Database Management System

Project

Hostel Room Allotment System

Review 3

Akul Garg **17BIT0298**

Submitted to

Prof. Bimal Kumar Ray

Question1:

Motivation:

For the past few years the number of educational institutions is increasing rapidly which leads to increase in the number of hostels to accommodate the students. The hostel room allotment for students used to be done manually through applications to Wardens of the hostels and the students need to be present physically for the same. This system is especially tedious in case of a large number of students and because of the requirement to fill out pen-paper forms. The administrators (Wardens and assistants) also need to process the forms which can be inefficient and tough.

The project aims to transfer most of the manual labour to a computerized system that will be compatible to the existing system with the system which is more user friendly to both the students and the administrators (Wardens) and more GUI oriented.

Breakdown of the System:

The online room allotment system should basically consist of three distinct sections:

1) *Log-In Section:* The first section of the Hostel Room Allotment system, which makes sure the system is accessed only by authorized users. Basically consists of a form with required fields including a user ID (the institute e-mail ID) and password. This is checked against the established institute database.
2) *Input Section:* Once the user has been authenticated, he is required to fill out various details through certain forms. The forms should cover the fields described under the '*Requirements from the user (student) side'* section. It is essential the information provided here is valid and consistent with the information provided in the institute. This information will be used to allot hostel rooms to the students and may not be changed in the future, barring exceptions.
3) *Processing Section:* Once the student has filled out all the required information and successfully submitted it, the room will be reserved for that particular student. The system should take care of this section and no user interaction should alter this phase in any way.

Data Requirements:

The following are required for the project from the admin (warden) side:

1) **Hostel Information:** List of hostels used for accommodation.
2) **Seating Capacity of hostels:** How many seats are available in each hostel?
3) **Hostel Allotment information:** Which batch of each branch has been allotted to which hostel?
4) **Batch strength Information:** How many students are there in each batch?
5) **Student Information:** For each student, the information about the deposition of the current session's Mess Fees is needed.
6) For each student, we need the information about the clearance of last session's Mess Dues. No student is allotted a room until all his Mess Dues have been cleared.
7) Access to other credentials of each student including his institute e-mail ID information.
8) The range of rooms the students of a particular batch can occupy.

The following are required for the project from the user (student) side:

1) **Log In credentials:** No student is allowed to access the forms without proper authentication using the institute e-mail ID and password. Each student may only register using his institute ID.
2) **Course enrolled in (course)**: What course the student is pursuing, e.g. B. Tech, M. Tech, Integrated Dual Degree, PHD etc.
3) **Department enrolled in (Dept):** What department the student belongs to, e.g. Department of Computer Science and Engineering.
4) **Session enrolled in (year):** The session in which the student has enrolled in, e.g. 2017-18 5) **Part enrolled in**: The year the student has enrolled in this session, e.g. Part II, Part III etc.
6) **Student Name (Name)**: Name of the student. Must be consistent with the name provided at the time of registration in the institute.
7) **Student Roll Number (roll)**: The unique roll number provided to the student at the time of registration.
8) **Mess Fee Receipt Reference Number(receipt)**: The reference number of the Mess Fee receipt used to pay the Mess Charge Advanced.
9) **Previous hostel allotment information (prev_r):** Includes information about the previous hostel the applicant resided in and his room number.
10) **No-dues certificate:** The student must upload a copy of his no-dues certificate as obtained from his previous hostel warden.

11) **Room Number**: The students will be given a range of rooms to choose from in the allotted hostels. The students may fill the number of the room of their preference and these will be allotted on a First Come First Serve basis.
12) **Contact Information (cont):** The contact information of the student. Includes mobile number for contact purposes.
13) **Guardian Information (g_name, g_cont, g_add):** The address and contact information of the guardian of the student, e.g. contact information of the father/mother of the student.
14) **Local Guardian Information (l_name,L_cont,l_add):** The address and contact information of the local guardian of the student to refer to in case of any emergencies.
15) **Bank account information (Bank):** The bank account information of the student. Used to deposit the leftover funds after the semester/session.

Rules and Functional requirements from the system:

1) The student should apply online using their institute e-mail IDs only. Personal IDs and unauthenticated users should not be allowed to apply for hostel room allotment.
2) Student's registration details should be able to access by warden on passing the roll number of student.
3) In case the login credentials don't match with the ones already present in the database then an error message should be displayed and student should be asked to enter credentials again.
4) After several number of failed attempts student should be asked to contact warden to reset login id and password.
5) The students of a particular batch may only apply in the hostel allotted to them as decided by the administration. It is essential that this information be provided by the administrators and the students cannot alter this once it is decided.
6) The student may only fill the room preference from the range of rooms allotted to the particular batch the student belongs in.
7) The rooms should be allotted strictly on a First Come First Serve basis.
8) No student is allotted a room until his Mess fees and College tuition fees has been paid. The Mess Fees and tuition fees receipt reference number provided by the student should be consistent.
9) The student who has not paid fees should not be allowed to login into the system displaying a proper message asking to pay the dues.

10) After payment of dues warden should produce a no dues certification. Student should be allowed to enter the system only after submission of no dues certificate.
11) There should be no clashes in the room allotment. In case of any discrepancy, the decision of the Hostel Warden is final and binding, other factors notwithstanding.
12) The students may fill only one choice and it is advised that the students fill their choice carefully.
13) Once the student has filled in his/her choice, the choice may not be changed. All applications for change of choice will be dealt by the Head Warden separately.
14) After submitting the online request, the student must take the printout of the Hostel Allotment Form and submit the duly signed form in Hostel office along with the supporting documents.
15) The room will be considered 'allotted' only when the form has been properly submitted in the office of the respective hostel. The rooms will be considered 'reserved' and not 'allotted' if the student has filled out the online form and has not submitted the hard copy of the form along with the required documents in the hostel office.
16) All other discrepancies will be dealt with by the Hostel Warden or Dean of Student Affairs as applicable. Their decision will be final and binding.

Scenarios of removal of old data

1. If student is not able to submit no-dues certificate before counselling date, his record should be deleted by warden (admin).
2. If a student wants to withdraw from counselling before counselling date, his record will be deleted after contacting warden (admin).

Scenarios of data modification

1 If a student wants to change his/her room, they can apply for room change to the head warden and warden can modify the room number of the particular student.

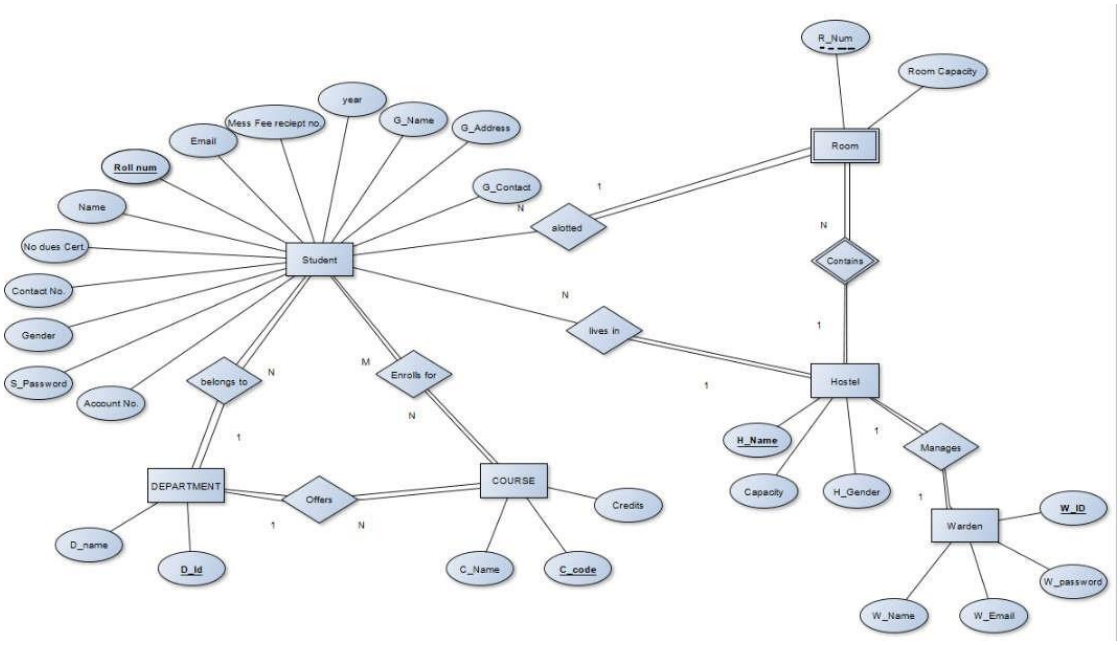
2 If a student submits no dues certificate before time then warden should modify his status and he should be able to login.

