UNIVERSITY DATABASE MANAGEMENT SYSTEM

DBMS PROJECT REPORT

SUBMITTED BY

JASHAN ARORA	102003206
DISHIKA GUPTA	102003231
KANISHKA GUPTA	102003232
GARIMA MINOCHA	102003234

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

SUBMITTED TO

DR. VARUN SRIVASTAVA

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

TABLE OF CONTENTS

INTRODUCTION

Project Statement

University Database Management System creates, manages and performs all the activities related to the database of a given University. The database consists of information about the university, students, faculties, hostels and academic programs.
The main aim of this project is to manage the database in such a way that information about academic activities can be retrieved easily, efficiently and accurately.

The aim was to design and build a medium-large scale database in SQL Server which stores university data and contains multiple triggers and procedures.
The data used for courses was taken from a Thapar university website.
We have Created the tables in MYSQL Server. There were 15 tables in total.
University Database Management Project Focuses on managing the data associated with the Academic and Research department of the University.
This report consists of the problem statement, E-R Diagram, the way of normalizing tables, an overview of the procedures, views, triggers etc and some of the complex queries.

This DBMS project is our rendition of the Thapar database management system based in PL/SQL queries, dealing with functionalities including cursors, functions, procedures, triggers and exception handling.

Cursors:
A cursor in SQL is a temporary work area created in system memory when a SQL statement is executed. A SQL cursor is a set of rows together with a pointer that identifies a current row. It is a database object to retrieve data from a result set one row at a time.

Procedure:
A subprogram is a program unit/module that performs a particular task. These subprograms are combined to form larger programs. This is basically called the 'Modular design'. A subprogram can be invoked by another subprogram or program which is called the calling program.

Functions:
A function is same as a procedure except that it returns a value. Therefore, all the discussions of the previous chapter are true for functions too.

Triggers:
Triggers are stored programs, which are automatically executed or fired when some events occur. Triggers are, in fact, written to be executed in response to any of the following events

A database manipulation (DML) statement (DELETE, INSERT, or UPDATE)
A database definition (DDL) statement (CREATE, ALTER, or DROP).
A database operation (SERVERERROR, LOGON, LOGOFF, STARTUP, or SHUTDOWN).

Exception handling:
An exception is an error condition during a program execution. PL/SQL supports programmers to catch such conditions using EXCEPTION block in the program and an appropriate action is taken against the error condition

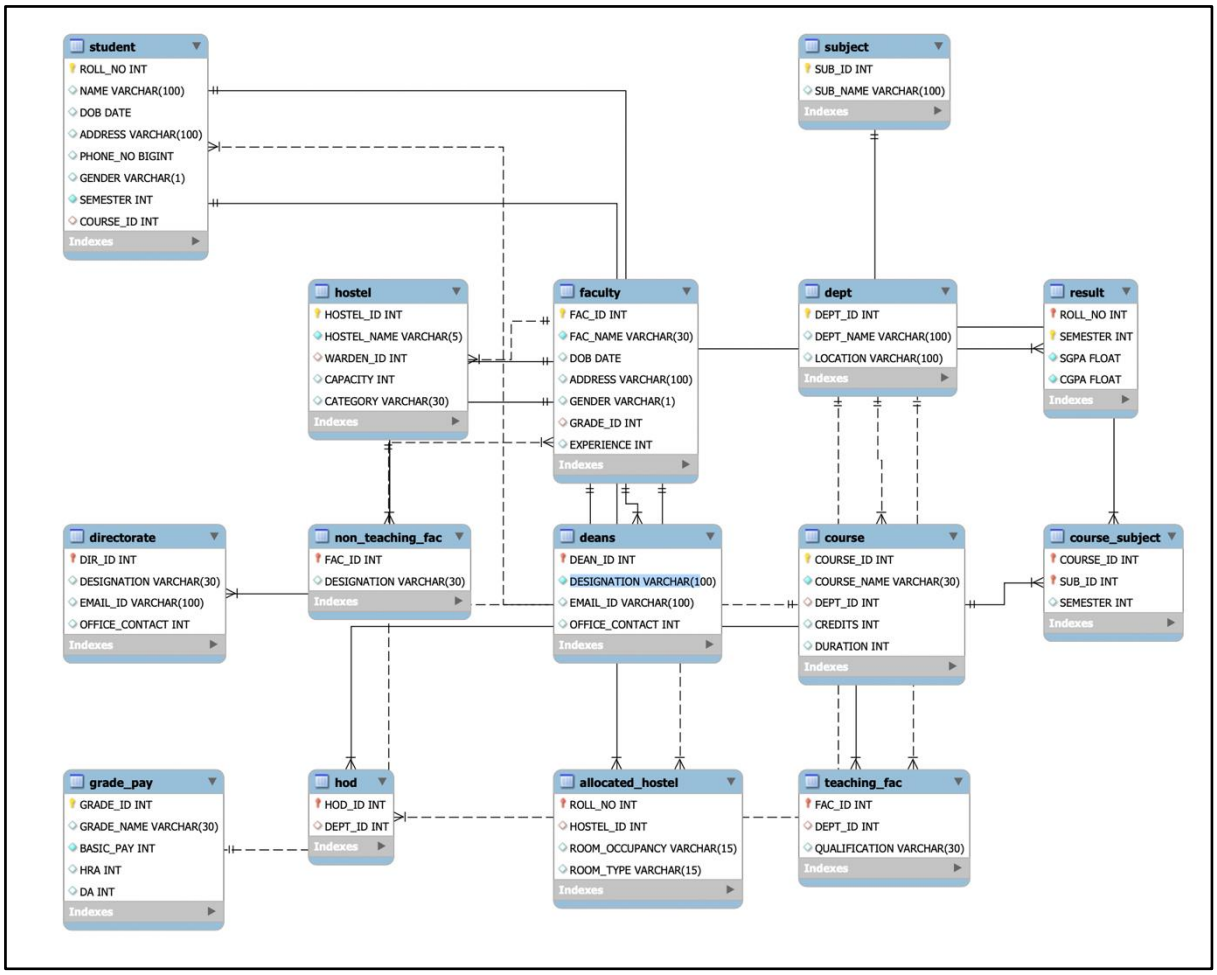
The Tables we created are as follows :

STUDENT - stores basic attributes of student like DOB, roll number etc
GRADE_PAY - stores the salary details of staff
DEPT - stores attributes like number, names and location of several departments
FACULTY - stores basic info of all staff
TEACHING_FAC - stores information for the teaching staff like qualifications, etc
NON_TEACHNG_FAC - stores information for the non teaching staff like designations, etc
HOD - details information for the heads of various departments
DEANS - details attributes of the deans of the various fields with attributes like contact information
DIRECTORATE - details information about the members of the directorate
COURSE - stores information about the various fields of engineering courses offered
SUBJECT - stores information about the various subjects that every course entails
COURSE_SUBJECT - relays information about courses offered in a courses
HOSTEL - names of hostels, wardens etc
ALLOCATED_HOSTEL - relays information about the allotment of hostels for each student opting for hostels
RESULT - stores information about the result of all students semester by semester

We created procedures and functions to store and relay information like AddGradePay, AddHostel, AddDept, etc.

We have also applied 20 sets of queries on this database to fetch information and create unique context areas which include :
Top 10 students with respect to students, Heads of departments with courses, number of students from the same city, etc.

E-R DIAGRAM AND RELATIONAL MODEL



CREATION OF NORMALIZED TABLES

```sql
CREATE DATABASE College;
USE COLLEGE;

CREATE TABLE STUDENT(
ROLL_NO INT AUTO_INCREMENT,
CONSTRAINT ROLLNO_PK PRIMARY KEY(ROLL_NO),
NAME VARCHAR(100) NOT NULL,
DOB DATE,
ADDRESS VARCHAR(100),
PHONE_NO BIGINT ,
CONSTRAINT PHONE_U UNIQUE(PHONE_NO),
GENDER VARCHAR(1),
SEMESTER INT NOT NULL,
COURSE_ID INT ,
CONSTRAINT COURSEID_FK FOREIGN KEY(COURSE_ID) REFERENCES COURSE(COURSE_ID)
);

CREATE TABLE GRADE_PAY(
GRADE_ID INT AUTO_INCREMENT,
CONSTRAINT GRADEID_PK PRIMARY KEY(GRADE_ID),
GRADE_NAME VARCHAR(30) NOT NULL,
BASIC_PAY INT NOT NULL,
HRA INT,
DA INT
);

