Vivekanand Education Society's

Institute of Technology

(Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)

Since 1962

Department of Artificial Intelligence and Data Science

Database Management Systems Labs Journal

Roll No.	37
Name	Manav Pahilwani
Class	D6AD
Subject	DBMS Lab
Grade:	

1. Lab Objectives:

	Description
1	To learn the basic concepts of Object-oriented Programming
2	To study Java Programming Language.
3	To study various concepts of Java Programming like multithreading, exception Handling, packages etc
4	To explain components of GUI based programming

2. Lab Outcome:

LO	Description
LO 1	Design ER /EER diagrams and convert to relational models for the real world application.
LO 2	Apply DDL, DML, DCL and TCL commands
LO 3	Write simple and complex queries
LO 4	Use PL / SQL Constructs.
LO 5	Demonstrate the concept of concurrent transactions execution and frontend-backend connectivity

3. LO/PO Mapping

LO	PO1	PO2	PO3	PO4	PO5	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
LO1	**1**	**3**	**2**	**1**	**1**	**1**	**2**	**1**	**1**	**2**	**2**	**1**
LO2	**-**	**3**	**3**	**2**	**2**	**1**	**2**	**1**	**2**	**3**	**2**	**2**
LO3	**-**	**3**	**3**	**2**	**2**	**1**	**2**	**1**	**2**	**3**	**1**	**3**
LO4	**-**	**2**	**2**	**1**	**2**	**-**	**1**	**1**	**-**	**1**	**-**	**2**
LO5	**1**	**3**	**2**	**2**	**2**	**1**	**2**	**1**	**1**	**3**	**1**	**2**

DBMS Lab

Name: Manav Pahilwani

INDEX

List of Experiments:

Sr No	List of Experiments	LO
1	Identify the case study and detail **statement of problem** and draw **Entity-Relationship (ER) / Extended Entity-Relationship (EER)** Model Using Draw.io.	LO1
2.	Mapping **ER/EER to the Relational schema model and creating a schema diagram for your system.**	LO1, LO2
3	Create a database using **Data Definition Language (DDL)** and apply required **Integrity Constraints** for the specified system.	LO2, LO3
4	Populate database using **DML Commands** for your specified System.	LO2, LO3
5	Perform **Simple queries, string manipulation operations.**	LO2, LO3
6	Write **Nested queries using (in, not in some any exist nt exits, with clause)**	LO2, LO3
7	Implement various types of **Joins and Views.**	LO3
8	Demonstrate **DCL** and **TCL** commands.	LO2
9	Implementation of **Functions and Stored Procedure** in PL-SQL.	LO4
10	Implement different types **of triggers.**	LO4
11	Implementation and demonstration of **Transaction and Concurrency control** techniques using locks.	LO2, LO4, LO5
12	Implement Database Connectivity (JDBC.ODBC)	LO2, LO3, LO5

Page No.
Date

Experiment :- 01.

Consider a university database, for a university registers office,
Courses include title, credits, number.
Students include course number, semester, time, class, room.
Instructor includes id no, name, department title.

ER :- ER model stands for Entity-Relationship model, It is a high level data model. It is used to define data elements & relationships for a specific system.

Components of ER diagram :-
Entity :- It may be an object, class, person or place, represented by triangles.
Weak entity :- Depends on another entity, doesn't contain any primary key of its own, represented by double rectangle.

Attribute :- It is used to define the property of an entity, represented by an eclipse.
e.g id, age, name etc.
Key attribute :- Key attribute is used to represent the main characteristics of an entity. It represents a primary key, represented by an eclipse with underlined text.
Composite attribute :- An attribute composed of many other attributes, represented by an eclipse, and are connected with an eclipse.
Multivalued attribute :- An attribute that can have more than one value, represented by double oval. e.g phone no.

Teacher's Sign.: _____

Derived attribute :- An attribute that can be derived from other attributes, can be represented by a dotted attribute.

Relationship :-
A relationship is used to describe the relation between entities. Diamond or rhombus is used to represent relationship.

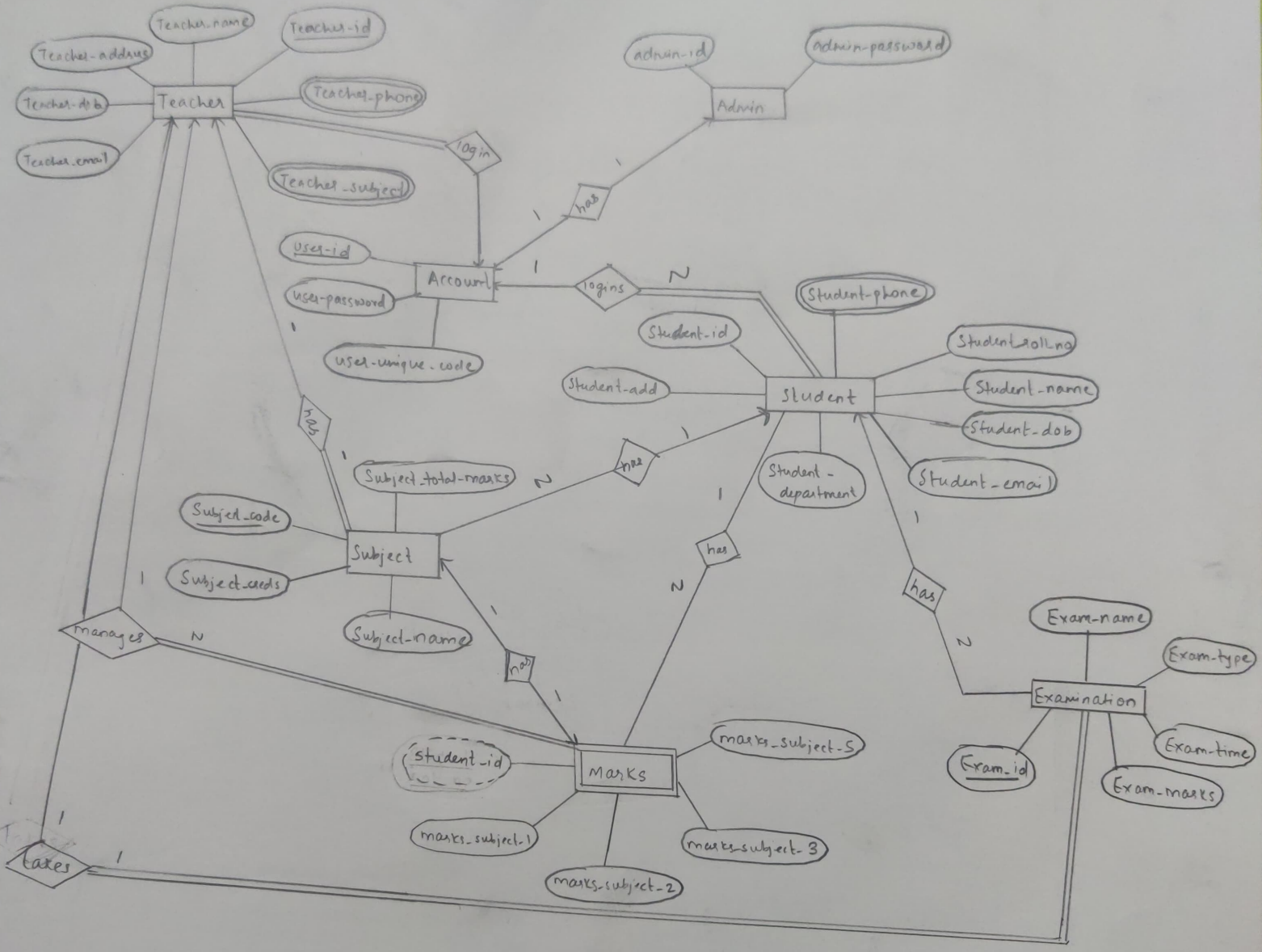
a. One to one :- one instance of an entity is associated with the relationship.
e.g a female can marry to one male, vice-versa.

b. One to many :- When only one ~~is~~ instance of the entity on the left, & more than one instance of an entity on the right. e.g one scientist can have many inventions, ~~only~~ ~~one~~ an invention can be claimed by 2 scientist.

c. Many to one :- many instances of entity on the left, and only one instance of an entity on the right.
e.g student enrolls for only one course, but a course may have many students.

d. Many to many :- More than one instance of the entity on the left, and more than one instance of the entity on the right. e.g Employee can be assign many projects, a project can have many employees.

ER Model for College Management Process



Experiment :- 02 (ER to Relational Schema)

All the table contents and their Commands :-

1. create table Admin (admin_id varchar(10) primary key not null, admin_password varchar(50) not null);
insert into Admin values ('Trustee','NewPassword');
insert into Admin values ("Principal","IamPrincipal");

2. create table Account(user_id int primary key, user_password varchar(20), user_unique_code varchar(20) not null);
insert into Account values(1001,'IamNew','HiIam1001');
insert into Account values(1002,'IamNew1','HiIam1002');
insert into Account values(1003,'IamNew2','HiIam1003');
insert into Account values(1004,'IamNew3','HiIam1004');

3. create table subject (subject_code int primary key, subject_creds int, subject_name varchar(30), subject_total_marks int);
insert into subject values(101, 150,'DBMS',125);
insert into subject values(102, 120,'OS',125);
insert into subject values(103, 120,'MP',125);
insert into subject values(104, 150,'AOA',125);
insert into subject values(105, 200,'Maths',125);

4. create table Teacher(teacher_id int primary key, teacher_name varchar(30), teacher_address varchar(50), teacher_dob varchar(20), teacher_email varchar(40), teacher_phoneno varchar(20), teacher_subject varchar(30), subject_code int foreign key (subject_code) references subject);
insert into teacher
values(111,'XYZ','Kurla','15-08-1980','xyz@ves.ac.in','9988776655','DBMS',101);
insert into teacher
values(222,'ABC','Thane','18-04-1975','abc@ves.ac.in','9182736455','OS',102);
insert into teacher
values(333,'PQR','Dadar','23-05-1979','pqr@ves.ac.in','9283746556','MP',103);
insert into teacher
values(444,'RTY','Chembur','18-11-1989','rty@ves.ac.in','9123456789','AOA',104);
insert into teacher
values(555,'ASD','Sion','20-08-1979','asd@ves.ac.in','9564728273','Maths',105);

