

IT252 DBMS FINAL PROJECT REPORT

"Hostel Management System"

Team Members:

1)Rahul Maheshwari - 191CH039

2)Viren Varma - 191ME293

3)Shreyas Rao - 191EE224

1)PROBLEM STATEMENT:

Hostel is the Place where Students Can Live for a Short Period of Time For their Studies and have No Risk from Outsiders.For the Past Few Years The Number of educational institutions are increasing Rapidly.Thereby the Number of Hostels are also increasing for the accommodation of the Students studying in that Particular institution,& hence there is a lot of Strain on the Person who are running the hostel & software's are Not Usually used in this context. This Particular project deals with the problems on managing a hostel and avoids the Problems which occur when carried it Manually.

“Hostel Management System” is the database where we will Be having Each and every single Details of Student,Staff:Cleaners, Warden,Hostel Rooms,Fee Records,Accommodation related stuff Etc.

Hostel Management will be taking details of Students such as their **Name,Email,Phone Number,Date of Birth,Age,Nationality, Address (comprises Temporary and Permanent Address)** and will Provide them a **Unique Student ID** which will be retained till they don't leave the Hostel permanently.

Once the Student Details are entered in The database and the Hostel Fees is Paid By Student a Fee Receipt will be Generated which will be having **Amount paid,Account Number,Date of Payment** and **Receipt Code** for every Student.This Receipt Printout should be With students For Future Reference and as a Proof.The Receipt will be having a Signature of Warden.

Fee Receipt is the Only Proof By Which Student can Enter into Hostel for the first time and will Accommodate to Some Room.

Hostel Management will Be Keeping Wardens for each Hostel and keep their Details Like **Name, Email,Phone,Age,Date of birth, City with**

Pincode,Address(comprises Temporary and Permanent)and will give them a **Unique Staff ID**. If Students Face Any Issue Regarding the facilities of Hostel,Ragging Issues etc. then they can Contact their Hostel Warden Immediately.

And all The Responsibilities of Hostel will be Managed by The Wardens.

For a Particular Hostel,Hostel Management will have a Record of Hostel Name,Number of Rooms & Number of WashRooms in That Hostel and **Unique Hostel Number**.Once the Hostel Fees is Paid,Student will be accommodated to his/her Hostel Rooms. Hostel Management will Keep Track of **Date of Allotment** of each and every student while Accommodation.

Hostel Should be Clean 24x7 for which Hostel Management will be Assigning Cleaners and will have their Details Like Name, Email, Phone,Age,Date of Birth,City with PinCode and will Give them a **Unique Staff ID**.Keeping Hostel Hygienic will be the Duty of Cleaners.

All Wardens of That Particular Hostel & Cleaners will Come Under One Category Called as Staff Members where Hostel Management will be having Their Age,Address,Date of Birth,City with Pincode and **Unique Staff ID**.

If At all Hostel Management Require Some Details of any Member then They can Directly Go To Staff Members and can Search That Particular Person very Efficiently and in a Much Quicker Way.

However All This Information is Not maintained at a Single Location because Different Sector Maintains Their Own Information System.Therefore,Distributed Scenario comes into Picture that allows control & Data Flow between Multiple sites to share Information.Based on this three sites are identified.(i)Fees Department will keep track of

Fee Payment,(ii)Staff Management will Keep track of Members like Wardens,Cleaners & (iii)Admin Section will keep Track of Students,Student Complaints,their Suggestions,their Room Numbers,their Login IDs and Passwords etc.

2)ACTORS:People Who will Interact with the Database

i)Students

ii)Wardens

iii)Admin Section

3)Some Sample Queries:

➤ A Staff Management Can

- ★ Get Email IDs of all Cleaners and Wardens.

- ★ Get a List of Cleaners for a Particular Hostel.

➤ A Admin Section Can

- ★ Get a List of Complaints/Suggestions From Students.

- ★ Get a Track of Login IDs and Passwords

- ★ Get a Floor on which Students are Staying and their Room Numbers.

- ★ Get a Survey of Hostel.

- ★ Get The Number of WashRooms and Rooms on a Particular Floor.

➤ A Fees Department Can

- ★ Verify the Transaction of Each Student.

- ★ Track the Accommodation Date of Students.

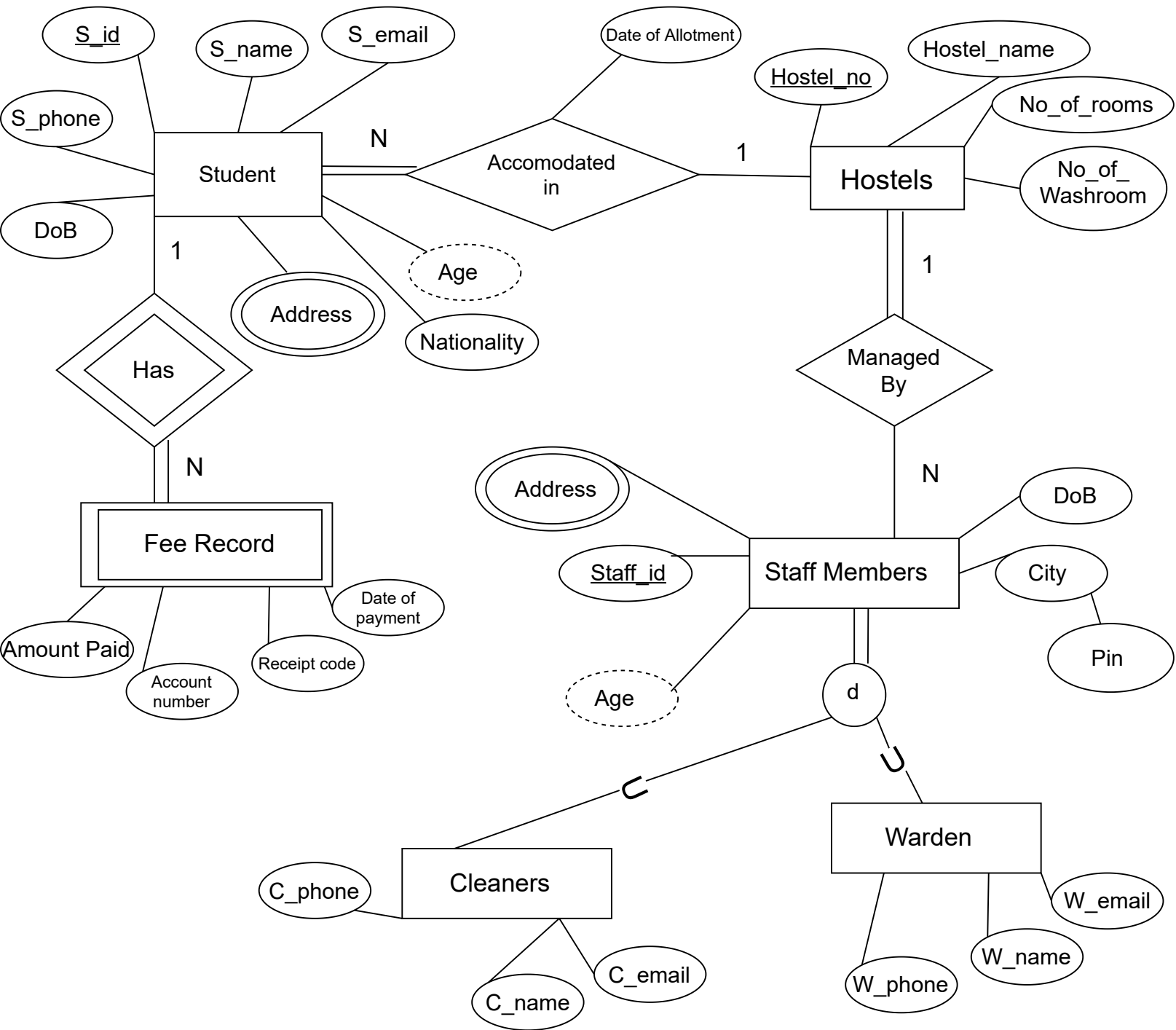
- ★ Get Details of Account Number.

➤ A Student Can

- ★ Request Fee Details.

- ★ Request Contacts of Staff Members.

Hostel Management System EER Diagram



Relational Data Model

STUDENT

S-ID	S-Name	S-Phone	S-DOB	S-DOA	S-Email	S-Address	Nationality	S-Age	S-Hostel
------	--------	---------	-------	-------	---------	-----------	-------------	-------	----------

HOSTEL

Hostel-NO	Hostel-Name	Rooms	WashRooms	Warden
-----------	-------------	-------	-----------	--------

STAFF

Staff-ID	Staff-Address	Staff-Age	Staff-DOB	City	PIN
----------	---------------	-----------	-----------	------	-----

CLEANER

C-Staff-ID	C-Phone	C-Name	C-Email
------------	---------	--------	---------

WARDEN

W-Staff-ID	W-Name	W-Phone	W-Email
------------	--------	---------	---------

FEE RECORD

Amount Paid	AccNo	ReceiptCode	DOP	S-ID
-------------	-------	-------------	-----	------

STUDENT ADDRESS

S-ID	S-Address
------	-----------

STAFF ADDRESS

Staff-ID	Staff-Address
----------	---------------

BASIC DEFINITIONS

1NF: First normal form is a property of a relation in a relational database. A relation is in first normal form if and only if the domain of each attribute contains only atomic values, and the value of each attribute contains only a single value from that domain.

2NF: A relation is in the second normal form if it fulfills the following two requirements: It is in first normal form. It does not have any non-prime attribute that is functionally dependent on any proper subset of any candidate key of the relation (partial dependency)

3NF: A relation is in third normal form, if there is no transitive dependency for non-prime attributes as well as it is in second normal form.

Partial Dependencies are when one of the primary keys determines another attribute or attributes or we can say Proper Subset of Candidate Key determines Non-Prime Attributes

Transitive Dependencies are when a NonPrime attribute determines another NonPrime attribute.

NORMALIZATION

1)Student:

S_ID,S_Name,S_Phone,S_DOB,S_Email,S_Address,Nationality,S_Age,
S_HostelNo,S_DOA

FD:

- a)S_ID->S_Name,S_DOB,S_EMail,S_Address,S_Age,S_Phone,Nationality
S_HostelNo,S_DOA
- b)S_Address->Nationality
- c)S_DOB->S_Age

Normalization:

Here we can see S_ID is Determining each and every Attribute So it is Candidate Key and Prime Attribute as well and All other attributes are Non-Prime Attributes as No other Candidate Key is there.

