**DBMS PROJECT(UCS310)**

****

**HOSTEL MANAGEMENT SYSTEM**

**Submitted By:**

1.Nipun Tank(102153011)

2.Yuvam Sharma(102153012)

3.Manav Singh(102283005)

4.Manu Aggarwal(102103318)

5.Jasreet Kaur(102103317)

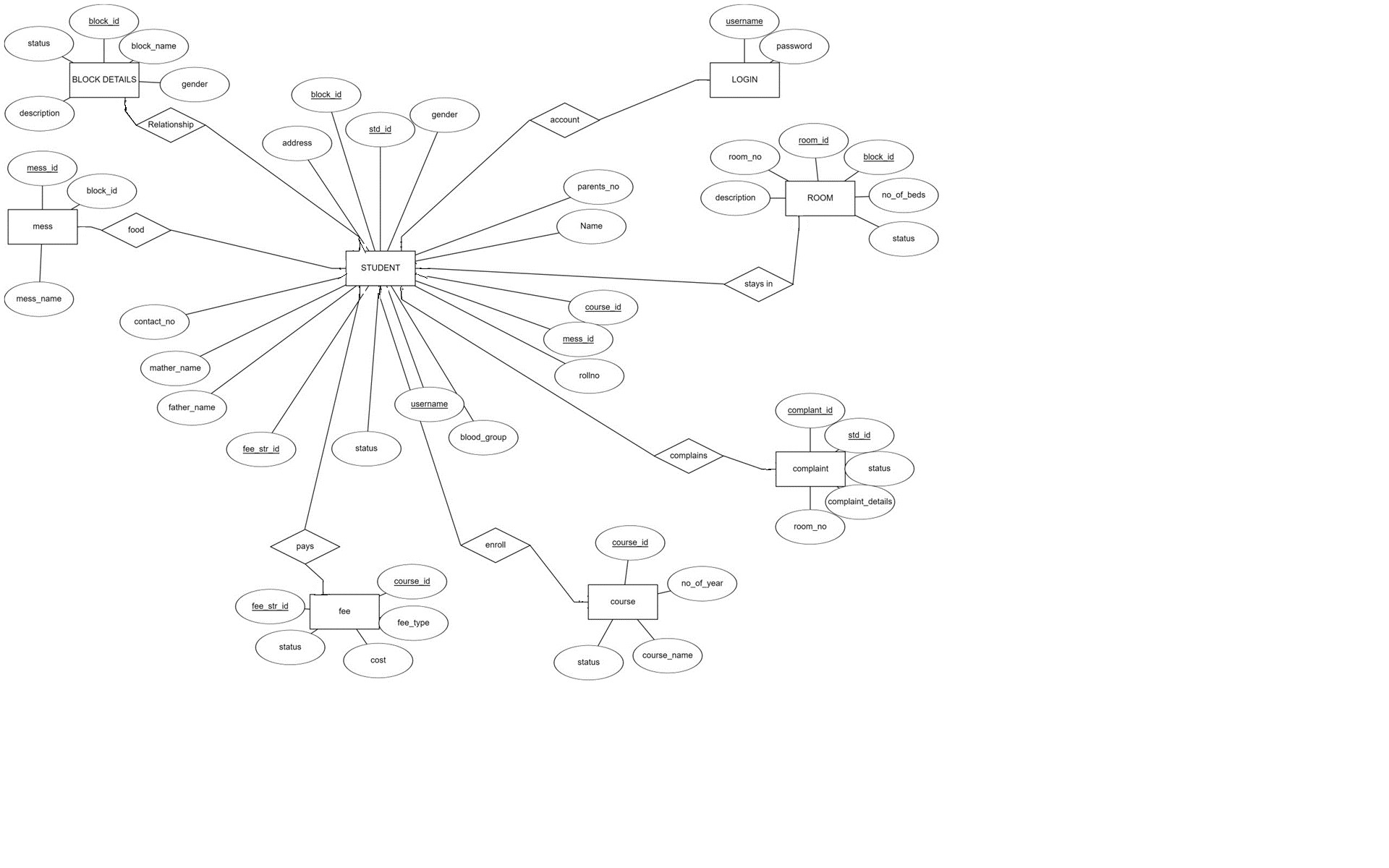
# OBJECTIVES OF PROJECT

This software product the hostel management to improve their services for all the students of the hostel. This also reduce the manual work of the persons in admin penal and the bundle of registers that were search when to find the information of a previous student, because through this system you can store the data of those students who had left the hostel .Through this you can check the personal profile of all the current students within few minutes the data base of the system will help you to check a particular one.