5. create table Students(student_id int not null primary key, student_full_name varchar(50), student_phoneno varchar(50), student_email varchar(50), student_address varchar(100), student_rollno int, student_dob date, student_department varchar(10));
insert into students values(612, 'Manav Inder Pahilwani','9588660397','2020.manav.pahilwani@ves.ac.in','Flat 4 & 5 Achal Residency, Malegaon',37,12-12-2002,3001);
insert into students values(614, 'Om Gaydhane','8369000474','2020.om.gaydhane@ves.ac.in','Airoli',15,14-02-2002,3001);
insert into students values(645, 'Madhusudhana Naidu','9321451547','2020.madhusudhana.naidu@ves.ac.in','Seawoods',36,31-4-2002,3001);

```
insert into students values(669, 'Akshat
Tiwari','9876543210','2020.akshant.tiwari@ves.ac.in','Chembur',62,21-08-2002,3001);
```

6. create table Examination (exam_id int primary key, exam_name varchar(10), exam_time varchar(20),exam_marks int,exam_mode varchar(10));

```
insert into examination values(320, 'Maths-4','21-05-21 3:00:00',80,'ONLINE');
insert into examination values(310, 'DBMS','23-05-21 3:00:00',80,'ONLINE');
insert into examination values(330, 'AOA','25-05-21 3:00:00',80,'ONLINE');
insert into examination values(300, 'OS','27-05-21 3:00:00',80,'ONLINE');
insert into examination values(350, 'MP','29-05-21 3:00:00',80,'ONLINE');
```

7. create table Marks(student_id int primary key foreign key(student_id) references students, marks_sub1 int, marks_sub2 int, marks_sub3 int, marks_sub4 int, marks_sub5 int);

```
insert into marks values(612, 80,78,75,76,70);
insert into marks values(614, 79,65,78,80,78);
insert into marks values(645, 67,58,70,80,80);
insert into marks values(669, 60,60,60,60,59);
```

Experiment :- 02 (ER to Relational Schema)

```
1> select * from Admin;
2> go
admin_id   admin_password
---------- --------------------------------------------------
```

```
1> select * from Admin;
2> go
admin_id   admin_password
---------- --------------------------------------------------
Principal  IamPrincipal
Trustee    NewPassword
```

```
1> select * from Teacher;
2> go
teacher_id  teacher_name                teacher_address                                 teacher_dob      teacher_email                        teacher_phoneno teacher_subject
---------- ------------------------------ ---------------------------------------------- ---------------- ------------------------------------ --------------- ------------------------------

(0 rows affected)
1>
```

```
1> insert into Account values(1001,'IamNew','HiIam1001'
2> );
3> go

(1 rows affected)
1> insert into Account values(1002,'IamNew1','HiIam1002');
2> go

(1 rows affected)
1> insert into Account values(1003,'IamNew2','HiIam1003');
2> go

(1 rows affected)
1> insert into Account values(1004,'IamNew3','HiIam1004');
2> go

(1 rows affected)
1> select * from Account;
2> go
user_id     user_password        user_unique_code
----------- -------------------- --------------------
       1001 IamNew               HiIam1001
       1002 IamNew1              HiIam1002
       1003 IamNew2              HiIam1003
       1004 IamNew3              HiIam1004

(4 rows affected)
```

```
1> insert into examination values(320, 'Maths-4','21-05-21 3:00:00',80,'ONLINE');
2> go

(1 rows affected)
1> insert into examination values(310, 'DBMS','23-05-21 3:00:00',80,'ONLINE');
2> go

(1 rows affected)
1> insert into examination values(330, 'AOA','25-05-21 3:00:00',80,'ONLINE');
2> go

(1 rows affected)
1>  insert into examination values(300, 'OS','27-05-21 3:00:00',80,'ONLINE');
2> go

(1 rows affected)
1>  insert into examination values(350, 'MP','29-05-21 3:00:00',80,'ONLINE');
2> go

(1 rows affected)
1> select * from examination;
2> go
exam_id     exam_name  exam_time             exam_marks  exam_mode
----------- ---------- --------------------- ----------- ----------
        300 OS         27-05-21 3:00:00               80 ONLINE
        310 DBMS       23-05-21 3:00:00               80 ONLINE
        320 Maths-4    21-05-21 3:00:00               80 ONLINE
        330 AOA        25-05-21 3:00:00               80 ONLINE
        350 MP         29-05-21 3:00:00               80 ONLINE

(5 rows affected)
```

```
1> select * from Teacher
2> ;
3> select * from Students;
4> select * from Examination;
5> select * from account;
6> select * from marks;
7> select * from admin;
8> select * from subject;
9> go
teacher_id  teacher_name                teacher_address                                   teacher_dob      teacher_email                       teacher_phoneno teacher_subject             subject_code
----------- --------------------------- ------------------------------------------------- ---------------- ----------------------------------- --------------- --------------------------- ------------


(0 rows affected)
student_id  student_full_name                          student_phoneno student_email                                 student_address
                 student_rollno student_dob      student_department
----------- ---------------------------------------- --------------- ------------------------------------------- ----------------------------------------------------------------------------------------------------
-------------------- -------------- ---------------- ------------------

(0 rows affected)
exam_id     exam_name  exam_time             exam_marks  exam_mode
----------- ---------- --------------------- ----------- ----------

(0 rows affected)
user_id     user_password       user_unique_code
----------- ------------------- -------------------

(0 rows affected)
student_id  marks_sub1  marks_sub2  marks_sub3  marks_sub4  marks_sub5
----------- ----------- ----------- ----------- ----------- -----------

(0 rows affected)
admin_id   admin_password
---------- --------------------------------------------------
Principal  IamPrincipal
Trustee    NewPassword

(2 rows affected)
subject_code subject_creds subject_name                subject_total_marks
------------ ------------ ------------------------------ -------------------

(0 rows affected)
```

```
1> insert into marks values(612, 80,78,75,76,70);
2> go

(1 rows affected)
1> insert into marks values(614, 79,65,78,80,78);
2> go

(1 rows affected)
1> insert into marks values(645, 67,58,70,80,80);
2> go

(1 rows affected)
1> insert into marks values(669, 60,60,60,60,59);
2> go

(1 rows affected)
1> select * from marks;
2> go
student_id  marks_sub1  marks_sub2  marks_sub3  marks_sub4  marks_sub5
----------- ----------- ----------- ----------- ----------- -----------
         612          80          78          75          76          70
         614          79          65          78          80          78
         645          67          58          70          80          80
         669          60          60          60          60          59
```

```
1> select * from Admin;
2> go
admin_id   admin_password
---------- ------------------------------------------------------------
Principal  IamPrincipal
Trustee    NewPassword

(2 rows affected)
1> select * from Account;
2> go
user_id     user_password       user_unique_code
----------- ------------------- -------------------
      1001 IamNew              Hilam1001
      1002 IamNew1             Hilam1002
      1003 IamNew2             Hilam1003
      1004 IamNew3             Hilam1004

(4 rows affected)
1> select * from teacher;
2> go
teacher_id  teacher_name                teacher_address                                    teacher_dob         teacher_email                      teacher_phoneno     teacher_subject             sub
ject_code
----------- --------------------------- -------------------------------------------------- ------------------- ---------------------------------- ------------------- --------------------------- ---
----------
        111 XYZ                         Kurla                                              15-08-1980          xyz@ves.ac.in                      9988776655          DBMS
     101
        222 ABC                         Thane                                              18-04-1975          abc@ves.ac.in                      9182736455          OS
     102
        333 PQR                         Dadar                                              23-05-1979          pqr@ves.ac.in                      9283746556          MP
     103
        444 RTY                         Chembur                                            18-11-1989          rty@ves.ac.in                      9123456789          AOA
     104
        555 ASD                         Sion                                               20-08-1979          asd@ves.ac.in                      9564728273          Maths
     105

(5 rows affected)
1> select * from Subject
2> ;
3> go
subject_code subject_creds subject_name                 subject_total_marks
------------ ------------- ---------------------------- -------------------
         101           150 DBMS                                         125
         102           120 OS                                           125
         103           120 MP                                           125
         104           150 AOA                                          125
         105           200 Maths                                        125
```

