution:

1. Use the alter table syntax.
2. Modify the column width and its data type.

Ans: SQL> alter table emp modify(job varchar2(12));

Table altered.

SQL> alter table emp modify(job varchar(13));

Table altered.

Q4: Create dept table with the following structure.

```
Name Type
- - - - - - - - ------------------------

DEPTNO NUMBER(2)
DNAME VARCHAR2(10)
LOC VARCHAR2(10)
```

Deptno as the primarykey

Solution:

1. Understand create table syntax.
2. Decide the name of the table.
3. Decide the name of each column and its data type.
4. Use the create table syntax to create the said tables.
5. Create primary key constraint for each table as understand from logical table structure.

Ans:

SQL> create table dept(deptno number(2) primary key,dname varchar2(10),loc varchar2(10));

Table created.

Q5: create the emp1 table with ename and empno, add constraints to check the empno value while entering (i.e) empno > 100.

Solution:

1. Learn alter table syntax.
2. Define the new constraint [columns name type]
3. Use the alter table syntax for adding constraints.

ANS:

SQL> create table emp1(ename varchar2(10),empno number(6) constraint check(empno>100));
Table created.

Q6: drop a column experience to the emp table.

Solution:

1. Learn alter table syntax. Use the alter table syntax to drop the column.

Ans:
SQL> alter table emp drop column experience; Table altered.

Q7: Truncate the emp table and drop the dept table

Solution:

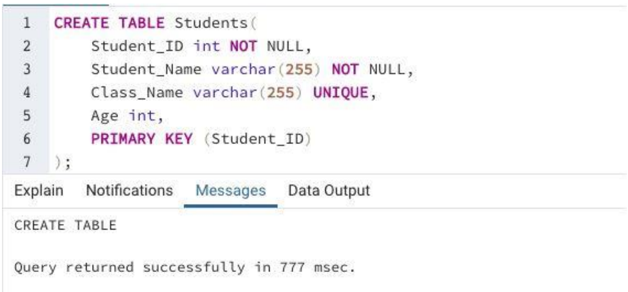
1. Learn drop, truncate table syntax.

Ans: SQL> truncate table emp; Table truncated.

8. TEST CASES

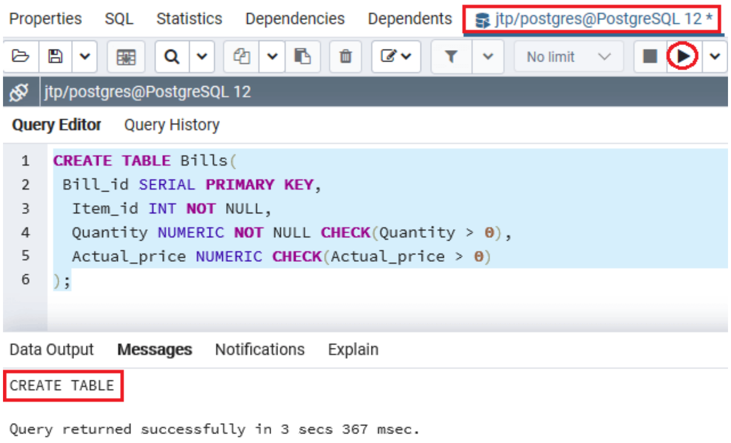
1. Apply References constraint University Database Department table.
2. Add primary key to an existing database table

9. SAMPLE OUTPUTS:



```
CREATE TABLE Students(
    Student_ID int NOT NULL,
    Student_Name varchar(255) NOT NULL,
    Class_Name varchar(255) UNIQUE,
    Age int,
    PRIMARY KEY (Student_ID)
);
```

```
CREATE TABLE

Query returned successfully in 777 msec.
```



```
CREATE TABLE Bills(
 Bill_id SERIAL PRIMARY KEY,
  Item_id INT NOT NULL,
  Quantity NUMERIC NOT NULL CHECK(Quantity > 0),
  Actual_price NUMERIC CHECK(Actual_price > 0)
);
```

```
CREATE TABLE

Query returned successfully in 3 secs 367 msec.
```

10.PRACTICE QUESTIONS

1. Define DDL
2. What are constraints?
3. Categories of SQL Command.
4. Difference between truncate and drop.
5. Define primary and referential integrity.

11.EXERCISE QUESTIONS

1.Consider the schema:

employee(employee-name, street, city)
works(employee-name, company-name, salary)
company(company-name, city)
manages(employee-name, manager-name)

Give an SQL DDL definition for the tables of this database. Identify referential integrity constraints that should hold and include them in the DDL definition.

PRACTICAL 3 – INBULT FUNCTIONS IN SQL

1.PRACTICAL SIGNIFICANCE :

In SQL a built-in function is a piece for programming that takes zero or more inputs and returns a value.Sql is designed with specific functions which can used without the use of writing the query which are called inbuilt functions through which we can perfome minute operations without the need to write complex queries and also use these inbuilt functions while writing huge queries.

2. RELEVANT PROGRAM OUTCOMES : PO1,PO2,PO3,PO5,PO12

3. COMPETENCY AND PRACTICAL SKILLS :

The practical is expected to develop the following skills :

1. while working on this experiment one should know some of the usage of functions or meanings of date functions,mathematical functions such as ceil,floor etc.

4. PREREQUISITES:

1. Make sure that we have an active db created.

5. RESOURCES REQUIRED :

1.systems should be installed with a licensed database software such as oracle,mysql etc

6. PRECAUTIONS:

1. Check Whether the computer is getting proper power or not.

2. Ensure the keyboard, mouse and monitor are properly working.

3. Ensure that there are no power fluctuations while executing the program.

4. Safe working conditions help prevent injury to people and damage to computer equipment.

5. A safe work space is clean, organized, and properly lighted. Everyone must understand and follow safety procedures.

6. Follow electrical safety guidelines to prevent electrical fires, injuries, and fatalities in the home and the workplace. Power supplies and CRT monitors contain high voltage.

7.DESCRIPTION- ALGORITHM/QUERY/PROGRAM

CHARACTER/STRING FUNCTION:

```
SQL> select upper('welcome') from dual;



-------


WELCOME


SQL> select upper('hai') from dual;

---

HAI


SQL> select lower('HAI') from dual;

LOW

---

hai


SQL> select initcap('hello world') from dual;

INITCAP('Hello

---------


Hello World


SQL> select ltrim('    hai') from dual;

LTR

---
```

```
hai


SQL> select rtrim('hai      ')from dual;
SQL> select rpad('hai',3,'*')from dual;

RPAD('

----


hai***


SQL> select lpad('hai',3,'*')from dual;

LPAD('

----


***hai


SQL> select replace('Dany','y','ie')from dual;

REPLACE

-----


Danie


SQL> select translate('cold','ld','ol')from dual;

TRANSL

----


cool
```

```
SQL> select rpad('hai',3,'*')from dual;

RPAD('

----

hai***


SQL> select lpad('hai',3,'*')from dual;

LPAD('

----

***hai


SQL> select replace('Dany','y','ie')from dual;

REPLACE

-----

Danie


SQL> select translate('cold','ld','ol')from dual;

TRANSL

----

cool
```

DATE & TIME FUNCTION

```
SQL> select sysdate from dual;
SYSDATE
------

07-APR-10

SQL> select round(sysdate)from dual;
ROUND(SYS
------

07-APR-10

SQL> select add_months(sysdate,3)from dual;
ADD_MONTH
------

07-JUL-10

SQL> select last_day(sysdate)from dual;
LAST_DAY(
------

30-APR-10

SQL> select sysdate+20 from dual;
SYSDATE+2
------

27-APR-10

SQL> select next_day(sysdate,'tuesday')from dual;

NEXT_DAY(

------

13-APR-10
```