The system will help you to check the mess bills of every student and the student’s hostel dues. The students of the hostel will be recognized from the ID number allocated at the room rental time. In the last this system will improve the management work in the hostel.

* To automate each and every activity of the manual system, which increases its throughput
* To provide a quick response with very accurate information as and when required
* To make the present manual system more interactive, speedy and user friendly
* To avail any information, whatever and whenever needed.
* Reduce the cost of maintenance

**ER DIAGRAM**



The following are the tables that are involved in the proposed system

**LOGIN**

**ATTRIBUTE**

**DATATYPE**

**CONSTRAIN**

**T**

**Username**

Varchar(20)

primary key

**Password**

Varchar(20)

Not Null

## BLOCKDETAILS

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **ATTRIBUTES** | **DATATYPE** | **CONSTRAINT** |
| **block\_id** | int (10) | Primary key |
| **Block\_name** | varchar (20) | Not Null |
| **Gender** | varchar (10) | Not Null |
| **Description** | float (10) | Not Null |
| **Status** | varchar (20) | Not Null |

## STUDENT

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **ATTRIBUTES** | **DATATYPE** | **CONSTRAINTS** |
| **Stid** | int (10) | Primary key |
| **Courseid** | int (10) | Secondarykey |
| Block id | int (10) | Secondary key |
| Mess id | int (5) | Secondary key |
| **Fee\_str\_id** | int (10) | Secondary key |
| **Course\_id** | varchar (10) | Secondary key |
| Room\_id | Int (10) | Secondary key |
| **Name** | Varchar(20) | Not Null |
| **Rollno** | Varchar(20) | Not Null |
| **Father\_name** | Varchar(20) | Not Null |
| **Mother\_name** | Varchar(20) | Not Null |
| **Gender** | Varchar(10) | Not Null |
| **Address** | Varchar(40) | Not Null |
| **Contact\_no** | int (10) | Not Null |
| **Parents\_no** | int (10) | Not Null |
| **Blood\_group** | Varchar(5) | Not Null |
| username | Varchar(20) | Not Null |

## ROOM DETAILS

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **ATTRIBUTES** | **DATATYPE** | **CONSTRAINT** |
| **Room\_id** | int (10) | Primary key |
| **Block\_id** | int (10) | Secondary key |
| **Room\_no** | int (5) | Not Null |
| **No\_of\_beds** | int (10) | Not Null |
| **Description** | Varchar(20) | Not Null |
| **Status** | varchar (20) | Not Null |

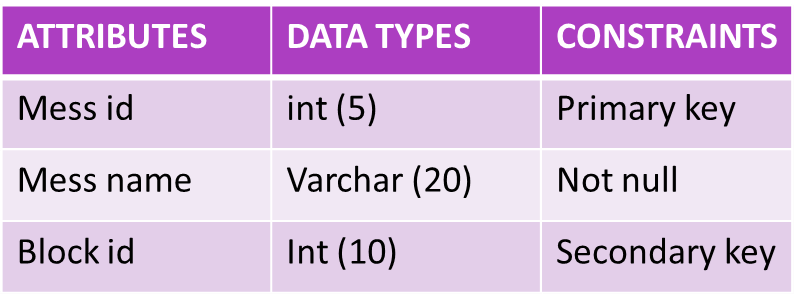
## FEEDETAILS

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **ATTRIBUTES** | **DATATYPE** | **CONSTRAINTS** |
| **Fee\_str\_id** | int (10) | Primary key |
| **course\_id** | varchar (10) | Secondary key |
| **Fee\_type** | varchar(20) | Not Null |
| **Cost** | float (10) | Not Null |
| **Status** | varchar(20) | Not Null |