Experiment :- 02 (ER to Relational Schema)

```
1> select * from students;
2> go
student_id  student_full_name                           student_phoneno                                  student_email                                 student_address
                                                    student_rollno student_dob          student_department
----------- ------------------------------------------- ---------- -------------------- ------------------
----------------------------------------------------------- ------------ -------------------- ------------------
        612 Manav Inder Pahilwani                       9588660397                                       2020.manav.pahilwani@ves.ac.in                Flat 4 & 5 Achal Residency, Malegaon
                                                                37 -2002             3001
        614 Om Gaydhane                                 8369000474                                       2020.om.gaydhane@ves.ac.in                    Airoli
                                                                15 -1990             3001
        645 Madhusudhana Naidu                          9321451547                                       2020.madhusudhana.naidu@ves.ac.in             Seawoods
                                                                36 -1975             3003
        669 Akshat Tiwari                               9876543210                                       2020.akshant.tiwari@ves.ac.in                 Chembur
                                                                62 -1989             3001

(4 rows affected)
1> select * from examination;
2> go
exam_id     exam_name  exam_time           exam_marks  exam_mode
----------- ---------- ------------------- ----------- ----------
        300 OS         27-05-21 3:00:00             80 ONLINE
        310 DBMS       23-05-21 3:00:00             80 ONLINE
        320 Maths-4    21-05-21 3:00:00             80 ONLINE
        330 AOA        25-05-21 3:00:00             80 ONLINE
        350 MP         29-05-21 3:00:00             80 ONLINE

(5 rows affected)
1> select * from marks;
2> go
student_id  marks_sub1  marks_sub2  marks_sub3  marks_sub4  marks_sub5
----------- ----------- ----------- ----------- ----------- -----------
        612          80          78          75          76          70
        614          79          65          78          80          78
        645          67          58          70          80          80
        669          60          60          60          60          59

(4 rows affected)
1>
```

```
1> insert into students values(612, 'Manav Inder Pahilwani','9588660397','2020.manav.pahilwani@ves.ac.in','Flat 4 & 5 Achal Residency, Malegaon',37,12-12-2002,3001);
2> go

(1 rows affected)
1> insert into students values(614, 'Om Gaydhane','8369000474','2020.om.gaydhane@ves.ac.in','Airoli',15,14-02-2002,3001);
2> go

(1 rows affected)
1> insert into students values(645, 'Madhusudhana Naidu','9321451547','2020.madhusudhana.naidu@ves.ac.in','Seawoods',36,31-4-2002,3003);
2> go

(1 rows affected)
1> insert into students values(669, 'Akshat Tiwari','9876543210','2020.akshant.tiwari@ves.ac.in','Chembur',62,21-08-2002,3001);
2> go

(1 rows affected)
1> select * from students;
2> go
student_id  student_full_name                           student_phoneno                                  student_email                                 student_address
                                                    student_rollno student_dob          student_department
----------- ------------------------------------------- ---------- -------------------- ------------------
----------------------------------------------------------- ------------ -------------------- ------------------
        612 Manav Inder Pahilwani                       9588660397                                       2020.manav.pahilwani@ves.ac.in                Flat 4 & 5 Achal Residency, Malegaon
                                                                37 -2002             3001
        614 Om Gaydhane                                 8369000474                                       2020.om.gaydhane@ves.ac.in                    Airoli
                                                                15 -1990             3001
        645 Madhusudhana Naidu                          9321451547                                       2020.madhusudhana.naidu@ves.ac.in             Seawoods
                                                                36 -1975             3003
        669 Akshat Tiwari                               9876543210                                       2020.akshant.tiwari@ves.ac.in                 Chembur
                                                                62 -1989             3001
```

```
1> insert into subject values(101, 150,'DBMS',125);
2> go

(1 rows affected)
1> insert into subject values(102, 120,'OS',125);
2> go

(1 rows affected)
1> insert into subject values(103, 120,'MP',125);
2> go

(1 rows affected)
1> insert into subject values(104, 150,'AOA',125);
2> go

(1 rows affected)
1> insert into subject values(105, 200,'Maths',125);
2> go

(1 rows affected)
1> select * from subject;
2> go
subject_code subject_creds subject_name                    subject_total_marks
------------ ------------- ------------------------------ -------------------
         101           150 DBMS                                           125
         102           120 OS                                             125
         103           120 MP                                             125
         104           150 AOA                                            125
         105           200 Maths                                          125

(5 rows affected)
```

```
1> insert into teacher values(111,'XYZ','Kurla','15-08-1980','xyz@ves.ac.in','9988776655','DBMS',101);
2> go

(1 rows affected)
1> insert into teacher values(222,'ABC','Thane','18-04-1975','abc@ves.ac.in','9182736455','OS',102);
2> go

(1 rows affected)
1> insert into teacher values(333,'PQR','Dadar','23-05-1979','pqr@ves.ac.in','9283746556','MP',103);
2> go

(1 rows affected)
1> insert into teacher values(444,'RTY','Chembur','18-11-1989','rty@ves.ac.in','9123456789','AOA',104);
2> go

(1 rows affected)
1> insert into teacher values(555,'ASD','Sion','20-08-1979','asd@ves.ac.in','9564728273','Maths',105);
2> go

(1 rows affected)
1> select * from teacher;
2> go
teacher_id  teacher_name                teacher_address                           teacher_dob         teacher_email                    teacher_phoneno     teacher_subject             sub
ject_code
---------- ---------------------------- ----------------------------------------- ------------------- -------------------------------- ------------------- --------------------------- ---
--------
       111 XYZ                          Kurla                                     15-08-1980          xyz@ves.ac.in                    9988776655          DBMS
     101
       222 ABC                          Thane                                     18-04-1975          abc@ves.ac.in                    9182736455          OS
     102
       333 PQR                          Dadar                                     23-05-1979          pqr@ves.ac.in                    9283746556          MP
     103
       444 RTY                          Chembur                                   18-11-1989          rty@ves.ac.in                    9123456789          AOA
     104
       555 ASD                          Sion                                      20-08-1979          asd@ves.ac.in                    9564728273          Maths
     105

(5 rows affected)
1>
```

```
1> create table Examination (exam_id int primary key, exam_name varchar(10), exam_time time,exam_marks int,exam_mode varchar(10));
2> go
1> select * from Examination
2> ;
3> go
exam_id     exam_name  exam_time             exam_marks  exam_mode
----------- ---------- --------------------- ----------- ----------

(0 rows affected)
1>
```

Experiment :- 02 (ER to Relational Schema)

```
1> select * from students where student_full_name like '%____%'
2> go
student_id  student_full_name                                   student_phoneno                                      student_email                                      student_address
                                                   student_rollno student_dob          student_department
----------- --------------------------------------------------- ---------------------------------------------------- -------------------------------------------------- --------------------------------------------------
--------------------------------------------------- -------------- -------------------- ------------------
        612 Manav Inder Pahilwani                               9588660397                                           2020.manav.pahilwani@ves.ac.in                     Flat 4 & 5 Achal Residency, Malegaon
                                                               37 -2002             3001
        614 Om Gaydhane                                         8369000474                                           2020.om.gaydhane@ves.ac.in                         Airoli
                                                               15 -1990             3001
        645 Madhusudhana Naidu                                  9321451547                                           2020.madhusudhana.naidu@ves.ac.in                  Seawoods
                                                               36 -1975             3003
        669 Akshat Tiwari                                       9876543210                                           2020.akshant.tiwari@ves.ac.in                      Chembur
                                                               62 -1989             3001

(4 rows affected)
```

```
1> select * from students where student_full_name like 'M%'
2> go
student_id  student_full_name                                   student_phoneno                                      student_email                                      student_address
                                                   student_rollno student_dob          student_department
----------- --------------------------------------------------- ---------------------------------------------------- -------------------------------------------------- --------------------------------------------------
--------------------------------------------------- -------------- -------------------- ------------------
        612 Manav Inder Pahilwani                               9588660397                                           2020.manav.pahilwani@ves.ac.in                     Flat 4 & 5 Achal Residency, Malegaon
                                                               37 -2002             3001
        645 Madhusudhana Naidu                                  9321451547                                           2020.madhusudhana.naidu@ves.ac.in                  Seawoods
                                                               36 -1975             3003

(2 rows affected)
```

```
1> select upper(student_full_name) from students;
2> go

--------------------------------------------------
MANAV INDER PAHILWANI
OM GAYDHANE
MADHUSUDHANA NAIDU
AKSHAT TIWARI

(4 rows affected)
```

```
1> select lower(student_full_name) from students;
2> go

--------------------------------------------------
manav inder pahilwani
om gaydhane
madhusudhana naidu
akshat tiwari

(4 rows affected)
```

```
1> select * from marks where marks_sub1 not between 70 and 80;
2> go
student_id  marks_sub1  marks_sub2  marks_sub3  marks_sub4  marks_sub5
----------- ----------- ----------- ----------- ----------- -----------
        645          67          58          70          80          80
        669          60          60          60          60          59
```

```
1> select * from teacher as t, subject as s where t.subject_code=s.subject_code;
2> go
teacher_id  teacher_name                teacher_address                                  teacher_dob          teach
er_email                          teacher_phoneno      teacher_subject              subject_code subject_code subjec
t_creds subject_name                 subject_total_marks
---------- ------------------------------ ---------------------------------------------------- -------------------- -----
---------------------------------- ------------------- ------------------------------ ------------ ------------ ------
------- ------------------------------ -------------------
       111 XYZ                            Kurla                                            15-08-1980           xyz@v
es.ac.in                          9988776655           DBMS                                   101          101
    150 DBMS                                       125
       222 ABC                            Thane                                            18-04-1975           abc@v
es.ac.in                          9182736455           OS                                     102          102
    120 OS                                         125
       333 PQR                            Dadar                                            23-05-1979           pqr@v
es.ac.in                          9283746556           MP                                     103          103
    120 MP                                         125
       444 RTY                            Chembur                                          18-11-1989           rty@v
es.ac.in                          9123456789           AOA                                    104          104
    150 AOA                                        125
       555 ASD                            Sion                                             20-08-1979           asd@v
es.ac.in                          9564728273           Maths                                  105          105
    200 Maths                                      125

(5 rows affected)
```

```
1> select sum(marks_sub1), max(marks_sub1),student_id from marks group by student_id;
2> go
                        student_id
----------- ----------- -----------
         80          80         612
         79          79         614
         67          67         645
         60          60         669
```

```
1> select avg(marks_sub1), avg(marks_sub2),avg(marks_sub3), avg(marks_sub4), avg(marks_sub4) from marks;
2> go

----------- ----------- ----------- ----------- -----------
        71          65          70          74          74
```

```
1> select student_id, max(marks_sub1) from marks group by student_id;
2> go
student_id
----------- -----------
        612          80
        614          79
        645          67
        669          60
```

```
1> select * from teacher where subject_code in(select subject_code from subject) order by teacher_id;
2> go
teacher_id  teacher_name                 teacher_address                                  teacher_dob          teach
er_email                          teacher_phoneno      teacher_subject               subject_code
----------- ------------------------------ ---------------------------------------------- -------------------- -----
----------------------------------- -------------------- ------------------------------ ------------

        111 XYZ                          Kurla                                          15-08-1980           xyz@v
es.ac.in                          9988776655           DBMS                             101
        222 ABC                          Thane                                          18-04-1975           abc@v
es.ac.in                          9182736455           OS                               102
        333 PQR                          Dadar                                          23-05-1979           pqr@v
es.ac.in                          9283746556           MP                               103
        444 RTY                          Chembur                                        18-11-1989           rty@v
es.ac.in                          9123456789           AOA                              104
        555 ASD                          Sion                                           20-08-1979           asd@v
es.ac.in                          9564728273           Maths                            105

(5 rows affected)
```

```
1> create table Marks( student_id int primary key foreign key(student_id) references students, marks_sub1 int, marks_sub2 int, marks_sub3 int, marks_sub4 int, marks_sub5 int);
2> go
1> select * from table Marks;
2> go
Msg 156, Level 15, State 1, Server MANNS-LAPTOP, Line 1
Incorrect syntax near the keyword 'table'.
1> select * from Marks;
2> go
student_id  marks_sub1  marks_sub2  marks_sub3  marks_sub4  marks_sub5
----------- ----------- ----------- ----------- ----------- -----------

(0 rows affected)
```

Experiment 3

Aim :- Create a database using DDL & apply integrity constraints for the specified system.

Theory :-

DDL Queries are used:

a. Create the structure of the database.
b. Change / Modify the database.
c. Delete a table from database.
d. Change name of the table.
e. Delete rows of table.

CREATE Query :-

It is used to create a table in database.

ALTER Query :-

It is used to change the structure of an existing table.

Two types of queries :-

1] ALTER Add
2] ALTER Modify

a. **ALTER add :-** This is used to add a new column to an existing table.

b. **ALTER modify :-** It is used when the datatype or size of existing table is to be modified.

c. **Drop :-** This Query is used to completely delete a table from database.

d. Truncate :- It deletes only rows of the table.

e. Rename :- It is used to rename the table.