3. After the allotment of rooms for the new session, room details of all the students should be modified.
4. The availability status such as number of rooms allotted and number of rooms left should be modified as soon as a room is allotted.

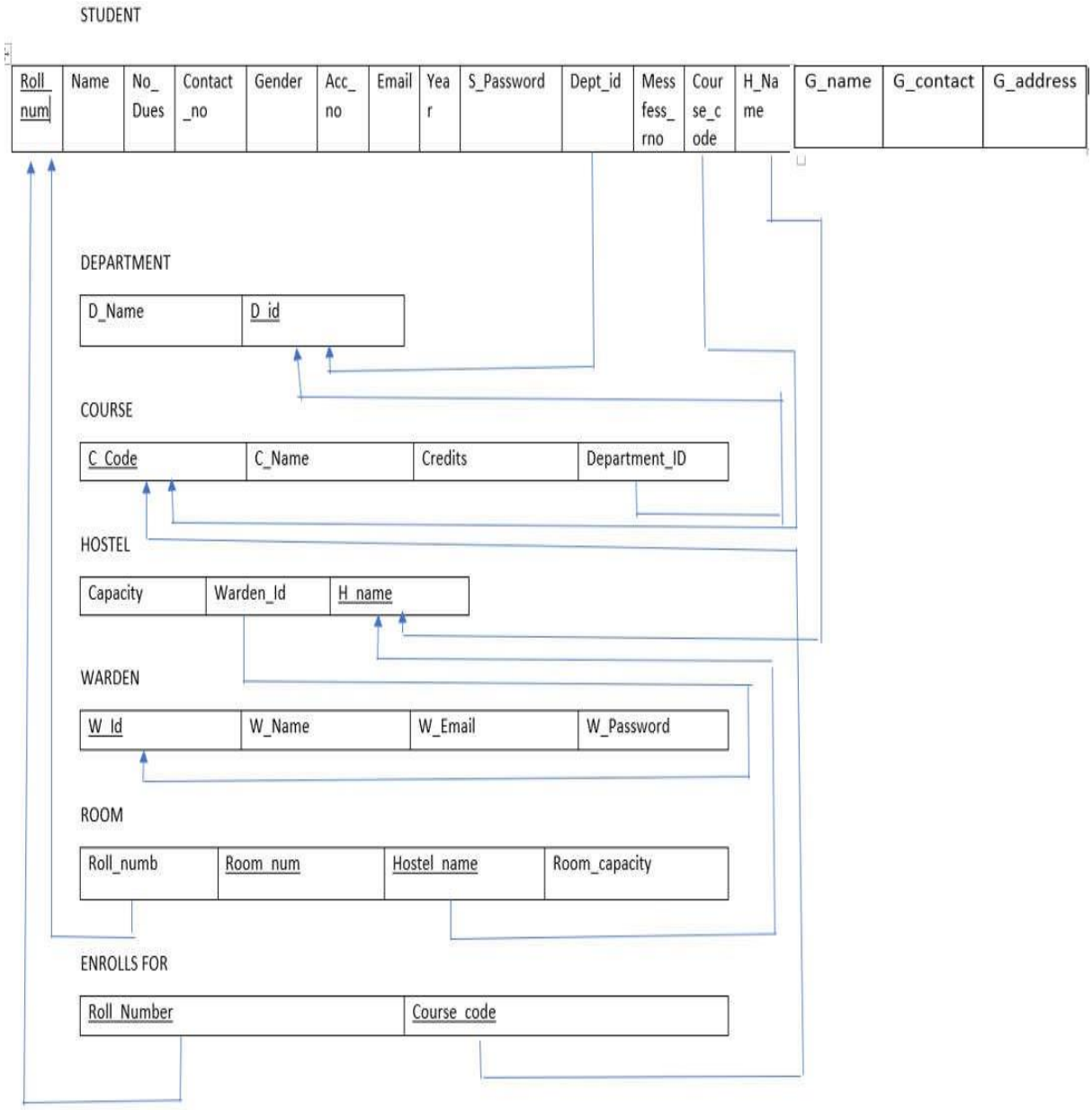
Scenarios of data retrieval

1. Warden can access the student details such as branch details, course details, previous hostel details etc.
2. Warden should have student's banking details to manage payment process and should have account of student's fee dues.
3. Students should be able to access the hostel details i.e., number of rooms, room's capacity before the commencement of counselling.
4. Students should get live updates about the availability of rooms as soon as a room is allotted.
5. After the completion of counselling the warden should get a complete list of students and the rooms allotted to them.

ER diagram



ER to Relational Mapping



Implementation of Relational Database Schema

Creation of Tables

Department

```
mysql> CREATE DATABASE hostelRoomAlloc;
Query OK, 1 row affected (0.17 sec)

mysql> USE hostelRoomAlloc;
Database changed
mysql> CREATE TABLE `department` (
    ->   `D_Id` varchar(10) NOT NULL,
    ->   `D_Name` varchar(40) NOT NULL,
    ->   PRIMARY KEY (`D_Id`),
    ->   UNIQUE KEY `D_Id` (D_Id)
    -> );
Query OK, 0 rows affected (0.83 sec)
```

Course, Warden and Hostel

```
mysql> CREATE TABLE `course` (
    ->   `C_code` varchar(10) NOT NULL,
    ->   `C_name` varchar(40) NOT NULL,
    ->   `Credits` int(10) unsigned NOT NULL,
    ->   `Department_ID` varchar(10) NOT NULL,
    ->   PRIMARY KEY (`C_code`),
    ->   UNIQUE KEY `C_code` (`C_code`),
    ->   KEY `fk_dept_id` (`Department_ID`),
    ->   CONSTRAINT `fk_dept_id` FOREIGN KEY (`Department_ID`) REFERENCES `department` (`D_Id`)
    -> );
Query OK, 0 rows affected (0.60 sec)

mysql> CREATE TABLE `warden` (
    ->   `w_id` varchar(9) NOT NULL,
    ->   `w_email` varchar(50) NOT NULL,
    ->   `w_name` varchar(40) NOT NULL,
    ->   `w_password` varchar(8) NOT NULL,
    ->   PRIMARY KEY (`w_id`),
    ->   UNIQUE KEY `w_id` (`w_id`)
    -> );
Query OK, 0 rows affected (0.45 sec)

mysql> CREATE TABLE `hostel` (
    ->   `h_name` varchar(15) NOT NULL,
    ->   `capacity` int(11) unsigned NOT NULL,
    ->   `h_warden_id` varchar(9) NOT NULL,
    ->   PRIMARY KEY (`h_name`),
    ->   UNIQUE KEY `h_name` (`h_name`),
    ->   KEY `fk_w_id` (`h_warden_id`),
    ->   CONSTRAINT `fk_w_id` FOREIGN KEY (`h_warden_id`) REFERENCES `warden` (`w_id`)
    -> );
Query OK, 0 rows affected (0.68 sec)
```