```

```sql
CREATE TABLE DEPT(
DEPT_ID INT AUTO_INCREMENT,
CONSTRAINT DEPTID_PK PRIMARY KEY(DEPT_ID),
DEPT_NAME VARCHAR(100) NOT NULL,
LOCATION VARCHAR(100)
);

CREATE TABLE FACULTY(
FAC_ID INT AUTO_INCREMENT,
CONSTRAINT FACID_PK PRIMARY KEY(FAC_ID),
FAC_NAME VARCHAR(30) NOT NULL,
DOB DATE,
EXPERIENCE INT,
ADDRESS VARCHAR(100),
GENDER VARCHAR(1),
GRADE_ID INT,
CONSTRAINT GRADEID_FK FOREIGN KEY(GRADE_ID) REFERENCES GRADE_PAY(GRADE_ID)
);

CREATE TABLE TEACHING_FAC(
FAC_ID INT,
CONSTRAINT FACID_PK1 PRIMARY KEY(FAC_ID),
CONSTRAINT FACID_FK FOREIGN KEY(FAC_ID) REFERENCES FACULTY(FAC_ID),
DEPT_ID INT,
CONSTRAINT DEPTID_FK FOREIGN KEY(DEPT_ID) REFERENCES DEPT(DEPT_ID),
QUALIFICATION VARCHAR(30)
);

```

```sql
CREATE TABLE NON_TEACHING_FAC(
FAC_ID INT,
CONSTRAINT FACID_PK2 PRIMARY KEY(FAC_ID),
CONSTRAINT FACID_FK1 FOREIGN KEY(FAC_ID) REFERENCES FACULTY(FAC_ID),
DESIGNATION VARCHAR(30)
);

CREATE TABLE HOD(
HOD_ID INT,
CONSTRAINT HODID_PK PRIMARY KEY(HOD_ID),
CONSTRAINT FACID_FK2 FOREIGN KEY(HOD_ID) REFERENCES FACULTY(FAC_ID),
DEPT_ID INT,
CONSTRAINT DEPTID_FK1 FOREIGN KEY(DEPT_ID) REFERENCES DEPT(DEPT_ID)
);

CREATE TABLE DEANS(
DEAN_ID INT,
CONSTRAINT DEANID_PK PRIMARY KEY(DEAN_ID),
CONSTRAINT FACID_FK3 FOREIGN KEY(DEAN_ID) REFERENCES FACULTY(FAC_ID),
DESIGNATION VARCHAR(100) NOT NULL,
EMAIL_ID VARCHAR(100),
CONSTRAINT EMAIL_U UNIQUE(EMAIL_ID),
OFFICE_CONTACT INT,
CONSTRAINT CONTACT_U UNIQUE(OFFICE_CONTACT)
);
```

```sql
CREATE TABLE DIRECTORATE(
DIR_ID INT,
CONSTRAINT DIRID_PK PRIMARY KEY(DIR_ID),
CONSTRAINT FACID_FK4 FOREIGN KEY(DIR_ID) REFERENCES FACULTY(FAC_ID),
DESIGNATION VARCHAR(30),
EMAIL_ID VARCHAR(100),
CONSTRAINT EMAIL_U1 UNIQUE(EMAIL_ID),
OFFICE_CONTACT INT,
CONSTRAINT CONTACT_U1 UNIQUE(OFFICE_CONTACT)
);

CREATE TABLE COURSE(
COURSE_ID INT AUTO_INCREMENT,
CONSTRAINT COURSEID_PK PRIMARY KEY(COURSE_ID),
COURSE_NAME VARCHAR(30) NOT NULL,
DEPT_ID INT,
CONSTRAINT DEPTID_FK2 FOREIGN KEY(DEPT_ID) REFERENCES DEPT(DEPT_ID),
CREDITS INT,
DURATION INT
);

CREATE TABLE SUBJECT(
SUB_ID INT AUTO_INCREMENT,
CONSTRAINT SUBID_PK PRIMARY KEY(SUB_ID),
SUB_NAME VARCHAR(100)
);
```

```sql
CREATE TABLE COURSE_SUBJECT(
COURSE_ID INT,
CONSTRAINT COURSEID_FK1 FOREIGN KEY(COURSE_ID) REFERENCES COURSE(COURSE_ID),
SUB_ID INT,
CONSTRAINT SUBID_FK FOREIGN KEY(SUB_ID) REFERENCES SUBJECT(SUB_ID),
CONSTRAINT COURSESUBJECT_PK PRIMARY KEY(COURSE_ID,SUB_ID),
SEMESTER INT
);

CREATE TABLE HOSTEL(
HOSTEL_ID INT AUTO_INCREMENT,
CONSTRAINT HOSTELID_PK PRIMARY KEY(HOSTEL_ID),
HOSTEL_NAME VARCHAR(5) NOT NULL,
CONSTRAINT HNAME_U UNIQUE(HOSTEL_NAME),
WARDEN_ID INT,
CONSTRAINT FACID_FK6 FOREIGN KEY(WARDEN_ID) REFERENCES FACULTY(FAC_ID),
CAPACITY INT,
CATEGORY VARCHAR(30),
CONSTRAINT CATEGORY_C CHECK(CATEGORY IN ('GIRLS','BOYS'))
);

```

```sql
CREATE TABLE ALLOCATED_HOSTEL(
ROLL_NO INT,
CONSTRAINT ROLLNO_PK1 PRIMARY KEY(ROLL_NO),
CONSTRAINT ROLLNO_FK FOREIGN KEY(ROLL_NO) REFERENCES STUDENT(ROLL_NO),
HOSTEL_ID INT,
CONSTRAINT HOSTELID_FK FOREIGN KEY(HOSTEL_ID) REFERENCES HOSTEL(HOSTEL_ID),
ROOM_OCCUPANCY VARCHAR(15),
CONSTRAINT ROOMOCCP_C CHECK(ROOM_OCCUPANCY IN ('1S','2S','3S')),
ROOM_TYPE VARCHAR(15),
CONSTRAINT ROOMTYPE_C CHECK(ROOM_TYPE IN ('AC','NON-AC'))
);

CREATE TABLE RESULT(
ROLL_NO INT,
CONSTRAINT ROLLNO_FK1 FOREIGN KEY(ROLL_NO) REFERENCES STUDENT(ROLL_NO),
SEMESTER INT ,
CONSTRAINT ROLLSEM_PK PRIMARY KEY(ROLL_NO,SEMESTER),
SGPA FLOAT NOT NULL,
CONSTRAINT SGPA_C CHECK(SGPA<=10),
CGPA FLOAT NOT NULL,
CONSTRAINT CGPA_C CHECK(CGPA<=10)
);

```

CREATING PROCEDURES FOR INSERTING RECORDS IN TABLES

```
1  USE COLLEGE;
2
3  DELIMITER &&
4  CREATE PROCEDURE AddStudent(
5  NAME VARCHAR(100) ,
6  DOB DATE,
7  ADDRESS VARCHAR(100),
8  PHONE_NO BIGINT ,
9  GENDER VARCHAR(1),
10 SEMESTER INT,
11 COURSE_ID INT
12 )
13 BEGIN
14     INSERT INTO STUDENT(NAME ,DOB,ADDRESS,PHONE_NO,GENDER,SEMESTER,COURSE_ID)
15     VALUES(NAME ,DOB,ADDRESS,PHONE_NO,GENDER,SEMESTER,COURSE_ID);
16 END &&
17
18 DELIMITER &&
19 CREATE PROCEDURE AddGradePay(
20 GRADE_NAME VARCHAR(30) ,
21 BASIC_PAY INT ,
22 HRA INT,
23 DA INT
24 )
25 BEGIN
26     INSERT INTO GRADE_PAY(GRADE_NAME,BASIC_PAY,HRA,DA)
27     VALUES(GRADE_NAME,BASIC_PAY,HRA,DA);
28 END &&
29
```

```
30 DELIMITER &&
31 CREATE PROCEDURE AddDept(
32 DEPT_NAME VARCHAR(100) ,
33 LOCATION VARCHAR(100)
34 )
35 BEGIN
36     INSERT INTO DEPT(DEPT_NAME,LOCATION)
37     VALUES(DEPT_NAME,LOCATION);
38 END &&
39
40 DELIMITER &&
41 CREATE PROCEDURE AddFaculty(
42 FAC_NAME VARCHAR(30) ,
43 DOB DATE,
44 EXPERIENCE INT,
45 ADDRESS VARCHAR(100),
46 GENDER VARCHAR(1),
47 GRADE_ID INT
48 )
49 BEGIN
50     INSERT INTO FACULTY(FAC_NAME,DOB,EXPERIENCE,ADDRESS,GENDER,GRADE_ID)
51     VALUES(FAC_NAME,DOB,EXPERIENCE,ADDRESS,GENDER,GRADE_ID);
52 END &&
53
```

```
DELIMITER &&
CREATE PROCEDURE AddCourse(
COURSE_NAME VARCHAR(30),
DEPT_ID INT,
CREDITS INT,
DURATION INT
)
BEGIN
    INSERT INTO COURSE(COURSE_NAME,DEPT_ID,CREDITS,DURATION)
    VALUES(COURSE_NAME,DEPT_ID,CREDITS,DURATION);
END &&

DELIMITER &&
CREATE PROCEDURE AddSubject(
SUB_NAME VARCHAR(100)
)
BEGIN
    INSERT INTO SUBJECT(SUB_NAME)
    VALUES(SUB_NAME);
END &&
```

```
DELIMITER &&
CREATE PROCEDURE AddHostel(
HOSTEL_NAME VARCHAR(5) ,
WARDEN_ID INT,
CAPACITY INT,
CATEGORY VARCHAR(30)
)
BEGIN
    INSERT INTO HOSTEL(HOSTEL_NAME,WARDEN_ID,CAPACITY,CATEGORY)
    VALUES(HOSTEL_NAME,WARDEN_ID,CAPACITY,CATEGORY);
END &&
```