Conclusion :- Thus we learnt about all DDL Queries
Create,
Alter,
Drop,
Truncate,
Rename in detail. ~~with~~

Experiment - 3

Aim: Create a database using Data Definition Language (DDL) and apply integrity constraints for the specified System

Name :- Manav Pahilwani
Class :- D6AD
Roll No:- 37

Query to Create tablespace, create user, grant privilege to user :-

```
Enter user-name: system
Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> create tablespace Manav
  2  datafile 'Manav.dat'
  3  size 50M
  4  online;

Tablespace created.

SQL> create user Manav identified by Hamilton
  2  ;

User created.

SQL> Grant DBA to Manav
  2  ;

Grant succeeded.
```

DDL Queries with output :-

1. Customer Table:-

```
SQL> create table customer(cid int, cname varchar(20), address char(10));

Table created.

SQL> Alter table customer ADD phoneNo numeric(10);

Table altered.

SQL> Alter table customer Modify address varchar(20);

Table altered.
```

Create employee table :-

```
SQL> create table employee(ssn int primary key, ename varchar(20) not null, salary int, dno int);

Table created.
```

2. Department Table :-

```
SQL> create table dept1(dno int primary key, dname varchar(20),mgrssn int references employee, startdate date);

Table created.
```

```
SQL> alter table employee add constraint fk_dnp foreign key(dno) references dept1(dno);

Table altered.
```

3. Deptlocation Table :-

```
SQL> create table deptloc(dno int,dloc varchar(20), primary key(dno, dloc), foreign key(dno)references dept1(dno));

Table created.
```

4. Project Table and Table dependent :-

```
SQL> create table project(pno int primary key, pname varchar(20), dno int references dept1(dno));

Table created.

SQL> create table dependent(ssn int, depname varchar(20),relation varchar(20), primary key(ssn,depname),foreign key(ssn)references employee(ssn));

Table created.
```

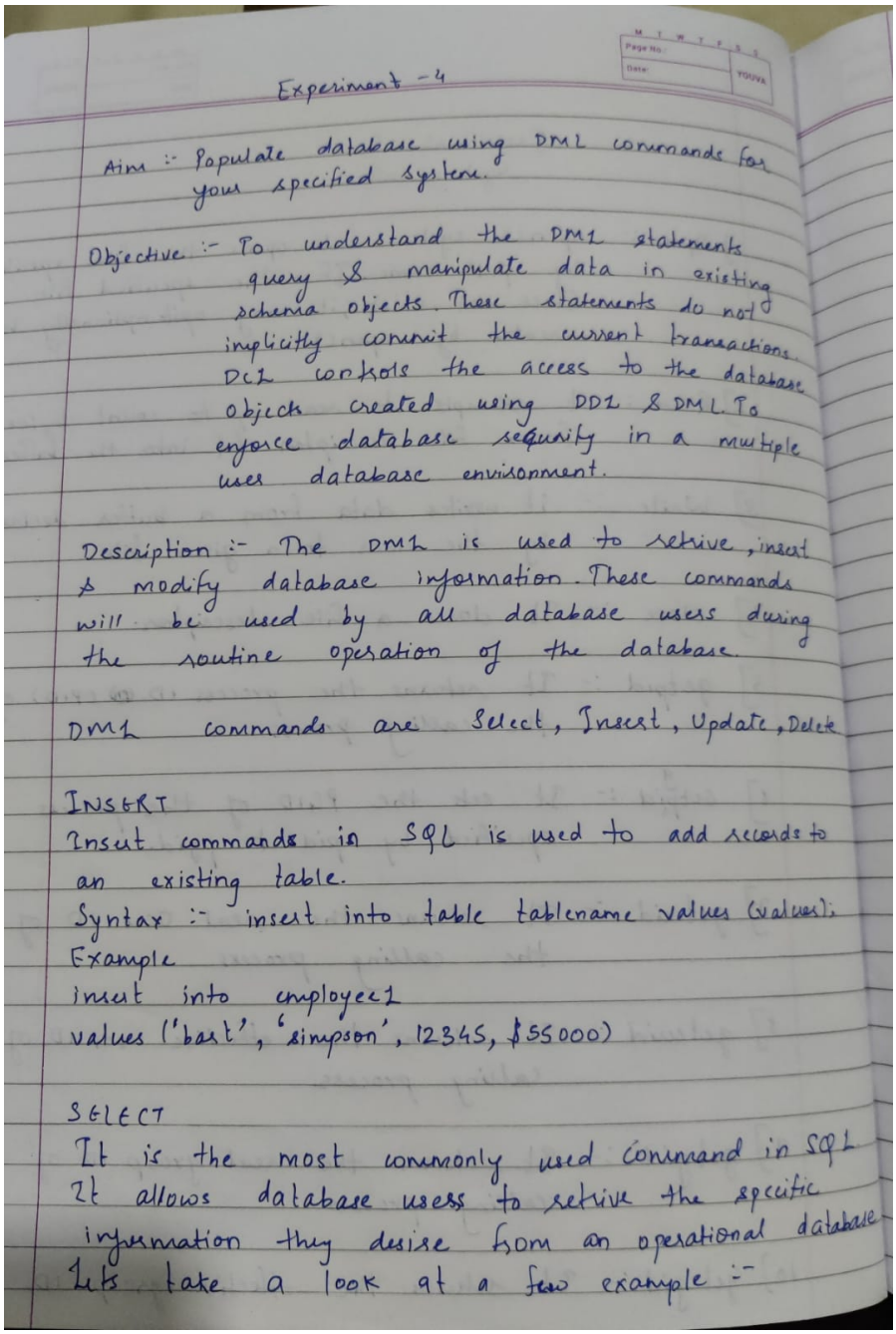
```
SQL> alter table dependent drop primary key;

Table altered.

SQL> alter table dependent add constraint pk_ssn primary key(ssn, depname);

Table altered.
```

Experiment - 4

Name :- Manav Pahilwani
Roll No :- 37
Class :- D6AD

Aim :- Populate database using DML Commands for your specified System.

Theory :-



Experiment - 4

Aim :- Populate database using DML commands for your specified system.

Objective :- To understand the DML statements query & manipulate data in existing schema objects. These statements do not implicitly commit the current transactions. DCL controls the access to the database objects created using DDL & DML. To enforce database security in a multiple user database environment.

Description :- The DML is used to retrive, insert & modify database information. These commands will be used by all database users during the routine operation of the database.

DML commands are Select, Insert, Update, Delete

INSERT
Insert commands in SQL is used to add records to an existing table.
Syntax :- insert into table tablename values (value);
Example
insert into employee1
values ('bart', 'simpson', 12345, $55000)

SELECT
It is the most commonly used command in SQL
It allows database users to retrive the specific information they desire from an operational database
Lets take a look at a few example :-

Syntax :- select <attribute list> from <list of tables>
where predicate;

Select * from employee1

Alternatively users may want to limit the attributes that are retrieved from the database.
The following SQL command would retrive only that information

Select last_name from employee1.

The where clause can be used to limit the records that are retrieved to ~~that~~ those that meet specified criteria.

select *FROM employee1 where salary > 50000.

Update :-
The update command can be used to modify information contained within a table, either in bulk or individually.
~~syntax :- update employee1 sala set salary = salary * 1.03.~~
syntax :- update ~~employee1~~ tablename set predicate;

Update employee1 set salary = salary * 1.03.
update employee1 set salary = salary + 5000.

Delete :- Delete command where clause can be used to remove his record from the employeeh table.
Syntax :- delete from tablename where predicate;

Delete from employee where employee_id = 12345;

Code Snippets :-

Employee Table :-

```
SQL> create table employee1(first_name varchar(20), last_name varchar(20), employee_id int unique, salary int);

Table created.

SQL> desc employee1
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 FIRST_NAME                                         VARCHAR2(20)
 LAST_NAME                                          VARCHAR2(20)
 EMPLOYEE_ID                                        NUMBER(38)
 SALARY                                             NUMBER(38)
```

```
SQL> insert into employee1 values('bart','simpson',1234,55000);
```

```
SQL> update employee1 set salary = salary*1.03;

1 row updated.
```

Book Table :-

```
SQL> desc book
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BOOK_ID                                   NOT NULL NUMBER(38)
 BOOK_NAME                                          VARCHAR2(150)
 BOOK_ISSDN                                         NUMBER(38)
 BOOK_AUTHOR                                        VARCHAR2(50)
 BOOK_QTY                                           NUMBER(38)
 BOOK_AVAILABILITY                                  VARCHAR2(10)
 BOOK_PUBLICATION                                   VARCHAR2(150)
 DEPT_CODE                                          VARCHAR2(10)
 REF_ID                                             NUMBER(38)
```

```
   BOOK_ID
----------
BOOK_NAME
--------------------------------------------------------------------------------
BOOK_ISSDN BOOK_AUTHOR                                      BOOK_QTY
---------- -------------------------------------------------- ----------
BOOK_AVAIL
----------
BOOK_PUBLICATION
--------------------------------------------------------------------------------
DEPT_CODE      REF_ID
---------- ----------
      4001

   BOOK_ID
----------
BOOK_NAME
--------------------------------------------------------------------------------
BOOK_ISSDN BOOK_AUTHOR                                      BOOK_QTY
---------- -------------------------------------------------- ----------
BOOK_AVAIL
----------
BOOK_PUBLICATION
--------------------------------------------------------------------------------
DEPT_CODE      REF_ID
---------- ----------
Database Management System
```

```
   BOOK_ID
----------
BOOK_NAME
--------------------------------------------------------------------------------
BOOK_ISSDN BOOK_AUTHOR                                      BOOK_QTY
---------- -------------------------------------------------- ----------
BOOK_AVAIL
----------
BOOK_PUBLICATION
--------------------------------------------------------------------------------
DEPT_CODE      REF_ID
---------- ----------
      1234 G.K.Gupta
```

```
BOOK_NAME
--------------------------------------------------------------------------------
BOOK_ISSDN BOOK_AUTHOR                                          BOOK_QTY
---------- -------------------------------------------------- ----------
BOOK_AVAIL
----------
BOOK_PUBLICATION
--------------------------------------------------------------------------------
DEPT_CODE      REF_ID
---------- ----------
      4002

   BOOK_ID
----------
BOOK_NAME
--------------------------------------------------------------------------------
BOOK_ISSDN BOOK_AUTHOR                                          BOOK_QTY
---------- -------------------------------------------------- ----------
BOOK_AVAIL
----------
BOOK_PUBLICATION
--------------------------------------------------------------------------------
DEPT_CODE      REF_ID
---------- ----------
Advanced Java

   BOOK_ID
----------
BOOK_NAME
--------------------------------------------------------------------------------
BOOK_ISSDN BOOK_AUTHOR                                          BOOK_QTY
---------- -------------------------------------------------- ----------
BOOK_AVAIL
----------
BOOK_PUBLICATION
--------------------------------------------------------------------------------
DEPT_CODE      REF_ID
---------- ----------
       156 Schildt                                                  56
```

```
   BOOK_ID
----------
BOOK_NAME
--------------------------------------------------------------------------------
BOOK_ISSDN BOOK_AUTHOR                                              BOOK_QTY
---------- -------------------------------------------------- ----------
BOOK_AVAIL
----------
BOOK_PUBLICATION
--------------------------------------------------------------------------------
DEPT_CODE      REF_ID
---------- ----------
Y

   BOOK_ID
----------
BOOK_NAME
--------------------------------------------------------------------------------
BOOK_ISSDN BOOK_AUTHOR                                              BOOK_QTY
---------- -------------------------------------------------- ----------
BOOK_AVAIL
----------
BOOK_PUBLICATION
--------------------------------------------------------------------------------
DEPT_CODE      REF_ID
---------- ----------
Tata McGraw Hill
```

Records Table :-

```
SQL> select * from records;

   BOOK_ID DEPT_CODE
---------- ----------
      4001 3001
      4005 3002
      4003 3003
      4002 3001
      4004 3005
```

Delete Function :-

```
SQL> delete from records where book_id = 4001;

1 row deleted.
```

Update Function :-

```
SQL> update student set f_name = 'manav' , l_name = 'P' where lib_id = '1005';

1 row updated.

SQL>
```

```
SQL> desc stud_book
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 LIB_ID                                    NOT NULL VARCHAR2(50)
 BOOK_ID                                   NOT NULL NUMBER(38)
 DATE_OF_ISSUE                                      DATE
 DATE_OF_RETURN                                     DATE
```

Conclusion: Thus we have implemented different DML commands successfully.