So Here There are two Transitive Dependencies S_Address->Nationality and S_DOB->S_Age. So this Table is **Not in 3NF**.

To make this Table Normalized in 3NF we can split in this Form:

Student(S_ID,S_Name,S_Phone,S_DOB,S_Email,S_Address,S_HostelNo,
S_DOA)

StudentAddress(S_Address,Nationality)

StudentDOB(S_DOB,S_Age)

S_Address and S_DOB are Foreign Key in Student

So Now **All three tables are in 3NF**.

2)Hostel:

Hostel_No,Hostel_Name,Rooms,WashRooms,Warden

FD:

a)Hostel_No->Hostel_Name,Rooms,WashRooms,Warden

b)Hostel_Name->Hostel_No,Rooms,WashRooms,Warden

c)Rooms->WashRooms

Normalization:

Here we can see Hostel_No and Hostel_Name both can determine all attributes hence both are Candidate key and Prime attribute as well. Rest of the Attributes are Non_Prime attributes Hence we can see one Transitive Dependency Rooms->WashRooms as Both are Non-Prime Attributes. So this Table is **Not in 3NF**.

To make this Table Normalized in 3NF we can split in this Form:

Hostel(Hostel_No,Warden,Hostel_Name,Rooms)

Rooms(Rooms,WashRooms)

Rooms is Foreign Key in Hostel.

So Now **Both tables are in 3NF**.

3)Staff:

Staff_ID,Staff_Address,Staff_Age,Staff_DOB,City,PIN

FD:

a)Staff_ID->Staff_Address,Staff_Age,Staff_DOB,City,PIN

b)City->PIN

Normalization:

Here Staff_ID is Super Key as it Alone determines each and every Attribute and it is also the Prime attribute and other than this all are Non-Prime Attributes.

So, Here we can see there is Transitive Dependency as City and PIN both are Non-Prime Attributes and City \rightarrow PIN. So this Table is **Not in 3NF**.

To make this Table Normalized in 3NF we can split in this Form:

Staff(Staff_ID, Staff_Address, Staff_Age, Staff_DOB, PIN)
PinCode(City, PIN)

PIN is Foreign Key in Staff.

So Now **Both tables are in 3NF**.

4) Cleaners:

C_Staff_ID, C_Name, C_Phone, C_Email

FD:

a) C_Staff_ID \rightarrow C_Name, C_Phone, C_Email

Normalization:

BCNF, because the Left attribute is SuperKey and there is No Partial Dependency and Transitive Dependency.

Cleaners(C_Staff_ID, C_Name, C_Phone, C_Email)

5)Warden:

W_Staff_ID,W_Name,W_Phone,W_Email

FD:

a)W_Staff_ID->W_Name,W_Phone,W_Email

Normalization:

BCNF, because the Left attribute is SuperKey and there is No Partial Dependency and Transitive Dependency.

Warden(W_Staff_ID,W_Name,W_Phone,W_Email)

6)Fee Record:

Amount_Paid,AccNo,ReceiptCode,DOP,S_ID

FD:

a)S_ID->Amount_Paid,ReceiptCode,DOP,AccNo

b)ReceiptCode->DOP,Amount_Paid,S_ID,AccNo

Normalization:

BCNF,because Prime attributes are S_ID,ReceiptCode and AccNo as they are Part of Candidate Keys (S_ID is Super Key so it is Candidate Key and ReceiptCode determine all Attributes so it is also Candidate Key) and DOP and Amount_Paid and Acc No are Non Prime Attributes so as S_ID,ReceiptCode Both are Super Key means Left side is Super Key concludes that this table is in **BCNF**.

StudentFees(S_ID,ReceiptCode,Amount_Paid,AccNo,DOP)

7)Student Address:

S_ID,S_Address

FD:

a)S_ID->S_Address

Normalization:

BCNF, because the Left attribute is SuperKey and there is No Partial Dependency and Transitive Dependency.

StudentAdd(S_ID,S_Address)

8)Staff Address:

Staff_ID,Staff_Address

FD:

a)Staff_ID->Staff_Address

Normalization:

BCNF, because the Left attribute is SuperKey and there is No Partial Dependency and Transitive Dependency.

StaffAdd(Staff_ID,Staff_Address)

Schema

&

Values

```
mysql> show Tables;
```

Tables_in_Hostel_Management_System	
Cleaners	
Hostel	
PinCode	
Rooms	
Staff	
StaffAdd	
Student	
StudentAdd	
StudentAddress	
StudentDOB	
StudentFees	
Warden	

```
12 rows in set (0.00 sec)
```

```
mysql> desc Staff;
```

Field	Type	Null	Key	Default	Extra
Staff_ID	int(6)	NO	PRI	NULL	
Staff_Address	varchar(30)	YES		NULL	
Staff_Age	int(3)	YES		NULL	
Staff_DOB	date	YES		NULL	
PIN	int(6)	YES	MUL	NULL	

```
5 rows in set (0.00 sec)
```

```
mysql> select * from Staff;
```

Staff_ID	Staff_Address	Staff_Age	Staff_DOB	PIN
190001	Address 1	41	1990-01-01	4007030
190002	Address 2	43	1988-01-01	4007030
190003	Address 3	45	1986-01-01	4007031
190004	Address 4	40	1991-01-01	4008510
190005	Address 5	38	1993-01-01	4008510
190006	Address 6	38	1993-01-01	4008510
190007	Address 7	46	1985-01-01	4007031
190008	Address 8	46	1985-01-01	4007031
190009	Address 9	41	1990-01-01	4007030
190010	Address 10	41	1990-01-01	4007030
190011	Address 11	42	1989-01-01	4007030
190012	Address 12	51	1980-01-01	4008131

```
12 rows in set (0.00 sec)
```

```
mysql> desc Cleaners;
```

```

+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| C_Staff_ID | int(6)        | YES  | MUL | NULL     |       |
| C_Name      | varchar(30)   | YES  |     | NULL     |       |
| C_Phone     | bigint(13)    | YES  |     | NULL     |       |
| C_Email     | varchar(30)   | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```

mysql> select * from Cleaners;

```

+-----+-----+-----+-----+
| C_Staff_ID | C_Name      | C_Phone      | C_Email          |
+-----+-----+-----+-----+
| 190005     | Paulim      | 9895661234   | paulim@gmail.com |
| 190006     | Paul        | 9891261234   | paul@gmail.com   |
| 190007     | Bhavesh     | 9891223434   | bhavesh@gmail.com |
| 190008     | Avesh       | 9845223434   | avesh@gmail.com  |
| 190009     | Vesh        | 9841233434   | vesh@gmail.com   |
| 190010     | Josh        | 9856233434   | josh@gmail.com   |
| 190011     | Jesh        | 9851233434   | jesh@gmail.com   |
| 190012     | Jehan       | 9851003434   | jehan@gmail.com  |
+-----+-----+-----+-----+
8 rows in set (0.00 sec)

```

mysql> Desc Warden;

```

+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| W_Staff    | int(6)        | YES  | MUL | NULL     |       |
| W_Name     | varchar(30)   | YES  |     | NULL     |       |
| W_Phone    | bigint(13)    | YES  |     | NULL     |       |
| W_Email    | varchar(30)   | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```

mysql> select * from Warden;

```

+-----+-----+-----+-----+
| W_Staff | W_Name      | W_Phone      | W_Email          |
+-----+-----+-----+-----+
| 190001  | Dhruv       | 9869452323   | dhruv@gmail.com  |
| 190002  | Mandeep     | 9869346723   | mandeep@gmail.com |
| 190003  | Sandeep     | 9869341234   | Sandeep@gmail.com |
| 190004  | Rahul       | 9898761234   | Rahul@gmail.com   |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```

```
mysql> desc StaffAdd;
```

Field	Type	Null	Key	Default	Extra
Staff_ID	int(6)	YES	MUL	NULL	
Staff_Address	varchar(50)	YES		NULL	

2 rows in set (0.00 sec)

```
mysql> select * from StaffAdd;
```

Staff_ID	Staff_Address
190001	NULL
190002	NULL
190003	Alternate Address 3
190004	Alternate Address 4
190005	NULL
190006	NULL
190007	NULL
190008	NULL
190009	Alternate Address 9
190010	Alternate Address 10
190011	Alternate Address 11
190012	NULL

12 rows in set (0.00 sec)

```
mysql> desc PinCode;
```

Field	Type	Null	Key	Default	Extra
City	varchar(30)	YES		NULL	
PIN	int(6)	NO	PRI	NULL	

2 rows in set (0.00 sec)

```
mysql> select * from PinCode;
```

City	PIN
Mumbai	4007030
Pune	4007031
Lucknow	4008131
Bangalore	4008510

4 rows in set (0.00 sec)

```
mysql> desc Rooms;
```

Field	Type	Null	Key	Default	Extra
-------	------	------	-----	---------	-------

Field	Type	Null	Key	Default	Extra
Rooms	int(3)	NO	PRI	NULL	
WashRooms	int(3)	YES		NULL	

2 rows in set (0.00 sec)

```
mysql> select * from Rooms;
```

Rooms	WashRooms
30	6
40	8
50	10
60	12

4 rows in set (0.00 sec)

```
mysql> desc Hostel;
```

Field	Type	Null	Key	Default	Extra
Hostel_No	int(2)	NO	PRI	NULL	
Hostel_name	varchar(30)	YES		NULL	
Rooms	int(3)	YES	MUL	NULL	
Warden	int(6)	YES	MUL	NULL	

4 rows in set (0.01 sec)

```
mysql> select * from Hostel;
```

Hostel_No	Hostel_name	Rooms	Warden
10	Hostel A	30	190001
20	Hostel B	40	190002
30	Hostel C	50	190003
40	Hostel D	60	190004

4 rows in set (0.00 sec)

```
mysql> desc Student;
```

Field	Type	Null	Key	Default	Extra
S_ID	int(6)	NO	PRI	NULL	
S_Name	varchar(30)	YES		NULL	
S_DOB	date	YES	MUL	NULL	
S_Phone	bigint(13)	YES		NULL	
S_Email	varchar(30)	YES		NULL	

S_Address	varchar(50)	YES	MUL	NULL		
S_HostelNo	int(2)	YES	MUL	NULL		
S_DOA	date	YES		NULL		

8 rows in set (0.00 sec)