Student

```
mysql> CREATE TABLE `student` (
    ->   `name` varchar(40) NOT NULL,
    ->   `roll_num` varchar(9) NOT NULL,
    ->   `email` varchar(50) NOT NULL,
    ->   `s_password` varchar(8) NOT NULL,
    ->   `hostel_name` varchar(15) NOT NULL,
    ->   `mess_fee_receipt_num` varchar(20) NOT NULL,
    ->   `no_dues_certificate` char(1) NOT NULL,
    ->   `contact_num` varchar(13) NOT NULL,
    ->   `gender` enum('M','F') NOT NULL,
    ->   `dept_id` varchar(10) NOT NULL,
    ->   `course_code` varchar(10) NOT NULL,
    ->   `year` int(10) unsigned NOT NULL,
    ->   `guardian_name` varchar(50) NOT NULL,
    ->   `guardian_contact` varchar(13) NOT NULL,
    ->   `guardian_address` varchar(300) NOT NULL,
    ->   `bank_account_number` varchar(100) DEFAULT NULL,
    ->   PRIMARY KEY (`roll_num`),
    ->   UNIQUE KEY `roll_num` (`roll_num`),
    ->   UNIQUE KEY `email` (`email`),
    ->   UNIQUE KEY `mess_fee_receipt_num` (`mess_fee_receipt_num`),
    ->   KEY `fk_stud_dept` (`dept_id`),
    ->   KEY `fk_stud_id` (`roll_num`),
    ->   KEY `fk_stud_host` (`hostel_name`),
    ->   CONSTRAINT `fk_stud_dept` FOREIGN KEY (`dept_id`) REFERENCES `department` (`D_Id`),
    ->   CONSTRAINT `fk_stud_course` FOREIGN KEY (`course_code`) REFERENCES `course` (`C_code`),
    ->   CONSTRAINT `fk_stud_host` FOREIGN KEY (`hostel_name`) REFERENCES `hostel` (`h_name`)
    -> );
Query OK, 0 rows affected (0.94 sec)
```

Room

```
mysql> CREATE TABLE `room` (
    ->   `r_num` int(10) unsigned NOT NULL,
    ->   `r_capacity` int(10) NOT NULL,
    ->   `host_name` varchar(15) NOT NULL,
    ->   `roll_numb` varchar(9) NOT NULL,
    ->   PRIMARY KEY (`host_name`,`r_num`),
    ->   KEY `fk_roll` (`roll_numb`),
    ->   CONSTRAINT `fk_roll` FOREIGN KEY (`roll_numb`) REFERENCES `student` (`roll_num`)
    -> );
Query OK, 0 rows affected (0.58 sec)
```

Enrols for

```
mysql> CREATE TABLE `enrolls_for` (
    ->   `roll_number` varchar(9) NOT NULL,
    ->   `crse_code` varchar(10) NOT NULL,
    ->   PRIMARY KEY (`roll_number`, `crse_code`),
    ->   KEY `fk_roll_num` (`roll_number`),
    ->   KEY `fk_course` (`crse_code`),
    ->   CONSTRAINT `fk_roll_num` FOREIGN KEY (`roll_number`) REFERENCES `student` (`roll_num`),
    ->   CONSTRAINT `fk_course` FOREIGN KEY (`crse_code`) REFERENCES `course` (`C_code`)
    -> );
Query OK, 0 rows affected (0.94 sec)
```

Insertion into Tables

Department

```
mysql> INSERT INTO `department` VALUES
    ->   ('SCOPE','Computer Science and Engineering'),
    ->   ('SELECT','Electrical Engineering'),
    ->   ('SITE','Information Technology'),
    ->   ('SMEC','Mechanical Engineering');
Query OK, 4 rows affected (0.15 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

Course and Warden

```
mysql> INSERT INTO `course` VALUES
    ->   ('ITE1003', 'Database Management', 4, 'SCOPE'),
    ->   ('ITE1002', 'Web Technologies', 3, 'SITE'),
    ->   ('ITE1001', 'Environmental Science', 2, 'SITE'),
    ->   ('ITE1006', 'Mechanics', 3, 'SMEC'),
    ->   ('ITE1007', 'Engineering Physics', 4, 'SCOPE'),
    ->   ('ITE2003', 'Electrical Engineering', 3, 'SELECT'),
    ->   ('ITE1005', 'Software Engineering', 3, 'SITE'),
    ->   ('ITE2002', 'Calculus for Engineers', 4, 'SMEC'),
    ->   ('ITE1102', 'Chemistry for Engineers', 4, 'SCOPE'),
    ->   ('ITE2001', 'Computer Architecture', 3, 'SELECT'),
    ->   ('ITE1009', 'Digital Logic', 2, 'SITE'),
    ->   ('ITE1014', 'Ethics and Values', 1, 'SMEC'),
    ->   ('ITE2007', 'Statistics for Engineers', 4, 'SCOPE'),
    ->   ('ITE1123', 'Surveying', 3, 'SELECT'),
    ->   ('ITE1213', 'Innovative Projects', 2, 'SITE'),
    ->   ('ITE2004', 'Soft Skills', 1, 'SMEC');
Query OK, 16 rows affected (0.19 sec)
Records: 16  Duplicates: 0  Warnings: 0

mysql> INSERT INTO `warden` VALUES
    ->   ('101', 'ABC@vit.ac.in', 'ABC', 'A23@xpos'),
    ->   ('102', 'DEF@vit.ac.in', 'DEF', 'E76!iosr'),
    ->   ('103', 'GHI@vit.ac.in', 'GHI', 'B34@uios'),
    ->   ('104', 'JKL@vit.ac.in', 'JKL', 'S76!ifsr'),
    ->   ('105', 'MNO@vit.ac.in', 'MNO', 'Q63@xtys'),
    ->   ('106', 'PQR@vit.ac.in', 'PQR', 'ZS6!lkjr'),
    ->   ('107', 'STU@vit.ac.in', 'STU', '!w3@xpos'),
    ->   ('108', 'VWX@vit.ac.in', 'VWX', 'E51!iowe'),
    ->   ('109', 'YZA@vit.ac.in', 'YZA', 'T24@yhjs'),
    ->   ('110', 'BCD@vit.ac.in', 'BCD', 'FF6$mvgr'),
    ->   ('111', 'EFG@vit.ac.in', 'EFG', 'C23@klms'),
    ->   ('112', 'HIJ@vit.ac.in', 'HIJ', 'X16!itsr'),
    ->   ('113', 'KLM@vit.ac.in', 'KLM', '133@slka'),
    ->   ('114', 'NOP@vit.ac.in', 'NOP', '346!kfnc'),
    ->   ('115', 'QRS@vit.ac.in', 'QRS', 'Qp3@vbxs'),
    ->   ('116', 'TUV@vit.ac.in', 'TUV', 'E86!ilpo');
Query OK, 16 rows affected (0.14 sec)
Records: 16  Duplicates: 0  Warnings: 0
```

Hostel and Student