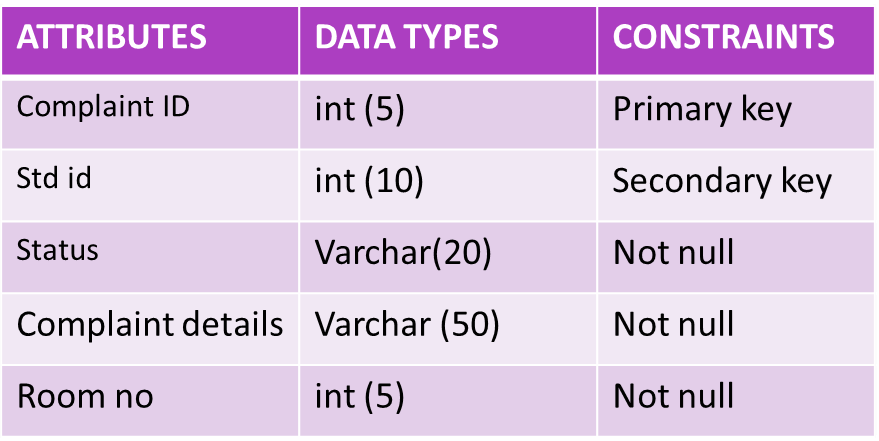
## COURSE

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTES** | **DATATYPE** | **CONSTRAINT** |
| **course\_id** | varchar (10) | Primary key |
| **course\_name** | varchar (20) | Not Null |
| **No\_of\_year** | int (10) | Not Null |
| **Status** | Varchar(20) | Not Null |