Page No.
Date

Experiment 5

Aim :- Perform Simple Queries, string manipulation operations.

Objective :- To understand simple queries & string manipulation operations

Description :-

SQL where clause is used to select rows satisfying given predicate.

```
SELECT column1...
From table
Where (condition).
```

Where clause consist of 5 search conditions

Compare → (<, >, =, <=, >=).

Range → (Between / Not Between)

Set membership : (IN / NOT IN)

Pattern Match :- (LIKE / NOT LIKE)

NULL :- (IS NULL / IS NOT NULL)

Simple Queries :-

Between :- checks value in a range

IN :- Tests whether a data value matches one of a list value.

LIKE :- % represents any sequence of zero or more characters. _ (underscore) represents a single character.

Teacher's Sign.: _____

Page No.
Date

String Manipulation Soft Operations.

1] Ascii :- converts single character string to ascii value.
2] Bit_length :- Returns the length, in bits.
3] Char :- Convert numeric value between 0 & 255 to character value.
4] Char_length :- Returns the length, in no. of character
5] Concat :- Concatinate two character strings
6] Insert :- Insert specific character string into specified location.
7] Upper :- Converts a string to uppercase.
8] Position :- Returns the numeric position of str in a character expression.
9] Repeat :- Repeat a specific expression n times.
10] Space :- Insert blank spaces.

Conclusion :- Thus we have successfully performed Simple queries, string manipulation op. on our system.

Teacher's Sign.:

Name :- Manav Pahilwani **Roll No**:- 37 **Class**:- D6AD

Experiment - 05

Aim :- To perform simple queries , string manipulation operations.

Code Snippets :-

Where clause:-

Comparison :-

```
1> select * from employee where salary > 50000;
2> go
ssn         ename                salary                dno
----------- -------------------- --------------------- -----------
         12 Manav P.                         110000.0000        3001
         21 Paul                              70000.0000        3004

(2 rows affected)
1> select * from employee where salary <= 50000;
2> go
ssn         ename                salary                dno
----------- -------------------- --------------------- -----------
         15 Harry                             50000.0000        3003
         26 John                              30000.0000        3001
```

Between :-

```
1> select * from employee where salary between 30000 and 90000;
2> go
ssn         ename                salary                dno
----------- -------------------- --------------------- -----------
         15 Harry                             50000.0000        3003
         21 Paul                              70000.0000        3004
         26 John                              30000.0000        3001

(3 rows affected)
```

Not Between :-

```
1> select * from employee where salary not between 40000 and 80000;
2> go
ssn         ename                salary                dno
----------- -------------------- --------------------- -----------
         12 Manav P.                        110000.0000        3001
         26 John                             30000.0000        3001

(2 rows affected)
```

IN :-

```
1> select * from employee where dno in ('3001');
2> go
ssn         ename                salary                dno
----------- -------------------- --------------------- -----------
         12 Manav P.                        110000.0000        3001
         26 John                             30000.0000        3001

(2 rows affected)
```

Not in :-

```
1> select * from employee where dno not in ('3003');
2> go
ssn         ename                salary                dno
----------- -------------------- --------------------- -----------
         12 Manav P.                        110000.0000        3001
         21 Paul                             70000.0000        3004
         26 John                             30000.0000        3001

(3 rows affected)
1>
```

Like / Not Like :-

```
1> select * from employee where ename like '%o%';
2> go
ssn         ename                salary                dno
----------- -------------------- --------------------- -----------
         26 John                             30000.0000        3001
```

```
1> select * from employee where ename not like '%z%';
2> go
ssn         ename                salary                dno
----------- -------------------- --------------------- -----------
         12 Manav P.                        110000.0000        3001
         15 Harry                            50000.0000        3003
         21 Paul                             70000.0000        3004
         26 John                             30000.0000        3001

(4 rows affected)
```

String Manipulation Operations :-

```
1> select ASCII('t');
2> go

-----------
        116
```

```
1> select CONCAT_WS('_','Hello','Good','Morning');
2> go

------------------
Hello_Good_Morning
```

```
1> select trim('        Good        ');
2> go

--------------------
Good
```

```
1> select lower("Hello Good Morning");
2> go

------------------
hello good morning
```

```
1> select upper("Hello Good Morning");
2> go

------------------
HELLO GOOD MORNING
```

```
1> select upper(ename) from employee;
2> go

--------------------
MANAV P.
HARRY
PAUL
JOHN

(4 rows affected)
```

```
1> select UNICODE("H");
2> go

-----------
         72

(1 rows affected)
```

```
1> select left('abcdef',3);
2> go

---
abc

(1 rows affected)
```

```
1> select left(ename, 3) from employee;
2> go

---
Man
Har
Pau
Joh

(4 rows affected)
```

```
1> select right(ename, 3) from employee;
2> go

---
 P.
rry
aul
ohn

(4 rows affected)
```

```
1> select ename, reverse(ename) from employee;
2> go
ename
-------------------- --------------------
Manav P.             .P vanaM
Harry                yrraH
Paul                 luaP
John                 nhoJ

(4 rows affected)
```

Page No.
Date

Experiment :- 06

Aim :- Write nested query for our system.

Objective :- To return data that will be used in the main query as a condition to further restrict the data to be retrived.

Description :- Nested query is a query within the another sql query & embedded within the where clause. The data that is returned by sub-query will be used by main query. It can be used with select, insert, update, delete & operating like =, <, >, >=, <=, <>, IN, between etc.

Rules :- must be enclosed within the parenthesis
can only have one column.
orders by cannot be used in subquery.

Aggregate functions :-

SUM() :- It returns the sum of values from given set.
Avg() :- It returns the average of values from given set.
Count() :- It returns the number of values from a given set.
Min() :- It returns the smallest value from a given set.
Max() :- It returns the largest value from a given set.

Group by clause :- It is used to create groups or group the records based on the criteria it applies aggregate function on subgroup of some tuples where subgroups are formed on some column value.

Teacher's Sign.:

having clause :- this clause is applied with group by, not with where.

select column... from table_name
group by ___ having ___ ;

co-related queries :- Subquery that uses value from outer query. It contains a reference to value from row related by outer query

Nested subquery :- runs only one for outer nesting query. It does not contain any reference to the outer query.

exists & non exists :- used to check whether the result of correlated nested query is empty or not, exists returns true if there is atleast one true, not exist returns true if there are no tuples in the output of the subquery

Derived relations :- allows a subquery to be used in the from clause eg select ssn from (select ssn, salary from emp, where salary <40000);

Complex queries :- are hard to write as single sql block needs to compose multiple SQL blocks to express complex query.

1] Derived clause :- occurs as nested select statements in the from clause of an outer select statement

2] with clause :- The with clause provides a way of defining a temporary view whose definition is available only to the query in which the clause occurs.

Teacher's Sign.: _____

Name :- Manav Pahilwani
Roll No :- 37
Class :- D6AD

Experiment :- 06

Aim : Write a Nested Query for your system.

Code Snippets :-

Sub-Query :-

“=” operator :-

```
1> select * from book where dno = (select dno from department where dname = 'CMPN');
2> go
book_id     book_name                                                                                      book_Issd   book_author
                                                       book_qty    book_avail      book_publication
                           dno         ref_id
----------- ------------------------------------------------------------------------------------------ ----------- ------------------
------------------------------------------------------- ----------- -------------- ------------------------------------------------------
------------------------------ ----------- -----------
       4002 Advanced Java                                                                                     156 Schildt
                                                                56 Y               Tata McGraw
Hill                                                                                            3001           1

(1 rows affected)
```

```
1> select * from book where dno = (select dno from department where dname = 'CMPN');
2> go
```

Null/ Not Null :-

```
1> select f_name, l_name from student where no_of_books is NULL;
2> go
f_name                                              l_name
--------------------------------------------------- ------------------------------------------
Joshua                                              Weissman
```

```
1> select f_name from student where no_of_books is not null;
2> go
f_name
--------------------------------------------------
Paul
Manav
Om
Yash

(4 rows affected)
```

Not in :-

```
1> select * from student where dno not in('3001');
2> go
lib_id                                             f_name                                                    l_
name                                              dno         no_of_books academic_yr
------------------------------------------------- ----------------------------------------------------- --
------------------------------------------------- ----------- ----------- --------------------
201041                                             Manav                                                     P.
                                                        3005           1 2021
201043                                             Joshua                                                    We
issman                                                  3002        NULL 2019
201047                                             Om                                                        G.
                                                        3005           3 2021
201056                                             Yash                                                      S.
                                                        3002           0 2020

(4 rows affected)
1>
```

Between :-

```
1> select * from book where book_qty between 0 and 50;
2> go
book_id     book_name
                                                        book_Issd   book_author
                                                                book_qty    book_avail      book_pub
lication
                                    dno         ref_id
----------- ----------------------------------------------------------------------------------------
----------------------------------------------------------- ----------- ------------------------------
------------------------------------------------------------------- ----------- --------------- --------
----------------------------------------------------------------------------------------------------------
------------------------------------ ----------- -----------
       4003 Microprocessors 8086
                                                                 7 Uffenbeck
                                                                        42 Y              Wiley

                                           3003           3
       4004 Microprocessor
                                                                 7 Uffenbeck
                                                                        42 Y              Wiley

                                           3003           3
       4006 Python
                                                               127 G
                                                                        45 Y              Indian P
ublications
                                           3004           4
       4007 Lets C
                                                               987 B
                                                                         0 N              Indian P
ublications
                                           3005           5

(4 rows affected)
```