```
mysql> select * from Student;
```

S_ID	S_Name	S_DOB	S_Phone	S_Email	S_Address	S_HostelNo	S_DOA
201001	Ram	2001-01-01	9869411234	ram@gmail.com	1	10	2020-01-01
201002	Shyaam	2001-02-01	9869121234	shyam@gmail.com	2	10	2020-01-02
201003	Ramu	2001-03-01	9869821234	ramu@gmail.com	3	10	2020-01-03
201004	Shyamu	2001-04-01	9123821234	shyamu@gmail.com	4	10	2020-01-04
201005	Dhruvi	2001-05-01	9124567234	dhruvi@gmail.com	5	20	2020-01-04
201006	Dhruvill	2001-06-01	9124567321	dhruvill@gmail.com	6	20	2020-01-04
201007	Roy	2001-07-01	9124234321	roy@gmail.com	7	30	2020-01-05
201008	Rohit	2001-08-01	9124123321	rohit@gmail.com	8	30	2020-01-06
201009	Rohi	2001-09-01	9345123321	rohi@gmail.com	9	30	2020-01-07
201010	Devdutt	2001-10-01	9345456321	devdutt@gmail.com	10	40	2020-01-08
201011	Devdut	2001-11-01	9312456321	devdut@gmail.com	11	40	2020-01-08
201012	Dev	2001-12-01	9312006321	dev@gmail.com	12	40	2020-01-09

12 rows in set (0.00 sec)

```
mysql> desc StudentAdd;
```

Field	Type	Null	Key	Default	Extra
S_ID	int(6)	YES	MUL	NULL	
S_Address	varchar(50)	YES		NULL	

2 rows in set (0.00 sec)

```
mysql> select * from StudentAdd;
```

-----+

S_ID	S_Address
201001	NULL
201002	NULL
201003	NULL
201004	NULL
201005	NULL
201006	NULL
201007	NULL
201008	NULL
201009	NULL
201010	NULL
201011	NULL
201012	NULL

12 rows in set (0.00 sec)

```
mysql> desc StudentFees;
```

Field	Type	Null	Key	Default	Extra
Amount_Paid	int(10)	YES		NULL	
AccNo	bigint(13)	YES		NULL	
ReceiptCode	bigint(13)	NO	PRI	NULL	
DOP	date	YES		NULL	
S_ID	int(6)	NO	PRI	NULL	

5 rows in set (0.00 sec)

```
mysql> select * from StudentFees;
```

Amount_Paid	AccNo	ReceiptCode	DOP	S_ID
120000	100450010001	2003450001	2019-12-01	201001
120000	100450010002	2003450002	2019-12-02	201002
120000	100450010003	2003450003	2019-12-03	201003
70000	100450010004	2003450004	2019-12-04	201004
120000	100450010005	2003450005	2019-12-05	201005
120000	100450010006	2003450006	2019-12-06	201006
120000	100450010007	2003450007	2019-12-07	201007
120000	100450010008	2003450008	2019-12-08	201008
24000	100450010009	2003450009	2019-12-09	201009
200000	100450010010	2003450010	2019-12-10	201010
120000	100450010011	2003450011	2019-12-11	201011
140000	100450010012	2003450012	2019-12-12	201012

12 rows in set (0.00 sec)

```
mysql> desc StudentAddress;
```

```

+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| S_Address  | varchar(50) | NO   | PRI | NULL    |       |
| Nationality | varchar(30) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

```

```
mysql> select * from StudentAddress;
```

```

+-----+-----+
| S_Address | Nationality |
+-----+-----+
| S_Address 1 | Indian      |
| S_Address 10 | Indian      |
| S_Address 11 | Indian      |
| S_Address 12 | Indian      |
| S_Address 2 | Indian      |
| S_Address 3 | Indian      |
| S_Address 4 | Indian      |
| S_Address 5 | British     |
| S_Address 6 | British     |
| S_Address 7 | Nepali      |
| S_Address 8 | Indian      |
| S_Address 9 | Indian      |
+-----+-----+
12 rows in set (0.00 sec)

```

```
mysql> desc StudentDOB;
```

```

+-----+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| S_DOB | date  | NO   | PRI | NULL    |       |
| S_Age | int(3) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

```

```
mysql> select * from StudentDOB;
```

```

+-----+-----+
| S_DOB | S_Age |
+-----+-----+
| 2001-01-01 | 20 |
| 2001-02-01 | 20 |
| 2001-03-01 | 20 |
| 2001-04-01 | 19 |
| 2001-05-01 | 19 |
| 2001-06-01 | 19 |
| 2001-07-01 | 19 |
| 2001-08-01 | 19 |
| 2001-09-01 | 19 |
| 2001-10-01 | 19 |
| 2001-11-01 | 19 |

```

2001-12-01	19
+-----+-----+	

12 rows in set (0.00 sec)

Simple Queries

1) Counting staff members below the age of 40
select count(Staff_Age) from Staff where Staff_Age>=40

```
mysql> select count(Staff_Age) from Staff where Staff_Age>=40;
+-----+
| count(Staff_Age) |
+-----+
|                10 |
+-----+
1 row in set (0.00 sec)
```

2) Selecting Cleaner with phone numbers 9895661234 and 9891261234
select C_Name from Cleaners where C_Phone = 9895661234 or C_Phone = 9891261234

```
mysql> select C_Name from Cleaners where C_Phone = 9895661234 or C_Phone = 9891261234;
+-----+
| C_Name |
+-----+
| Paulim |
| Paul   |
+-----+
2 rows in set (0.04 sec)
```

3) Selecting information about a warden named Sandeep
select * from Warden where W_Name = 'Sandeep'

```
mysql> select * from Warden where W_Name = 'Sandeep';
+-----+-----+-----+-----+
| W_Staff | W_Name | W_Phone | W_Email |
+-----+-----+-----+-----+
| 190003 | Sandeep | 9869341234 | Sandeep@gmail.com |
+-----+-----+-----+-----+
1 row in set (0.05 sec)
```

4) Finding which city the pincode 4008510 comes under
select City from PinCode where PIN = 4008510

```
mysql> select City from PinCode where PIN = 4008510;
+-----+
| City |
+-----+
| Bangalore |
+-----+
1 row in set (0.00 sec)
```

5) Finding the warden of a hostel
select W_Name from Hostel left join Warden on
Hostel.Warden=Warden.W_Staff where Hostel.Hostel_Name = 'Hostel A'

```
mysql> select W_Name from Hostel left join Warden on
Hostel.Warden=Warden.W_Staff where Hostel.Hostel_Name = 'Hostel A';
+-----+
| W_Name |
+-----+
```

```
| Dhruv |
+-----+
1 row in set (0.00 sec)
```

6) Finding date of accomodation of a student

```
select S_DOA from Student where Student.S_Name = 'Devdutt'
```

```
mysql> select S_DOA from Student where Student.S_Name = 'Devdutt';
+-----+
| S_DOA |
+-----+
| 2020-01-08 |
+-----+
1 row in set (0.00 sec)
```

7) Selecting students who paid fees after 5th December 2019

```
select S_ID from StudentFees where DOP>'2019-12-05'
```

```
mysql> select S_ID from StudentFees where DOP>'2019-12-05';
+-----+
| S_ID |
+-----+
| 201006 |
| 201007 |
| 201008 |
| 201009 |
| 201010 |
| 201011 |
| 201012 |
+-----+
7 rows in set (0.00 sec)
```

8) Selecting international addresses

```
select * from StudentAddress where NOT(Nationality='Indian')
```

```
mysql> select * from StudentAddress where NOT(Nationality='Indian');
+-----+-----+
| S_Address | Nationality |
+-----+-----+
| S_Address 5 | British |
| S_Address 6 | British |
| S_Address 7 | Nepali |
+-----+-----+
3 rows in set (0.00 sec)
```

9) Selecting Staff whose address is unknown

```
select Staff_ID from StaffAdd where Staff_Address = NULL
```

```
mysql> select Staff_ID from StaffAdd where Staff_Address = NULL;
Empty set (0.05 sec)
```

10) Counting the number of students whose addresses are unknown

```
select count(S_ID) from StudentAdd where S_Address = NULL
```

```
mysql> select count(S_ID) from StudentAdd where S_Address = NULL;
```



```
+-----+
| count(S_ID) |
+-----+
|           0 |
+-----+
1 row in set (0.05 sec)
```

Activities Terminal Tue 08:31 viren@viren-VirtualBox: ~

File Edit View Search Terminal Help

mysql> show tables;

+-----+
| Tables_in_Hostel_Management_System |
+-----+
| Cleaners
| Hostel
| PinCode
| Rooms
| Staff
| StaffAdd
| Student
| StudentAdd
| StudentAddress
| StudentDOB
| StudentFees
| Warden
+-----+

12 rows in set (0.00 sec)

mysql> select count(Staff_Age) from Staff where Staff_Age>=40;

+-----+
| count(Staff_Age) |
+-----+
| 10 |
+-----+

1 row in set (0.00 sec)

mysql> select C_Name from Cleaners where C_Phone = 9895661234 or C_Phone = 9891261234;

+-----+
| C_Name |
+-----+
| Paulim |
| Paul |
+-----+

2 rows in set (0.04 sec)

Activities Terminal Tue 08:31 viren@viren-VirtualBox: ~

File Edit View Search Terminal Help

```
mysql> select C_Name from Cleaners where C_Phone = 9895661234 or C_Phone = 9891261234;
+-----+
| C_Name |
+-----+
| Paulim |
| Paul   |
+-----+
2 rows in set (0.04 sec)

mysql> select * from Warden where W_Name = 'Sandeep';
+-----+-----+-----+-----+
| W_Staff | W_Name | W_Phone | W_Email |
+-----+-----+-----+-----+
| 190003 | Sandeep | 9869341234 | Sandeep@gmail.com |
+-----+-----+-----+-----+
1 row in set (0.05 sec)

mysql> select City from PinCode where PIN = 4008510;
+-----+
| City |
+-----+
| Bangalore |
+-----+
1 row in set (0.00 sec)

mysql> select W_Name from Hostel left join Warden on Hostel.Warden=Warden.W_Staff where Hostel.Hostel_Name = 'Hostel A';
+-----+
| W_Name |
+-----+
| Dhruv |
+-----+
1 row in set (0.00 sec)

mysql> select S_DOA from Student where Student.S_Name = 'Devdutt';
```

Activities Terminal Tue 08:31 viren@viren-VirtualBox: ~

File Edit View Search Terminal Help