INSERTING RECORDS IN TABLES

1. STUDENT TABLE

```
insert into Student values(102003000,'Chirag','2001-03-01','Amritsar',9318270289,'M',4,1);
CALL AddStudent('Kushagar Bansal','2002-04-23','Patiala',9336271289,'M',4,1);
CALL AddStudent('Kirti Kapoor','2002-07-01','Karnal',7601271289,'F',4,1);
CALL AddStudent('Ashwin Nigam','2002-11-21','Jalandhar',9336212345,'M',4,1);
CALL AddStudent('Deeksha Aggarwal','2002-12-06','Delhi',7136279289,'F',4,1);
CALL AddStudent('Vidul Gupta','2001-07-06','Chandigarh',9336271221,'M',4,1);
insert into Student values(102103100,'Ashish Kakkar','2001-01-07','Patna',7658290289,'M',2,1);
CALL AddStudent('Karan Nasa','2003-07-23','Ludhiana',9236234189,'M',2,1);
CALL AddStudent('Sanidhya Vijay','2002-12-05','Dehradun',9876511289,'M',2,1);
CALL AddStudent('Shashank Singh','2001-01-01','Jammu',9753112345,'M',2,1);
insert into Student values(111905100,'Jatin Garg','2002-11-01','Ambala',7652190219,'M',6,1);
CALL AddStudent('Harsh Gupta','2001-11-01','Rudrapur',7178479212,'M',6,1);
CALL AddStudent('Riya','2001-09-01','Lucknow',9336274112,'F',6,1);
CALL AddStudent('Srishti Sharma','2001-10-10','Gurgaon',7236271113,'F',6,1);
insert into Student values(121805300,'Prachi Gupta','1999-12-01','Ambala',7652190234,'F',8,1);
CALL AddStudent('Kamya Tayal','1999-01-10','Ranchi',9651879212,'F',8,1);
CALL AddStudent('Rohan Grover','2000-10-10','Jalandhar',9871374112,'M',8,1);
CALL AddStudent('Ayush Modi','2000-04-16','Moradabad',7212345113,'M',8,1);


insert into Student values(131805000,'Anuj Gupta','2003-12-11','Lucknow',9318275149,'M',8,2);
CALL AddStudent('Tanishq Hans','2003-05-21','Patiala',9336277610,'M',8,2);
insert into Student values(131903100,'Aryan Kapoor','2000-02-07','Patna',7674290289,'M',6,2);
CALL AddStudent('Inayat Goyal','2000-01-23','Ludhiana',7236674189,'F',6,2);
insert into Student values(132005100,'Dhruv Mishra','2001-11-26','Ambala',7052191499,'M',4,2);
CALL AddStudent('Tushar Kapoor','2001-10-23','Rudrapur',6178429201,'M',4,2);
insert into Student values(132105300,'Gohan Kohli','2003-12-21','Ambala',6782191334,'M',2,2);
CALL AddStudent('Diya Gupta','2003-04-10','Ranchi',9651000212,'F',2,2);

```

2. GRADE_PAY TABLE

```
CALL AddGradePay('Professor',175000,17500,30000);
CALL AddGradePay('Associate Professor',150000,15000,25000);
CALL AddGradePay('Assisant Professor-III',110000,11000,20000);
CALL AddGradePay('Assistant Professor-II',92000,9200,16000);
CALL AddGradePay('Assistant Professor-I',78000,7800,13000);
CALL AddGradePay('Lecturer',71500,0,0);
CALL AddGradePay('Registrar',190000,19000,34000);
CALL AddGradePay('Manager',100000,10000,18000);
CALL AddGradePay('Gazzetted Officer',145000,14500,23000);
```

3. DEPT TABLE

```
Call AddDept('Computer Science and Engineering','LC');
Call AddDept('Electronics and Communication Engineering','B Block');
Call AddDept('Electrical and Instrumentation Engineering','C Block');
Call AddDept('Mechanical Engineering','F Block');
Call AddDept('Chemical Engineering','E Block');
```

4. FACULTY TABLE

```
Call AddFaculty('RK SHARMA','1970-12-29',17,'PATIALA','M',1);
Call AddFaculty('SEEMA BAWA','1975-12-29',14,'JALANDHAR','F',1);
Call AddFaculty('RAJESH KUMAR','1981-09-28',10,'CHANDIGARH','M',2);
Call AddFaculty('MANINDER SINGH','1974-04-29',17,'PATIALA','M',1);
Call AddFaculty('INDERVEER CHANNA','1984-6-01',17,'LUDHIANA','F',2);
Call AddFaculty('SHALINI BATRA','1988-03-31',9,'NEW DELHI','F',2);
Call AddFaculty('SHREELEKHA PANDEY','1989-11-12',8,'KARNAL','F',3);
Call AddFaculty('NIDHI KALRA','1990-03-02',6,'SANGRUR','F',4);
Call AddFaculty('ANSHU PRASHAR','1991-01-01',5,'CHANDIGARH','M',5);
Call AddFaculty('SAHIL SHARMA','1989-09-22',5,'AMRITSAR','M',6);

Call AddFaculty('ALPANA AGGARWAL','1970-11-29',17,'PATIALA','F',1);
Call AddFaculty('AMIT KUMAR KOHLI','1975-12-02',14,'JALANDHAR','M',2);
Call AddFaculty('RAVI KUMAR','1981-01-11',12,'CHANDIGARH','M',3);
Call AddFaculty('AMIT MISHRA','1974-04-02',11,'PATIALA','M',4);
Call AddFaculty('KULBIR SINGH','1984-10-31',8,'LUDHIANA','M',5);
Call AddFaculty('AJAY KAKKAR','1988-09-23',7,'NEW DELHI','M',6);

Call AddFaculty('RS KALER','1970-06-02',17,'LUDHIANA','M',1);
Call AddFaculty('MUKESH SINGH','1980-11-27',9,'NEW DELHI','M',2);
Call AddFaculty('SHAKTI SINGH','1982-01-12',8,'KARNAL','M',3);
Call AddFaculty('SWATI SONDHI','1985-01-02',6,'SANGRUR','F',4);
Call AddFaculty('SANDEEP PANDEY','1988-05-01',5,'CHANDIGARH','M',5);
Call AddFaculty('NAVDEEP KAUR','1990-03-22',5,'AMRITSAR','F',6);

Call AddFaculty('AJAY BATISH','1970-06-02',20,'LUDHIANA','M',1);
Call AddFaculty('SK MOHAPATRA','1982-01-31',18,'NEW DELHI','M',2);
Call AddFaculty('TARUN KUMAR BERA','1984-01-22',8,'KARNAL','M',3);
Call AddFaculty('AS JAWANDA','1985-12-03',6,'SANGRUR','M',4);
```