Less Than :-

```
1> select * from book where book_qty < 45;
2> go
book_id     book_name
                                                    book_Issd   book_author
                                                             book_qty    book_avail      book_pub
lication
                                      dno         ref_id
----------- ------------------------------------------------------------------------------------
---------------------------------------------------------- ----------- ---------------------------
---------------------------------------------------------------- ----------- --------------- --------
---------------------------------------------------------------------------------------------------
------------------------------------ ----------- -----------
       4003 Microprocessors 8086
                                                                 7 Uffenbeck
                                                                         42 Y               Wiley

                                             3003           3
       4004 Microprocessor
                                                                 7 Uffenbeck
                                                                         42 Y               Wiley

                                             3003           3
       4007 Lets C
                                                               987 B
                                                                          0 N               Indian P
ublications
                                             3005           5

(3 rows affected)
```

Order by :-

```
1> select * from book order by book_name;
2> go
book_id     book_name
                                                    book_Issd   book_author
                                                            book_qty    book_avail      book_pub
lication
                                      dno         ref_id
----------- ------------------------------------------------------------------------------------
------------------------------------------------------------ ----------- -------------------------------
------------------------------------------------------------------- ----------- --------------- --------
------------------------------------------------------------------------------------------------------------
----------------------------------- ----------- -----------
       4002 Advanced Java
                                                                  156 Schildt
                                                                           56 Y               Tata McG
raw
Hill
                                    3001           1
       4007 Lets C
                                                                  987 B
                                                                            0 N               Indian P
ublications
                                          3005           5
       4004 Microprocessor
                                                                    7 Uffenbeck
                                                                           42 Y               Wiley


                                          3003           3
       4003 Microprocessors 8086
                                                                    7 Uffenbeck
                                                                           42 Y               Wiley


                                          3003           3
       4006 Python
                                                                  127 G
                                                                           45 Y               Indian P
ublications
                                          3004           4

(5 rows affected)
```

Group by and Having and Count() :-

```
1> select count(*) as c1 from book group by dno having count(*)>1;
2> go
c1
-----------
          2
```

Sum ():-

```
1> select sum(salary) as totalsal from employee group by dno;
2> go
totalsal
---------------------
          140000.0000
           50000.0000
           70000.0000

(3 rows affected)
```

```
1> select dno, sum(salary) as totalsal, count(*) as no_of_emp from employee group by dno;
2> go
dno         totalsal             no_of_emp
----------- -------------------- -----------
       3001          140000.0000           2
       3003           50000.0000           1
       3004           70000.0000           1
```

Avg () :-

```
1> select avg(salary) as avgsal from emp;
2> go
avgsal
----------------------
           48600.0000

(1 rows affected)
1>
```

Total () :-

```
1> select dno,avg(salary) as totalsal, count(*) as no_of_emp from emp group by dno;
2> go
dno         totalsal             no_of_emp
----------- -------------------- -----------
        101          43500.0000           2
        102          56000.0000           1
        103          50000.0000           2

(3 rows affected)
1>
```

```
1> select dno,count(emp_id) as no_of_emp from emp where salary between 40000 and 60000 group by dno havi
ng count(emp_id) >= 2
2> go
dno         no_of_emp
----------- -----------
        101           2

(1 rows affected)
1>
```

```
1> select * from emp as e1 where e1.salary > (select avg(salary) from emp as e2 where e1.dno = e2.dno);
2> go
emp_id      ename                                             salary               dno
----------- -------------------------------------------------- -------------------- -----------
          1 Joshua                                             45000.0000         101
          4 Harry                                              65000.0000         103

(2 rows affected)
1>
```

```
1> select * from emp as e1 where e1.salary>=all(select salary from emp as e2 where e1.dno = e2.dno);
2> go
emp_id      ename                                              salary               dno
----------- -------------------------------------------------- -------------------- -----------
          1 Joshua                                             45000.0000         101
          4 Harry                                              65000.0000         103
         10 Karan                                              56000.0000         102

(3 rows affected)
1>
```

```
1> select dno from(select dno,count(*) from emp group by dno) as deptinfo(dno,no_of_emp) where no_of_emp
>1
2> go
dno
-----------
        101
        103

(2 rows affected)
1>
```

With Clause :-

```
1> with temporarytable(averagevalue) as (select avg(salary) from employee)
2> select ssn,ename,salary from employee, temporarytable where employee.salary>temporarytable.averagevalue;
3> go
ssn         ename               salary
----------- ------------------- --------------------
         12 Manav P.                    110000.0000
         21 Paul                         70000.0000

(2 rows affected)
```

Page No.
Date

Experiment - 07

Aim :- Implement various types of joins & views.

Objective :- To combine records from two or more tables in a database.

Description :- Join command is used to combine field tables by using common values to each. It is used to retrive data from 2 or more tables based on some logical relationship between tables.

Inner Join :- Return rows when there is a match in both tables.

Left Join :- returns all rows from left table even if there are no matches in the right table.

Right Join :- returns all rows from right table even if no matches in the left table.

full join :- returns all rows present in both the tables

self join :- Used to join table to itself.

Outer join :- Returns the cartesian product of records from two or more joined tables.

Equi join :- returns rows satisfying the selection criteria from both joined tables are related

Conclusion :- We have successfully implemented the join in our system.

Teacher's Sign.:

Name :- Manav Pahilwani
Roll No :- 37
Class :- D6AD

Experiment :- 07

Aim :- Implement various types of Joins and Views

Code Snippets :-

Without using Join Keyword :-

```
1> select ename,e.dno,dname from employee as e,department as d where e.dno = d.dno;
2> go
ename                dno         dname
-------------------- ----------- --------------------
Manav P.                    3001 CMPN
Harry                       3003 ETRX
Paul                        3004 ELEX
John                        3001 CMPN

(4 rows affected)
```

With using Join keyword :-

```
1> select e.ename, e.dno, dname from (emp1 as e join dept1 as d on e.dno = d.dno);
2> go
ename                dno         dname
-------------------- ----------- --------------------
John                        101 CMPN
Smita                       101 CMPN
Smit                        101 CMPN
Neha                        102 IT
Nisha                       103 ETRX

(5 rows affected)
```

Left Outer Join :-

```
1> select e.ename,e.dno,dname from(employee as e left outer join department as d on e.dno=d.dno);
2> go
ename                dno         dname
-------------------- ----------- --------------------
Manav P.                    3001 CMPN
Harry                       3003 ETRX
Paul                        3004 ELEX
John                        3001 CMPN

(4 rows affected)
```

Right Outer Join :-

```
1> select e.ename, e.dno,dname from (employee as e right outer join department as d on e.dno = d.dno);
2> go
ename                dno         dname
-------------------- ----------- --------------------
Manav P.                    3001 CMPN
John                        3001 CMPN
NULL                        NULL INST
Harry                       3003 ETRX
Paul                        3004 ELEX
NULL                        NULL IT

(6 rows affected)
```

Full Outer Join :-

```
1> select e.ename,e.dno,dname from(employee as e full outer join department as d on e.dno = d.dno);
2> go
ename                dno         dname
-------------------- ----------- --------------------
Manav P.                    3001 CMPN
Harry                       3003 ETRX
Paul                        3004 ELEX
John                        3001 CMPN
NULL                        NULL INST
NULL                        NULL IT

(6 rows affected)
```

Self Join :-

```
1> select e.ename as empname,s.ename as supervisorname from emp1 as e,emp1 as s where e.superssn = s.ssn;
2> go
empname              supervisorname
-------------------- --------------------
John                 Neha
Smita                Neha
Smit                 Nisha
Nisha                Neha

(4 rows affected)
```

```
1> select * from emp1 as e,emp1 as s;
2> go
ssn         ename                salary               superssn    dno         ssn         ename                salary               superssn    dno
----------- -------------------- -------------------- ----------- ----------- ----------- -------------------- -------------------- ----------- -----------
          1 John                 45000.0000                  10         101           1 John                 45000.0000                  10         101
          2 Smita                42000.0000                  10         101           1 John                 45000.0000                  10         101
          5 Smit                 35000.0000                  15         101           1 John                 45000.0000                  10         101
         10 Neha                 25000.0000                   0         102           1 John                 45000.0000                  10         101
         15 Nisha                40000.0000                  10         103           1 John                 45000.0000                  10         101
          1 John                 45000.0000                  10         101           2 Smita                42000.0000                  10         101
          2 Smita                42000.0000                  10         101           2 Smita                42000.0000                  10         101
          5 Smit                 35000.0000                  15         101           2 Smita                42000.0000                  10         101
         10 Neha                 25000.0000                   0         102           2 Smita                42000.0000                  10         101
         15 Nisha                40000.0000                  10         103           2 Smita                42000.0000                  10         101
          1 John                 45000.0000                  10         101           5 Smit                 35000.0000                  15         101
          2 Smita                42000.0000                  10         101           5 Smit                 35000.0000                  15         101
          5 Smit                 35000.0000                  15         101           5 Smit                 35000.0000                  15         101
         10 Neha                 25000.0000                   0         102           5 Smit                 35000.0000                  15         101
         15 Nisha                40000.0000                  10         103           5 Smit                 35000.0000                  15         101
          1 John                 45000.0000                  10         101          10 Neha                 25000.0000                   0         102
          2 Smita                42000.0000                  10         101          10 Neha                 25000.0000                   0         102
          5 Smit                 35000.0000                  15         101          10 Neha                 25000.0000                   0         102
         10 Neha                 25000.0000                   0         102          10 Neha                 25000.0000                   0         102
         15 Nisha                40000.0000                  10         103          10 Neha                 25000.0000                   0         102
          1 John                 45000.0000                  10         101          15 Nisha                40000.0000                  10         103
          2 Smita                42000.0000                  10         101          15 Nisha                40000.0000                  10         103
          5 Smit                 35000.0000                  15         101          15 Nisha                40000.0000                  10         103
         10 Neha                 25000.0000                   0         102          15 Nisha                40000.0000                  10         103
         15 Nisha                40000.0000                  10         103          15 Nisha                40000.0000                  10         103
```

Equi Join :-

```
1> select ename,dname from emp1 as e,dept1 as d where e.dno = d.dno;
2> go
ename                dname
-------------------- --------------------
John                 CMPN
Smita                CMPN
Smit                 CMPN
Neha                 IT
Nisha                ETRX

(5 rows affected)
```

Experiment 8

Aim :- Conditional loops & cursors in PL/SQL

Theory :-

1] PL/SQL is a block structured language. The programs are logical blocks that can contain any number of nested blocks.