```
mysql> select W_Name from Hostel left join Warden on Hostel.Warden=Warden.W_Staff where Hostel.Hostel_Name = 'Hostel A';
```

W_Name
Dhruv

1 row in set (0.00 sec)

```
mysql> select S_DOA from Student where Student.S_Name = 'Devdutt';
```

S_DOA
2020-01-08

1 row in set (0.00 sec)

```
mysql> select S_ID from StudentFees where DOP>'2019-12-05';
```

S_ID
201006
201007
201008
201009
201010
201011
201012

7 rows in set (0.00 sec)

```
mysql> select * from StudentAddress where NOT(Nationality='Indian');
```

S_Address	Nationality
S_Address 5	British

Activities Terminal Tue 08:31 viren@viren-VirtualBox: ~

File Edit View Search Terminal Help

201012

7 rows in set (0.00 sec)

mysql> select * from StudentAddress where NOT(Nationality='Indian');

S_Address	Nationality
S_Address 5	British
S_Address 6	British
S_Address 7	Nepali

3 rows in set (0.00 sec)

mysql> select Staff_ID from StaffAdd where Staff_Address = NULL;

select Staff_ID from StaffAdd where Staff_Address = NULL;

mysql> select Staff_ID from StaffAdd where Staff_Address = NULL;

Empty set (0.05 sec)

mysql> select count(S_ID) from StudentAdd where S_Address = NULL ;

select count(S_ID) from StudentAdd where S_Address = NULL ;

mysql> select count(S_ID) from StudentAdd where S_Address = NULL;

count(S_ID)
0

1 row in set (0.05 sec)

mysql> ;

mysql>

Complex Queries

/*Q1: Select Name and Address of All Cleaners whose Age is Greater than 43.*/

```
mysql> select Cleaners.C_name, Staff.Staff_Age from Cleaners join Staff
where Staff.Staff_ID=Cleaners.C_Staff_ID and Staff.Staff_Age>43;
```

```
+-----+-----+
| C_name | Staff_Age |
+-----+-----+
| Bhavesh |          46 |
| Avesh   |          46 |
| Jehan   |          51 |
+-----+-----+
3 rows in set (0.00 sec)
```

/*Q2: select Warden's details older than the age of 40 via DOB*/

```
mysql> select Warden.W_name, Staff.Staff_DOB from Warden join Staff where
Staff.Staff_ID=Warden.W_Staff and Staff.Staff_DOB<'1990-01-01';
```

```
+-----+-----+
| W_name | Staff_DOB |
+-----+-----+
| Mandeep | 1988-01-01 |
| Sandeep | 1986-01-01 |
+-----+-----+
2 rows in set (0.00 sec)
```

/*Q3: Select Staff_ID and Age of the Staff Member which are having Multiple/Alternate Address.*/

```
mysql> select s1.Staff_ID,s1.Staff_Age from Staff s1,StaffAdd s2 where
s1.Staff_ID=s2.Staff_ID group by s1.Staff_ID having
count(s2.Staff_Address)>0;
```

```
+-----+-----+
| Staff_ID | Staff_Age |
+-----+-----+
| 190003 |          45 |
| 190004 |          40 |
| 190009 |          41 |
| 190010 |          41 |
| 190011 |          42 |
+-----+-----+
5 rows in set (0.00 sec)
```

/*Q4: Select Hostel Name and Hostel Number of Such Hostel which are having Number of Washrooms greater than 7.*/

```
mysql> select h.Hostel_no, h.Hostel_name from Hostel h, Rooms r where
h.Rooms=r.Rooms group by h.hostel_no having avg(r.Washrooms)>7;
```

```
+-----+-----+
| Hostel_no | Hostel_name |
+-----+-----+
```

20	Hostel B
30	Hostel C
40	Hostel D

3 rows in set (0.00 sec)

/*Q5: Select All the Hostel Names and Warden IDs such that even if one hostel is having Number of Rooms Less than 45.*/

```
mysql> select Hostel_name, Warden as Warden_ID from Hostel where exists
(select Rooms from Hostel where Rooms<45);
```

Hostel_name	Warden_ID
Hostel A	190001
Hostel B	190002
Hostel C	190003
Hostel D	190004

4 rows in set (0.00 sec)

/*Q6: Select All the Hostel Names and Warden names such that no hostel is having Number of Rooms more than 25.*/

```
mysql> select h.Hostel_name, Warden.W_Name as Warden_Name from Hostel
h,Warden where Warden.W_Staff=h.Warden and exists (select Rooms from
Hostel where Rooms<25);
Empty set (0.00 sec)
```

/*Q7: Select All the Hostel Names and Warden names such that no hostel is having Number of Rooms less than 25.*/

```
mysql> select h.Hostel_name, Warden.W_Name as Warden_Name from Hostel
h,Warden where Warden.W_Staff=h.Warden and not exists (select Rooms from
Hostel where Rooms<25);
```

Hostel_name	Warden_Name
Hostel A	Dhruv
Hostel B	Mandeep
Hostel C	Sandeep
Hostel D	Rahul

4 rows in set (0.00 sec)

/*Q8: Select Student Name and Student ID of all students who have paid Fees.*/

```
mysql> select s.S_Name,s.S_ID from Student s,StudentFees sf where sf.S_ID
= s.S_ID and sf.Amount_paid=any (Select Amount_paid from StudentFees where
Amount_paid is not null);
```

S_Name	S_ID
--------	------


```

+-----+-----+
| Ram    | 201001 |
| Shyaam | 201002 |
| Ramu    | 201003 |
| Shyamu  | 201004 |
| Dhruvi  | 201005 |
| Dhruvill | 201006 |
| Roy     | 201007 |
| Rohit   | 201008 |
| Rohi    | 201009 |
| Devdutt | 201010 |
| Devdut  | 201011 |
| Dev     | 201012 |
+-----+-----+
12 rows in set (0.00 sec)

```

```

/*Q9: Select Student Name and Student ID of all students who have paid
24000 as fees*/

```

```

mysql> select s.S_Name,s.S_ID from Student s,StudentFees sf where sf.S_ID
= s.S_ID and sf.Amount_paid=all (Select Amount_paid from StudentFees where
Amount_paid=24000);
+-----+-----+
| S_Name | S_ID  |
+-----+-----+
| Rohi   | 201009 |
+-----+-----+
1. row in set (0.00 sec)

```

```

/*Q10: Select Student Address and student Name Who belongs to INDIA and
NEPAL.*/

```

```

mysql> select s.S_Address,s.S_Name from Student s, StudentAddress sa where
s.S_Address=sa.S_Address and sa.Nationality in ("Indian","Nepali");
+-----+-----+
| S_Address | S_Name |
+-----+-----+
| S_Address 1 | Ram    |
| S_Address 10 | Devdutt |
| S_Address 11 | Devdut |
| S_Address 12 | Dev     |
| S_Address 2 | Shyaam |
| S_Address 3 | Ramu    |
| S_Address 4 | Shyamu  |
| S_Address 7 | Roy     |
| S_Address 8 | Rohit   |
| S_Address 9 | Rohi    |
+-----+-----+
10 rows in set (0.00 sec)

```



```
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> select Warden.W_name, Staff.Staff_DOB from Warden join Staff where Staff.Staff_ID=Warden.W_Staff and Staff.Staff_DOB<'1990-01-01';
+-----+-----+
| W_name | Staff_DOB |
+-----+-----+
| Mandeep | 1988-01-01 |
| Sandeep | 1986-01-01 |
+-----+-----+
2 rows in set (0.00 sec)

mysql>
```



```
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> select Hostel_name, Warden as Warden_ID from Hostel where exists (select Rooms from Hostel where Rooms<45);
+-----+-----+
| Hostel_name | Warden_ID |
+-----+-----+
| Hostel A    | 190001    |
| Hostel B    | 190002    |
| Hostel C    | 190003    |
| Hostel D    | 190004    |
+-----+-----+
4 rows in set (0.00 sec)

mysql>
```



```
mysql> 
mysql> 
mysql> 
mysql> 
mysql> 
mysql> 
mysql> 
mysql> 
mysql> 
mysql> 
mysql> 
mysql> 
mysql> 
mysql> 
mysql> 
mysql> 
mysql> 
mysql> 
mysql> 
mysql> select h.Hostel_name, Warden.W_Name as Warden_Name from Hostel h,Warden where Warden.W_Staff=h.Warden and not exists (select Rooms from Hostel where Rooms<25);
+-----+-----+
| Hostel_name | Warden_Name |
+-----+-----+
| Hostel A    | Dhruv       |
| Hostel B    | Mandeep     |
| Hostel C    | Sandeep     |
| Hostel D    | Rahul       |
+-----+-----+
4 rows in set (0.00 sec)

mysql>
```


Activities Terminal Wed 21:24 virtualBox: ~

File Edit View Search Terminal Help

mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> select s.S_Name,s.S_ID from Student s,StudentFees sf where sf.S_ID = s.S_ID and sf.Amount_paid=any (Select Amount_paid from StudentFees where Amount_paid is not null);
+-----+-----+
| S_Name | S_ID |
+-----+-----+
Ram	201001
Shyaam	201002
Ramu	201003
Shyanu	201004
Dhruvi	201005
Dhruvill	201006
Roy	201007
Rohit	201008
Rohi	201009
Devdutt	201010
Devdut	201011
Dev	201012
+-----+-----+
12 rows in set (0.00 sec)
mysql>

Activities Terminal Wed 21:25 virrtualBox: ~

File Edit View Search Terminal Help

mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> select s.S_Address,s.S_Name from Student s, StudentAddress sa where s.S_Address=sa.S_Address and sa.Nationality in ("Indian","Nepali");
+-----+-----+
| S_Address | S_Name |
+-----+-----+
S_Address 1	Ram
S_Address 10	Devdutt
S_Address 11	Devdut
S_Address 12	Dev
S_Address 2	Shyaam
S_Address 3	Ramu
S_Address 4	Shyamu
S_Address 7	Roy
S_Address 8	Rohit
S_Address 9	Rohl
+-----+-----+
10 rows in set (0.00 sec)
mysql>

VIEWS

```
/*Q1: create view Greater where Name and Address of All Cleaners whose Age
is Greater than
43.*/
```

```
mysql> create view Greater as select Cleaners.C_name, Staff.Staff_Age from
Cleaners join Staff where Staff.Staff_ID=Cleaners.C_Staff_ID and
Staff.Staff_Age>43;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select * from Greater;
```