NUMERIC FUNCTION

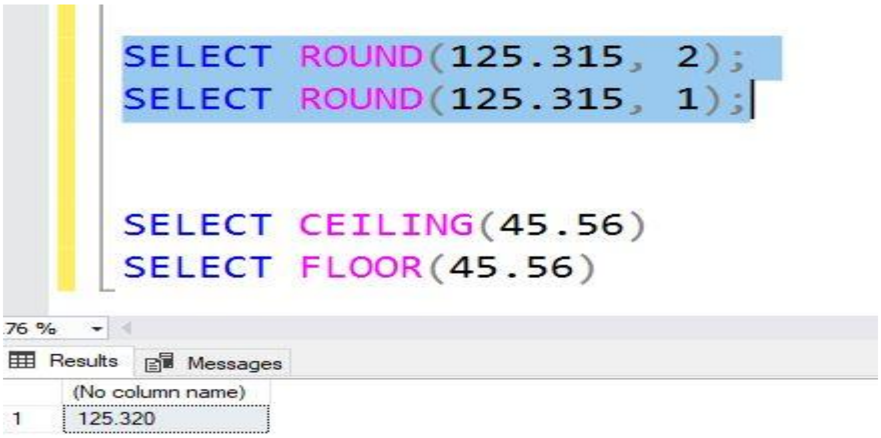
```
SQL> select round(15.6789)from dual;

ROUND(15.6789)

---------


        16


SQL> select ceil(23.20)from dual;

CEIL(23.20)

-------


      24


SQL> select floor(34.56)from dual;

FLOOR(34.56)

--------


       34


SQL> select trunc(15.56743)from dual;

TRUNC(15.56743)

----------


         15


SQL> select sign(-345)from dual;

SIGN(-345)

-------


     -1


SQL> select abs(-70)from dual;

 ABS(-70)

------
```

```
       70
```

MATH FUNCTION:

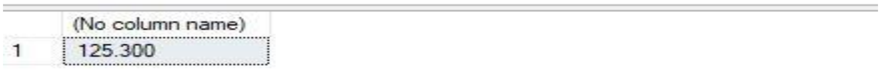
```
SQL> select abs(45) from dual;
  ABS(45)
------

       45

SQL> select power(10,12) from dual;
POWER(10,12)
--------

   1.000E+12

SQL> select mod(11,5) from dual;
MOD(11,5)
------

        1

SQL> select exp(10) from dual;
  EXP(10)
------

22026.466

SQL> select sqrt(225) from dual;
SQRT(225)
------

       15
```

8.SAMPLE OUTPUT



```
SELECT ROUND(125.315, 2);
SELECT ROUND(125.315, 1);


SELECT CEILING(45.56)
SELECT FLOOR(45.56)
```

```
  (No column name)
1 125.320
```



```
  (No column name)
1 125.300
```

9.PRACTICE QUESTIONS:

1.describe some inbuilt functions in mysql
2.what are the scalar functions used
3.describe some date functions and their significance

10.EXCERICE QUESTIONS

1.write a query to display current systems date
2.write a query to retrieve the substring from the create table
3. Write a query to round the marks to the integer value.

PRACTICAL 4: AGGREGATE FUNCTIONS

1.PRACTICAL SIGNIFICANCE :

This experiment is useful to calculate highest salaries or lowest salaries or count of the employees in a company or an organization or specific performance of an employees,apparantely sql has inbuilt aggregate functions to work with in the data base

2. RELEVANT PROGRAM OUTCOMES : PO1,PO2,PO3,PO5,PO12

3. COMPETENCY AND PRACTICAL SKILLS :

The practical is expected to develop the following skills :

1. In database management systems, an aggregate function is a function that performs calculations on a group of values and returns a single value. It is used to summarize the data. It performs calculations on multiple rows of a single column of a table to form a single value of more significant meaning.

4. PREREQUISITES:

1. Make sure that we have an active db created.

5. RESOURCES REQUIRED :

1.systems should be installed with a licensed database software such as oracle,mysql etc

6. PRECAUTIONS:

1. Check Whether the computer is getting proper power or not.

2. Ensure the keyboard, mouse and monitor are properly working.

3. Ensure that there are no power fluctuations while executing the program.

4. Safe working conditions help prevent injury to people and damage to computer equipment.

5. A safe work space is clean, organized, and properly lighted. Everyone must understand and follow safety procedures.

6. Follow electrical safety guidelines to prevent electrical fires, injuries, and fatalities in the home and the workplace. Power supplies and CRT monitors contain high voltage.

7.**Algorithm/circuit/Diagram/Description:**

Aggregate Functions

- SQL aggregation function is used to perform the calculations on multiple rows of a single column of a table. It returns a single value.
- It is also used to summarize the data. Types of SQL Aggregation Function

1. Count
2. Avg
3. Sum
4. Min
5. Max

Aggregate functions in DBMS take multiple rows from the table and return a value according to the query.

All the aggregate functions are used in Select statement.

Syntax −

```
SELECT <FUNCTION NAME> (<PARAMETER>) FROM <TABLE NAME>
```

AVG Function

This function returns the average value of the numeric column that is supplied as a parameter.

Example: Write a query to select average salary from employee table.

```
Select AVG(salary) from Employee
```

COUNT Function

The count function returns the number of rows in the result. It does not count the null values.

Example: Write a query to return number of rows where salary > 20000.

```
Select COUNT(*) from Employee where Salary > 20000;
```

Types −

- COUNT(*): Counts all the number of rows of the table including null.
- COUNT(COLUMN_NAME): count number of non-null values in column.
- COUNT(DISTINCT COLUMN_NAME): count number of distinct values in a column.

MAX Function

The MAX function is used to find maximum value in the column that is supplied as a parameter. It can be used on any type of data.

Example − Write a query to find the maximum salary in employee table.

```
Select MAX(salary) from Employee
```

SUM Function

This function sums up the values in the column supplied as a parameter.

Example: Write a query to get the total salary of employees.