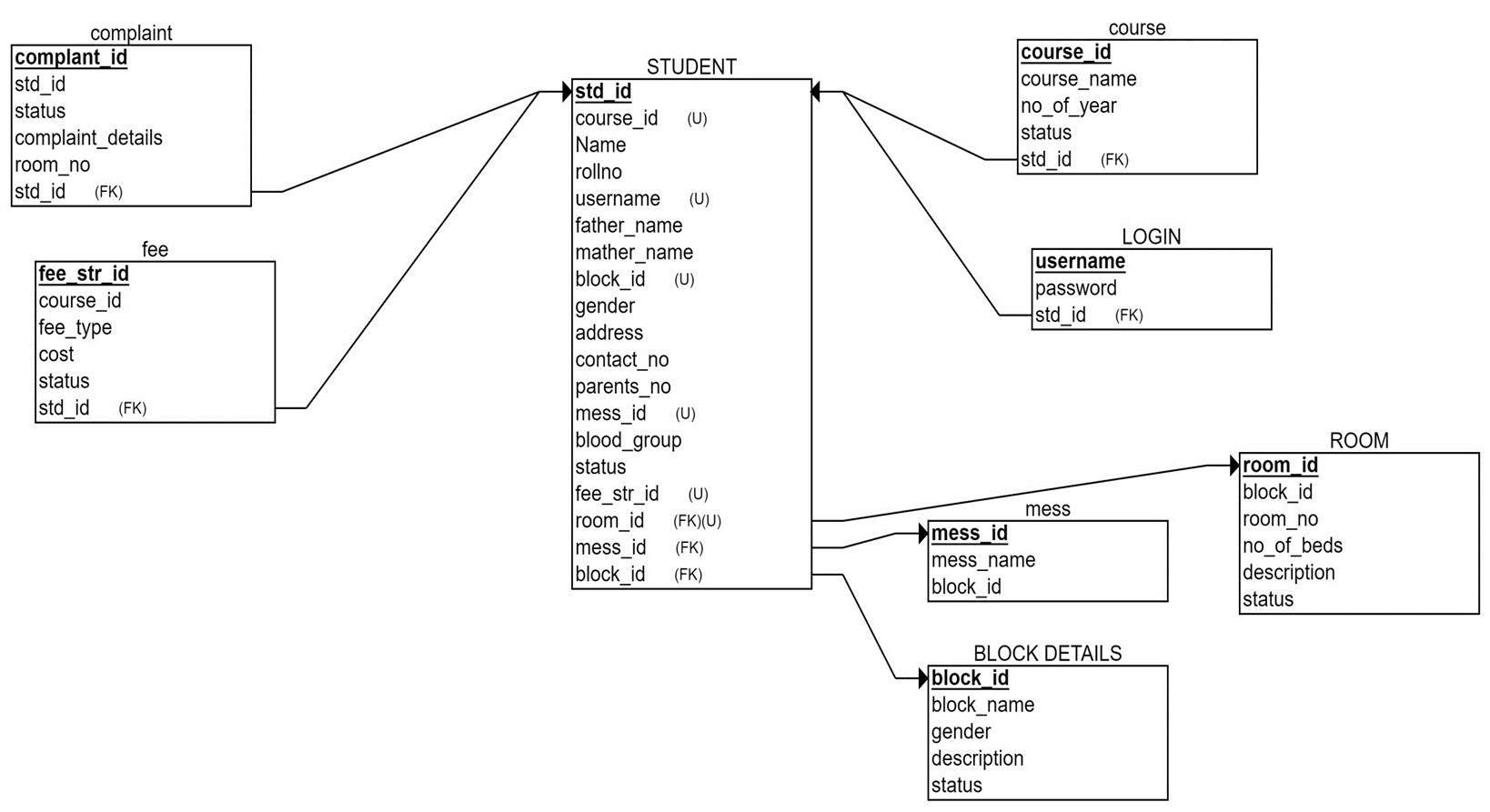
## MESS DETAILS



## COMPLAINT



## RELATIONAL SCHEMA



**Normalization:**

**1NF- First Normal Form**

If a relation contains a composite or multi-valued attribute, it violates the first normal form, or the relationship is in the first normal form if it does not contain any composite or multi-valued attribute. A relation is in its first normal form if every attribute in that relation is singled valued attribute. A table is in 1 NF iff:

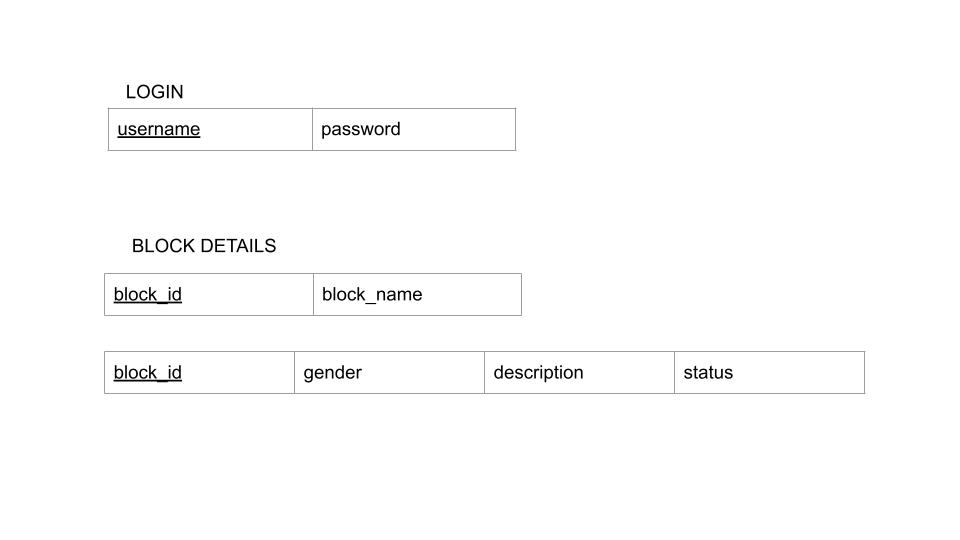
1. There are only Single Valued Attributes.  
2. Attribute Domain does not change.  
3. There is a unique name for every Attribute/Column. 4. The order in which data is stored does not matter.