```
mysql> INSERT INTO `hostel` VALUES
    ->   ('L-Block',400,'101'),
    ->   ('K-Block',600,'103');
Query OK, 2 rows affected (0.26 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> INSERT INTO `student` VALUES
    ->   ('ABCD', '17BCE0207', 'ABCD@vitstudent.ac.in', 'S23@xpos', 'L-Block', 1234, 'Y', '9123678123', 'M', 'SCOPE', 'ITE1003', 1, 'ABYZ', '8212312392', 'H.No.13, Gand
hi Nagar, Muzaffarnagar, UP', 'A123492310'),
    ->   ('DEFG', '17BCE0386', 'DEFG@vitstudent.ac.in', 'S76!iosr', 'K-Block', 2254, 'N', '9225678225', 'F', 'SELECT', 'ITE1002', 2, 'ABDJ', '8222522592', 'H.No.25, Raj
endra Nagar, Bhiwadi, MP', 'B123492520'),
    ->   ('HIJK', '17BCE0902', 'HIJK@vitstudent.ac.in', 'S34@uios', 'L-Block', 3554, 'Y', '9355678355', 'M', 'SITE', 'ITE1001', 3, 'ABPT', '8235535592', 'H.No.35, Suren
dra Nagar, Siyachin, J&K', 'C123495530'),
    ->   ('LMNO', '17BCE0920', 'LMNO@vitstudent.ac.in', 'S76!ifsr', 'K-Block', 7264, 'Y', '9726678726', 'F', 'SMEC', 'ITE1006', 4, 'ABQV', '8272672692', 'H.No.76, Rajpa
l Nagar, Shastri Nagar, Delhi', 'D127267890'),
    ->   ('PQRS', '17BEE0212', 'PQRS@vitstudent.ac.in', 'S63@xtys', 'L-Block', 4534, 'Y', '9453678453', 'M', 'SCOPE', 'ITE1007', 1, 'ABAB', '8245345392', 'H.No.43, Nare
ndra Nagar, Kanpur, UP', 'E123495340'),
    ->   ('TUVW', '17BEE0378', 'TUVW@vitstudent.ac.in', 'SS6!lkjr', 'K-Block', 5684, 'N', '9568678568', 'F', 'SELECT', 'ITE2003', 2, 'ABCD', '8256856892', 'H.No.58, Ind
ra Nagar, Rampuri, MP', 'F123496850'),
    ->   ('XYZA', '17BEE0908', 'XYZA@vitstudent.ac.in', 'Sw3@xpos', 'L-Block', 6234, 'N', '9623678623', 'M', 'SITE', 'ITE1005', 3, 'ABEF', '8262362392', 'H.No.63, Jawah
ar Nagar. Rajouri. J&K', 'G123492360'),
    ->   ('BCDE', '17BEE2002', 'BCDE@vitstudent.ac.in', 'S51!iowe', 'K-Block', 1434, 'Y', '9123678123', 'F', 'SMEC', 'ITE2002', 4, 'ABGH', '8212312392', 'H.No.13, Ram N
agar, Rohini, Delhi', 'G123492310'),
    ->   ('FGHI', '17BIT0102', 'FGHI@vitstudent.ac.in', 'S24@yhjs', 'L-Block', 3164, 'Y', '9316678316', 'M', 'SCOPE', 'ITE1102', 1, 'ABIJ', '8231631692', 'H.No.36, Mamt
a Nagar, Saharanpur, UP', 'H123491630'),
    ->   ('JKLM', '17BIT0120', 'JKLM@vitstudent.ac.in', 'SF6$mvgr', 'K-Block', 2044, 'Y', '9204678204', 'F', 'SELECT', 'ITE2001', 2, 'ABKL', '8220420492', 'H.No.24, Bha
gat Nagar, Laxmanpuri, MP', 'I123490420'),
    ->   ('NOPQ', '17BIT0298', 'NOPQ@vitstudent.ac.in', 'S23@klms', 'L-Block', 5544, 'N', '9554678554', 'M', 'SITE', 'ITE1009', 3, 'ABMN', '8255455492', 'H.No.54, Ashok
 Nagar, Rajghat, J&K', 'J12349545O'),
    ->   ('RSTU', '17BIT0378', 'RSTU@vitstudent.ac.in', 'S16!itsr', 'K-Block', 7594, 'Y', '9759678759', 'F', 'SMEC', 'ITE1014', 4, 'ABOP', '8275975992', 'H.No.79, Gandh
i Nagar, Hauz Khas, Delhi', 'K123495970'),
    ->   ('VWXY', '17BME0156', 'VWXY@vitstudent.ac.in', 'S33@slka', 'L-Block', 1024, 'N', '9102678102', 'M', 'SCOPE', 'ITE2007', 1, 'ABQR', '8210210292', 'H.No.12, Mahe
ndra Nagar, Meerut, UP', 'L123490210'),
    ->   ('ZABC', '17BME0897', 'ZABC@vitstudent.ac.in', 'S46!kfnc', 'K-Block', 4044, 'Y', '9404678404', 'F', 'SELECT', 'ITE1123', 2, 'ABST', '8240440492', 'H.No.44, Raj
endra Nagar, Beejpur, MP', 'M123490440'),
    ->   ('WEFG', '17BME2001', 'WEFG@vitstudent.ac.in', 'Sp3@vbxs', 'L-Block', 5124, 'Y', '9512678512', 'M', 'SITE', 'ITE1213', 3, 'ABUV', '8251251292', 'H.No.52, Gandh
i Nagar, Krishnapur, J&K', 'N123491250'),
    ->   ('XHIJ', '17BME2019', 'XHIJ@vitstudent.ac.in', 'S86!ilpo', 'K-Block', 7024, 'Y', '9702678702', 'F', 'SMEC', 'ITE2004', 4, 'ABWX', '8270270292', 'H.No.72, Pradi
p Nagar, Indirapuram, Delhi', 'O12345027678');
Query OK, 16 rows affected (0.32 sec)
Records: 16  Duplicates: 0  Warnings: 0
```

Room

```
mysql> INSERT INTO `room` VALUES
    ->   (123,4,'L-Block', '17BCE0207'),
    ->   (225,6,'K-Block', '17BCE0386'),
    ->   (355,4,'L-Block', '17BCE0902'),
    ->   (726,6,'K-Block', '17BCE0920'),
    ->   (453,4,'L-Block', '17BEE0212'),
    ->   (568,6,'K-Block', '17BEE0378'),
    ->   (623,4,'L-Block', '17BEE0908'),
    ->   (123,6,'K-Block', '17BEE2002'),
    ->   (316,4,'L-Block', '17BIT0102'),
    ->   (204,6,'K-Block', '17BIT0120'),
    ->   (554,4,'L-Block', '17BIT0298'),
    ->   (759,6,'K-Block', '17BIT0378'),
    ->   (102,4,'L-Block', '17BME0156'),
    ->   (404,6,'K-Block', '17BME0897'),
    ->   (512,4,'L-Block', '17BME2001'),
    ->   (702,6,'K-Block', '17BME2019');
Query OK, 16 rows affected (0.14 sec)
Records: 16  Duplicates: 0  Warnings: 0
```

Display of Tables

Student

mysql> select * from student;

name	roll_num	email	s_password	hostel_name	mess_fee_receipt_num	no_dues_certificate	contact_num	gender	dept_id	course_code	year	guardian_name	guardian_contact	guardian_address	bank_account_number
ABCD	17BCE0207	ABCD@vitstudent.ac.in	S23@xpos	L-Block	1234	Y	9123678123	M	SCOPE	ITE1003	1	ABYZ	8212312392	H.No.13, Gandhi Nagar, Muzaffarnagar, UP	A123492310
DEFG	17BCE0386	DEFG@vitstudent.ac.in	S76!iosr	K-Block	2254	N	9225678225	F	SELECT	ITE1002	2	ABDJ	8222522592	H.No.25, Rajendra Nagar, Bhiwadi, MP	B123492520
HIJK	17BCE0902	HIJK@vitstudent.ac.in	S34@uios	L-Block	3554	Y	9355678355	M	SITE	ITE1001	3	ABPT	8235535592	H.No.35, Surendra Nagar, Siyachin, J&K	C123495530
LMNO	17BCE0920	LMNO@vitstudent.ac.in	S76!ifsr	K-Block	7264	Y	9726678726	F	SMEC	ITE1006	4	ABQV	8272672692	H.No.76, Rajpal Nagar, Shastri Nagar, Delhi	D127267890
PQRS	17BEE0212	PQRS@vitstudent.ac.in	S63@xtys	L-Block	4534	Y	9453678453	M	SCOPE	ITE1007	1	ABAB	8245345392	H.No.43, Narendra Nagar, Kanpur, UP	E123495340
TUVW	17BEE0378	TUVW@vitstudent.ac.in	S56!lkjr	K-Block	5684	N	9568678568	F	SELECT	ITE2003	2	ABCD	8256856892	H.No.58, Indra Nagar, Rampuri, MP	F123496850
XYZA	17BEE0908	XYZA@vitstudent.ac.in	Sw3@xpos	L-Block	6234	N	9623678623	M	SITE	ITE1005	3	ABEF	8262362392	H.No.63, Jawahar Nagar, Rajouri, J&K	G123492360
BCDE	17BEE2002	BCDE@vitstudent.ac.in	S51!iowe	K-Block	1434	Y	9123678123	F	SMEC	ITE2002	4	ABGH	8212312392	H.No.13, Ram Nagar, Rohini, Delhi	G123492310
FGHI	17BIT0102	FGHI@vitstudent.ac.in	S24@yhjs	L-Block	3164	Y	9316678316	M	SCOPE	ITE1102	1	ABIJ	8231631692	H.No.36, Mamta Nagar, Saharanpur, UP	H123491630
JKLM	17BIT0120	JKLM@vitstudent.ac.in	SF6$mvgr	K-Block	2044	Y	9204678204	F	SELECT	ITE2001	2	ABKL	8220420492	H.No.24, Bhagat Nagar, Laxmanpuri, MP	I123490420
NOPQ	17BIT0298	NOPQ@vitstudent.ac.in	S23@klms	L-Block	5544	N	9554678554	M	SITE	ITE1009	3	ABMN	8255455492	H.No.54, Ashok Nagar, Rajghat, J&K	J123495450
RSTU	17BIT0378	RSTU@vitstudent.ac.in	S16!itsr	K-Block	7594	Y	9759678759	F	SMEC	ITE1014	4	ABOP	8275975992	H.No.79, Gandhi Nagar, Hauz Khas, Delhi	K123495970
VWXY	17BME0156	VWXY@vitstudent.ac.in	S33@slka	L-Block	1024	N	9102678102	M	SCOPE	ITE2007	1	ABQR	8210210292	H.No.12, Mahendra Nagar, Meerut, UP	L123490210
ZABC	17BME0897	ZABC@vitstudent.ac.in	S46!kfnc	K-Block	4044	Y	9404678404	F	SELECT	ITE1123	2	ABST	8240440492	H.No.44, Rajendra Nagar, Beejpur, MP	M123490440
WEFG	17BME2001	WEFG@vitstudent.ac.in	Sp3@vbxs	L-Block	5124	Y	9512678512	M	SITE	ITE1213	3	ABUV	8251251292	H.No.52, Gandhi Nagar, Krishnapur, J&K	N123491250
XHIJ	17BME2019	XHIJ@vitstudent.ac.in	S86!ilpo	K-Block	7024	Y	9702678702	F	SMEC	ITE2004	4	ABWX	8270270292	H.No.72, Pradip Nagar, Indirapuram, Delhi	O12345027678

16 rows in set (0.00 sec)

Department

mysql> select * from department;

D_Id	D_Name
SCOPE	Computer Science and Engineering
SELECT	Electrical Engineering
SITE	Information Technology
SMEC	Mechanical Engineering

4 rows in set (0.00 sec)

Course