5. TEACHING_FAC TABLE

```
INSERT INTO TEACHING_FAC VALUES(1,1,'PHD');
INSERT INTO TEACHING_FAC VALUES(2,1,'PHD');
INSERT INTO TEACHING_FAC VALUES(3,1,'DOCTORATE');
INSERT INTO TEACHING_FAC VALUES(4,1,'PHD');
INSERT INTO TEACHING_FAC VALUES(5,1,'PHD');
INSERT INTO TEACHING_FAC VALUES(6,1,'PHD');
INSERT INTO TEACHING_FAC VALUES(7,1,'PHD');
INSERT INTO TEACHING_FAC VALUES(8,1,'PHD');
INSERT INTO TEACHING_FAC VALUES(9,1,'M TECH');
INSERT INTO TEACHING_FAC VALUES(10,1,'M TECH');

INSERT INTO TEACHING_FAC VALUES(11,2,'DOCTORATE');
INSERT INTO TEACHING_FAC VALUES(12,2,'PHD');
INSERT INTO TEACHING_FAC VALUES(13,2,'DOCTORATE');
INSERT INTO TEACHING_FAC VALUES(14,2,'PHD');
INSERT INTO TEACHING_FAC VALUES(15,2,'B TECH');
INSERT INTO TEACHING_FAC VALUES(16,2,'M TECH');

INSERT INTO TEACHING_FAC VALUES(17,3,'DOCTORATE');
INSERT INTO TEACHING_FAC VALUES(18,3,'PHD');
INSERT INTO TEACHING_FAC VALUES(19,3,'DOCTORATE');
INSERT INTO TEACHING_FAC VALUES(20,3,'M TECH');
INSERT INTO TEACHING_FAC VALUES(21,3,'B TECH');
INSERT INTO TEACHING_FAC VALUES(22,3,'PHD');

INSERT INTO TEACHING_FAC VALUES(23,4,'DOCTORATE');
INSERT INTO TEACHING_FAC VALUES(24,4,'PHD');
INSERT INTO TEACHING_FAC VALUES(25,4,'DOCTORATE');
INSERT INTO TEACHING_FAC VALUES(26,4,'PHD');
```

6. NON_TEACHING_FAC TABLE

```
INSERT INTO NON_TEACHING_FAC VALUES(36,'REGISTRAR');
INSERT INTO NON_TEACHING_FAC VALUES(37,'ASSISTANT REGISTRAR');
INSERT INTO NON_TEACHING_FAC VALUES(38,'LAW OFFICER');
INSERT INTO NON_TEACHING_FAC VALUES(39,'FINANCE OFFICER');
INSERT INTO NON_TEACHING_FAC VALUES(40,'SENIOR MANAGER (AUDITS)');
INSERT INTO NON_TEACHING_FAC VALUES(41,'SENIOR MANAGER (S&P)');
INSERT INTO NON_TEACHING_FAC VALUES(42,'ACCOUNTS MANAGER');
```

7. HOD TABLE

```
insert into HOD values(4,1);
insert into HOD values(11,2);
insert into HOD values(17,3);
insert into HOD values(25,4);
insert into HOD values(29,5);
```

8. DEANS TABLE

```
INSERT INTO DEANS VALUES(4,'Dean of Academic Affairs','doaa@thapar.edu',2393022);
INSERT INTO DEANS VALUES(5,'Dean of Student Affairs','dosa@thapar.edu',2393039);
INSERT INTO DEANS VALUES(17,'Dean of Faculty Affairs','dofa@thapar.edu',2393914);
INSERT INTO DEANS VALUES(3,'Dean of Research and Sponsered Projects','dorsp@thapar.edu',2393038);
```

9. DIRECTORATE TABLE

```
INSERT INTO DIRECTORATE VALUES(35,'Director','director@thapar.edu',2393001);
INSERT INTO DIRECTORATE VALUES(23,'Deputy Director','deputydirector@thapar.edu',2393521);
```

10. COURSE TABLE

```
Call AddCourse('BE COE',1,220,4);
Call AddCourse('BE CSE',1,225,4);
Call AddCourse('BE COBS',1,195,4);
Call AddCourse('ME SOFT ENGG',1,120,2);
Call AddCourse('BE ENC',2,220,4);
Call AddCourse('BE ECE',2,220,4);
Call AddCourse('ME ECE',2,110,2);
Call AddCourse('BE EIC',3,195,4);
Call AddCourse('ME POWER SYSTEMS',3,115,2);
Call AddCourse('BE MECH ENGG',4,210,4);
Call AddCourse('BE MECHATRONICS',4,205,4);
Call AddCourse('ME CAD/CAM',4,120,2);
Call AddCourse('BE CHEM ENGG',5,185,4);
Call AddCourse('ME CHEM ENGG',5,110,2);
```

11. SUBJECT TABLE

```
Call AddSubject('Applied Chemistry');
Call AddSubject('Computer Programming');
Call AddSubject('Electrical Engineering');
Call AddSubject('Energy and Environment');
Call AddSubject('Mathematics-I');
Call AddSubject('Mechanics');
Call AddSubject('Applied Physics');
Call AddSubject('Object Oriented Programming');
Call AddSubject('Electronics Engineering');
Call AddSubject('Engineering Drawing');
Call AddSubject('Professional Communication');
Call AddSubject('Mathematics-II');
Call AddSubject('Data Structures');
Call AddSubject('Design and Analysis of Algorithms');
Call AddSubject('Digital Electronics');
Call AddSubject('Software Engineering');
Call AddSubject('Computer Graphics');
Call AddSubject('Compiler Construction');
Call AddSubject('Project Semester');
Call AddSubject('Machine Learning');
Call AddSubject('Parallel Computing');
Call AddSubject('Advanced Software Engineering ');
Call AddSubject('VSLI Design');
Call AddSubject('Optimisation Techniques');
Call AddSubject('Optimisation Process');
Call AddSubject('Operating Systems');
Call AddSubject('Capstone Project');
Call AddSubject('Signal and System');
Call AddSubject('Digital System Design');
```

12. COURSE_SUBJECT TABLE

```
INSERT INTO COURSE_SUBJECT VALUES(1,1,1);
INSERT INTO COURSE_SUBJECT VALUES(1,2,1);
INSERT INTO COURSE_SUBJECT VALUES(1,3,1);
INSERT INTO COURSE_SUBJECT VALUES(1,4,1);
INSERT INTO COURSE_SUBJECT VALUES(1,5,1);
INSERT INTO COURSE_SUBJECT VALUES(1,6,1);
INSERT INTO COURSE_SUBJECT VALUES(1,7,2);
INSERT INTO COURSE_SUBJECT VALUES(1,8,2);
INSERT INTO COURSE_SUBJECT VALUES(1,9,2);
INSERT INTO COURSE_SUBJECT VALUES(1,10,2);
INSERT INTO COURSE_SUBJECT VALUES(1,11,2);
INSERT INTO COURSE_SUBJECT VALUES(1,12,2);
INSERT INTO COURSE_SUBJECT VALUES(1,13,3);
INSERT INTO COURSE_SUBJECT VALUES(1,14,4);
INSERT INTO COURSE_SUBJECT VALUES(1,15,5);
INSERT INTO COURSE_SUBJECT VALUES(1,16,6);
INSERT INTO COURSE_SUBJECT VALUES(1,17,7);
INSERT INTO COURSE_SUBJECT VALUES(1,18,8);

INSERT INTO COURSE_SUBJECT VALUES(2,1,2);
INSERT INTO COURSE_SUBJECT VALUES(2,2,2);
INSERT INTO COURSE_SUBJECT VALUES(2,3,2);
INSERT INTO COURSE_SUBJECT VALUES(2,4,2);
INSERT INTO COURSE_SUBJECT VALUES(2,5,2);
INSERT INTO COURSE_SUBJECT VALUES(2,6,2);
INSERT INTO COURSE_SUBJECT VALUES(2,7,1);
INSERT INTO COURSE_SUBJECT VALUES(2,8,1);
INSERT INTO COURSE_SUBJECT VALUES(2,9,1);
INSERT INTO COURSE_SUBJECT VALUES(2,10,1);
```

13. HOSTEL TABLE

```
Call AddHostel('M',29,1000,'BOYS');
Call AddHostel('O',3,900,'BOYS');
Call AddHostel('K',14,300,'BOYS');
Call AddHostel('N',20,800,'GIRLS');
Call AddHostel('G',32,200,'GIRLS');
```