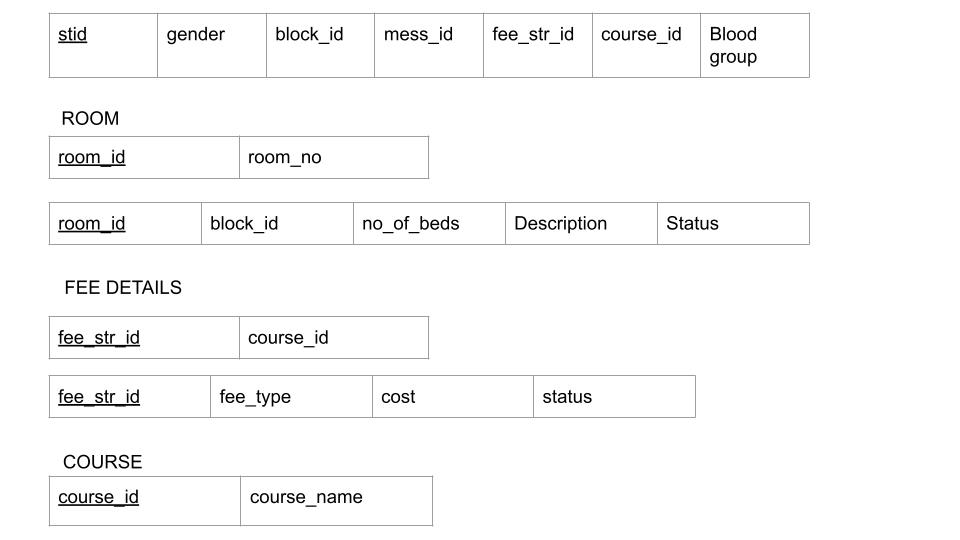
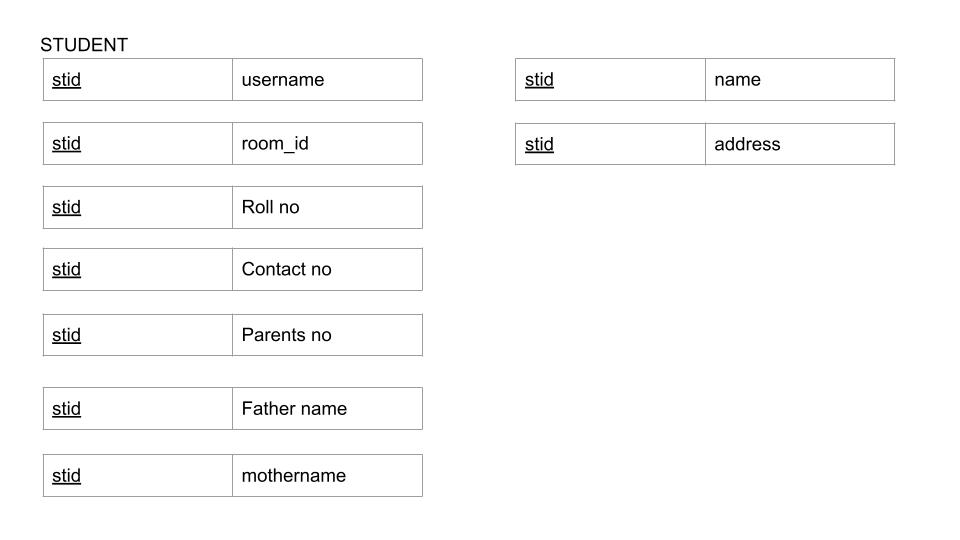
There are no multivalued attributes in the tables so the tables are already in 1NF form.