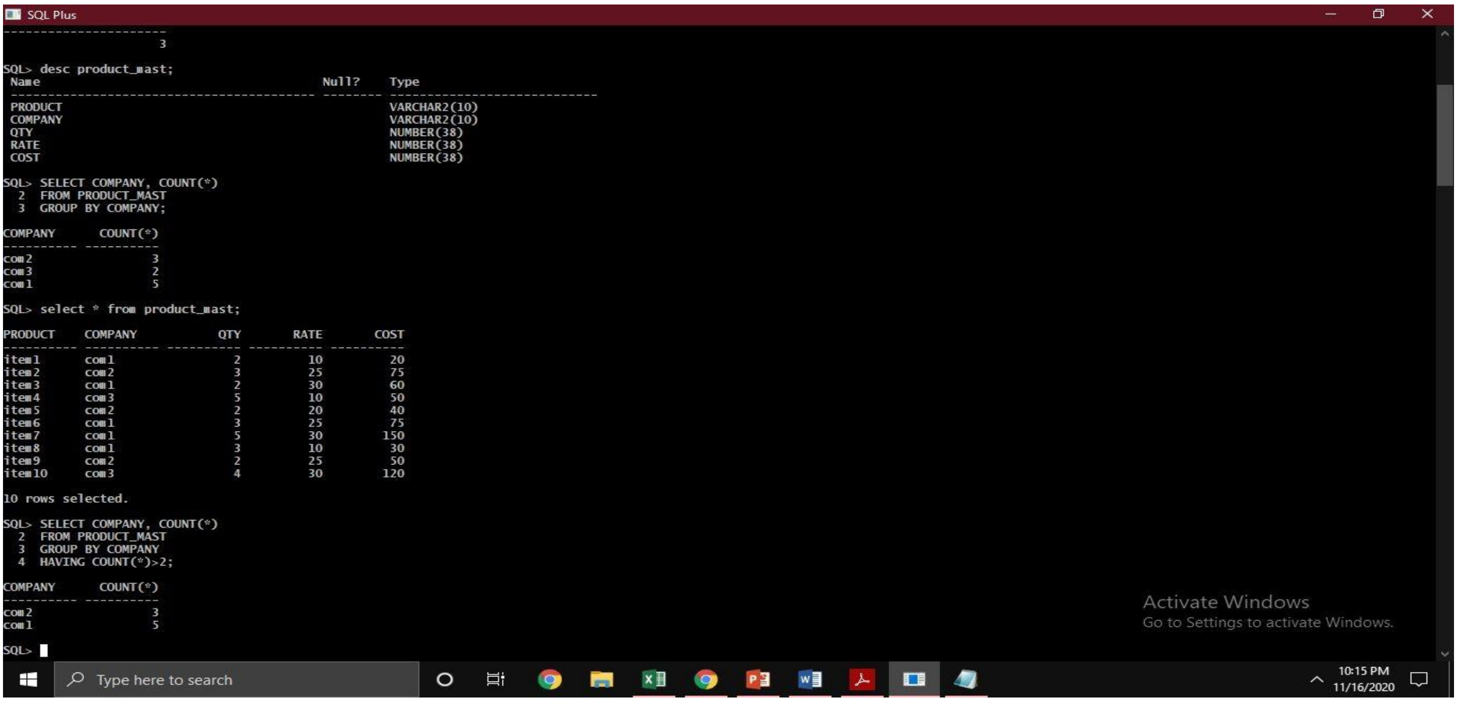
8.PRACTICAL QUESTIONS

1. what are aggregate functions and why do we use them
2. describe having and group by clause

9.EXERCISE QUESTIONS

1. Write a query to list the number of jobs available in the employees table.
2. Write a query to get the total salaries payable to employee

10..SAMPLE OUTPUTS:



```
----------------------
                     3

SQL> desc product_mast;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PRODUCT                                            VARCHAR2(10)
 COMPANY                                            VARCHAR2(10)
 QTY                                                NUMBER(38)
 RATE                                               NUMBER(38)
 COST                                               NUMBER(38)

SQL> SELECT COMPANY, COUNT(*)
  2  FROM PRODUCT_MAST
  3  GROUP BY COMPANY;

COMPANY      COUNT(*)
---------- ----------
com2                3
com3                2
com1                5

SQL> select * from product_mast;

PRODUCT    COMPANY           QTY       RATE       COST
---------- ---------- ---------- ---------- ----------
item1      com1                2         10         20
item2      com2                3         25         75
item3      com1                2         30         60
item4      com3                5         10         50
item5      com2                2         20         40
item6      com1                3         25         75
item7      com1                5         30        150
item8      com1                3         10         30
item9      com2                2         25         50
item10     com3                4         30        120

10 rows selected.

SQL> SELECT COMPANY, COUNT(*)
  2  FROM PRODUCT_MAST
  3  GROUP BY COMPANY
  4  HAVING COUNT(*)>2;

COMPANY      COUNT(*)
---------- ----------
com2                3
com1                5

SQL> 
```

PRACTICAL QUESTION 5 – NESTED QUERIES AND JOIN QUERIES

1.PRACTICAL SIGNIFICANCE :

A Subquery or Inner query or a Nested query is a query within another SQL query and embedded within the WHERE clause. A subquery is used to return data that will be used in the main query as a condition to further restrict the data to be retrieved.

The SQL Joins clause is used to combine records from two or more tables in a database. A JOIN is a means for combining fields from two tables by using values common to each.

2. RELEVANT PROGRAM OUTCOMES : PO1,PO2,PO3,PO5,PO12

3. COMPETENCY AND PRACTICAL SKILLS :

The practical is expected to develop the following skills :

1.helps in processing of query effectively also reduces the complexity of writing more number of queries for retrieval of specific operations.

2.joins are used to combine two table using a specified matching column so that a relatable table can be viewed at a time.

4. PREREQUISITES:

1. Make sure that we have an active db created.

2. Subqueries are required to have names, which are added after parentheses the same way you would add an alias to a normal table. In this case, we've used the name "sub."

3. while performing joins make sure there are more than on table.

5. RESOURCES REQUIRED :

1.systems should be installed with a licensed database software such as oracle,mysql etc

6. PRECAUTIONS:

1. Check Whether the computer is getting proper power or not.

2. Ensure the keyboard, mouse and monitor are properly working.

3. Ensure that there are no power fluctuations while executing the program.

4. Safe working conditions help prevent injury to people and damage to computer equipment.

5. A safe work space is clean, organized, and properly lighted. Everyone must understand and follow safety procedures.

6. Follow electrical safety guidelines to prevent electrical fires, injuries, and fatalities in the home and the workplace. Power supplies and CRT monitors contain high voltage.

7.ALGORITHM/CIRCUIT/DIAGRAM/DESCRIPTION

1. A Subquery or Inner query or a Nested query is a query within another SQL query and embedded within the WHERE clause.
2. A subquery is used to return data that will be used in the main query as a condition to further restrict the data to be retrieved.
3. Subqueries can be used with the SELECT, INSERT, UPDATE, and DELETE statements along with the operators like =, <, >, >=, <=, IN, BETWEEN, etc.

There are a few rules that subqueries must follow −

1. Subqueries must be enclosed within parentheses.
2. A subquery can have only one column in the SELECT clause, unless multiple columns are in the main query for the subquery to compare its selected columns.
3. An ORDER BY command cannot be used in a subquery, although the main query can use an ORDER BY. The GROUP BY command can be used to perform the same function as the ORDER BY in a subquery.
4. Subqueries that return more than one row can only be used with multiple value operators such as the IN operator.
5. A subquery cannot be immediately enclosed in a set function.
6. The BETWEEN operator cannot be used with a subquery. However, the BETWEEN operator can be used within the subquery.

Q1: Display all employee names and salary whose salary is greater than minimum salary of the company and job title starts with ‗M‘.

Solution:

1. Use select from clause.

2. Use like operator to match job and in select clause to get the result.

Ans: SQL> select ename,sal from emp where sal>(select min(sal) from emp where job like 'A%');

```
ENAME SAL

----------- -------

Arjun 12000

Gugan 20000

Karthik 15000
```

Q2: Issue a query to find all the employees who work in the same job as Arjun.

Ans: SQL> select * from emp;

```
EMPNO ENAME JOB DEPTNO SAL

------ ----------------------- --------------------

 1 Mathi AP 1 10000

2 Arjun ASP 2 12000

3 Gugan ASP 2 20000

4 Karthik AP 1 15000
```

SQL> select ename from emp where job=(select job from emp where ename='Arjun');

```
ENAME

 ---------
```

Arjun

Gugan

SQL JOIN

As the name shows, JOIN means to combine something. In case of SQL, JOIN means "to combine two or more tables".

In SQL, JOIN clause is used to combine the records from two or more tables in a database.

Types of SQL JOIN

1. INNER JOIN
2. LEFT JOIN
3. RIGHT JOIN
4. FULL JOIN

Sample Table

EMPLOYEE

EMP_ID	EMP_NAME	CITY	SALARY	AGE
1	Angelina	Chicago	200000	30
2	Robert	Austin	300000	26
3	Christian	Denver	100000	42
4	Kristen	Washington	500000	29
5	Russell	Los angels	200000	36
6	Marry	Canada	600000	48

PROJECT_NO	EMP_ID	DEPARTMENT
101	1	Testing
102	2	Development
103	3	Designing

104	4	Development

1. **INNER JOIN**

In SQL, INNER JOIN selects records that have matching values in both tables as long as the condition is satisfied. It returns the combination of all rows from both the tables where the condition satisfies.

Syntax

```
SELECT table1.column1, table1.column2, table2.column1,....
FROM table1
INNER JOIN table2
ON table1.matching_column = table2.matching_column;
```

Query

```
SELECT EMPLOYEE.EMP_NAME, PROJECT.DEPARTMENT
FROM EMPLOYEE
INNER JOIN PROJECT
ON PROJECT.EMP_ID = EMPLOYEE.EMP_ID;
```

Output

EMP_NAME	DEPARTMENT
Angelina	Testing
Robert	Development
Christian	Designing
Kristen	Development

2. **LEFT JOIN**

The SQL left join returns all the values from left table and the matching values from the right table. If there is no matching join value, it will return NULL.