2] It includes procedural language elements like condition & loops. It allows declaration of constants, variable, procedures & functions, types & variable of those types & triggers.

3]

Declare :-
    declaration section.
Begin :-
    executable section
Exception :-
    errors handling section
END;

4] DDL is not allowed in PL/SQL block.

5] Cursors :-

i] Oracle creates a certain portion in the memory for every SQL Query that is executed.

ii] Using PL/SQL this portion in the memory can be given a name, it is basically a pointer to the context area, & thus represents a structure in memory.

iii] Cursors are of 2 types
    i] Implicit
    ii] Explicit.

Conclusion :- Thus through this experiment we have implemented conditional / loops & cursors in PL/SQL

Experiment :- 08

Queries: -

To give raises to all employees earning a salary less than 1500.

```
SQL> DECLARE
  2      CURSOR employee_cur IS
  3        SELECT employee_id,
  4               salary
  5        FROM   emp_temp
  6  WHERE department_id = 101
  7  FOR UPDATE;
  8      incr_sal NUMBER;
  9  BEGIN
 10      FOR employee_rec IN employee_cur LOOP
 11          IF employee_rec.salary < 15000 THEN
 12  incr_sal := .15;
 13  else
 14  incr_sal := .0;
 15  end if;
 16  UPDATE emp_temp
 17          SET    salary = salary + salary * incr_sal
 18          WHERE  CURRENT OF employee_cur;
 19      END LOOP;
 20  END;
 21  /
```

```
SQL> select * from emp_temp;

EMPLOYEE_ID FIRST_NAME           LAST_NAME           DEPARTMENT_ID     SALARY
----------- -------------------- -------------------- ------------- ----------
          1 manav                pahilwani                     101      16500
          2 Om                   Gaydhane                      101      15870
          3 Madhu                Naidu                         101      20900
```

```
SQL> select * from emp_temp;

EMPLOYEE_ID FIRST_NAME           LAST_NAME           DEPARTMENT_ID     SALARY
----------- -------------------- -------------------- ------------- ----------
          1 manav                pahilwani                     101      16500
          2 Om                   Gaydhane                      101      13800
          3 Madhu                Naidu                         101      20900
```

2. To set job of all clerks with salary greater than 1300 as “Senior Clerk”

```
declare
cursor empcursor is select salary, job from emp;
sal emp.salary%type;
job emp.job%type;
begin
open empcursor;
loop
fetch empcursor into sal,job;
exit when empcursor%notfound;
if sal>1300 and job like 'Programmer' then
update emp set job = 'Senior Clerk' where sal>1300 and job like 'Programmer';
end if;
end loop;
close empcursor;
end;
/
select * from emp;
```

ID	ENAME	SALARY	JOB
1	Manav	1500	Senior Clerk
2	Rohan	1750	Senior Clerk
3	Jaz	1800	Senior Clerk
4	Atish	500	Senior Clerk
5	Hamilton	3500	Senior Clerk
6	Josh	1200	Clerk

Date
Page

Experiment 9

Aim :- Triggess / functions and Procedure.

Theory :-

Trigger

1. A trigger is a piece of code that is run before or after a database table is modified.
2. A trigger is used to
   - prevent invalid transactions
   - keep an audit trail of table.
   - ensure rollback if database found inconsistent.
3. A row trigges is fired each time a row in the table is affected. It uses FOR EACH ROW clause, statement triggers are fired once on behalf of the statement, independent of the no. of rows the triggess statement affects.

Procedure & functions :-

1. PL / SQL allows writing subprograms.
2. If we want to update data on table, a procedure is preffered while for retriving info., a function is preferred.
3. A select SQL query can call a function but cannot call a procedure.
4. Procedure may return one or more values through parameters or may not return values at all. A function always return a value using the return statement.

SYNTAX for TRIGGER.

CREATE [OR REPLACE] TRIGGER trigger_name
{ BEFORE / AFTER / INSTEAD OF }
{ INSERT / UPDATE / DELETE }
[OF COL_NAME].

Conclusion :- Triggers / functions and Procedures have been implemented.

Create a trigger that fires before insert or delete of a row in emp table and displays the count of rows

```
create trigger ans
before insert or delete on employee
declare
val number;
begin
select count(*) into val from employee;
dbms_output.put_line('Number of rows are; '|| val);
end;
/
```

```
Trigger created.
```

```
insert into employee values(20,'Hari',49000,101);
```

```
1 row(s) inserted.
Number of rows are; 13
Value Inserted
```

Create a trigger that stops user from entering Dept no in emp table if that dept no doesn't exist in dept table. Trigger should display the contents of dept table.

```
Department does not exist
```

```
create trigger checkers
before insert on employee
for each row
declare
val int;
begin
select count(*) into val from depart d where d.deptno = :new.deptno;
if val = 0 then
dbms_output.put_line('Department does not exist');
else
dbms_output.put_line('Value Inserted');
end if;
end;
/
```

```
Trigger created.
```

```
Department does not exist
```

Write a procedure that:
Accepts department number and percentage of raise in sal
Updates the sal of all those employee under that department

```
1  create procedure bonus(deptno in int, inc in float)
2  as
3  begin
4  update employee set salary = salary + salary*inc where deptno = deptno;
5  dbms_output.put_line('Value updated');
6  end;
7  /
```

```
Procedure created.
```

```
1  EXEC bonus(102, 0.15);
```

```
Statement processed.
Value updated
```

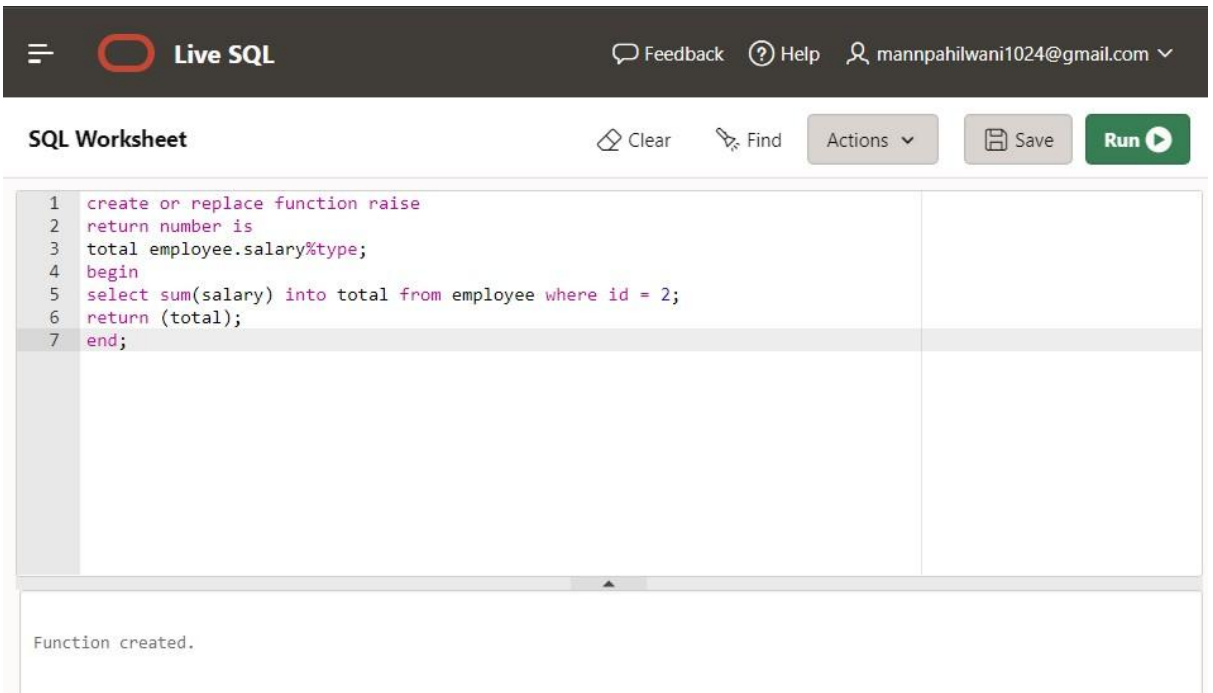
```
1  select * from employee where deptno = 102;
```

ID	ENAME	SALARY	DEPTNO
5	NA	74750	102
6	XYZ	51750	102
9	NA	77050	102
4	Rox	55545	102

Download CSV

4 rows selected.

Write a function that accepts :
Dept no and returns total sal of all employees in that department.



SQL Worksheet

Clear Find Actions Save Run

```
1  declare
2  c float;
3  begin
4  c := raise();
5  dbms_output.put_line('Total Salary =' ||c);
6  end;
```

```
Statement processed.
Total Salary =4000
```

Date: / /

Exp - 10.

Aim :- Demonstrate DCL & TCL commands

Objective :- To execute commands that control databases and transactions.

Descriptions :-

Data Control Language (DCL) :- Commands that control a database, including administering privileges & committing data. Used to create roles, permissions & referential integrity as well it is used to control access to databases by securing it.

DCL Commands :- GRANT, REVOKE

Grant :- Give users access privileges to the database. Only Database Administrators or owners of the database object can provide / remove privileges on a database object It is used to assign permissions to users

E.g grant insert on emp to user1;
grant all on emp to public; (assign all permissions to all users)

Revoke :- Withdraws users access to database given that with the GRANT command.

Syntax :- revoke <permission> on <object_name> from username;

Date: / /

Transaction Control :-

A transaction can be defined as a grp of tasks. A single task is the min. processing unit which cannot be divided further.

Let's take e.g. of a bank employee transfering ₹500 from A's acc. to B's acc.

A's acc.
Open_Acc (A)
Old_Balance = A.balance
New_Bal = Old.Bal. − 500
A.balance = New_bal.
Close_Acc.(A).

B's acc.
Open_Acc. (B)
Old_Bal. = B.balance
New_Bal. = Old_Bal. + 500
B. Bal.