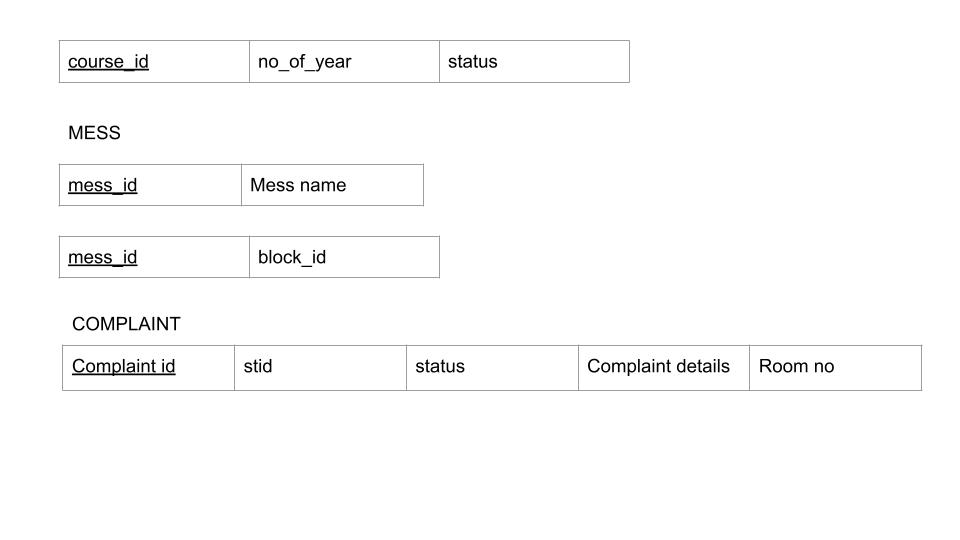
**2NF- Second Normal Form**

To be in the second normal form, a relation must be in the first normal form and the relation must not contain any partial dependency. A relation is in 2NF if it has No Partial Dependency, i.e., no non-prime attribute (attributes that are not part of any candidate key) is dependent on any proper subset of any candidate key of the table. Student Table.

Based on this the tables should be represented as:



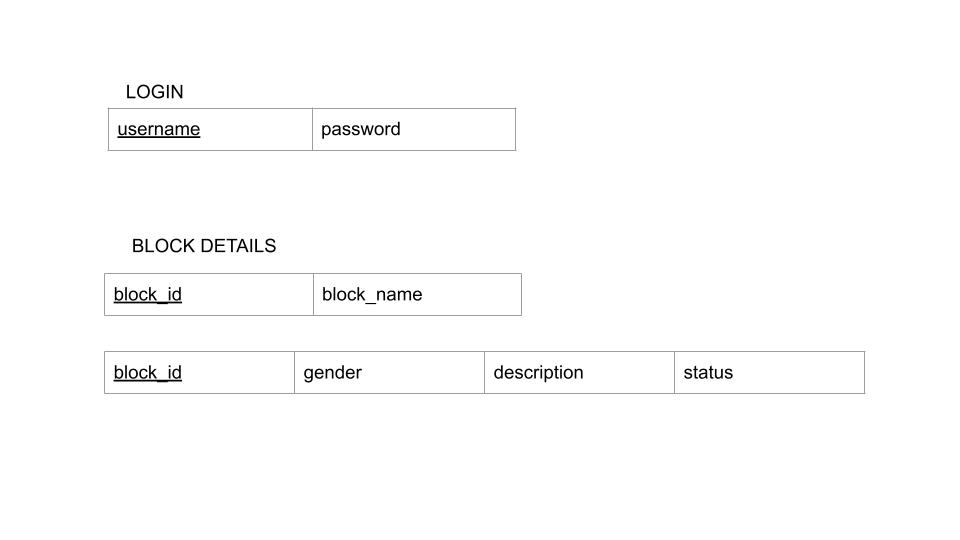


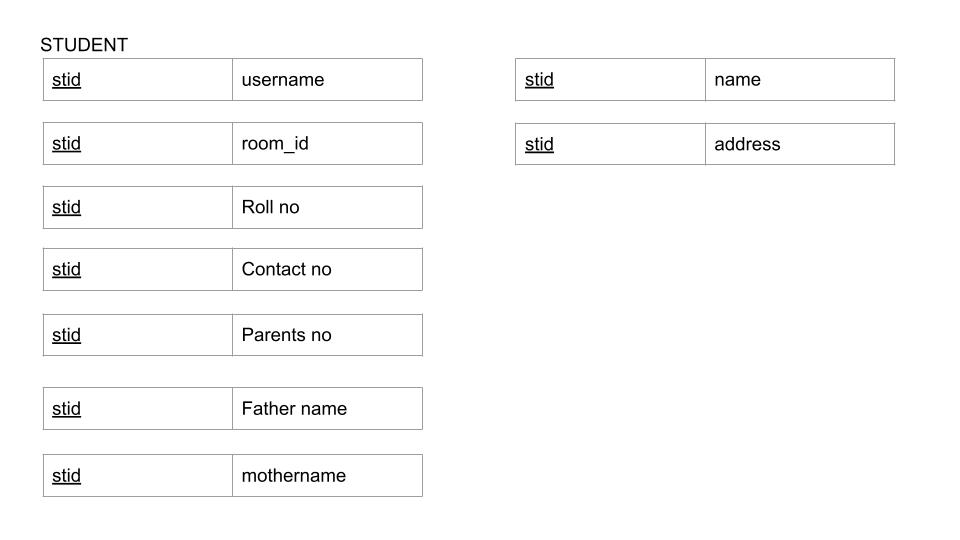


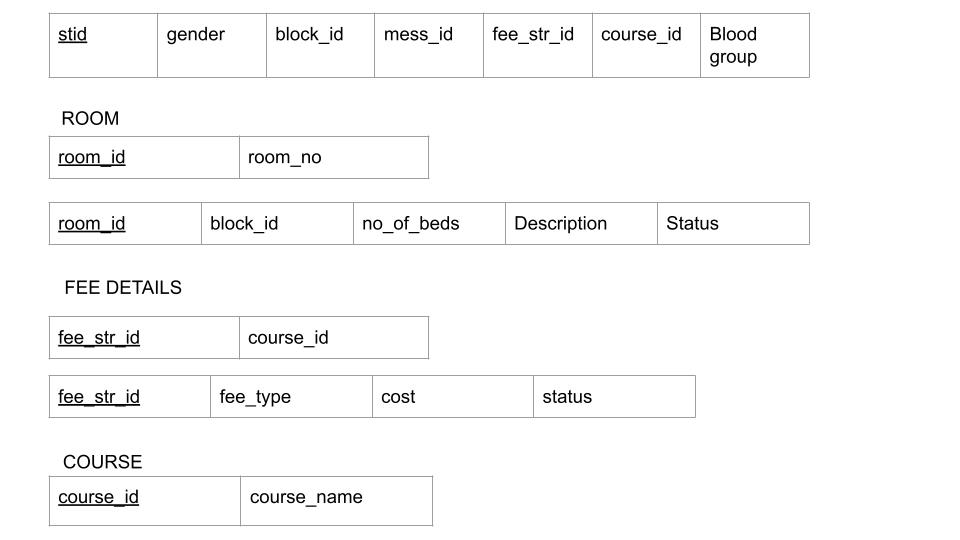
**3NF- Third Normal Form**

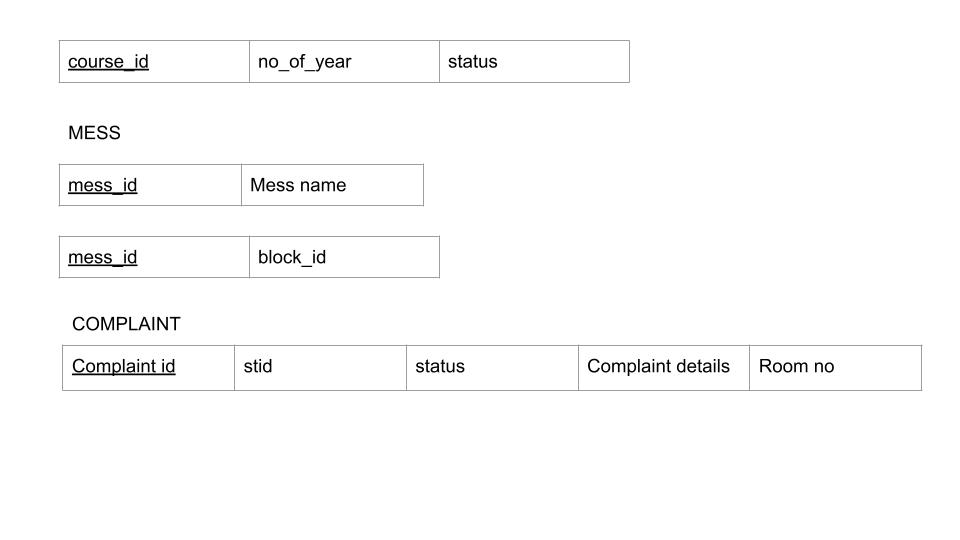
A relation that is in First and Second Normal Form and in which no non-primary key attribute is transitively dependent on the primary key, then it is in Third Normal Form (3NF). If A->B and B->C are two FDs then A->C is called transitive dependency.