SYNTAX

```
SELECT table1.column1, table1.column2, table2.column1,....
```

```
FROM table1
LEFT JOIN table2
ON table1.matching_column = table2.matching_column;
```

Query

```
SELECT EMPLOYEE.EMP_NAME, PROJECT.DEPARTMENT
FROM EMPLOYEE
LEFT JOIN PROJECT
ON PROJECT.EMP_ID = EMPLOYEE.EMP_ID;
```

Output

EMP_NAME	DEPARTMENT
Angelina	Testing
Robert	Development
Christian	Designing
Kristen	Development
Russell	NULL
Marry	NULL

3. **RIGHT JOIN**

In SQL, RIGHT JOIN returns all the values from the values from the rows of right table and the matched values from the left table. If there is no matching in both tables, it will return NULL.

Syntax

```
SELECT table1.column1, table1.column2, table2.column1,....
FROM table1
RIGHT JOIN table2
ON table1.matching_column = table2.matching_column;
```

Query

```
SELECT EMPLOYEE.EMP_NAME, PROJECT.DEPARTMENT
FROM EMPLOYEE
RIGHT JOIN PROJECT
ON PROJECT.EMP_ID = EMPLOYEE.EMP_ID;
```

Output

EMP_NAME	DEPARTMENT
Angelina	Testing
Robert	Development
Christian	Designing
Kristen	Development

4. FULL JOIN

In SQL, FULL JOIN is the result of a combination of both left and right outer join. Join tables have all the records from both tables. It puts NULL on the place of matches not found.

Syntax

```
SELECT table1.column1, table1.column2, table2.column1,....
FROM table1
FULL JOIN table2
ON table1.matching_column = table2.matching_column;
```

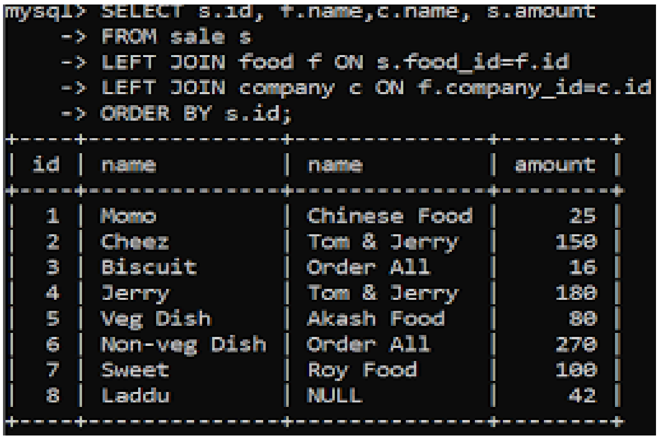
Query

```
SELECT EMPLOYEE.EMP_NAME, PROJECT.DEPARTMENT
FROM EMPLOYEE
FULL JOIN PROJECT
ON PROJECT.EMP_ID = EMPLOYEE.EMP_ID;
```

Output

EMP_NAME	DEPARTMENT
Angelina	Testing
Robert	Development
Christian	Designing
Kristen	Development
Russell	NULL
Marry	NULL

8.SAMPLE OUTPUTS:



```
mysql> SELECT s.id, f.name,c.name, s.amount
    -> FROM sale s
    -> LEFT JOIN food f ON s.food_id=f.id
    -> LEFT JOIN company c ON f.company_id=c.id
    -> ORDER BY s.id;
+----+----------------+----------------+--------+
| id | name           | name           | amount |
+----+----------------+----------------+--------+
|  1 | Momo           | Chinese Food   |     25 |
|  2 | Cheez          | Tom & Jerry    |    150 |
|  3 | Biscuit        | Order All      |     16 |
|  4 | Jerry          | Tom & Jerry    |    180 |
|  5 | Veg Dish       | Akash Food     |     80 |
|  6 | Non-veg Dish   | Order All      |    270 |
|  7 | Sweet          | Roy Food       |    100 |
|  8 | Laddu          | NULL           |     42 |
+----+----------------+----------------+--------+
```

9.**PRACTICE QUESTIONS**

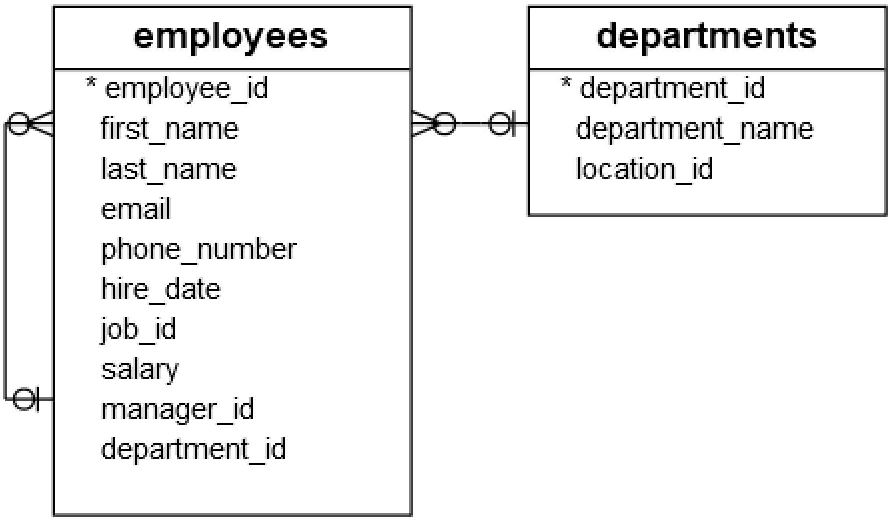
1. What is a SQL Join?
2. What is inner join,outer join,full outer join
3. What is a subquery?

10.EXERCISE RELATED QUESTIONS

1.Write a SQL statement to prepare a list with salesman name, customer name and their cities for the salesmen and customer who belongs to the same city
SALEMAN TABLE : (salesman_id ,name,city,commission)
CUSTOMER TABLE : (customer_id,cust_name, city,grade,salesman_id)

- Write a SQL statement to make a list with order no, purchase amount, customer name and their cities for those orders which order amount between 500 and 2000.
- Write a SQL statement to know which salesman are working for which customer
- Write a SQL statement to know which salesman are working for which customer

2.Consider the following employees and departments tables from the sample database:



1. find all departments located at the location whose id is 1700:
2. find all employees that belong to the location 1700 by using the department id list of the previous query:
3. find all employees who do not locate at the location 1700:
4. find the employees who have the highest salary:

PRACTICAL QUESTION 7– CREATION AND DROPPING OF VIEWS.

1.PRACTICAL SIGNIFICANCE :

A Subquery or Inner query or a Nested query is a query within another SQL query and embedded within the WHERE clause. A subquery is used to return data that will be used in the main query as a condition to further restrict the data to be retrieved.

The SQL Joins clause is used to combine records from two or more tables in a database. A JOIN is a means for combining fields from two tables by using values common to each.

2. RELEVANT PROGRAM OUTCOMES : PO1,PO2,PO3,PO5,PO12

3. COMPETENCY AND PRACTICAL SKILLS :

The practical is expected to develop the following skills :

1.helps in processing of query effectively also reduces the complexity of writing more number of queries for retrieval of specific operations.

2.joins are used to combine two table using a specified matching column so that a relatable table can be viewed at a time.

4. PREREQUISITES:

1. Make sure that we have an active db created.

2. Subqueries are required to have names, which are added after parentheses the same way you would add an alias to a normal table. In this case, we've used the name "sub."

3. While performing joins make sure there are more than on table.

5. RESOURCES REQUIRED :

1.Systems should be installed with a licensed database software such as oracle,mysql etc

6. PRECAUTIONS:

1. Check Whether the computer is getting proper power or not.

2. Ensure the keyboard, mouse and monitor are properly working.

3. Ensure that there are no power fluctuations while executing the program.

4. Safe working conditions help prevent injury to people and damage to computer equipment.

5. A safe work space is clean, organized, and properly lighted. Everyone must understand and follow safety procedures.

6. Follow electrical safety guidelines to prevent electrical fires, injuries, and fatalities in the home and the workplace. Power supplies and CRT monitors contain high voltage.