GRANT:

```
SQL> grant select, update on emp to flash
  2  ;

Grant succeeded.
```

REVOKE:

```
SQL> revoke select,update on emp from flash;

Revoke succeeded.
```

TCL COMMANDS: COMMIT, ROLLBACK, SAVEPOINT---

```
SQL> select * from customer;

       CID CNAME                     PHNO
---------- -------------------- ----------
        12 ramesh                   99201
        12 ramesh                   98021
        13 suresh                   86547
        14 mahesh                  675320
```

COMMIT:

```
SQL> delete from customer
  2  where cid=12;

2 rows deleted.

SQL> commit;

Commit complete.

SQL> select * from customer;

       CID CNAME                     PHNO
---------- -------------------- ----------
        13 suresh                   86547
        14 mahesh                  675320
```

ROLLBACK:

```
SQL> delete from customer
  2  where cid=12;

2 rows deleted.

SQL> rollback;

Rollback complete.
```

SAVEPOINT:

```
SQL> insert into customer values (12,'ramesh',99201);

1 row created.

SQL> insert into customer values(12,'ramesh',98021);

1 row created.

SQL> savepoint customers;

Savepoint created.

SQL> select * from customer;

       CID CNAME                     PHNO
---------- -------------------- ----------
        12 ramesh                   99201
        13 suresh                   86547
        14 mahesh                  675320
        12 ramesh                   98021

SQL> delete from customer where cid=12;

2 rows deleted.

SQL> select * from customer;

       CID CNAME                     PHNO
---------- -------------------- ----------
        13 suresh                   86547
        14 mahesh                  675320

SQL> rollback to customers;

Rollback complete.

SQL> select * from customer;

       CID CNAME                     PHNO
---------- -------------------- ----------
        12 ramesh                   99201
        13 suresh                   86547
        14 mahesh                  675320
        12 ramesh                   98021
```

Exp - 11

Transaction Control :-

A transaction can be defined as a grp of tasks A single task is the min. processing unit which cannot be divided fusthes.

Let's take e.g. of a bank employee transfering ₹500 from A's acc. to B's acc.

A's acc.
Open_Acc (A)
Old_Balance := A.balance
New_Bal = Old.Bal. ∓500
A.balance = New_bal.
Close_Acc.(A).

B's acc.
Open_Acc. (B)
Old_Bal. = B.balance
New_Bal. = Old_Bal. +500
B Bal.

Date: / /

Concurrency Control

1] Concurrency control manages the transactions simultaneously without letting them interfere with each another.

2] The main objective of concurrency control is to allow many users perform different operations at the same time.

3] Using more than one transaction concurrently improves the performance of system.

4] If we are not able to perform the operations concurrently, then there can be serious problems such as loss of data integrity & consistency.

5] It reduces waiting time of transaction.

```
SQL> select * from emp;

NAME                    SALARY        SSN        DNO   SUPERSSN
-------------------- ---------- ---------- ---------- ----------


subrato                  55000          1          3          1
manas                    51750          2          2          2
jayesh                   65000          3          1          3
sarthak                46287.5          4          4          4
arnav                    50000          5          5          5
```

- CHANGING SALARY OF SSN=1
- COMMIT,SAVING IT
- CHANGING SALARY OF SSN=2
- ROLLING BACK THE CHANGE

```
SQL> set serveroutput on;
SQL>  DECLARE
  2   salary emp.salary%type;
  3  BEGIN
  4  dbms_output.put_line('Updating salary of employee,ssn=1');
  5  update emp set salary=salary+10000 where ssn=1;
  6   dbms_output.put_line('Committing transaction');
  7  COMMIT;
  8   dbms_output.put_line('Adding a savepoint');
  9  SAVEPOINT sal1;
 10  dbms_output.put_line('Updating salary of employee,ssn=2');
 11  update emp set salary=salary+10000 where ssn=2;
 12  dbms_output.put_line('Rolling back to savepoint');
 13  ROLLBACK TO sal1;
 14  END;
 15  /
Updating salary of employee,ssn=1
Committing transaction
Adding a savepoint
Updating salary of employee,ssn=2
Rolling back to savepoint

PL/SQL procedure successfully completed.
```

- WE CAN SEE THAT ONLY SALARY OF EMP SSN=1 GETS UPDATED

```
SQL> select * from emp;

NAME                    SALARY        SSN        DNO   SUPERSSN
-------------------- ---------- ---------- ---------- ----------


subrato                  65000          1          3          1
manas                    51750          2          2          2
jayesh                   65000          3          1          3
sarthak                46287.5          4          4          4
arnav                    50000          5          5          5
```

```
SQL>  set serveroutput on;
SQL> DECLARE
  2   rollno student.sno%type;
  3  s_age student.age%type;
  4  snm student.sname%type;
  5  s_cr student.course%type;
  6  BEGIN
  7  rollno:=&sno;
  8  s_age:=&age;
  9  snm:='&name';
 10  s_cr:='&course':
 11  insert into student values (rollno,s_age,snm,s_cr);
 12  dbms_output.put_line('one record inserted');
 13  COMMIT;
 14  SAVEPOINT save1;
 15   rollno:=&sno;
 16   s_age:=&age;
 17  snm:='&name';
 18  s_cr:='&course':
 19  INSERT into student values (rollno,s_age,snm,s_cr);
 20   dbms_output.put_line('one record inserted');
 21  ROLLBACK to SAVEPOINT save1;
 22  END;
 23  /
```

```
Enter value for sno: 60
old   7: rollno:=&sno;
new   7: rollno:=60;
Enter value for age: 19
old   8: s_age:=&age;
new   8: s_age:=19;
Enter value for name: subrato
old   9: snm:='&name';
new   9: snm:='subrato';
Enter value for course: ai
old  10: s_cr:='&course':
new  10: s_cr:='ai':
Enter value for sno: 61
old  15:  rollno:=&sno;
new  15:  rollno:=61;
Enter value for age: 19
old  16:  s_age:=&age;
new  16:  s_age:=19;
Enter value for name: vivek
old  17: snm:='&name';
new  17: snm:='vivek';
Enter value for course: it
old  18: s_cr:='&course':
new  18: s_cr:='it':
```

```
SQL> desc employee;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPLOYEE_ID                               NOT NULL NUMBER(38)
 ENAME                                     NOT NULL VARCHAR2(20)
 SALARY                                             NUMBER
 FK_EMPLOYEE                                        NUMBER(38)
 D_NO                                               NUMBER(38)

SQL> lock table employee in share mode;

Table(s) Locked.

SQL> commit;

Commit complete.
```

```
SQL> lock table employee in exclusive mode;

Table(s) Locked.

SQL> rollback;

Rollback complete.
```

```
SQL> lock table system.Donor in exclusive mode;

Table(s) Locked.

SQL> lock table system.Receiver in exclusive mode;
lock table system.Receiver in exclusive mode
*
ERROR at line 1:
ORA-00060: deadlock detected while waiting for resource


SQL> rollback;

Rollback complete.
```

```
SQL> insert into emp values('mahi',30000,6,6,6);

1 row created.
```

```
SQL> savepoint s1;

Savepoint created.

SQL> update emp
  2  set salary=salary+1000;

13 rows updated.

SQL> select * from emp;

NAME                    SALARY        SSN        DNO   SUPERSSN
-------------------- ---------- ---------- ---------- ----------


subrato                  56000          1          3          1
manas                    52750          2          2          2
jayesh                   66000          3          1          3
sarthak                47287.5          4          4          4
arnav                    51000          5          5          5




NAME                    SALARY        SSN        DNO   SUPERSSN
-------------------- ---------- ---------- ---------- ----------
Ramesh                   11000          1          2          1
mahi                     31000          6          6          6

13 rows selected.

SQL> rollback to s1;

Rollback complete.
```

```
SQL> select * from emp;

NAME                    SALARY        SSN        DNO   SUPERSSN
-------------------- ---------- ---------- ---------- ----------


subrato                  55000          1          3          1
manas                    51750          2          2          2
jayesh                   65000          3          1          3
sarthak                46287.5          4          4          4
arnav                    50000          5          5          5




NAME                    SALARY        SSN        DNO   SUPERSSN
-------------------- ---------- ---------- ---------- ----------
Ramesh                   10000          1          2          1
mahi                     30000          6          6          6
```

Experiment - 12

Aim :- Implement Database Connectivity (JDBC, ODBC)

Objective

Description :-

JDBC Architecture :-

In the two tier model, a Java app talks directly to the data source. This requires a JDBC driver that can communicate with the particular data source being accessed. The data source may be located on another machine to which the user is connected via network. This is referred to as a client/server config, with the user's machine as the client, & the machine housing the data source as the server. The network can be an intranet.

In the three tier model, commands are sent to a 'middle tier' of services, which then sends the commands to the data source. The data source processes the commands & sends the results back to the middle tier, which then sends them to the user.

Until recently, the middle tier has often been written in languages such as C or C++, which offer fast performance. However, with the introduction of optimizing compilers that translate Java bytecode into efficient machine-specific code & technologies such as Enterprise JavaBeans, the Java platform is fast becoming a standard platform of middle-tier development.

Date: / /

Fundamental Steps in JDBC

1] Import JDBC packages
2] Load & register the JDBC drivers
3] Open a connection to the database.
4] Create a statement object to perform a query.
5] Execute the statement object & return a query resultset.
6] Process the resultset.
7] Close the resultset & statement objects.
8] Close the connection.

```java
import java.sql.*;
public class jdbc {

    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost:3306/";
    static final String USER = "java";
    static final String PASS = "subo";
public static void main(String[] args) {
Connection conn = null;
Statement stmt = null;
try{

Class.forName("com.mysql.cj.jdbc.Driver");

System.out.println("Connecting to database...");
conn = DriverManager.getConnection(DB_URL, USER, PASS);

System.out.println("Creating database...");
stmt = conn.createStatement();

String sql1 = ("CREATE DATABASE STUDENTS1");
stmt.executeUpdate(sql1);
System.out.println("Database created successfully...");
}

catch(SQLException se){
se.printStackTrace();
}

catch(Exception e){
e.printStackTrace();
}

finally{
    try{
        if(stmt!=null)
        stmt.close();
        }

    catch(SQLException se2){
}
try{
    if(conn!=null)
    conn.close();
    }
    catch(SQLException se){
    se.printStackTrace();}
}

System.out.println("Goodbye!");
}
}
```

```
Connecting to database...
Creating database...
java.sql.SQLSyntaxErrorException: Access denied for user 'java'@'localhost' to database 'students1'
        at com.mysql.cj.jdbc.exceptions.SQLError.createSQLException(SQLError.java:120)
        at com.mysql.cj.jdbc.exceptions.SQLExceptionsMapping.translateException(SQLExceptionsMapping.java:122)
        at com.mysql.cj.jdbc.StatementImpl.executeUpdateInternal(StatementImpl.java:1340)
        at com.mysql.cj.jdbc.StatementImpl.executeLargeUpdate(StatementImpl.java:2089)
        at com.mysql.cj.jdbc.StatementImpl.executeUpdate(StatementImpl.java:1251)
        at jdbc.main(jdbc.java:22)
Goodbye!
```

```
Connecting to database...
Creating database...
Goodbye!
```