```
+-----+-----+
| C_name | Staff_Age |
+-----+-----+
| Bhavesh | 46 |
| Avesh | 46 |
| Jehan | 51 |
+-----+-----+
3 rows in set (0.00 sec)
```

```
/*Q2: create view Alternate where Staff_ID and Age of the Staff Member
which are having
Multiple/Alternate Address.*/
```

```
mysql> create view Alternate as select s1.Staff_ID,s1.Staff_Age from Staff
s1,StaffAdd s2 where s1.Staff_ID=s2.Staff_ID group by s1.Staff_ID having
count(s2.Staff_Address)>0;
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> select * from Alternate;
```

```
+-----+-----+
| Staff_ID | Staff_Age |
+-----+-----+
| 190003 | 45 |
| 190004 | 40 |
| 190009 | 41 |
| 190010 | 41 |
| 190011 | 42 |
+-----+-----+
5 rows in set (0.00 sec)
```

```
/*Q3: create view WashRoom having Hostel Name and Hostel Number of Such
Hostel which are having Number of Washrooms greater than 7.*/
```

```
mysql> create view WashRoom as select h.Hostel_no, h.Hostel_name from
Hostel h, Rooms r where
h.Rooms=r.Rooms group by h.hostel_no having avg(r.Washrooms)>7;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select * from WashRoom;
```

```
+-----+-----+
| Hostel_no | Hostel_name |
+-----+-----+
| 20 | Hostel B |
| 30 | Hostel C |
+-----+-----+
```

```
|          40 | Hostel D      |
+-----+-----+
3 rows in set (0.05 sec)
```

/*Q4: create view Roomview having the Hostel Names and Warden names such that no hostel is having Number of Rooms more than 25.*/

```
mysql> create view Roomview as select h.Hostel_name, Warden.W_Name as
Warden_Name from Hostel h,Warden where Warden.W_Staff=h.Warden and exists
(select Rooms from Hostel where Rooms<25);
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> select * from Roomview;
Empty set (0.00 sec)
```

/*Q5: create view Nation having Student Address and student Name Who belongs to INDIA and NEPAL.*/

```
mysql> create view Nation as select s.S_Address,s.S_Name from Student s,
StudentAddress sa where
s.S_Address=sa.S_Address and sa.Nationality in ("Indian","Nepali");
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> select * from Nation;
```

```
+-----+-----+
| S_Address | S_Name |
+-----+-----+
| S_Address 1 | Ram    |
| S_Address 10 | Devdutt |
| S_Address 11 | Devdut |
| S_Address 12 | Dev    |
| S_Address 2 | Shyaam |
| S_Address 3 | Ramu    |
| S_Address 4 | Shyamu  |
| S_Address 7 | Roy     |
| S_Address 8 | Rohit   |
| S_Address 9 | Rohi    |
+-----+-----+
10 rows in set (0.00 sec)
```

File Edit View Search Terminal Help

mysql>

mysql>

mysql>

mysql>

mysql> create view Greater as select Cleaners.C_name, Staff.Staff_Age from Cleaners join Staff where Staff.Staff_ID=Cleaners.C_Staff_ID and Staff.Staff_Age>43;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from Greater;

C_name	Staff_Age
Bhaves	46
Avesh	46
Jehan	51

3 rows in set (0.00 sec)

mysql> create view Alternate as select s1.Staff_ID,s1.Staff_Age from Staff s1,StaffAdd s2 where s1.Staff_ID=s2.Staff_ID group by s1.Staff_ID having count(s2.Staff_Address)>0;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from Alternate;

Staff_ID	Staff_Age
190003	45
190004	40
190009	41
190010	41
190011	42

5 rows in set (0.00 sec)

mysql>

File Edit View Search Terminal Help

```
taff_ID having count(s2.Staff_Address)>0;  
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> select * from Alternate;
```

```
+-----+-----+  
| Staff_ID | Staff_Age |  
+-----+-----+  
| 190003 | 45 |  
| 190004 | 40 |  
| 190009 | 41 |  
| 190010 | 41 |  
| 190011 | 42 |  
+-----+-----+  
5 rows in set (0.00 sec)
```

```
mysql> create view WashRoom as select h.Hostel_no, h.Hostel_name from Hostel h, Rooms r where  
-> h.Rooms=r.Rooms group by h.hostel_no having avg(r.Washrooms)>7;  
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> select * from WashRoom;
```

```
+-----+-----+  
| Hostel_no | Hostel_name |  
+-----+-----+  
| 20 | Hostel B |  
| 30 | Hostel C |  
| 40 | Hostel D |  
+-----+-----+  
3 rows in set (0.00 sec)
```

```
mysql> create view Roomview as select h.Hostel_name, Warden.W_Name as Warden_Name from Hostel h,Warden where Warden.W_Staff=h.Warden  
and exists (select Rooms from Hostel where Rooms<25);  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select * from Roomview;  
Empty set (0.00 sec)
```


File Edit View Search Terminal Help

	20		Hostel B	
	30		Hostel C	
	40		Hostel D	

+-----+
3 rows in set (0.00 sec)

```
mysql> create view Roomview as select h.Hostel_name, Warden.W_Name as Warden_Name from Hostel h,Warden where Warden.W_Staff=h.Warden
and exists (select Rooms from Hostel where Rooms<25);
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select * from Roomview;
Empty set (0.00 sec)
```

```
mysql> create view Nation as select s.S_Address,s.S_Name from Student s, StudentAddress sa where
-> s.S_Address=sa.S_Address and sa.Nationality in ("Indian","Nepali");
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> select * from Nation;
```

S_Address	S_Name
S_Address 1	Ram
S_Address 10	Devdutt
S_Address 11	Devdut
S_Address 12	Dev
S_Address 2	Shyaam
S_Address 3	Ramu
S_Address 4	Shyamu
S_Address 7	Roy
S_Address 8	Rohit
S_Address 9	Rohi

+-----+
10 rows in set (0.00 sec)

```
mysql> 
```

STORED PROCEDURES

/*Q1: showing all the staff members and then Counting staff members below the age of 50 using stored procedure called get_Regular_citizen();*/

```
mysql> DELIMITER //
mysql> CREATE PROCEDURE get_Regular_citizen()
-> BEGIN
->     SELECT * FROM Staff;
->     select count(Staff_Age) from Staff where Staff_Age<50;
-> END //
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> DELIMITER ;
mysql> call get_Regular_citizen()
-> ;
```

```
+-----+-----+-----+-----+-----+
| Staff_ID | Staff_Address | Staff_Age | Staff_DOB | PIN |
+-----+-----+-----+-----+-----+
| 190001 | Address 1 | 41 | 1990-01-01 | 4007030 |
| 190002 | Address 2 | 43 | 1988-01-01 | 4007030 |
| 190003 | Address 3 | 45 | 1986-01-01 | 4007031 |
| 190004 | Address 4 | 40 | 1991-01-01 | 4008510 |
| 190005 | Address 5 | 38 | 1993-01-01 | 4008510 |
| 190006 | Address 6 | 38 | 1993-01-01 | 4008510 |
| 190007 | Address 7 | 46 | 1985-01-01 | 4007031 |
| 190008 | Address 8 | 46 | 1985-01-01 | 4007031 |
| 190009 | Address 9 | 41 | 1990-01-01 | 4007030 |
| 190010 | Address 10 | 41 | 1990-01-01 | 4007030 |
| 190011 | Address 11 | 42 | 1989-01-01 | 4007030 |
| 190012 | Address 12 | 51 | 1980-01-01 | 4008131 |
+-----+-----+-----+-----+-----+
12 rows in set (0.00 sec)
```

```
+-----+
| count(Staff_Age) |
+-----+
| 11 |
+-----+
1. row in set (0.00 sec)
```

Query OK, 0 rows affected (0.00 sec)

/*Q2: Selecting Cleaners who have C_Staff_ID less that 190010 And then Selecting Cleaner with phone numbers 9895661234 and 9891261234 using procedure get_Cleaner_numbers()*/

```
mysql> DELIMITER //
mysql> CREATE PROCEDURE get_Cleaner_numbers()
-> BEGIN
->     SELECT * FROM Cleaners where C_Staff_ID<190010;
->     select C_Name from Cleaners where C_Phone = 9895661234 or
C_Phone = 9891261234;
-> END //
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> DELIMITER ;
mysql> call get_Cleaner_numbers()
-> ;
```

```
+-----+-----+-----+-----+
| C_Staff_ID | C_Name | C_Phone | C_Email |
+-----+-----+-----+-----+
| 190005 | Paulim | 9895661234 | paulim@gmail.com |
| 190006 | Paul | 9891261234 | paul@gmail.com |
| 190007 | Bhavesh | 9891223434 | bhavesh@gmail.com |
| 190008 | Avesh | 9845223434 | avesh@gmail.com |
| 190009 | Vesh | 9841233434 | vesh@gmail.com |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
+-----+
| C_Name |
+-----+
| Paulim |
| Paul |
+-----+
2. rows in set (0.00 sec)
```

Query OK, 0 rows affected (0.00 sec)

/*Q3: Select Student Name and Student ID of all students who have paid Fees and Select Student Name and Student ID of all students who have paid 24000 as fees using procedure get_fee_details()*/

```
mysql> DELIMITER //
mysql> CREATE PROCEDURE get_fee_details()
-> BEGIN
-> select s.S_Name,s.S_ID from Student s,StudentFees sf where
sf.S_ID = s.S_ID and sf.Amount_paid=any (Select Amount_paid from
StudentFees where Amount_paid is not null);
-> select s.S_Name,s.S_ID from Student s,StudentFees sf where
sf.S_ID = s.S_ID and sf.Amount_paid=all (Select Amount_paid from
StudentFees where Amount_paid=24000);
-> END //
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> DELIMITER ;
mysql> call get_fee_details()
-> ;
```

```
+-----+-----+
| S_Name | S_ID |
+-----+-----+
| Ram | 201001 |
| Shyaam | 201002 |
| Ramu | 201003 |
| Shyamu | 201004 |
| Dhruvi | 201005 |
| Dhruvill | 201006 |
| Roy | 201007 |
```

Rohit	201008
Rohi	201009
Devdutt	201010
Devdut	201011
Dev	201012

12 rows in set (0.00 sec)

S_Name	S_ID
Rohi	201009

1. row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

/*Q4: Displaying all Addresses that are unknown in Staff Student and the international Address of Students too using get_Address()*/