```
mysql> select * from course;
+---------+--------------------------+---------+---------------+
| C_code  | C_name                   | Credits | Department_ID |
+---------+--------------------------+---------+---------------+
| ITE1001 | Environmental Science    |       2 | SITE          |
| ITE1002 | Web Technologies         |       3 | SITE          |
| ITE1003 | Database Management      |       4 | SCOPE         |
| ITE1005 | Software Engineering     |       3 | SITE          |
| ITE1006 | Mechanics                |       3 | SMEC          |
| ITE1007 | Engineering Physics      |       4 | SCOPE         |
| ITE1009 | Digital Logic            |       2 | SITE          |
| ITE1014 | Ethics and Values        |       1 | SMEC          |
| ITE1102 | Chemistry for Engineers  |       4 | SCOPE         |
| ITE1123 | Surveying                |       3 | SELECT        |
| ITE1213 | Innovative Projects      |       2 | SITE          |
| ITE2001 | Computer Architecture    |       3 | SELECT        |
| ITE2002 | Calculus for Engineers   |       4 | SMEC          |
| ITE2003 | Electrical Engineering   |       3 | SELECT        |
| ITE2004 | Soft Skills              |       1 | SMEC          |
| ITE2007 | Statistics for Engineers |       4 | SCOPE         |
+---------+--------------------------+---------+---------------+
16 rows in set (0.00 sec)
```

Hostel

```
mysql> select * from hostel;
+---------+----------+-------------+
| h_name  | capacity | h_warden_id |
+---------+----------+-------------+
| K-Block |      600 | 103         |
| L-Block |      400 | 101         |
+---------+----------+-------------+
2 rows in set (0.00 sec)
```

Warden

```
mysql> select * from warden;
+------+--------------+--------+------------+
| w_id | w_email      | w_name | w_password |
+------+--------------+--------+------------+
| 101  | ABC@vit.ac.in | ABC    | A23@xpos   |
| 102  | DEF@vit.ac.in | DEF    | E76!iosr   |
| 103  | GHI@vit.ac.in | GHI    | B34@uios   |
| 104  | JKL@vit.ac.in | JKL    | S76!ifsr   |
| 105  | MNO@vit.ac.in | MNO    | Q63@xtys   |
| 106  | PQR@vit.ac.in | PQR    | ZS6!lkjr   |
| 107  | STU@vit.ac.in | STU    | !w3@xpos   |
| 108  | VWX@vit.ac.in | VWX    | E51!iowe   |
| 109  | YZA@vit.ac.in | YZA    | T24@yhjs   |
| 110  | BCD@vit.ac.in | BCD    | FF6$mvgr   |
| 111  | EFG@vit.ac.in | EFG    | C23@klms   |
| 112  | HIJ@vit.ac.in | HIJ    | X16!itsr   |
| 113  | KLM@vit.ac.in | KLM    | 133@slka   |
| 114  | NOP@vit.ac.in | NOP    | 346!kfnc   |
| 115  | QRS@vit.ac.in | QRS    | Qp3@vbxs   |
| 116  | TUV@vit.ac.in | TUV    | E86!ilpo   |
+------+--------------+--------+------------+
16 rows in set (0.00 sec)
```

Room

```
mysql> select * from room;
+-------+------------+-----------+-----------+
| r_num | r_capacity | host_name | roll_numb |
+-------+------------+-----------+-----------+
|   123 |          6 | K-Block   | 17BEE2002 |
|   204 |          6 | K-Block   | 17BIT0120 |
|   225 |          6 | K-Block   | 17BCE0386 |
|   404 |          6 | K-Block   | 17BME0897 |
|   568 |          6 | K-Block   | 17BEE0378 |
|   702 |          6 | K-Block   | 17BME2019 |
|   726 |          6 | K-Block   | 17BCE0920 |
|   759 |          6 | K-Block   | 17BIT0378 |
|   102 |          4 | L-Block   | 17BME0156 |
|   123 |          4 | L-Block   | 17BCE0207 |
|   316 |          4 | L-Block   | 17BIT0102 |
|   355 |          4 | L-Block   | 17BCE0902 |
|   453 |          4 | L-Block   | 17BEE0212 |
|   512 |          4 | L-Block   | 17BME2001 |
|   554 |          4 | L-Block   | 17BIT0298 |
|   623 |          4 | L-Block   | 17BEE0908 |
+-------+------------+-----------+-----------+
16 rows in set (0.00 sec)
```

Enrols for

```
mysql> select * from enrolls_for;
+-------------+-----------+
| roll_number | crse_code |
+-------------+-----------+
| 17BCE0207   | ITE1003   |
| 17BCE0386   | ITE1002   |
| 17BCE0902   | ITE1001   |
| 17BCE0920   | ITE1006   |
| 17BEE0212   | ITE1007   |
| 17BEE0378   | ITE2003   |
| 17BEE0908   | ITE1005   |
| 17BEE2002   | ITE2002   |
| 17BIT0102   | ITE1102   |
| 17BIT0120   | ITE2001   |
| 17BIT0298   | ITE1009   |
| 17BIT0378   | ITE1014   |
| 17BME0156   | ITE2007   |
| 17BME0897   | ITE1123   |
| 17BME2001   | ITE1213   |
| 17BME2019   | ITE2004   |
+-------------+-----------+
16 rows in set (0.00 sec)
```

SQL Queries for the Implementation of functional Requirements

Update

Changing department name to CSE where department Id is SCOPE

```
SQL> update department
  2  set d_name='CSE' where d_id='SCOPE';

1 row updated.

SQL> select * from department;

D_ID       D_NAME
---------- ----------------------------------------
SCOPE      CSE
SELECT     Electrical Engineering
SITE       Information Technology
SMEC       Mechanical Engineering
```

Changing the name of warden with Id 101

```
SQL> update warden
  2  set w_name='hitler'
  3  where w_id='101';

1 row updated.