14. ALLOCATED_HOSTEL TABLE

```
INSERT INTO ALLOCATED_HOSTEL VALUES(152102132,4,'2S','AC');
INSERT INTO ALLOCATED_HOSTEL VALUES(122107812,5,'1S','NON-AC');
INSERT INTO ALLOCATED_HOSTEL VALUES(142105301,4,'3S','AC');
INSERT INTO ALLOCATED_HOSTEL VALUES(132106437,5,'1S','AC');
INSERT INTO ALLOCATED_HOSTEL VALUES(102003002,4,'2S','AC');
INSERT INTO ALLOCATED_HOSTEL VALUES(142005101,5,'3S','NON-AC');
INSERT INTO ALLOCATED_HOSTEL VALUES(152005000,5,'1S','NON-AC');
INSERT INTO ALLOCATED_HOSTEL VALUES(152005001,5,'2S','AC');
INSERT INTO ALLOCATED_HOSTEL VALUES(101904970,5,'3S','NON-AC');
INSERT INTO ALLOCATED_HOSTEL VALUES(121905402,4,'1S','NON-AC');
INSERT INTO ALLOCATED_HOSTEL VALUES(111905102,5,'1S','AC');
INSERT INTO ALLOCATED_HOSTEL VALUES(132005002,4,'2S','AC');
INSERT INTO ALLOCATED_HOSTEL VALUES(111905103,5,'3S','AC');
INSERT INTO ALLOCATED_HOSTEL VALUES(121805481,4,'2S','NON-AC');
INSERT INTO ALLOCATED_HOSTEL VALUES(161805000,5,'3S','AC');
INSERT INTO ALLOCATED_HOSTEL VALUES(141804582,4,'1S','NON-AC');
INSERT INTO ALLOCATED_HOSTEL VALUES(141805000,5,'3S','AC');
INSERT INTO ALLOCATED_HOSTEL VALUES(121805301,4,'2S','AC');
INSERT INTO ALLOCATED_HOSTEL VALUES(121805300,5,'2S','AC');
INSERT INTO ALLOCATED_HOSTEL VALUES(162105300,1,'2S','AC');
INSERT INTO ALLOCATED_HOSTEL VALUES(152103101,2,'1S','AC');
INSERT INTO ALLOCATED_HOSTEL VALUES(152103100,3,'3S','NON-AC');
INSERT INTO ALLOCATED_HOSTEL VALUES(132105300,1,'2S','NON-AC');
INSERT INTO ALLOCATED_HOSTEL VALUES(102103100,2,'1S','AC');
INSERT INTO ALLOCATED_HOSTEL VALUES(102103101,3,'3S','NON-AC');
INSERT INTO ALLOCATED_HOSTEL VALUES(102004960,1,'1S','AC');
INSERT INTO ALLOCATED_HOSTEL VALUES(102003000,2,'2S','NON-AC');
INSERT INTO ALLOCATED_HOSTEL VALUES(102003003,3,'3S','NON-AC');
INSERT INTO ALLOCATED_HOSTEL VALUES(102003005,1,'2S','AC');
```

15. RESULT TABLE

```
INSERT INTO RESULT VALUES(162105300,1,9.15,9.12);
INSERT INTO RESULT VALUES(152103101,1,8.34,9.10);
INSERT INTO RESULT VALUES(152103100,1,8.98,9.12);
INSERT INTO RESULT VALUES(152102132,1,8.90,8.78);
INSERT INTO RESULT VALUES(132105300,1,9.50,9.12);
INSERT INTO RESULT VALUES(122107812,1,7.15,7.90);
INSERT INTO RESULT VALUES(142105301,1,8.67,8.63);
INSERT INTO RESULT VALUES(132106437,1,9.01,9.24);
INSERT INTO RESULT VALUES(172105300,1,7.65,7.98);
INSERT INTO RESULT VALUES(102102947,1,8.90,8.98);
INSERT INTO RESULT VALUES(102103100,1,9.95,10);
INSERT INTO RESULT VALUES(102103101,1,9.15,9.85);
INSERT INTO RESULT VALUES(102103102,1,8.01,8.65);
INSERT INTO RESULT VALUES(102103103,1,7.96,8.12);
INSERT INTO RESULT VALUES(142105300,1,9.45,9.63);
INSERT INTO RESULT VALUES(142103221,1,8.54,8.88);
INSERT INTO RESULT VALUES(132105301,1,9.21,9.43);
INSERT INTO RESULT VALUES(192105300,1,8.12,8.67);
INSERT INTO RESULT VALUES(182109117,1,7.65,7.89);
INSERT INTO RESULT VALUES(182103100,1,8.90,9.12);

INSERT INTO RESULT VALUES(102004960,3,6.98,7.12);
INSERT INTO RESULT VALUES(172005100,3,5.98,6.10);
INSERT INTO RESULT VALUES(102003000,3,7.58,7.98);
INSERT INTO RESULT VALUES(102003001,3,9.58,9.12);
INSERT INTO RESULT VALUES(102003002,3,8.98,9.01);
INSERT INTO RESULT VALUES(102003003,3,7.71,7.98);
INSERT INTO RESULT VALUES(102003004,3,8.53,9.61);
INSERT INTO RESULT VALUES(102003005,3,6.71,6.98);
```

SQL QUERIES

1. Displaying details of HOD of all the Departments.

Query:

```
select dept_name AS "DEPARTMENT" , fac_name AS "HOD" ,
TIMESTAMPDIFF(YEAR,DOB,CURDATE()) AS "AGE"  , address AS "CITY" ,
experience AS "EXPERIENCE" from hod h, faculty f, dept d where h.hod_id=f.fac_id and
d.dept_id=h.dept_id ;
```

Output:

DEPARTMENT	HOD	AGE	CITY	EXPERIENCE
Computer Science and Engineering	MANINDER SINGH	48	PATIALA	17
Electronics and Communication Engineering	ALPANA AGGARWAL	51	PATIALA	17
Electrical and Instrumentation Engineering	RS KALER	51	LUDHIANA	17
Mechanical Engineering	TARUN KUMAR BERA	38	KARNAL	8
Chemical Engineering	RAJEEV MEHTA	55	LUDHIANA	22

2. Displaying No of Students in a Particular Hostel.

Query:

```
select h.hostel_id AS "HOSTEL ID", CONCAT("HOSTEL - ",h.hostel_name) AS
"HOSTEL NAME" ,h.category AS "GIRLS/BOYS" , count(*) AS "TOTAL
RESIDENTS" from student s , hostel h , allocated_hostel a where h.hostel_id=a.hostel_id
and s.roll_no=a.roll_no group by h.hostel_id,h.hostel_name,h.category order by
h.hostel_id;
```

Output:

HOSTEL ID	HOSTEL NAME	GIRLS/BOYS	TOTAL RESIDENTS
1	HOSTEL - M	BOYS	9
2	HOSTEL - O	BOYS	9
3	HOSTEL - K	BOYS	8
4	HOSTEL - N	GIRLS	8
5	HOSTEL - G	GIRLS	11

3. Displaying Faculty ID, Faculty Name, Designation , Salary of Faculties whose Monthly salary is greater than or equal to 200000.

Query:

```
select f.fac_id AS "FACULTY ID", f.fac_name "FACULTY
NAME",UPPER(g.grade_name) as "DESIGNATION",
(1+f.experience/100)*(g.basic_pay+g.hra+g.da) AS "MONTHLY SALARY" from
faculty f , grade_pay g where f.grade_id=g.grade_id and
(1+f.experience/100)*(g.basic_pay+g.hra+g.da)>=200000 order by fac_id;
```

Output:

FACULTY ID	FACULTY NAME	DESIGNATION	MONTHLY SALARY
1	RK SHARMA	PROFESSOR	260325.0000
2	SEEMA BAWA	PROFESSOR	253650.0000
3	RAJESH KUMAR	ASSOCIATE PROFESSOR	209000.0000
4	MANINDER SINGH	PROFESSOR	260325.0000
5	INDERVEER CHANNA	ASSOCIATE PROFESSOR	222300.0000
6	SHALINI BATRA	ASSOCIATE PROFESSOR	207100.0000
11	ALPANA AGGARWAL	PROFESSOR	260325.0000
12	AMIT KUMAR KOHLI	ASSOCIATE PROFESSOR	216600.0000
17	RS KALER	PROFESSOR	260325.0000
18	MUKESH SINGH	ASSOCIATE PROFESSOR	207100.0000
23	AJAY BATISH	PROFESSOR	267000.0000
24	SK MOHAPATRA	ASSOCIATE PROFESSOR	224200.0000
29	RAJEEV MEHTA	PROFESSOR	271450.0000
30	HARIPADA BHUNIA	ASSOCIATE PROFESSOR	226100.0000
35	PRAKASH GOPLAN	PROFESSOR	280350.0000
36	GURBINDER SINGH	REGISTRAR	291600.0000
37	RUCHI GUPTA	REGISTRAR	279450.0000
38	AMANPREET SINGH	GAZZETTED OFFICER	208050.0000
39	PANKAJ SINHA	GAZZETTED OFFICER	211700.0000

4. Displaying Total Staff in a Particular Department.

Query:

```
select d.dept_id AS "DEPARTMENT ID", d.dept_name AS "DEPARTMENT NAME" ,
count(*) AS "TOTAL STAFF" from teaching_fac f, dept d where f.dept_id=d.dept_id
group by d.dept_id,d.dept_name;
```

Output:

DEPARTMENT ID	DEPARTMENT NAME	TOTAL STAFF
1	Computer Science and Engineering	10
2	Electronics and Communication Engineering	6
3	Electrical and Instrumentation Engineering	6
4	Mechanical Engineering	6
5	Chemical Engineering	6

5. Displaying Name of Faculty of a Particular Department who is having maximum work Experience in that Department.

Query:

```
select d.dept_id AS "DEPARTMENT ID", d.dept_name AS "DEPARTMENT NAME"
,f.fac_name AS "NAME",f.experience AS "EXPERIENCE"  from faculty f,teaching_fac
t, dept d where t.dept_id=d.dept_id and f.fac_id=t.fac_id and (d.dept_id,f.experience) in
(select d.dept_id , max(f.experience) from dept d , faculty f , teaching_fac t where
d.dept_id=t.dept_id and t.fac_id=f.fac_id group by d.dept_id);
```

Output:

DEPARTMENT ID	DEPARTMENT NAME	NAME	EXPERIENCE
1	Computer Science and Engineering	RK SHARMA	17
1	Computer Science and Engineering	MANINDER SINGH	17
1	Computer Science and Engineering	INDERVEER CHANNA	17
2	Electronics and Communication Engineering	ALPANA AGGARWAL	17
3	Electrical and Instrumentation Engineering	RS KALER	17
4	Mechanical Engineering	AJAY BATISH	20
5	Chemical Engineering	RAJEEV MEHTA	22

6. Displaying details of Students who are day scholars along with the course in which they are enrolled.

Query:

```
select s.roll_no AS "ROLL NO" , s.name AS "NAME" ,
TIMESTAMPDIFF(YEAR,s.DOB,CURDATE()) AS "AGE", s.address AS "CITY" ,
s.phone_no AS "PHONE NO" , c.course_name AS "COURSE"  from student s, course c
where s.roll_no not in (select a.roll_no from allocated_hostel a) and
c.course_id=s.course_id order by s.roll_no;
```

Output:

ROLL NO	NAME	AGE	CITY	PHONE NO	COURSE
101801871	Akshat Khosla	22	Kolkata	7108025987	BE MECH ENGG
102003001	Kushagar Bansal	20	Patiala	9336271289	BE COE
102003004	Deeksha Aggarwal	19	Delhi	7136279289	BE COE
102102947	Akshita Garg	18	Faridabad	9412015310	BE MECH ENGG
102103102	Sanidhya Vijay	19	Dehradun	9876511289	BE COE
102103103	Shashank Singh	21	Jammu	9753112345	BE COE
111905101	Harsh Gupta	20	Rudrapur	7178479212	BE COE
121805303	Ayush Modi	22	Moradabad	7212345113	BE COE
131805000	Anuj Gupta	18	Lucknow	9318275149	BE CSE
131903100	Aryan Kapoor	22	Patna	7674290289	BE CSE
132005100	Dhruv Mishra	20	Ambala	7052191499	BE CSE
132105301	Diya Gupta	19	Ranchi	9651000212	BE CSE
141805001	Kartik Sood	23	Ranchi	8136217626	BE COBS
141901232	Devansh Garg	21	Karnatka	90012912282	BE CHEM ENGG
141903100	Shiti Garg	21	Patna	9514299489	BE COBS
142002178	Khushi Garg	20	Bangalore	9678291009	BE CHEM ENGG
142103221	Rohit Bisht	19	Jodhpur	9619151200	BE CHEM ENGG
142105300	Riya Kohli	18	Ambala	9712111300	BE COBS
152004718	Kunal Madaan	24	Agra	9108113123	ME CHEM ENGG
171903100	Shrey	22	Agra	9004290181	BE ECE
172005100	Chahat Goyal	21	Kanpur	7098290169	BE ECE
172105300	Ananya Singh	18	Faridabad	9712015311	BE ECE
182103100	Neha Aggarwal	24	Agra	9014290132	ME ECE
182109117	Naina Madaan	23	Agra	9014970230	ME POWER SY...
192105300	Uday Uppal	18	Faridabad	9612015300	BE EIC

7. Displaying details of first 10 Students and their course who have highest CGPA.

Query:

```
select s.roll_no AS "ROLL NO" , s.name AS "NAME" , c.course_name AS "COURSE" ,
r.cgpa from student s, course c ,result r where s.roll_no=r.roll_no and
c.course_id=s.course_id and r.semester=(s.semester-1) order by r.cgpa desc LIMIT 10;
```

Output:

ROLL NO	NAME	COURSE	cgpa
102103100	Ashish Kakkar	BE COE	10
111905100	Jatin Garg	BE COE	10
161805000	Anshita	BE ENC	9.99
101904970	Taesha	BE MECH ENGG	9.98
152004718	Kunal Madaan	ME CHEM ENGG	9.98
102103101	Karan Nasa	BE COE	9.85
142105300	Riya Kohli	BE COBS	9.63
102003004	Deeksha Aggarwal	BE COE	9.61
131903100	Aryan Kapoor	BE CSE	9.54
132105301	Diya Gupta	BE CSE	9.43

8. Displaying No of students from a particular city.

Query:

select address AS "CITY" ,count(*) AS "TOTAL STUDENTS" from student group by address order by count(*) desc;

Output:

CITY	TOTAL STUDENTS
Ambala	8
Agra	6
Delhi	6
Patna	5
Ranchi	5
Kolkata	4
Ludhiana	4
Kanpur	3
Faridabad	3
Rudrapur	3
Patiala	2
Jalandhar	2
Lucknow	2
Karnatka	2
Bangalore	2
Jodhpur	2
Pune	2
Amritsar	1
Karnal	1
Chandigarh	1
Dehradun	1
Jammu	1
Gurgaon	1
Moradabad	1
Mumbai	1

9. Displaying all the Graduate courses offered by the university and their Corresponding Department.

Query:

select c.course_name AS "COURSE NAME" , d.dept_name AS "DEPARTMENT NAME" from dept d, course c where d.dept_id=c.dept_id and c.course_name like "BE%";

Output:

COURSE NAME	DEPARTMENT NAME
BE COE	Computer Science and Engineering
BE CSE	Computer Science and Engineering
BE COBS	Computer Science and Engineering
BE ENC	Electronics and Communication Engineering
BE ECE	Electronics and Communication Engineering
BE EIC	Electrical and Instrumentation Engineering
BE MECH ENGG	Mechanical Engineering
BE MECHATRONICS	Mechanical Engineering
BE CHEM ENGG	Chemical Engineering

10. Displaying all the Post Graduate courses offered by the university and their Corresponding Department.

Query:

```
select c.course_name AS "COURSE NAME" , d.dept_name AS "DEPARTMENT
NAME" from dept d, course c where d.dept_id=c.dept_id and c.duration = 2;
```

Output:

COURSE NAME	DEPARTMENT NAME
ME SOFT ENGG	Computer Science and Engineering
ME ECE	Electronics and Communication Engineering
ME POWER SYSTEMS	Electrical and Instrumentation Engineering
ME CAD/CAM	Mechanical Engineering
ME CHEM ENGG	Chemical Engineering

11. Displaying details of DEANS of the university.

Query:

```
select d.designation AS "DESIGNATION" , f.fac_name AS "NAME" , d.email_id AS
"EMAIL ID" , d.office_contact AS "CONTACT NO." from deans d,faculty f where
d.dean_id=f.fac_id;
```

Output:

DESIGNATION	NAME	EMAIL ID	CONTACT NO.
Dean of Research and Sponsered Projects	RAJESH KUMAR	dorsp@thapar.edu	2393038
Dean of Academic Affairs	MANINDER SINGH	doaa@thapar.edu	2393022
Dean of Student Affairs	INDERVEER CHANNA	dosa@thapar.edu	2393039
Dean of Faculty Affairs	RS KALER	dofa@thapar.edu	2393914

12. Displaying details of Directorates of the university.

Query:

```
select upper(d.designation) AS "DESIGNATION" , f.fac_name AS "NAME" , d.email_id
AS "EMAIL ID" , d.office_contact AS "CONTACT NO." from directorate d,faculty f
where d.dir_id=f.fac_id;
```

Output:

DESIGNATION	NAME	EMAIL ID	CONTACT NO.
DEPUTY DIRECTOR	AJAY BATISH	deputydirector@thapar.edu	2393521
DIRECTOR	PRAKASH GOPLAN	director@thapar.edu	2393001

13. Displaying details of Non-Teaching Staff of the University.

Query:

```
select f.fac_id AS "FACULTY ID", f.fac_name AS "NAME" ,t.designation AS
"DESIGNATION", TIMESTAMPDIFF(YEAR,f.DOB,CURDATE()) AS "AGE"  ,
f.address AS "CITY" , f.experience AS "EXPERIENCE" from faculty f,
non_teaching_fac t where t.fac_id=f.fac_id;
```