```
mysql> DELIMITER //
mysql> CREATE PROCEDURE get_Address()
-> BEGIN
->     select * from StudentAddress where NOT(Nationality='Indian');
->     select Staff_ID from StaffAdd where Staff_Address = NULL;
->     select * from StudentAdd where S_Address = NULL;
-> END //
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> DELIMITER ;
mysql> call get_Address();
```

S_Address	Nationality
S_Address 5	British
S_Address 6	British
S_Address 7	Nepali

3 rows in set (0.00 sec)

Empty set (0.00 sec)

Empty set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

/*Q5: Select All the Hostel Names and Warden IDs such that even if one hostel is having Number of Rooms Less than 45 and also the tables with rooms more that 25 and less that 25 rooms using get_Tables()*/

```
mysql> DELIMITER //
mysql> CREATE PROCEDURE get_Tables()
-> BEGIN
```

```

->      select Hostel_name, Warden as Warden_ID from Hostel where
exists (select Rooms from Hostel where Rooms<45);
->      select h.Hostel_name, Warden.W_Name as Warden_Name from Hostel
h,Warden where Warden.W_Staff=h.Warden and exists (select Rooms from
Hostel where Rooms<25);
->      select h.Hostel_name, Warden.W_Name as Warden_Name from Hostel
h,Warden where Warden.W_Staff=h.Warden and not exists (select Rooms from
Hostel where Rooms<25);
-> END //
Query OK, 0 rows affected (0.00 sec)

```

```

mysql> DELIMITER ;
mysql> call get_Tables();
+-----+-----+
| Hostel_name | Warden_ID |
+-----+-----+
| Hostel A    | 190001    |
| Hostel B    | 190002    |
| Hostel C    | 190003    |
| Hostel D    | 190004    |
+-----+-----+
4 rows in set (0.00 sec)

```

Empty set (0.00 sec)

```

+-----+-----+
| Hostel_name | Warden_Name |
+-----+-----+
| Hostel A    | Dhruv       |
| Hostel B    | Mandeep     |
| Hostel C    | Sandeep     |
| Hostel D    | Rahul       |
+-----+-----+
4 rows in set (0.00 sec)

```

Query OK, 0 rows affected (0.00 sec)

Activities Terminal Mon 01:26 viren@viren-VirtualBox: ~

File Edit View Search Terminal Help

```
mysql> DELIMITER //
mysql> CREATE PROCEDURE get_Regular_citizen()
-> BEGIN
->     SELECT * FROM Staff;
->     select count(Staff_Age) from Staff where Staff_Age<50;
-> END //
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER ;
mysql> call get_Regular_citizen()
-> ;
```

Staff_ID	Staff_Address	Staff_Age	Staff_DOB	PIN
190001	Address 1	41	1990-01-01	4007030
190002	Address 2	43	1988-01-01	4007030
190003	Address 3	45	1986-01-01	4007031
190004	Address 4	40	1991-01-01	4008510
190005	Address 5	38	1993-01-01	4008510
190006	Address 6	38	1993-01-01	4008510
190007	Address 7	46	1985-01-01	4007031
190008	Address 8	46	1985-01-01	4007031
190009	Address 9	41	1990-01-01	4007030
190010	Address 10	41	1990-01-01	4007030
190011	Address 11	42	1989-01-01	4007030
190012	Address 12	51	1980-01-01	4008131

```
12 rows in set (0.00 sec)

+-----+
| count(Staff_Age) |
+-----+
|          11      |
+-----+
1 row in set (0.00 sec)
```

```
Activities Terminal Mon 01:32 viren@viren-VirtualBox: ~
File Edit View Search Terminal Help
8 rows in set (0.00 sec)

mysql> DELIMITER //
mysql> CREATE PROCEDURE get_Cleaner_numbers()
-> BEGIN
->     SELECT * FROM Cleaners where C_Staff_ID<190010;
->     select C_Name from Cleaners where C_Phone = 9895661234 or C_Phone = 9891261234;
-> END //
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER ;
mysql> call get_Cleaner_numbers()
-> ;

+-----+-----+-----+-----+
| C_Staff_ID | C_Name | C_Phone | C_Email |
+-----+-----+-----+-----+
| 190005 | Paulm | 9895661234 | paulm@gmail.com |
| 190006 | Paul | 9891261234 | paul@gmail.com |
| 190007 | Bhavesh | 9891223434 | bhavesh@gmail.com |
| 190008 | Avesh | 9845223434 | avesh@gmail.com |
| 190009 | Vesh | 9841233434 | vesh@gmail.com |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

+-----+
| C_Name |
+-----+
| Paulm |
| Paul |
+-----+
2 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> 
```


Activities Terminal Mon 01:36 viren@viren-VirtualBox: ~

File Edit View Search Terminal Help

```
mysql> DELIMITER //
mysql> CREATE PROCEDURE get_fee_details()
-> BEGIN
->     select s.S_Name,s.S_ID from Student s,StudentFees sf where sf.S_ID = s.S_ID and sf.Amount_paid=any (Select Amount_paid fro
m StudentFees where Amount_paid is not null);
->     select s.S_Name,s.S_ID from Student s,StudentFees sf where sf.S_ID = s.S_ID and sf.Amount_paid=all (Select Amount_paid fro
m StudentFees where Amount_paid=24000);
-> END //
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER ;
mysql> call get_fee_details()
-> ;
```

S_Name	S_ID
Ram	201001
Shyaam	201002
Ramu	201003
Shyanu	201004
Dhruvi	201005
Dhruvill	201006
Roy	201007
Rohit	201008
Rohi	201009
Devdutt	201010
Devdut	201011
Dev	201012

12 rows in set (0.00 sec)

S_Name	S_ID
Rohi	201009

Activities Terminal Mon 01:36 viren@viren-VirtualBox: ~

File Edit View Search Terminal Help

m StudentFees where Amount_paid=24000);
-> END //

Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER ;
mysql> call get_fee_details()
-> ;

S_Nane	S_ID
Ram	201001
Shyaam	201002
Ramu	201003
Shyanu	201004
Dhruvi	201005
Dhruvill	201006
Roy	201007
Rohit	201008
Rohi	201009
Devdutt	201010
Devdut	201011
Dev	201012

12 rows in set (0.00 sec)

S_Nane	S_ID
Rohi	201009

1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql>

Activities Terminal Mon 01:42 viren@viren-VirtualBox: ~

File Edit View Search Terminal Help

```
mysql> DELIMITER //
mysql> CREATE PROCEDURE get_Address()
-> BEGIN
->   select * from StudentAddress where NOT(Nationality='Indian');
->   select Staff_ID from StaffAdd where Staff_Address = NULL;
->   select * from StudentAdd where S_Address = NULL;
-> END //
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER ;
mysql> call get_Address();
+-----+-----+
| S_Address | Nationality |
+-----+-----+
| S_Address 5 | British |
| S_Address 6 | British |
| S_Address 7 | Nepali |
+-----+-----+
3 rows in set (0.00 sec)

Empty set (0.00 sec)

Empty set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
```

```
Activities Terminal Mon 01:46 viren@viren-VirtualBox: ~
File Edit View Search Terminal Help
mysql>
mysql> DELIMITER //
mysql> CREATE PROCEDURE get_Tables()
-> BEGIN
->   select Hostel_name, Warden as Warden_ID from Hostel where exists (select Rooms from Hostel where Rooms<45);
->   select h.Hostel_name, Warden.W_Name as Warden_Name from Hostel h,Warden where Warden.W_Staff=h.Warden and exists (select
Rooms from Hostel where Rooms<25);
->   select h.Hostel_name, Warden.W_Name as Warden_Name from Hostel h,Warden where Warden.W_Staff=h.Warden and not exists (sel
ect Rooms from Hostel where Rooms<25);
-> END //
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER ;
mysql> call get_Tables();
+-----+-----+
| Hostel_name | Warden_ID |
+-----+-----+
| Hostel A   | 190001    |
| Hostel B   | 190002    |
| Hostel C   | 190003    |
| Hostel D   | 190004    |
+-----+-----+
4 rows in set (0.00 sec)

Empty set (0.00 sec)

+-----+-----+
| Hostel_name | Warden_Name |
+-----+-----+
| Hostel A   | Dhruv       |
| Hostel B   | Mandeep     |
| Hostel C   | Sandeep     |
| Hostel D   | Rahul       |
+-----+-----+
4 rows in set (0.00 sec)
```

Activities Terminal Mon 01:46 viren@viren-VirtualBox: ~

File Edit View Search Terminal Help

```
-> select Hostel_name, Warden as Warden_ID from Hostel where exists (select Rooms from Hostel where Rooms<45);
-> select h.Hostel_name, Warden.W_Name as Warden_Name from Hostel h,Warden where Warden.W_Staff=h.Warden and exists (select
Rooms from Hostel where Rooms<25);
-> select h.Hostel_name, Warden.W_Name as Warden_Name from Hostel h,Warden where Warden.W_Staff=h.Warden and not exists (sel
ect Rooms from Hostel where Rooms<25);
-> END //
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> DELIMITER ;
mysql> call get_Tables();
```

Hostel_name	Warden_ID
Hostel A	190001
Hostel B	190002
Hostel C	190003
Hostel D	190004

4 rows in set (0.00 sec)

Empty set (0.00 sec)

```
>_
```

Hostel_name	Warden_Name
Hostel A	Dhruv
Hostel B	Mandeep
Hostel C	Sandeep
Hostel D	Rahul

4 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

```
mysql>
```

STORED FUNCTIONS

/*Q1: Number of years it has been for each student post being enrolled to a Hostel using a function called no_of_years*/

```
mysql> DELIMITER //
```

```
mysql>
```

```
mysql> CREATE FUNCTION no_of_years(date1 date) RETURNS int DETERMINISTIC
-> BEGIN
->   DECLARE date2 DATE;
->   Select current_date()into date2;
->   RETURN year(date2)-year(date1);
-> END
->
-> //
```

Query OK, 0 rows affected (0.00 sec)