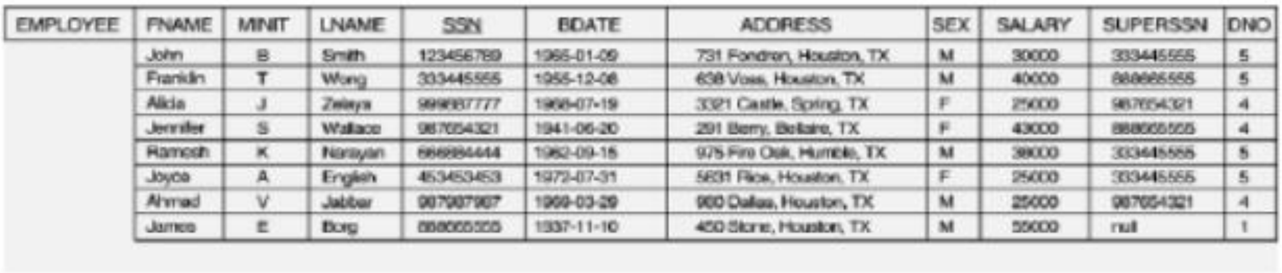
7.ALGORITHM/CIRCUIT/DIAGRAM/DESCRIPTION

Views Syntax:Create View As Select statement;
Example: SQL>Create View EmpView As Select * from Employee;
View created.
Syntax:Select columnname,columnname from ;
Example:
SQL>Select Empno,Ename,Salary from EmpView where Deptno in(10,30);

Updatable Views: Syntax for creating an Updatable View:

Create View Emp_vw As Select Empno,Ename,Deptno from Employee;
View created.
SQL>Insert into Emp_vw values(1126,'Brijesh',20);
SQL>Update Emp_vw set Deptno=30 where Empno=1125;
1 row updated.

For a given EMPLOYEE tables



EMPLOYEE	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
	John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
	Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
	Alicia	J	Zelaya	999887777	1968-07-19	3321 Castle, Spring, TX	F	25000	987654321	4
	Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
	Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
	Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
	Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
	James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	null	1

Perform the Following
1. Creating Views (With and Without Check Option),
2. Selecting from a View
3. Dropping Views

```
SQL> CREATE TABLE EMPLOYEE ( SSN VARCHAR2 (20) PRIMARY KEY, FNAME VARCHAR2
(20), LNAME VARCHAR2 (20), ADDRESS VARCHAR2 (20), SEX CHAR (1), SALARY INTEGER,
SUPERSSN REFERENCES EMPLOYEE (SSN), DNO REFERENCES DEPARTMENT (DNO));
```

```
 SQL> DESC EMPLOYEE;
Name  Null?  Type
 ---------------------------------------- --------
SSN NOT NULL VARCHAR2(20)
FNAME VARCHAR2(20)
LNAME VARCHAR2(20)
ADDRESS VARCHAR2(20)
SEX CHAR(1)
SALARY NUMBER(38)
SUPERSSN VARCHAR2(20)
DNO NUMBER(38)
```

```
 INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES
(‘RNSECE01‘,‘JOHN‘,‘SCOTT‘,‘BANGALORE‘,‘M‘, 450000);
```

POPULATE THE TABLES IN THE ABOVE WAY

Creating View
The query that defines the sales_staffview references only rows in department 5. Furthermore, the CHECK OPTION creates the view with the constraint (named sales_staff_cnst) that INSERT and UPDATE statements issued against the view cannot result in rows that the view cannot select.

1. Creating Views (With and Without Check Option)

   ```
    SQL> CREATE VIEW sales_staff AS
   2 SELECT fname, ssn, dno
   3 FROM employee
   4 WHERE dno =5
   5 WITH CHECK OPTION CONSTRAINT sales_staff_cnst;
   View created.
   ```

2. Selecting from a View

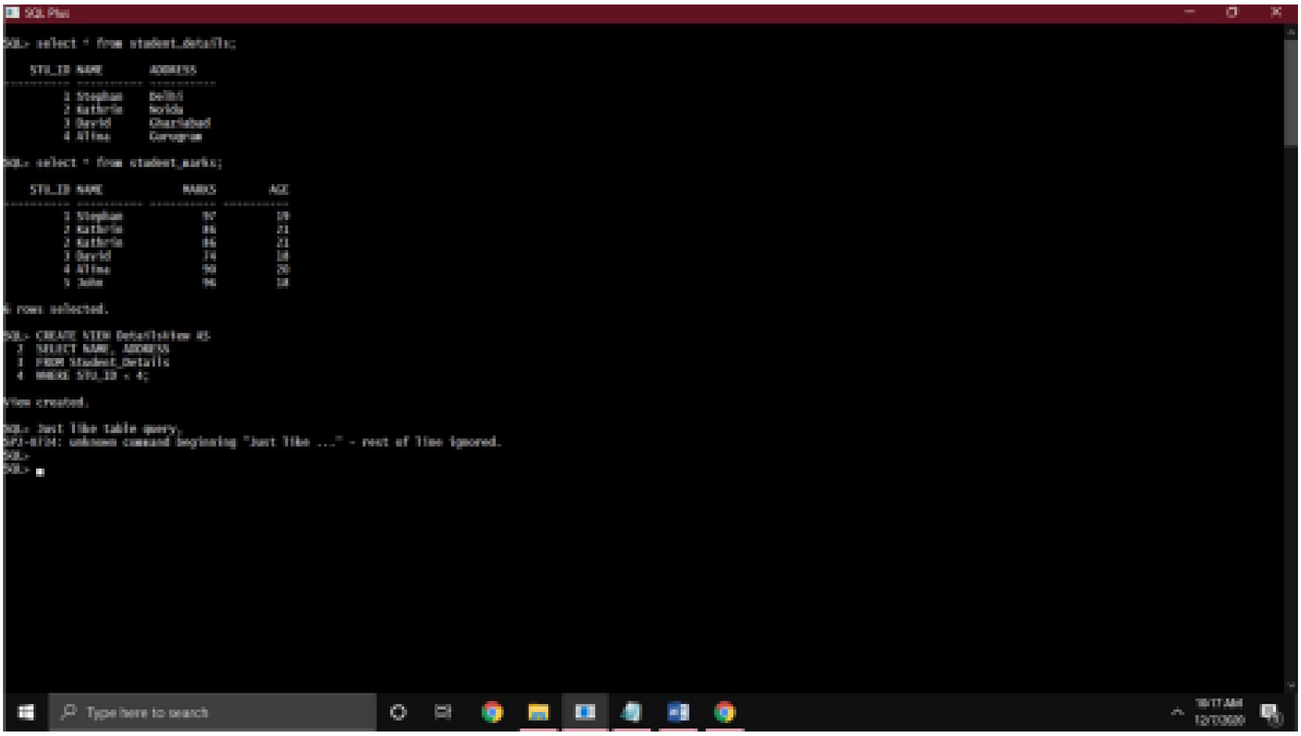
   ```
   SQL> select * from sales_staff;
   ```