SQL> select * from warden where w_id='101';

W_ID      W_EMAIL
--------- --------------------------------------------------
W_NAME                                    W_PASSWO
----------------------------------------- --------
101       ABC@vit.ac.in
hitler                                    A23@xpos
```

Changing the name of student with roll number 17BIT0902

```
SQL> update student
  2  set name='vishal'
  3  where roll_num='17BCE0902';

1 row updated.
```

```
SQL> select name,roll_num,email from student where roll_num='17BCE0902';

NAME                                      ROLL_NUM
----------------------------------------- ---------
EMAIL
--------------------------------------------------
vishal                                    17BCE0902
HIJK@vitstudent.ac.in
```

DELETE

Deleting the record of room number 512

```
SQL> select * from room;

     R_NUM R_CAPACITY HOST_NAME       ROLL_NUMB
---------- ---------- --------------- ---------
       355          4 L-Block         17BCE0902
       453          4 L-Block         17BEE0212
       568          6 K-Block         17BEE0378
       623          4 L-Block         17BEE0908
       316          4 L-Block         17BIT0102
       204          6 K-Block         17BIT0120
       554          4 L-Block         17BIT0298
       102          4 L-Block         17BME0156
       404          6 K-Block         17BME0897
       512          4 L-Block         17BME2001

10 rows selected.

SQL> delete from room where r_num='512';

1 row deleted.

SQL> select * from room;

     R_NUM R_CAPACITY HOST_NAME       ROLL_NUMB
---------- ---------- --------------- ---------
       355          4 L-Block         17BCE0902
       453          4 L-Block         17BEE0212
       568          6 K-Block         17BEE0378
       623          4 L-Block         17BEE0908
       316          4 L-Block         17BIT0102
       204          6 K-Block         17BIT0120
       554          4 L-Block         17BIT0298
       102          4 L-Block         17BME0156
       404          6 K-Block         17BME0897

9 rows selected.
```

Deleting the record of roll number 17BEE0908 from enrolls_for

```
SQL> delete from enrolls_for where roll_number='17BEE0908';

1 row deleted.

SQL> select * from enrolls_for;

ROLL_NUMB CRSE_CODE
--------- ----------
17BCE0902 ITE1001
17BEE0212 ITE1007
17BEE0378 ITE2003
17BIT0102 ITE1102
17BIT0120 ITE2001
17BIT0298 ITE1009
17BME0156 ITE2007
17BME0897 ITE1123
17BME2001 ITE1213

9 rows selected.
```

Deleting the record of room in which roll number 17BIT0102 lives

```
SQL> delete from room where roll_numb='17BIT0102';

1 row deleted.

SQL> select * from room;

     R_NUM R_CAPACITY HOST_NAME       ROLL_NUMB
---------- ---------- --------------- ---------
       355          4 L-Block         17BCE0902
       453          4 L-Block         17BEE0212
       568          6 K-Block         17BEE0378
       623          4 L-Block         17BEE0908
       204          6 K-Block         17BIT0120
       554          4 L-Block         17BIT0298
       102          4 L-Block         17BME0156
       404          6 K-Block         17BME0897

8 rows selected.
```

2. GREATER THAN(>)

Displaying all the 4-bed rooms

```
SQL> select r_num from room
  2  where r_capacity>4;

     R_NUM
----------
       568
       204
       404
```

3. LESS THAN (<)

Displaying all rooms having less than 6 beds

```
SQL> select r_num from room
  2  where r_capacity<6;

     R_NUM
----------
       355
       453
       623
       316
       554
       102
       512

7 rows selected.
```

4. NOT EQUAL (<>)

Displaying the record of rooms in hostels except K-Block

```
SQL> select * from room
  2  where host_name <> 'K-Block';

     R_NUM R_CAPACITY HOST_NAME       ROLL_NUMB
---------- ---------- --------------- ---------
       355          4 L-Block         17BCE0902
       453          4 L-Block         17BEE0212
       623          4 L-Block         17BEE0908
       316          4 L-Block         17BIT0102
       554          4 L-Block         17BIT0298
       102          4 L-Block         17BME0156
       512          4 L-Block         17BME2001

7 rows selected.
```

5. GREATER THAN OR EQUAL TO (>=)

Displaying the records of room with room number greater than or equal to 500

```
SQL> select * from room
  2  where r_num >= 500;

     R_NUM R_CAPACITY HOST_NAME       ROLL_NUMB
---------- ---------- --------------- ---------
       568          6 K-Block         17BEE0378
       623          4 L-Block         17BEE0908
       554          4 L-Block         17BIT0298
       512          4 L-Block         17BME2001
```

Aggregate functions

1. **SUM**

Calculating total credits

```
SQL> select sum(credits) from course;

SUM(CREDITS)
------------
          46
```

2. **MAX**

Displaying the maximum value for a credit

```
SQL> select max(credits) from course;

MAX(CREDITS)
------------
           4
```

3. **MIN**

Displaying the minimum value for a credit

```
SQL> select min(credits) from course;

MIN(CREDITS)
------------
           1
```

4. **COUNT**

Displaying total number of rooms

```
SQL> select count(r_num) from room;

COUNT(R_NUM)
------------
          16
```

5. **AVG**

Calculating the average room number

```
SQL> select avg(r_num) from room;

AVG(R_NUM)
----------
  421.8125
```

UNION

Displaying the list names of all students and wardens

```
SQL> select name from student
  2  union
  3  select w_name from warden;

NAME
----------------------------------------
ABC
ABCD
BCD
BCDE
DEF
DEFG
EFG
FGHI
GHI
HIJ
HIJK

NAME
----------------------------------------
JKL
JKLM
KLM
LMNO
MNO
NOP
NOPQ
PQR
PQRS
QRS
RSTU

NAME
----------------------------------------
STU
TUV
TUVW
VWX
VWXY
WEFG
XHIJ
XYZA
YZA
ZABC

32 rows selected.
```

MINUS

Displaying the list of department ids from course table excluding the department ids from department table where department name is CSE

```
SQL> select department_id from course
  2  minus
  3  select d_id from department where d_name='CSE';

DEPARTMENT
----------
SCOPE
SELECT
SITE
SMEC
```

INTERSECT

Displaying the department ids common between course and department table

```
SQL> select department_id from course
  2  intersect
  3  select d_id from department where d_name='CSE';

no rows selected

SQL> select department_id from course
  2  intersect
  3  select d_id from department;

DEPARTMENT
----------
SCOPE
SELECT
SITE
SMEC
```

GROUP BY AND HAVING

Displaying the number of courses having more than 3 Credits

```
SQL> select credits ,count(*) from course
  2  group by credits;

   CREDITS   COUNT(*)
---------- ----------
         1          2
         2          3
         4          4
         3          5

SQL> select credits ,count(*) from course
  2  group by credits
  3  having count(*) > 3
  4  ;

   CREDITS   COUNT(*)
---------- ----------
         4          4
         3          5
```

ORDER BY CLAUSE

Displaying the roll numbers from student table in increasing order

```
SQL> select d_id from department
  2  order by d_id;

D_ID
----------
SCOPE
SELECT
SITE
SMEC

SQL> select roll_num from student
  2  order by roll_num;

ROLL_NUM
---------
17BCE0902
17BEE0212
17BEE0378
17BEE0908
17BIT0102
17BIT0120
17BIT0298
17BME0156
17BME0897
17BME2001

10 rows selected.
```

JOINS

INNER JOIN

Displaying the names and roll numbers of students living in hostel room

```
SQL> select name,roll_num from student s join room r
  2  on s.roll_num=r.roll_numb
  3  ;

NAME                                     ROLL_NUM
---------------------------------------- ---------
HIJK                                     17BCE0902
PQRS                                     17BEE0212
TUVW                                     17BEE0378
XYZA                                     17BEE0908
FGHI                                     17BIT0102
JKLM                                     17BIT0120
NOPQ                                     17BIT0298
VWXY                                     17BME0156
ZABC                                     17BME0897
WEFG                                     17BME2001

10 rows selected.
```

Displaying the name, roll number, hostel name and room number of students who lives in hostel with names starting with 'L'

```
SQL> select name,roll_num,host_name,r_num from student s join room r
  2  on s.roll_num=r.roll_numb
  3  where host_name like 'L%';