```
mysql>
```

```
mysql> DELIMITER ;
```

```
mysql> select *,no_of_years(S_DOA) as Years from Student;
```

```
+-----+-----+-----+-----+-----+-----+-----+
| S_ID   | S_Name   | S_DOB      | S_Phone    | S_Email    | S_Address | S_HostelNo | S_DOA      | Years |
+-----+-----+-----+-----+-----+-----+-----+
| 201001 | Ram      | 2001-01-01 | 9869411234 | ram@gmail.com | S_Address 1 | 10 | 2020-01-01 | 1 |
| 201002 | Shyaam   | 2001-02-01 | 9869121234 | shyam@gmail.com | S_Address 2 | 10 | 2020-01-02 | 1 |
| 201003 | Ramu     | 2001-03-01 | 9869821234 | ramu@gmail.com | S_Address 3 | 10 | 2020-01-03 | 1 |
| 201004 | Shyamu   | 2001-04-01 | 9123821234 | shyamu@gmail.com | S_Address 4 | 10 | 2020-01-04 | 1 |
| 201005 | Dhruvi   | 2001-05-01 | 9124567234 | dhruvi@gmail.com | S_Address 5 | 20 | 2020-01-04 | 1 |
| 201006 | Dhruvill | 2001-06-01 | 9124567321 | dhruvill@gmail.com | S_Address 6 | 20 | 2020-01-04 | 1 |
| 201007 | Roy      | 2001-07-01 | 9124234321 | roy@gmail.com | S_Address 7 | 30 | 2020-01-05 | 1 |
| 201008 | Rohit    | 2001-08-01 | 9124123321 | rohit@gmail.com | S_Address 8 | 30 | 2020-01-06 | 1 |
| 201009 | Rohi     | 2001-09-01 | 9345123321 | rohi@gmail.com | S_Address 9 | 30 | 2020-01-07 | 1 |
| 201010 | Devdutt  | 2001-10-01 | 9345456321 | devdutt@gmail.com | S_Address 10 | 40 | 2020-01-08 | 1 |
| 201011 | Devdut   | 2001-11-01 | 9312456321 | devdut@gmail.com | S_Address 11 | 40 | 2020-01-08 | 1 |
| 201012 | Dev      | 2001-12-01 | 9312006321 | dev@gmail.com | S_Address 12 | 40 | 2020-01-09 | 1 |
+-----+-----+-----+-----+-----+-----+-----+
12 rows in set (0.00 sec)
```

/*Q2: Classifying if a staff member is a Senior Citizen or not which here we assume that as 50 or 50+ using a function*/

```
mysql> DELIMITER //
mysql> CREATE FUNCTION Senior_or_not(
->     age int
-> )
-> RETURNS VARCHAR(20)
-> DETERMINISTIC
-> BEGIN
->     DECLARE Age_status VARCHAR(20);
->     IF age >= 50 THEN
->         SET Age_status = 'Senior Citizen';
->     ELSEIF (age >= 18 AND
->         age <= 49) THEN
->         SET Age_status = 'Regular Worker';
->     ELSEIF age < 18 THEN
->         SET Age_status = 'Illegal';
->     END IF;
->     -- return the Age_status
->     RETURN (Age_status);
-> END //
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> DELIMITER ;
mysql> select *,Senior_or_not(Staff_age) as Age_Status from Staff;
+-----+-----+-----+-----+-----+-----+
-----+
| Staff_ID | Staff_Address | Staff_Age | Staff_DOB | PIN | Age_Status |
|
+-----+-----+-----+-----+-----+-----+
-----+
| 190001 | Address 1 | 41 | 1990-01-01 | 4007030 | Regular Worker |
| 190002 | Address 2 | 43 | 1988-01-01 | 4007030 | Regular Worker |
| 190003 | Address 3 | 45 | 1986-01-01 | 4007031 | Regular Worker |
| 190004 | Address 4 | 40 | 1991-01-01 | 4008510 | Regular Worker |
| 190005 | Address 5 | 38 | 1993-01-01 | 4008510 | Regular Worker |
| 190006 | Address 6 | 38 | 1993-01-01 | 4008510 | Regular Worker |
| 190007 | Address 7 | 46 | 1985-01-01 | 4007031 | Regular Worker |
| 190008 | Address 8 | 46 | 1985-01-01 | 4007031 | Regular Worker |
| 190009 | Address 9 | 41 | 1990-01-01 | 4007030 | Regular Worker |
| 190010 | Address 10 | 41 | 1990-01-01 | 4007030 | Regular Worker |
| 190011 | Address 11 | 42 | 1989-01-01 | 4007030 | Regular Worker |
```



```

| 190012 | Address 12 | 51 | 1980-01-01 | 4008131 | Senior
Citizen |
+-----+-----+-----+-----+-----+-----+
-----+
12 rows in set (0.00 sec)

```

/*Q3: Using a function determine the number of Washrooms a Hostel has,
Function name will be no_of_washrooms*/

```

mysql> DELIMITER //
mysql> CREATE FUNCTION no_of_washrooms(
-> Rooms int
-> )
-> RETURNS int
-> DETERMINISTIC
-> BEGIN
-> DECLARE washrooms int;
-> set washrooms = Rooms*0.20;
-> RETURN (washrooms);
-> END //

```

Query OK, 0 rows affected (0.00 sec)

```

mysql> DELIMITER ;
mysql> select *,no_of_washrooms(Rooms) from Hostel;
+-----+-----+-----+-----+-----+-----+
| Hostel_No | Hostel_name | Rooms | Warden | no_of_washrooms(Rooms) |
+-----+-----+-----+-----+-----+-----+
| 10 | Hostel A | 30 | 190001 | 6 |
| 20 | Hostel B | 40 | 190002 | 8 |
| 30 | Hostel C | 50 | 190003 | 10 |
| 40 | Hostel D | 60 | 190004 | 12 |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```

/*Q4: Months since the last payment by every student in StudentFees using
the function: time_in_months*/

```

mysql> DELIMITER //
mysql> CREATE FUNCTION time_in_years(date1 date) RETURNS int DETERMINISTIC
-> BEGIN
-> DECLARE date2 DATE;
-> Select current_date()into date2;
-> RETURN year(date2)-year(date1);
-> END
-> //

```

Query OK, 0 rows affected (0.00 sec)

```

mysql>
mysql> DELIMITER ;
mysql> select *,time_in_years(DOP) from StudentFees;
+-----+-----+-----+-----+-----+-----+
-----+
| Amount_Paid | AccNo | ReceiptCode | DOP | S_ID |
time_in_years(DOP) |

```

```

+-----+-----+-----+-----+-----+-----+
-----+
|      120000 | 100450010001 | 2003450001 | 2019-12-01 | 201001 |
2 |
|      120000 | 100450010002 | 2003450002 | 2019-12-02 | 201002 |
2 |
|      120000 | 100450010003 | 2003450003 | 2019-12-03 | 201003 |
2 |
|       70000 | 100450010004 | 2003450004 | 2019-12-04 | 201004 |
2 |
|      120000 | 100450010005 | 2003450005 | 2019-12-05 | 201005 |
2 |
|      120000 | 100450010006 | 2003450006 | 2019-12-06 | 201006 |
2 |
|      120000 | 100450010007 | 2003450007 | 2019-12-07 | 201007 |
2 |
|      120000 | 100450010008 | 2003450008 | 2019-12-08 | 201008 |
2 |
|       24000 | 100450010009 | 2003450009 | 2019-12-09 | 201009 |
2 |
|     200000 | 100450010010 | 2003450010 | 2019-12-10 | 201010 |
2 |
|      120000 | 100450010011 | 2003450011 | 2019-12-11 | 201011 |
2 |
|     140000 | 100450010012 | 2003450012 | 2019-12-12 | 201012 |
2 |
+-----+-----+-----+-----+-----+-----+
-----+
12 rows in set (0.00 sec)

```

/*Q5: Function Mess_fee to tell how much of the fees was given for mess food*/

```

mysql> DELIMITER //
mysql> CREATE FUNCTION Mess_fee(
->     Amount_paid int
-> )
-> RETURNS int
-> DETERMINISTIC
-> BEGIN
->     DECLARE messfee int;
->     set messfee = Amount_paid*0.40;
->     RETURN (messfee);
-> END //
Query OK, 0 rows affected (0.00 sec)

```

```

mysql> DELIMITER ;
mysql> select *,Mess_fee(Amount_Paid) from StudentFees;
+-----+-----+-----+-----+-----+-----+
-----+
| Amount_Paid | AccNo          | ReceiptCode | DOP           | S_ID   |
Mess_fee(Amount_Paid) |

```

```

+-----+-----+-----+-----+-----+-----+
-----+
|      120000 | 100450010001 | 2003450001 | 2019-12-01 | 201001 |
48000 |
|      120000 | 100450010002 | 2003450002 | 2019-12-02 | 201002 |
48000 |
|      120000 | 100450010003 | 2003450003 | 2019-12-03 | 201003 |
48000 |
|       70000 | 100450010004 | 2003450004 | 2019-12-04 | 201004 |
28000 |
|      120000 | 100450010005 | 2003450005 | 2019-12-05 | 201005 |
48000 |
|      120000 | 100450010006 | 2003450006 | 2019-12-06 | 201006 |
48000 |
|      120000 | 100450010007 | 2003450007 | 2019-12-07 | 201007 |
48000 |
|      120000 | 100450010008 | 2003450008 | 2019-12-08 | 201008 |
48000 |
|       24000 | 100450010009 | 2003450009 | 2019-12-09 | 201009 |
9600 |
|      200000 | 100450010010 | 2003450010 | 2019-12-10 | 201010 |
80000 |
|      120000 | 100450010011 | 2003450011 | 2019-12-11 | 201011 |
48000 |
|      140000 | 100450010012 | 2003450012 | 2019-12-12 | 201012 |
56000 |
+-----+-----+-----+-----+-----+-----+
-----+
12 rows in set (0.00 sec)

```

Activities Terminal Mon 00:00 viren@viren-VirtualBox: ~

File Edit View Search Terminal Help

mysql>
mysql> DELIMITER //
mysql>
mysql> CREATE FUNCTION no_of_years(date1 date) RETURNS int DETERMINISTIC
-> BEGIN
-> DECLARE date2 DATE;
-> Select current_date()into date2;
-> RETURN year(date2)-year(date1);
-> END
->
-> //
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> DELIMITER ;
mysql> select *,no_of_years(S_DOA) as Years from Student;