3. Drop View

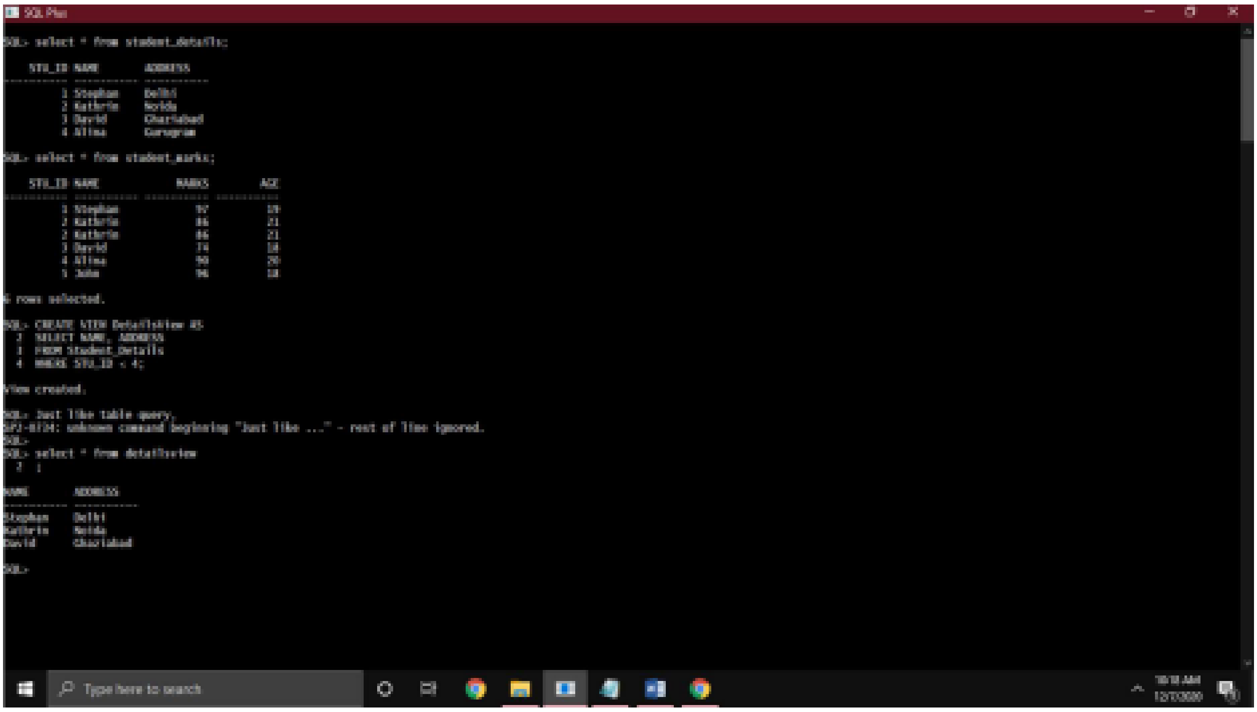
   ```
   SQL>DROP VIEW sales_staff;
   ```

8.TEST CASES
1. Creating an Updatable View
2. Create a Join view

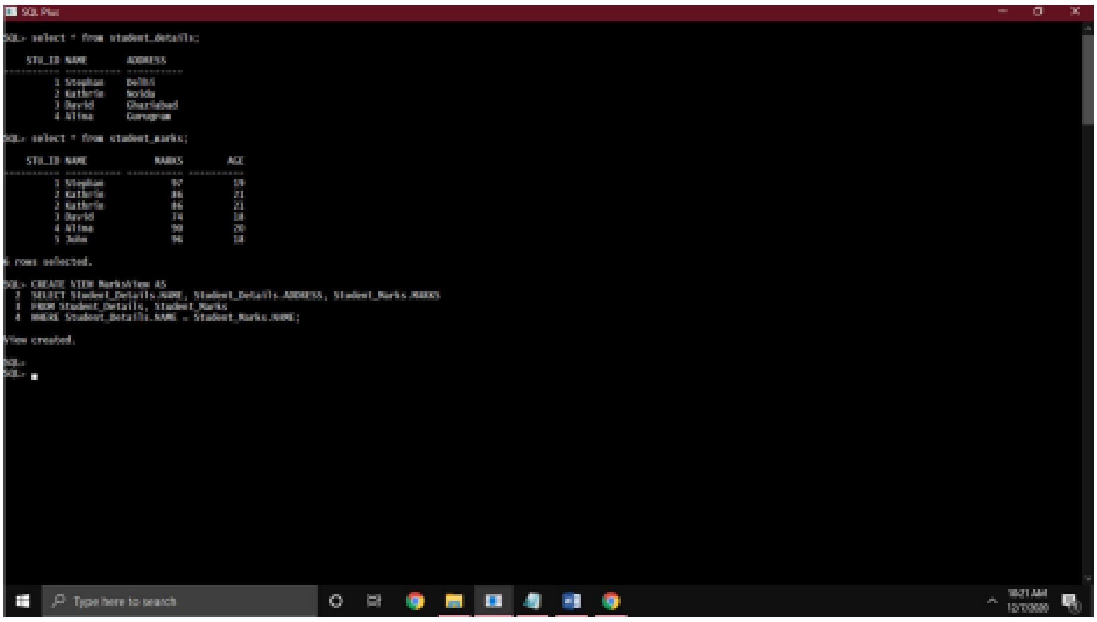
9.Sample Output



```
SQL> select * from student_details;

    STU_ID NAME       ADDRESS
---------- ---------- ----------
         1 Stephan    Delhi
         2 Kathrin    Noida
         3 David      Ghaziabad
         4 Alina      Gurugram

SQL> select * from student_marks;

    STU_ID NAME            MARKS        AGE
---------- ---------- ---------- ----------
         1 Stephan            97         19
         2 Kathrin            86         21
         2 Kathrin            86         21
         3 David              74         18
         4 Alina              90         20
         5 John               96         18

6 rows selected.

SQL> CREATE VIEW DetailsView AS
  2  SELECT NAME, ADDRESS
  3  FROM Student_Details
  4  WHERE STU_ID < 4;

View created.

SQL> Just like table query.
SP2-0734: unknown command beginning "Just like ..." - rest of line ignored.
SQL>
SQL>
```



```
SQL> select * from student_details;

    STU_ID NAME       ADDRESS
---------- ---------- ----------
         1 Stephan    Delhi
         2 Kathrin    Noida
         3 David      Ghaziabad
         4 Alina      Gurugram

SQL> select * from student_marks;

    STU_ID NAME            MARKS        AGE
---------- ---------- ---------- ----------
         1 Stephan            97         19
         2 Kathrin            86         21
         2 Kathrin            86         21
         3 David              74         18
         4 Alina              90         20
         5 John               96         18

6 rows selected.

SQL> CREATE VIEW DetailsView AS
  2  SELECT NAME, ADDRESS
  3  FROM Student_Details
  4  WHERE STU_ID < 4;

View created.

SQL> Just like table query.
SP2-0734: unknown command beginning "Just like ..." - rest of line ignored.
SQL>
SQL> select * from detailsview
  2  ;

NAME       ADDRESS
---------- ----------
Stephan    Delhi
Kathrin    Noida
David      Ghaziabad

SQL>
```



```
SQL> select * from student_details;

    STU_ID NAME       ADDRESS
---------- ---------- ----------
         1 Stephan    Delhi
         2 Katherin   Noida
         3 David      Ghaziabad
         4 Alina      Gurugram

SQL> select * from student_marks;

    STU_ID NAME            MARKS        AGE
---------- ---------- ---------- ----------
         1 Stephan            97         19
         2 Katherin           86         21
         2 Katherin           86         21
         3 David              74         18
         4 Alina              90         20
         5 John               96         18

6 rows selected.

SQL> CREATE VIEW MarksView AS
  2  SELECT Student_Details.NAME, Student_Details.ADDRESS, Student_Marks.MARKS
  3  FROM Student_Details, Student_Marks
  4  WHERE Student_Details.NAME = Student_Marks.NAME;

View created.

SQL>
SQL>
```

10.PRACTICE QUESTION

1.what is view
2.what kind of commands can be user with view
3. How can we update a view

11.EXERCISE QUESTIONS

1.How to destroy or drop a view

PRACTICAL QUESTION 7-: TRIGGERS

1.PRACTICAL SIGNIFICANCE :

Because a trigger resides in the database and anyone who has the required privilege can use it, a trigger lets you write a set of SQL statements that multiple applications can use. It lets you avoid redundant code when multiple programs need to perform the same database operation.