Output:

FACULTY ID	NAME	DESIGNATION	AGE	CITY	EXPERIENCE
36	GURBINDER SINGH	REGISTRAR	52	KAPURTHALA	20
37	RUCHI GUPTA	ASSISTANT REGISTRAR	47	LUDHIANA	15
38	AMANPREET SINGH	LAW OFFICER	43	SIRHIND	14
39	PANKAJ SINHA	FINANCE OFFICER	49	PATHANKOT	16
40	ASHISH MEHRA	SENIOR MANAGER (AUDITS)	42	HYDERABAD	12
41	P.S. JOLLY	SENIOR MANAGER (S&P)	45	PANIPAT	15
42	VINEET ARORA	ACCOUNTS MANAGER	40	MANDI	11

14. Displaying Subjects of the course BE-ENC.

Query:

```
select s.sub_id AS "SUBJECT ID", s.sub_name "SUBJECT NAME" from course c ,
subject s , course_subject cs where c.course_id=cs.course_id and s.sub_id=cs.sub_id and
c.course_id = (select course_id from course where course_name = 'BE ENC');
```

Output:

SUBJECT ID	SUBJECT NAME
1	Applied Chemistry
2	Computer Programming
3	Electrical Engineering
4	Energy and Environment
5	Mathematics-I
6	Mechanics
7	Applied Physics
8	Object Oriented Programming
9	Electronics Engineering
10	Engineering Drawing
11	Professional Communication
12	Mathematics-II
13	Data Structures
19	Project Semester
20	Machine Learning
25	Optimisation Process
26	Operating Systems
27	Capstone Project

15. Displaying Roll no and Name of Students residing in AC and 2-S room of Hostel with their Hostel Names.

Query:

```
select s.roll_no AS "ROLL NO" , upper(s.name) AS "NAME" , concat('HOSTEL -
',h.hostel_name) AS "HOSTEL" from student s , allocated_hostel a , hostel h
where a.roll_no=s.roll_no and a.hostel_id=h.hostel_id and a.room_type='AC' and
a.room_occupancy='2S';
```

Output:

ROLL NO	NAME	HOSTEL
102003002	KIRTI KAPOOR	HOSTEL - N
102003005	VIDUL GUPTA	HOSTEL - M
121805300	PRACHI GUPTA	HOSTEL - G
121805301	KAMYA TAYAL	HOSTEL - N
132005002	INAYAT GOYAL	HOSTEL - N
142005100	ARPIT SAGAR	HOSTEL - O
152005001	SIMRAN KAUR	HOSTEL - G
152102132	DIVYA GARG	HOSTEL - N
162105300	GIRIK GARG	HOSTEL - M
191805000	TARIK BHATEJA	HOSTEL - O

16. Displaying Department and where they are located.

Query:

```
select dept_name AS "DEPARTMENT" , location AS "LOCATION" from dept;
```

Output:

DEPARTMENT	LOCATION
Computer Science and Engineering	LC
Electronics and Communication Engineering	B Block
Electrical and Instrumentation Engineering	C Block
Mechanical Engineering	F Block
Chemical Engineering	E Block

17. Displaying credits of the courses along with their departments.

Query:

```
select c.course_id AS "COURSE ID" , c.course_name AS "COURSE NAME" ,
d.dept_name AS "DEPARTMENT", c.credits AS "CREDITS"  from course c , dept
d where c.dept_id=d.dept_id;
```

Output:

COURSE ID	COURSE NAME	DEPARTMENT	CREDITS
1	BE COE	Computer Science and Engineering	220
2	BE CSE	Computer Science and Engineering	225
3	BE COBS	Computer Science and Engineering	195
4	ME SOFT ENGG	Computer Science and Engineering	120
5	BE ENC	Electronics and Communication Engineering	220
6	BE ECE	Electronics and Communication Engineering	220
7	ME ECE	Electronics and Communication Engineering	110
8	BE EIC	Electrical and Instrumentation Engineering	195
9	ME POWER SYSTEMS	Electrical and Instrumentation Engineering	115
10	BE MECH ENGG	Mechanical Engineering	210
11	BE MECHATRONICS	Mechanical Engineering	205
12	ME CAD/CAM	Mechanical Engineering	120
13	BE CHEM ENGG	Chemical Engineering	185
14	ME CHEM ENGG	Chemical Engineering	110

18. Displaying Wardens of the Hostels with their capacity and category .

Query:

```
select concat('HOSTEL - ',h.hostel_name) AS "HOSTEL NAME" , f.fac_name
"WARDEN" , h.capacity AS "CAPACITY", h.category AS "CATEGORY" from
hostel h , faculty f where h.warden_id=f.fac_id;
```

Output:

HOSTEL NAME	WARDEN	CAPACITY	CATEGORY
HOSTEL - M	RAJEEV MEHTA	1000	BOYS
HOSTEL - O	RAJESH KUMAR	900	BOYS
HOSTEL - K	AMIT MISHRA	300	BOYS
HOSTEL - N	SWATI SONDHI	800	GIRLS
HOSTEL - G	NEETU SINGH	200	GIRLS

19. Displaying count of Staff members who have particular degree.

Query:

```
select qualification AS "QUALIFICATION",count(*) AS "NO. OF TEACHERS"
from teaching_fac group by qualification;
```

Output:

QUALIFICATION	NO. OF TEACHERS
PHD	15
DOCTORATE	9
M TECH	6
B TECH	4

20. Displaying count of students enrolled in specific course.

Query:

```
select c.course_name  AS "COURSE NAME",count(*) As 'NO OF STUDENTS'
from student s , course c where s.course_id = c.course_id group by
c.course_name;
```

Output:

COURSE NAME	NO OF STUDENTS
BE COE	18
BE CSE	8
BE COBS	8
ME SOFT ENGG	4
BE ENC	4
BE ECE	4
ME ECE	2
BE EIC	4
ME POWER SYSTEMS	2
BE MECH ENGG	4
BE MECHATRONICS	4
ME CAD/CAM	2
BE CHEM ENGG	4
ME CHEM ENGG	2

21. Displaying top 5 Faculties who has minimum Experience.

Query:

```
select f.fac_id AS "FACULTY ID", f.fac_name "FACULTY
NAME",UPPER(g.grade_name) as "DESIGNATION" ,f.experience AS
"EXPERIENCE" from faculty f , grade_pay g ,teaching_fac t where
f.grade_id=g.grade_id and f.fac_id=t.fac_id order by f.experience LIMIT 5;
```

Output:

FACULTY ID	FACULTY NAME	DESIGNATION	EXPERIENCE
28	NAVDEEP KUMMAR	LECTURER	4
27	SANDEEP KUMAR SHA...	ASSISTANT PROFESSOR-I	5
9	ANSHU PRASHAR	ASSISTANT PROFESSOR-I	5
10	SAHIL SHARMA	LECTURER	5
22	NAVDEEP KAUR	LECTURER	5

22. Displaying Faculty Members who are both HOD and DEAN of the University.

Query:

```
select f.fac_id AS "FACULTY ID", f.fac_name "FACULTY
NAME",UPPER(g.grade_name) as "DESIGNATION" ,f.experience AS
"EXPERIENCE" from faculty f , grade_pay g ,hod h,deans d where
f.grade_id=g.grade_id and f.fac_id=h.hod_id and d.dean_id=f.fac_id;
```

Output:

FACULTY ID	FACULTY NAME	DESIGNATION	EXPERIENCE
4	MANINDER SINGH	PROFESSOR	17
17	RS KALER	PROFESSOR	17

23. Displaying Average CGPA of the Semesters.

Query:

```
select semester AS "SEMESTER", ROUND(avg(sgpa),2) AS "AVERAGE SGPA"
from result group by semester;
```

Output:

SEMESTER	AVERAGE SGPA
7	7.78
5	8.04
3	7.91
1	8.66

24. Displaying count of Male and Female students studying in the University.

Query:

```
select CASE WHEN gender='M' THEN 'MALE' WHEN gender='F' THEN
'FEMALE' END AS "GENDER" , count(*) AS "NO. OF STUDENTS" from student
group by gender;
```

Output:

GENDER	NO. OF STUDENTS
MALE	41
FEMALE	29

25. Displaying count of Male and Female staff working in the University.

Query:

```
select CASE WHEN gender='M' THEN 'MALE' WHEN gender='F' THEN
'FEMALE' END AS "GENDER" , count(*) AS "COUNT OF FACULTY MEMBERS"
from faculty group by gender;
```

Output:

GENDER	COUNT OF FACULTY MEMBERS
MALE	30
FEMALE	12

PL/SQL QUERIES

1. Creating Trigger to display old Salary and new updated Salary after every Updation.

Query:

```
CREATE OR REPLACE TRIGGER Update_Salary
BEFORE UPDATE
ON GRADE PAY
FOR EACH ROW
DECLARE
SAL_DIFF NUMBER;
BEGIN
SAL_DIFF:=(:NEW.BASIC_PAY+:NEW.HRA+:NEW.DA)-
(:OLD.BASIC_PAY+:OLD.HRA+:OLD.DA);
DBMS_OUPUT.PUT_LINE('OLD SALARY:
'||:OLD.BASIC_PAY+:OLD.HRA+:OLD.DA);
DBMS_OUPUT.PUT_LINE('NEW SALARY:
'||:NEW.BASIC_PAY+:NEW.HRA+:NEW.DA);
DBMS_OUPUT.PUT_LINE('INCREMENT IN SALARY:  '||:SAL_DIFF);
END;

UPDATE GRADE_PAY SET BASIC_PAY=80000 , HRA=8000 , DA=13000 WHERE
GRADE_ID=6;
```

Output:

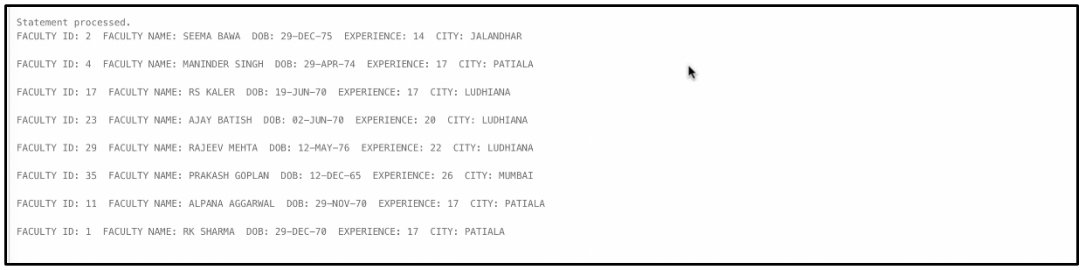
2. Creating Cursor to display details of all Professors.

Query:

```
DECLARE
CURSOR C1 IS SELECT * FROM FACULTY WHERE GRADE_ID=1;
BEGIN
FOR REC IN C1
LOOP
DBMS_OUTPUT.PUT_LINE('FACULTY ID: '||REC.FAC_ID||'  FACULTY NAME:
'||REC.FAC_NAME||'  DOB: '||REC.DOB||'  EXPERIENCE: '||REC.EXPERIENCE||'
CITY: '||REC.ADDRESS);
DBMS_OUTPUT.PUT_LINE('  ');
```

```
END LOOP;
END;
/
```

Output:

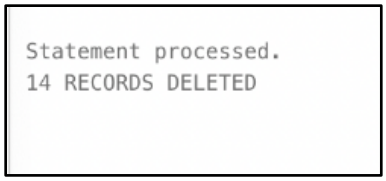


3. Using implicit Cursor to display No of Records deleted when deleting subjects of semester 3.

Query:

```
DECLARE
ROWS_DELETED NUMBER;
BEGIN
DELETE FROM COURSE_SUBJECT WHERE SEMESTER=3;
DBMS_OUTPUT.PUT_LINE(SQL%ROWCOUNT||' RECORDS DELETED');
END;
/
```

Output:



4. Creating Trigger for displaying Old Experience and new Experience whenever we update the experience.

Query:

```
CREATE OR REPLACE TRIGGER UPDATE_FAC
BEFORE UPDATE
ON FACULTY
FOR EACH ROW
BEGIN
IF ((:NEW.EXPERIENCE)<(:OLD.EXPERIENCE)) THEN
DBMS_OUTPUT.PUT_LINE('New Experience cannot be less than Old Experience');
:NEW.EXPERIENCE:=:OLD.EXPERIENCE;
ELSE
DBMS_OUTPUT.PUT_LINE('Old Experience: '||:OLD.EXPERIENCE);
DBMS_OUTPUT.PUT_LINE('New Experience: '||:NEW.EXPERIENCE);
END IF;
END;
```

/

Output:

Case 1:

```
BEGIN
UPDATE FACULTY SET EXPERIENCE=15 WHERE FAC_ID=32;
IF SQL%NOTFOUND THEN
DBMS_OUTPUT.PUT_LINE('Record not Found');
END IF;
END;
/
```

```
Statement processed.
Old Experience: 13
New Experience: 15
```

Case 2:

```
BEGIN
UPDATE FACULTY SET EXPERIENCE=15 WHERE FAC_ID=50;
IF SQL%NOTFOUND THEN
DBMS_OUTPUT.PUT_LINE('Record not Found');
END IF;
END;
/
```

```
Statement processed.
New Experience can not be less than Old Experience
```

Case 3:

```
BEGIN
UPDATE FACULTY SET EXPERIENCE=2 WHERE FAC_ID=13;
IF SQL%NOTFOUND THEN
DBMS_OUTPUT.PUT_LINE('Record not Found');
END IF;
END;
/
```

```
Statement processed.
Record not Found
```

5. Showing details of departments using cursor when given Dept_Id exists and showing error using exception handling when given Dept_Id doesn't exist .

Case 1:

Query:

```
DECLARE
CURSOR C(D_ID NUMBER) IS SELECT * FROM DEPT WHERE DEPT_ID=D_ID;
REC DEPT%ROWTYPE;
DNO NUMBER;
DEPT_NOFOUND EXCEPTION;
BEGIN
DNO:=5;
OPEN C(DNO);
LOOP
FETCH C INTO REC;
EXIT WHEN C%NOTFOUND;
DBMS_OUTPUT.PUT_LINE('DEPARTMENT ID: '||REC.DEPT_ID);
DBMS_OUTPUT.PUT_LINE('DEPARTMENT NAME: '||REC.DEPT_NAME);
DBMS_OUTPUT.PUT_LINE('LOCATION: '||REC.LOCATION);
END LOOP;
IF C%ROWCOUNT=0 THEN
RAISE DEPT_NOFOUND;
END IF;
CLOSE C;
EXCEPTION
WHEN DEPT_NOFOUND THEN
DBMS_OUTPUT.PUT_LINE('DEPARTMENT WITH DEPT_ID: '||DNO||' DOESNOT
EXISTS');
END;
/
```

Output:

```
Statement processed.
DEPARTMENT ID: 5
DEPARTMENT NAME: Chemical Engineering
LOCATION: E Block
```

Case 2:

```
DECLARE
CURSOR C(D_ID NUMBER) IS SELECT * FROM DEPT WHERE DEPT_ID=D_ID;
REC DEPT%ROWTYPE;
DNO NUMBER;
DEPT_NOFOUND EXCEPTION;
BEGIN
```

```
DNO:=12;
OPEN C(DNO);
LOOP
FETCH C INTO REC;
EXIT WHEN C%NOTFOUND;
DBMS_OUTPUT.PUT_LINE('DEPARTMENT ID: '||REC.DEPT_ID);
DBMS_OUTPUT.PUT_LINE('DEPARTMENT NAME: '||REC.DEPT_NAME);
DBMS_OUTPUT.PUT_LINE('LOCATION: '||REC.LOCATION);
END LOOP;
IF C%ROWCOUNT=0 THEN
RAISE DEPT_NOFOUND;
END IF;
CLOSE C;
EXCEPTION
WHEN DEPT_NOFOUND THEN
DBMS_OUTPUT.PUT_LINE('DEPARTMENT WITH DEPT_ID: '||DNO||' DOESNOT
EXISTS');
END;
/
```

Output:

```
Statement processed.
DEPARTMENT WITH DEPT_ID: 12 DOESNOT EXISTS
```

6. Showing use of a Function in PL_SQL by printing total no of employees working in the University.

Query:

```
CREATE OR REPLACE FUNCTION FAC_COUNT
RETURN NUMBER
AS
COUNT_F NUMBER;
BEGIN
SELECT COUNT(*) INTO COUNT_F FROM FACULTY;
RETURN COUNT_F;
END;

DECLARE
COUNT_F NUMBER;
BEGIN
COUNT_F:=FAC_COUNT();
DBMS_OUTPUT.PUT_LINE('TOTAL NUMBER OF EMPLOYEES CURRENTLY
WORKING IN UNIVERSITY = '||COUNT_F);
END;
```

Output:

```
Statement processed.
TOTAL NUMBER OF EMPLOYEES CURRENTLY WORKING IN UNIVERSITY = 42
```