S_ID	S_Name	S_DOB	S_Phone	S_Email	S_Address	S_HostelNo	S_DOA	Years
201001	Ram	2001-01-01	9869411234	ram@gmail.com	S_Address 1	10	2020-01-01	1
201002	Shyaam	2001-02-01	9869121234	shyam@gmail.com	S_Address 2	10	2020-01-02	1
201003	Ranu	2001-03-01	9869821234	ranu@gmail.com	S_Address 3	10	2020-01-03	1
201004	Shyamu	2001-04-01	9123821234	shyamu@gmail.com	S_Address 4	10	2020-01-04	1
201005	Dhruvi	2001-05-01	9124567234	dhruvi@gmail.com	S_Address 5	20	2020-01-04	1
201006	Dhruvill	2001-06-01	9124567321	dhruvill@gmail.com	S_Address 6	20	2020-01-04	1
201007	Roy	2001-07-01	9124234321	roy@gmail.com	S_Address 7	30	2020-01-05	1
201008	Rohit	2001-08-01	9124123321	rohit@gmail.com	S_Address 8	30	2020-01-06	1
201009	Rohi	2001-09-01	9345123321	rohi@gmail.com	S_Address 9	30	2020-01-07	1
201010	Devdutt	2001-10-01	9345456321	devdutt@gmail.com	S_Address 10	40	2020-01-08	1
201011	Devdut	2001-11-01	9312456321	devdut@gmail.com	S_Address 11	40	2020-01-08	1
201012	Dev	2001-12-01	9312006321	dev@gmail.com	S_Address 12	40	2020-01-09	1

12 rows in set (0.00 sec)

mysql>

```
mysql> DELIMITER //
mysql> CREATE FUNCTION Senior_or_not(
->     age int
-> )
-> RETURNS VARCHAR(20)
-> DETERMINISTIC
-> BEGIN
->     DECLARE Age_status VARCHAR(20);
->     IF age >= 50 THEN
->         SET Age_status = 'Senior Citizen';
->     ELSEIF (age >= 18 AND
->         age <= 49) THEN
->         SET Age_status = 'Regular Worker';
->     ELSEIF age < 18 THEN
->         SET Age_status = 'Illegal';
->     END IF;
->     -- return the Age_status
->     RETURN (Age_status);
-> END //
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> DELIMITER ;
mysql> select *,Senior_or_not(Staff_age) as Age_Status from Staff;
```

Staff_ID	Staff_Address	Staff_Age	Staff_DOB	PIN	Age_Status
190001	Address 1	41	1990-01-01	4007030	Regular Worker
190002	Address 2	43	1988-01-01	4007030	Regular Worker
190003	Address 3	45	1986-01-01	4007031	Regular Worker
190004	Address 4	40	1991-01-01	4008510	Regular Worker
190005	Address 5	38	1993-01-01	4008510	Regular Worker
190006	Address 6	38	1993-01-01	4008510	Regular Worker
190007	Address 7	46	1985-01-01	4007031	Regular Worker
190008	Address 8	46	1985-01-01	4007031	Regular Worker
190009	Address 9	41	1990-01-01	4007030	Regular Worker

File Edit View Search Terminal Help

```
-> DECLARE Age_status VARCHAR(20);
-> IF age >= 50 THEN
->     SET Age_status = 'Senior Citizen';
-> ELSEIF (age >= 18 AND
->     age <= 49) THEN
->     SET Age_status = 'Regular Worker';
-> ELSEIF age < 18 THEN
->     SET Age_status = 'Illegal';
-> END IF;
-> -- return the Age_status
-> RETURN (Age_status);
-> END //
```

Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER ;

mysql> select *,Senior_or_not(Staff_age) as Age_Status from Staff;

Staff_ID	Staff_Address	Staff_Age	Staff_DOB	PIN	Age_Status
190001	Address 1	41	1990-01-01	4007030	Regular Worker
190002	Address 2	43	1988-01-01	4007030	Regular Worker
190003	Address 3	45	1986-01-01	4007031	Regular Worker
190004	Address 4	40	1991-01-01	4008510	Regular Worker
190005	Address 5	38	1993-01-01	4008510	Regular Worker
190006	Address 6	38	1993-01-01	4008510	Regular Worker
190007	Address 7	46	1985-01-01	4007031	Regular Worker
190008	Address 8	46	1985-01-01	4007031	Regular Worker
190009	Address 9	41	1990-01-01	4007030	Regular Worker
190010	Address 10	41	1990-01-01	4007030	Regular Worker
190011	Address 11	42	1989-01-01	4007030	Regular Worker
190012	Address 12	51	1980-01-01	4008131	Senior Citizen

12 rows in set (0.00 sec)

mysql>

```
Activities Terminal Mon 00:41 viren@viren-VirtualBox: ~
File Edit View Search Terminal Help
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> DELIMITER //
mysql> CREATE FUNCTION no_of_washrooms(
-> Rooms int
-> )
-> RETURNS int
-> DETERMINISTIC
-> BEGIN
-> DECLARE washrooms int;
-> set washrooms = Rooms*0.20;
-> RETURN (washrooms);
-> END //
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER ;
mysql> select *,no_of_washrooms(Rooms) from Hostel;
+-----+-----+-----+-----+-----+
| Hostel_No | Hostel_name | Rooms | Warden | no_of_washrooms(Rooms) |
+-----+-----+-----+-----+-----+
| 10 | Hostel A | 30 | 190001 | 6 |
| 20 | Hostel B | 40 | 190002 | 8 |
| 30 | Hostel C | 50 | 190003 | 10 |
| 40 | Hostel D | 60 | 190004 | 12 |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```


Activities Terminal Mon 01:04 viren@viren-VirtualBox: ~

File Edit View Search Terminal Help

```
mysql>
mysql>
mysql>
mysql> DELIMITER //
mysql> CREATE FUNCTION time_in_years(date1 date) RETURNS int DETERMINISTIC
-> BEGIN
-> DECLARE date2 DATE;
-> Select current_date()into date2;
-> RETURN year(date2)-year(date1);
-> END
-> //
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> DELIMITER ;
mysql> select *,time_in_years(DOP) from StudentFees;
```

Amount_Paid	AccNo	ReceiptCode	DOP	S_ID	time_in_years(DOP)
120000	100450010001	2003450001	2019-12-01	201001	2
120000	100450010002	2003450002	2019-12-02	201002	2
120000	100450010003	2003450003	2019-12-03	201003	2
70000	100450010004	2003450004	2019-12-04	201004	2
120000	100450010005	2003450005	2019-12-05	201005	2
120000	100450010006	2003450006	2019-12-06	201006	2
120000	100450010007	2003450007	2019-12-07	201007	2
120000	100450010008	2003450008	2019-12-08	201008	2
240000	100450010009	2003450009	2019-12-09	201009	2
200000	100450010010	2003450010	2019-12-10	201010	2
120000	100450010011	2003450011	2019-12-11	201011	2
140000	100450010012	2003450012	2019-12-12	201012	2

```
12 rows in set (0.00 sec)

mysql>
```


Activities Terminal Mon 01:05 viren@viren-VirtualBox: ~

File Edit View Search Terminal Help

mysql>
mysql> DELIMITER //
mysql> CREATE FUNCTION Mess_fee(
-> Amount_paid int
->)
-> RETURNS int
-> DETERMINISTIC
-> BEGIN
-> DECLARE messfee int;
-> set messfee = Amount_paid*0.40;
-> RETURN (messfee);
-> END //
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER ;
mysql> select *,Mess_fee(Amount_Paid) from StudentFees;
+-----+-----+-----+-----+-----+-----+
| Amount_Paid | AccNo | ReceiptCode | DOP | S_ID | Mess_fee(Amount_Paid) |
+-----+-----+-----+-----+-----+-----+
120000	100450010001	2003450001	2019-12-01	201001	48000
120000	100450010002	2003450002	2019-12-02	201002	48000
120000	100450010003	2003450003	2019-12-03	201003	48000
70000	100450010004	2003450004	2019-12-04	201004	28000
120000	100450010005	2003450005	2019-12-05	201005	48000
120000	100450010006	2003450006	2019-12-06	201006	48000
120000	100450010007	2003450007	2019-12-07	201007	48000
120000	100450010008	2003450008	2019-12-08	201008	48000
24000	100450010009	2003450009	2019-12-09	201009	9600
200000	100450010010	2003450010	2019-12-10	201010	80000
120000	100450010011	2003450011	2019-12-11	201011	48000
140000	100450010012	2003450012	2019-12-12	201012	56000
+-----+-----+-----+-----+-----+-----+
12 rows in set (0.00 sec)

mysql>

TRIGGERS

```

# Registering a warden before he/she becomes the warden of a hostel
delimiter $$
create trigger insert_hostel_warden
before insert on Hostel
for each row
begin
if new.Warden not in(select W_Staff from Warden) then
insert into Warden values(new.Warden,concat(new.Hostel_Name,'
Warden'),NULL,NULL);
end if;
end $$
delimiter ;

```

```

# Updating student records before updating fee records to ensure that the
student who has paid fees is present in the student records
delimiter $$
create trigger fee_payment
before update on StudentFees
for each row
begin
if new.S_ID not in (select S_ID from Student) then
insert into Students
values(new.S_ID,NULL,NULL,NULL,NULL,NULL,NULL,NULL);
end if;
end $$
delimiter ;

```

```

# Removing a cleaner from staff records if he chooses to quit
delimiter $$
CREATE TRIGGER Remove_Cleaner
AFTER delete ON Cleaners
FOR EACH ROW
BEGIN
delete from Staff where Staff.Staff_ID = old.C_Staff_ID;
END$$
delimiter ;

```

```
Activities Terminal Tue 19:36 viren@viren-VirtualBox: ~
File Edit View Search Terminal Help
mysql> show tables;
+-----+
| Tables_in_Hostel_Management_System |
+-----+
| Alternate
| Cleaners
| Greater
| Hostel
| Nation
| PinCode
| Rooms
| Roomview
| Staff
| StaffAdd
| Student
| StudentAdd
| StudentAddress
| StudentDOB
| StudentFees
| Warden
| WashRoom
+-----+
17 rows in set (0.00 sec)

mysql> delimiter $$
mysql> create trigger insert_hostel_warden
-> before insert on Hostel
-> for each row
-> begin
-> if new.Warden not in(select W_Staff from Warden) then
-> insert into Warden values(new.Warden,concat(new.Hostel_Name,' Warden'),NULL,NULL);
-> end if;
-> end $$
Query OK, 0 rows affected (0.03 sec)

mysql> delimiter ;
```

[illegible]

```
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> delimiter $$
mysql> CREATE TRIGGER Remove_Cleaner
-> AFTER delete ON Cleaners
-> FOR EACH ROW
-> BEGIN
-> delete from Staff where Staff.Staff_ID = old.C_Staff_ID;
-> END$$
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
mysql>
```