NAME                                 ROLL_NUM  HOST_NAME            R_NUM
------------------------------------ --------- --------------- ----------
HIJK                                 17BCE0902 L-Block                355
PQRS                                 17BEE0212 L-Block                453
XYZA                                 17BEE0908 L-Block                623
FGHI                                 17BIT0102 L-Block                316
NOPQ                                 17BIT0298 L-Block                554
VWXY                                 17BME0156 L-Block                102
WEFG                                 17BME2001 L-Block                512

7 rows selected.
```

LEFT OUTER JOIN

Selecting all the course names and roll numbers of the students who applied for them

```
SQL> select c_name,roll_number from course c left outer join enrolls_for e
  2  on c.c_code=e.crse_code
  3  ;

C_NAME                               ROLL_NUMB
------------------------------------ ---------
Environmental Science                17BCE0902
Software Engineering                 17BEE0908
Mechanics
Engineering Physics                  17BEE0212
Digital Logic                        17BIT0298
Ethics and Values
Chemistry for Engineers              17BIT0102
Surveying                            17BME0897
Innovative Projects                  17BME2001
Computer Architecture                17BIT0120
Calculus for Engineers

C_NAME                               ROLL_NUMB
------------------------------------ ---------
Electrical Engineering               17BEE0378
Soft Skills
Statistics for Engineers             17BME0156

14 rows selected.
```

RIGHT OUTER JOIN

Displaying name, due status and department id for student living in hostel rooms

```
SQL> select name,no_dues_certificate,dept_id from student s right outer join room r
  2  on s.roll_num=r.roll_numb;

NAME                                N DEPT_ID
----------------------------------- - ----------
HIJK                                Y SITE
PQRS                                Y SCOPE
TUVW                                N SELECT
XYZA                                N SITE
FGHI                                Y SCOPE
JKLM                                Y SELECT
NOPQ                                N SITE
VWXY                                N SCOPE
ZABC                                Y SELECT
WEFG                                Y SITE

10 rows selected.
```

NESTED QUERIES

Displaying the course code for courses which are enrolled by students living in K-block

```
SQL> select crse_code from enrolls_for where roll_number in
  2  (select roll_num from student where hostel_name='K-Block');

CRSE_CODE
----------
ITE2003
ITE2001
ITE1123
```

Another approach for above query

```
SQL> select c_name ,department_id from course where c_code in(
  2  select crse_code from enrolls_for where roll_number in
  3  (select roll_num from student where hostel_name='K-Block')
  4  );

C_NAME                              DEPARTMENT
----------------------------------- ----------
Surveying                           SELECT
Computer Architecture               SELECT
Electrical Engineering              SELECT
```

Displaying the name of students who registered for a course

```
SQL> select name from student where roll_num in(select roll_number from course c left outer join enrolls_for e
  2  on c.c_code=e.crse_code);

NAME
-------------------------------------
XYZA
TUVW
JKLM
HIJK
PQRS
FGHI
VWXY
NOPQ
ZABC
WEFG

10 rows selected.
```

PL/SQL Function to authenticate a student through its user_id and password

```
SQL> create or replace function student_login(user_id student.email%type,password student.s_password%type)
  2  return number is
  3  s_id student.email%type;
  4  s_pass student.s_password%type;
  5  status number;
  6  cursor c_stud is
  7  select email,s_password from student;
  8  begin
  9  status:=0;
 10  open c_stud;
 11  loop
 12  fetch c_stud into s_id,s_pass;
 13  exit when c_stud%notfound;
 14  if s_id=user_id and s_pass=password then
 15  status:=status+1;
 16  end if;
 17  end loop;
 18  return status;
 19  end;
 20  /

Function created.
```

Output

```
SQL> declare
  2  user_id student.email%type:='ABCD@vitstudent.ac.in';
  3  password student.s_password%type:='S23@xpos';
  4  status varchar(10);
  5  begin
  6  status := student_login(user_id,password);
  7  if status=1
  8  then
  9  dbms_output.put_line('Valid User');
 10  else
 11  dbms_output.put_line('Invalid User');
 12  end if;
 13  end;
 14  /
Valid User

PL/SQL procedure successfully completed.

SQL> declare
  2  user_id student.email%type:='ABCD@vitstudent.ac.in';
  3  password student.s_password%type:='S23@jnos';
  4  status varchar(10);
  5  begin
  6  status := student_login(user_id,password);
  7  if status=1
  8  then
  9  dbms_output.put_line('Valid User');
 10  else
 11  dbms_output.put_line('Invalid User');
 12  end if;
 13  end;
 14  /
Invalid User

PL/SQL procedure successfully completed.
```

PL/SQL Procedure to list out the students in a hostel

```
SQL> create or replace procedure hostel_contains(h_name room.host_name%type)
  2  as
  3  r_roll room.roll_numb%type;
  4  r_host room.host_name%type;
  5  cursor c_host is
  6  select host_name,roll_numb from room;
  7  begin
  8  open c_host;
  9  loop
 10  fetch c_host into r_host,r_roll;
 11  exit when c_host%notfound;
 12  if r_host=h_name
 13  then
 14  dbms_output.put_line(r_roll);
 15  end if;
 16  end loop;
 17  close c_host;
 18  end;
 19  /

Procedure created.
```

Output

```
SQL> set serveroutput on;
SQL> declare
  2  r_host room.host_name%type;
  3  begin
  4  r_host:='L-Block';
  5  hostel_contains(r_host);
  6  end;
  7  /
17BCE0207
17BCE0902
17BEE0212
17BEE0908
17BIT0102
17BIT0298
17BME0156
17BME2001
```

A table Eligible_students to contain the students whose payment dues are clear

```
SQL> create table eligible_students(
  2  roll_num varchar(9) references student(roll_num),
  3  name varchar(40),
  4  email varchar(50),
  5  no_dues_certificate char(1),
  6  contact_num varchar(13),
  7  gender char(1),
  8  year int
  9  );

Table created.
```

A cursor to insert the data in the above table

```
SQL> declare
  2  s_roll student.roll_num%type;
  3  s_name student.name%type;
  4  s_email student.email%type;
  5  s_no_dues student.no_dues_certificate%type;
  6  s_contact student.contact_num%type;
  7  s_gender student.gender%type;
  8  s_year student.year%type;
  9  cursor c_eligibility is
 10  select roll_num,name,email,no_dues_certificate,contact_num,gender,year from student;
 11  begin
 12  open c_eligibility;
 13  loop
 14  fetch c_eligibility into s_roll,s_name,s_email,s_no_dues,s_contact,s_gender,s_year;
 15  exit when c_eligibility%notfound;
 16  if s_no_dues='Y'
 17  then
 18  insert into eligible_students
 19  values(s_roll,s_name,s_email,s_no_dues,s_contact,s_gender,s_year);
 20  end if;
 21  end loop;
 22  close c_eligibility;
 23  end;
 24  /

PL/SQL procedure successfully completed.
```

Business rules

1. **After the status of no dues certificate for a student is updated the details of the student should be inserted in eligible_students table Trigger for rule 1**

```
SQL> create trigger eligibility
  2  after update on student
  3  for each row
  4  declare
  5  sroll student.roll_num%type;
  6  sname student.name%type;
  7  semail student.email%type;
  8  sno_dues student.no_dues_certificate%type;
  9  scontact student.contact_num%type;
 10  sgender student.gender%type;
 11  syear student.year%type;
 12  begin
 13  sroll:=:old.roll_num;
 14  sname:=:old.name;
 15  semail:=:old.email;
 16  sno_dues:=:new.no_dues_certificate;
 17  scontact:=:old.contact_num;
 18  sgender:=:old.gender;
 19  syear:=:old.year;
 20  if :new.no_dues_certificate != :old.no_dues_certificate
 21  then
 22  insert into eligible_students
 23  values(sroll,sname,semail,sno_dues,scontact,sgender,syear);
 24  end if;
 25  end;
 26  /
```

Output

Before Trigger

```
SQL> select * from eligible_students;

ROLL_NUM  NAME
--------- ----------------------------------------
EMAIL                                              N CONTACT_NUM   G       YEAR
-------------------------------------------------- - ------------ - ----------
17BCE0207 ABCD
ABCD@vitstudent.ac.in                              Y 9467834376    M          1