2. RELEVANT PROGRAM OUTCOMES : PO1,PO2,PO3,PO5,PO12

3. COMPETENCY AND PRACTICAL SKILLS :

The practical is expected to develop the following skills :

1. The trigger is mostly used for maintaining the integrity of the information on the database

4. PREREQUISITES:

1. Make sure that we have an active db created.

2. Triggers can be defined on the table, view, schema, or database with which the event is associated

5. RESOURCES REQUIRED :

1.systems should be installed with a licensed database software such as oracle,mysql etc

6. PRECAUTIONS:

1. Check Whether the computer is getting proper power or not.

2. Ensure the keyboard, mouse and monitor are properly working.

3. Ensure that there are no power fluctuations while executing the program.

4. Safe working conditions help prevent injury to people and damage to computer equipment.

5. A safe work space is clean, organized, and properly lighted. Everyone must understand and follow safety procedures.

6. Follow electrical safety guidelines to prevent electrical fires, injuries, and fatalities in the home and the workplace. Power supplies and CRT monitors contain high voltage.

7.ALGORITHM/PROCEDURE,QUERY:

TRIGGER FOR DISPLAYING GRADE OF THE STUDENT

```
SQL> create table stdn(rollno number(3),name varchar(2),m1 number(3),m2 number(3),m3
number(3),tot num

ber(3),avrg number(3),result varchar(10));

Table created.

SQL> create or replace trigger t1 before insert on stdn

  2  for each row

  3  begin

  4  :new.tot:=:new.m1+:new.m2+:new.m3;

  5  :new.avrg:=:new.tot/3;

  6  if(:new.m1>=50 and :new.m2>=50 and :new.m3>=50) then

  7  :new.result:='pass';

  8  else

  9  :new.result:='Fail';

 10  end if;

 11  end;

 12 /

Trigger created.

SQL> insert into stdn values(101,'SM',67,89,99,",",");

1 row created.

SQL> select * from stdn;
```

```
ROLLNO NA       M1       M2       M3      TOT     AVRG RESULT
------ ------ ------ ------ ------ ------ ------ ------------

   101        SM       67       89        9 9     255       85       pass
```

PROGRAM TO INDICATE INVALID CONDITION USING TRIGGER

```
SQL> create table emp (name varchar(10),empno number(3),age number(3));

Table created.
SQL>
  1  create or replace trigger t2 before insert on emp
  2  for each row
  3  when(new.age>100)
  4  begin
  5  RAISE_APPLICATION_ERROR(-20998,'INVALID ERROR');
  6* end;
SQL> /

Trigger created.

SQL> insert into emp values('nithya',101,24);

1 row created.

SQL> insert into emp values('nithya',101,103);
insert into emp values('nithya',101,103)
        *
```

```
ERROR at line 1:

ORA-20998: INVALID ERROR

ORA-06512: at "SCOTT.T2", line 2

ORA-04088: error during execution of trigger 'SCOTT.T2'
```

RESULT:

Thus triggers were implemented successfully.

8. PRACTICE QUESTIONS:

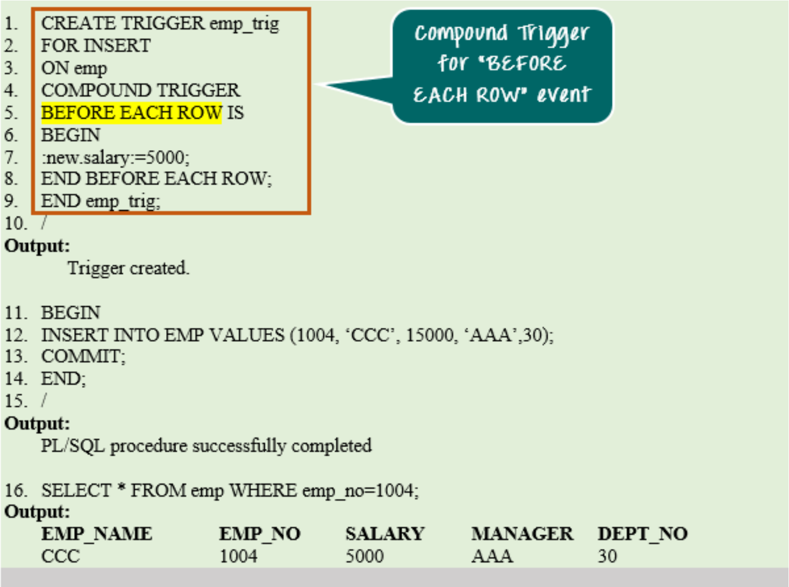
- what is a trigger ?
- define the syntax for triggers ?
- when do we use **BEFORE | AFTER ?**
- specify **on [table_name]**:
- why do we use [for each row]:?
- what is meant by action, event and conditions ?

9.EXERCISE QUESTIONS

Highschooler(ID int, name text, grade int); Friend(ID1 int, ID2 int); Likes(ID1 int, ID2 int);

- Write a trigger that automatically deletes students when they graduate, i.e., when their grade is updated to exceed 12.

10.outputs:



Compound Trigger for "BEFORE EACH ROW" event

```
CREATE TRIGGER emp_trig
FOR INSERT
ON emp
COMPOUND TRIGGER
BEFORE EACH ROW IS
BEGIN
:new.salary:=5000;
END BEFORE EACH ROW;
END emp_trig;
/
```

Output:

```
    Trigger created.
```

```
BEGIN
INSERT INTO EMP VALUES (1004, ‘CCC’, 15000, ‘AAA’,30);
COMMIT;
END;
/
```

Output:

```
PL/SQL procedure successfully completed
```

```
SELECT * FROM emp WHERE emp_no=1004;
```

Output:

```
EMP_NAME      EMP_NO     SALARY     MANAGER     DEPT_NO
CCC           1004       5000       AAA         30
```

PRACTICAL QUESTION 8: STORED PROCEDURE

1.PRACTICAL SIGNIFICANCE :

A stored procedure provides an important layer of security between the user interface and the database. It supports security through data access controls because end users may enter or change data, but do not write procedures.

2. RELEVANT PROGRAM OUTCOMES : PO1,PO2,PO3,PO5,PO12

3. COMPETENCY AND PRACTICAL SKILLS :

The practical is expected to develop the following skills :

1. By grouping SQL statements, a stored procedure allows them to be executed with a single call. This minimizes the use of slow networks, reduces network traffic, and improves round-trip response time. OLTP applications, in particular, benefit because result set processing eliminates network bottlenecks.

4. PREREQUISITES:

1. Make sure that we have an active db created and connected.

5. RESOURCES REQUIRED :

1.systems should be installed with a licensed database software such as oracle,mysql etc

6. PRECAUTIONS:

1. Check Whether the computer is getting proper power or not.

2. Ensure the keyboard, mouse and monitor are properly working.

3. Ensure that there are no power fluctuations while executing the program.

4. Safe working conditions help prevent injury to people and damage to computer equipment.

5. A safe work space is clean, organized, and properly lighted. Everyone must understand and follow safety procedures.

6. Follow electrical safety guidelines to prevent electrical fires, injuries, and fatalities in the home and the workplace. Power supplies and CRT monitors contain high voltage.

7.**DESCRIPTION -ALGORITHM/PROCEDURE,QUERY:**

SQL Stored Procedures for SQL Server

What is a Stored Procedure?

A stored procedure is a prepared SQL code that you can save, so the code can be reused over and over again.

So if you have an SQL query that you write over and over again, save it as a stored procedure, and then just call it to execute it.

You can also pass parameters to a stored procedure, so that the stored procedure can act based on the parameter value(s) that is passed.

Stored Procedure Syntax

```
CREATE PROCEDURE procedure_name
AS
sql_statement
GO;
```

Execute a Stored Procedure

```
EXEC procedure_name;
```

<u>PROCEDURE TO INSERT NUMBER</u>

```
SQL> create table emp1(id number(3),First_name varchar2(20));

Table created.

SQL> insert into emp1 values(101,'Nithya');

1 row created.

SQL> insert into emp1 values(102,'Maya');
```

```
1 row created.

SQL> select * from emp1;

      ID FIRST_NAME
------ -------------

      101  Nithya
      102  Maya

SQL> set serveroutput on;
SQL> create or replace
  2  procedure insert_num(p_num number)is
  3  begin
  4  insert into emp1(id,First_name) values(p_num,user);
  5  end insert_num;
  6 /
Procedure created.
SQL> exec insert_num(3);

PL/SQL procedure successfully completed.

SQL> select * from emp1;

       ID FIRST_NAME
------ -------------

      101   Nithya
```

```
       102   Maya

       103   SCOTT
```

8.EXCERCISE QUESTION : FUNCTION TO FIND FACTORIAL

```
SQL> create or replace function fact(n number)
  2  return number is
  3  i number(10);
  4  f number:=1;
  5  begin
  6  for i in 1..N loop
  7  f:=f*i;
  8  end loop;
  9  return f;
 10  end;
 11 /

Function created.
SQL> select fact(2) from dual;

  FACT(2)
------

     2
```

RESULT:

Thus procedures and functions were implemented successfully.

9.SAMPLE OUTPUT

Function created.

```
SQL> select fact(2) from dual;


          FACT(2)
  ------
              2
```

10.PRACTICE QUESTIONS:

1. What Is Stored Procedure?
2. Explain About The Process Which Takes Place To Execute A Stored Routine?
3. Where are The Procedures Are Stored In Database?

MINI PROJECT : DATABASE DESIGN AND IMPLEMENTATION PAY ROLL PROCESSING

STEPS:

1. Create a database for payroll processing which request the using SQL

2. Establish ODBC connection

3. In the administrator tools open data source ODBC

4. Click add button and select oracle in ORA home 90, click finish

5. A window will appear given the data source home as oracle and select TNS source name as lion and give the used id as SWTT

6. ADODC CONTROL FOR SALARY FORM:-

7. The above procedure must be follow except the table , A select the table as salary

8. Write appropriate Program in form each from created in VB from each from created in VB form project.

```
SQL>create table emp(eno number primary key,enamr varchar(20),age number,addr
varchar(20),DOB date,phno number(10));
Table created.

SQL>create table salary(eno number,edesig varchar(10),basic number,da number,hra
number,pf number,mc number,met number,foreign key(eno) references emp);
Table created.
TRIGGER to calculate DA,HRA,PF,MC
SQL> create or replace trigger employ
 2 after insert on salary
3 declare
4 cursor cur is select eno,basic from salary;
5 begin
6 for cur1 in cur loop
7 update salary set
8 hra=basic*0.1,da=basic*0.07,pf=basic*0.05,mc=basic*0.03 where hra=0; 9 end loop;
10 end;
11 / Trigger created.
```

PROGRAM FOR FORM 1

```
Private Sub emp_Click() Form
2.Show End
Sub Private
Sub exit_Click()
Unload Me
End Sub Private
Sub salary_Click()
Form3.Show
End Sub
```

PROGRAM FOR FORM 2

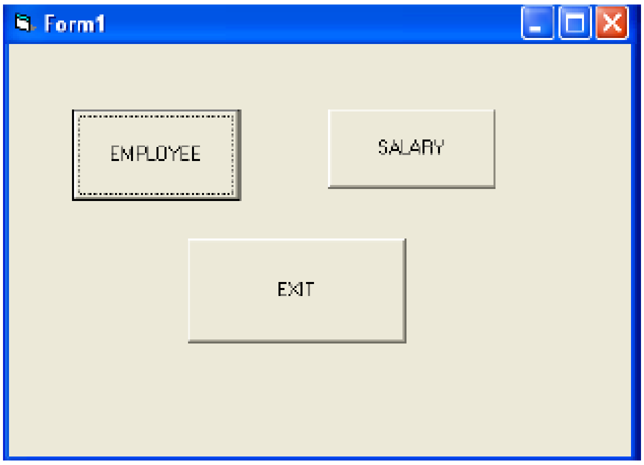
```
Private Sub add_Click()
Adodc1.Recordset.AddNew MsgBox "Record added"

End Sub Private
Sub clear_Click()
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
End Sub Private Sub delte_Click()
Adodc1.Recordset.Delete MsgBox "Record Deleted"
If Adodc1.Recordset.EOF = True
Then Adodc1.Recordset.MovePrevious
End If
End
Sub Private Sub exit_Click()
Unload Me
End Sub
Private Sub main_Click()
Form1.Show
End Sub
Private Sub modify_Click()
Adodc1.Recordset.Update
End Sub
```

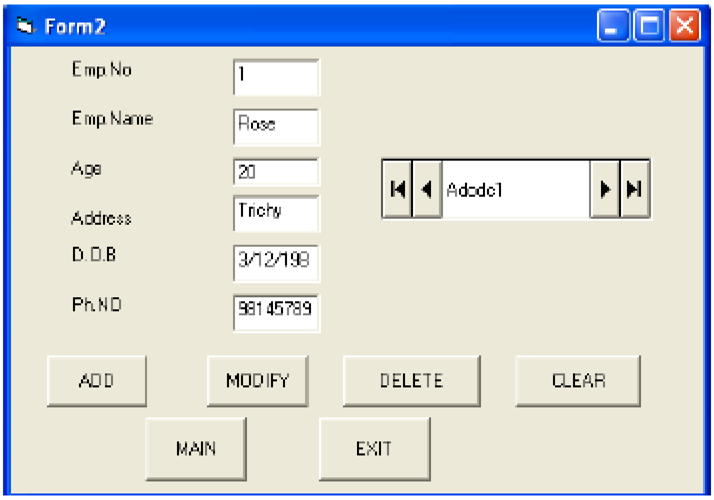
PROGRAM FOR FORM 3

```
Private Sub add_Click()
Adodc1.Recordset.AddNew MsgBox "Record added"
End Sub
Private Sub
clear_Click()
Text1.Text = ""
```

```
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
End Sub
Private Sub delte_Click()
Adodc1.Recordset.Delete MsgBox "Record Deleted"
If Adodc1.Recordset.EOF = True
Then Adodc1.Recordset.MovePrevious
End If
End Sub
Private Sub exit_Click()
Unload Me
End Sub
Private Sub main_Click()
Form1.Show
End Sub
Private Sub
modify_Click()
Adodc1.Recordset.Update
End Sub
```

Output:





Emp No 1

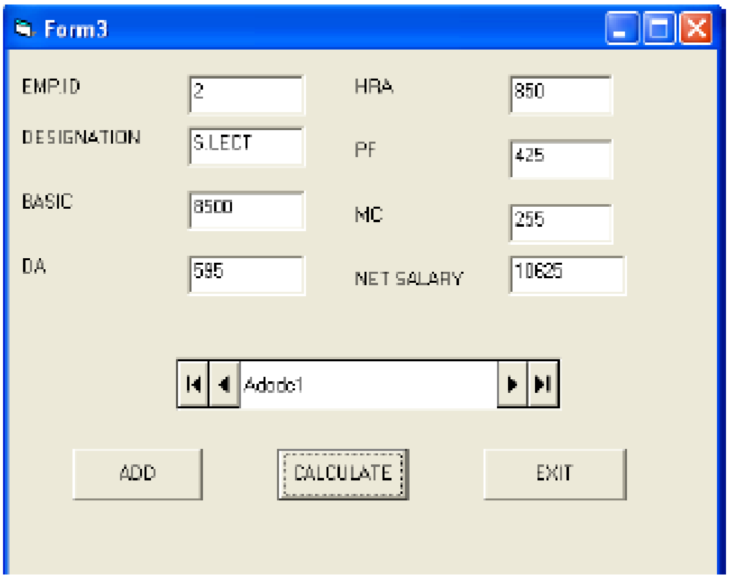
Emp Name Rose

Age 20

Address Trichy

D.O.B 3/12/198

Ph.NO 98145789



EMP.ID 2

DESIGNATION S.LECT

BASIC 8500

DA 595

HRA 850

PF 425

MC 255

NET SALARY 10625

RESULT:

Thus payroll system was designed and implemented successfully.