17BCE0902 HIJK
HIJK@vitstudent.ac.in                              Y 9355678355    M          3

17BEE0212 PQRS
PQRS@vitstudent.ac.in                              Y 9453678453    M          1


ROLL_NUM  NAME
--------- ----------------------------------------
EMAIL                                              N CONTACT_NUM   G       YEAR
-------------------------------------------------- - ------------ - ----------
17BIT0102 FGHI
FGHI@vitstudent.ac.in                              Y 9316678316    M          1

17BIT0120 JKLM
JKLM@vitstudent.ac.in                              Y 9204678204    F          2

17BIT0298 NOPQ
NOPQ@vitstudent.ac.in                              Y 9554678554    M          3


ROLL_NUM  NAME
--------- ----------------------------------------
EMAIL                                              N CONTACT_NUM   G       YEAR
-------------------------------------------------- - ------------ - ----------
17BME0897 ZABC
ZABC@vitstudent.ac.in                              Y 9404678404    F          2

17BME2001 WEFG
WEFG@vitstudent.ac.in                              Y 9512678512    M          3

17BCE0920 LMNO
LMNO@vitstudent.ac.in                              Y 9726678726    F          4


ROLL_NUM  NAME
--------- ----------------------------------------
EMAIL                                              N CONTACT_NUM   G       YEAR
-------------------------------------------------- - ------------ - ----------
17BEE2002 BCDE
BCDE@vitstudent.ac.in                              Y 9123678123    F          4

17BIT0378 RSTU
RSTU@vitstudent.ac.in                              Y 9759678759    F          4

17BME2019 XHIJ
XHIJ@vitstudent.ac.in                              Y 9702678702    F          4


12 rows selected.
```

After trigger

```
SQL> update student
  2  set no_dues_certificate='Y'
  3  where roll_num='17BME0156';

1 row updated.
```

```
SQL> select * from eligible_students;

ROLL_NUM  NAME
--------- ----------------------------------------
EMAIL                                              N CONTACT_NUM   G       YEAR
-------------------------------------------------- - ------------ - ----------
17BCE0207 ABCD
ABCD@vitstudent.ac.in                              Y 9467834376   M          1

17BCE0902 HIJK
HIJK@vitstudent.ac.in                              Y 9355678355   M          3

17BEE0212 PQRS
PQRS@vitstudent.ac.in                              Y 9453678453   M          1


ROLL_NUM  NAME
--------- ----------------------------------------
EMAIL                                              N CONTACT_NUM   G       YEAR
-------------------------------------------------- - ------------ - ----------
17BIT0102 FGHI
FGHI@vitstudent.ac.in                              Y 9316678316   M          1

17BIT0120 JKLM
JKLM@vitstudent.ac.in                              Y 9204678204   F          2

17BIT0298 NOPQ
NOPQ@vitstudent.ac.in                              Y 9554678554   M          3


ROLL_NUM  NAME
--------- ----------------------------------------
EMAIL                                              N CONTACT_NUM   G       YEAR
-------------------------------------------------- - ------------ - ----------
17BME0897 ZABC
ZABC@vitstudent.ac.in                              Y 9404678404   F          2

17BME2001 WEFG
WEFG@vitstudent.ac.in                              Y 9512678512   M          3

17BCE0920 LMNO
LMNO@vitstudent.ac.in                              Y 9726678726   F          4


ROLL_NUM  NAME
--------- ----------------------------------------
EMAIL                                              N CONTACT_NUM   G       YEAR
-------------------------------------------------- - ------------ - ----------
17BEE2002 BCDE
BCDE@vitstudent.ac.in                              Y 9123678123   F          4

17BIT0378 RSTU
RSTU@vitstudent.ac.in                              Y 9759678759   F          4

17BME2019 XHIJ
XHIJ@vitstudent.ac.in                              Y 9702678702   F          4


ROLL_NUM  NAME
--------- ----------------------------------------
EMAIL                                              N CONTACT_NUM   G       YEAR
-------------------------------------------------- - ------------ - ----------
17BME0156 VWXY
VWXY@vitstudent.ac.in                              Y 9102678102   M          1


13 rows selected.
```

2. **When the room of a student is updated then hostel details in the student table should be updated accordingly, when course id of a student is updated the enrols for table should be updated accordingly**

and when contact number of a student is changed the old and new contact numbers should be displayed.

Trigger for rule 2

```
SQL> create trigger stud_change
  2  after update on student
  3  for each row
  4  begin
  5  if :new.hostel_name!=:old.hostel_name then
  6  update room
  7  set host_name=:new.hostel_name
  8  where roll_numb=:new.roll_num;
  9  end if;
 10  if :new.course_code!=:old.course_code then
 11  update enrolls_for
 12  set crse_code=:new.course_code
 13  where roll_number=:new.roll_num;
 14  end if;
 15  if :new.contact_num!=:old.contact_num then
 16  dbms_output.put_line('Old contact number '||:old.contact_num);
 17  dbms_output.put_line('New contact number '||:new.contact_num);
 18  end if;
 19  end;
 20  /

Trigger created.
```

Output

Before Trigger

```
SQL> select roll_num,hostel_name from student;

ROLL_NUM  HOSTEL_NAME
--------- ---------------
17BCE0207 L-Block
17BCE0386 K-Block
17BCE0902 L-Block
17BEE0212 L-Block
17BEE0378 K-Block
17BEE0908 L-Block
17BIT0102 L-Block
17BIT0120 K-Block
17BIT0298 L-Block
17BME0156 L-Block
17BME0897 K-Block

ROLL_NUM  HOSTEL_NAME
--------- ---------------
17BME2001 L-Block
17BCE0920 K-Block
17BEE2002 K-Block
17BIT0378 K-Block
17BME2019 K-Block

16 rows selected.

SQL> select roll_numb,host_name from room;

ROLL_NUMB HOST_NAME
--------- ---------------
17BCE0207 L-Block
17BCE0386 K-Block
17BCE0902 L-Block
17BCE0920 K-Block
17BEE0212 L-Block
17BEE0378 K-Block
17BEE0908 L-Block
17BEE2002 K-Block
17BIT0102 L-Block
17BIT0120 K-Block
17BIT0298 L-Block

ROLL_NUMB HOST_NAME
--------- ---------------
17BIT0378 K-Block
17BME0156 L-Block
17BME0897 K-Block
17BME2001 L-Block
17BME2019 K-Block

16 rows selected.
```

After Trigger

```
SQL> update room
  2  set host_name='L-Block'
  3  where roll_numb='17BME2019';

1 row updated.
```

```
SQL> select roll_numb,host_name from room;

ROLL_NUMB HOST_NAME
--------- ---------------
17BCE0207 L-Block
17BCE0386 K-Block
17BCE0902 L-Block
17BCE0920 K-Block
17BEE0212 L-Block
17BEE0378 K-Block
17BEE0908 L-Block
17BEE2002 K-Block
17BIT0102 L-Block
17BIT0120 K-Block
17BIT0298 L-Block

ROLL_NUMB HOST_NAME
--------- ---------------
17BIT0378 K-Block
17BME0156 L-Block
17BME0897 K-Block
17BME2001 L-Block
17BME2019 L-Block

16 rows selected.

SQL> select roll_num,hostel_name from student;

ROLL_NUM  HOSTEL_NAME
--------- ---------------
17BCE0207 L-Block
17BCE0386 K-Block
17BCE0902 L-Block
17BEE0212 L-Block
17BEE0378 K-Block
17BEE0908 L-Block
17BIT0102 L-Block
17BIT0120 K-Block
17BIT0298 L-Block
17BME0156 L-Block
17BME0897 K-Block

ROLL_NUM  HOSTEL_NAME
--------- ---------------
17BME2001 L-Block
17BCE0920 K-Block
17BEE2002 K-Block
17BIT0378 K-Block
17BME2019 L-Block

16 rows selected.
```

```
SQL> update student
  2  set contact_num='9467834376'
  3  where roll_num='17BCE0207';
Old contact number 9567834376
New contact number 9467834376

1 row updated.
```