Based on this the tables should be represented as:



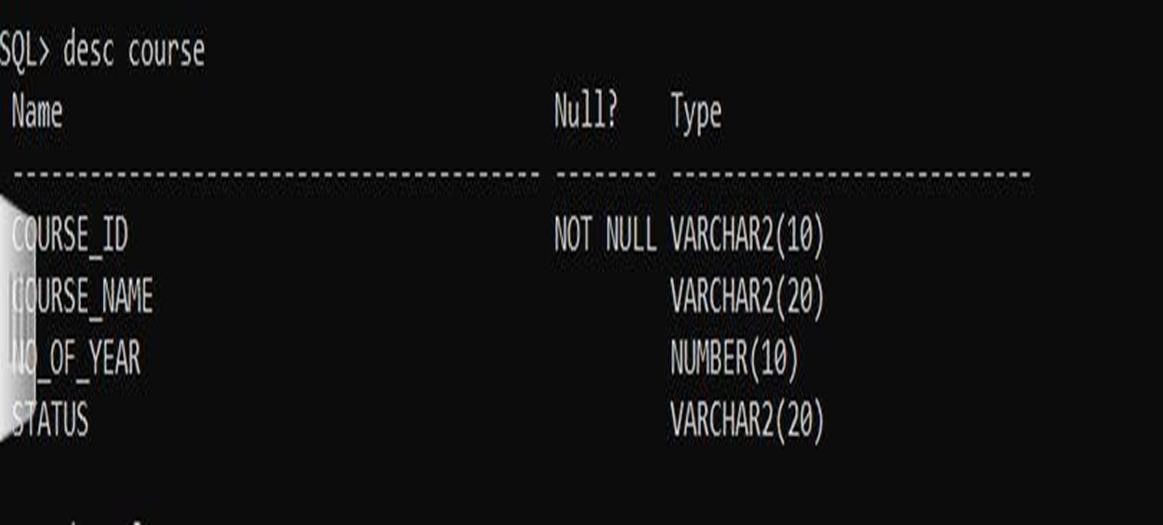




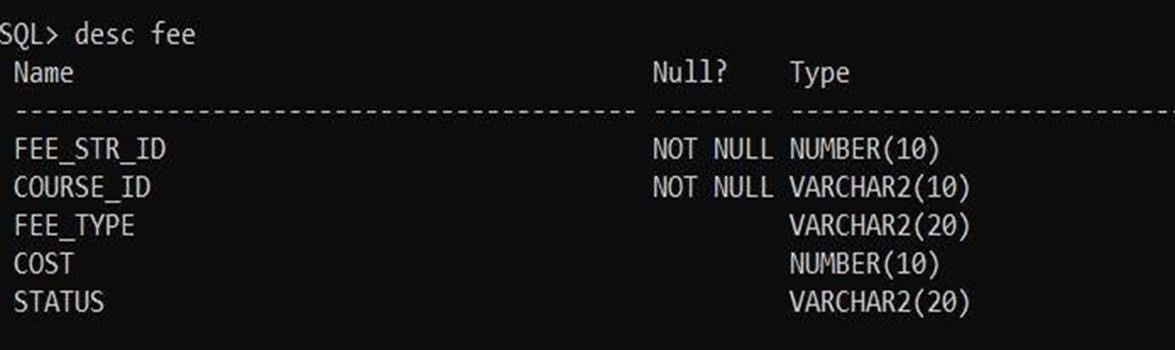


## TABLE CREATION

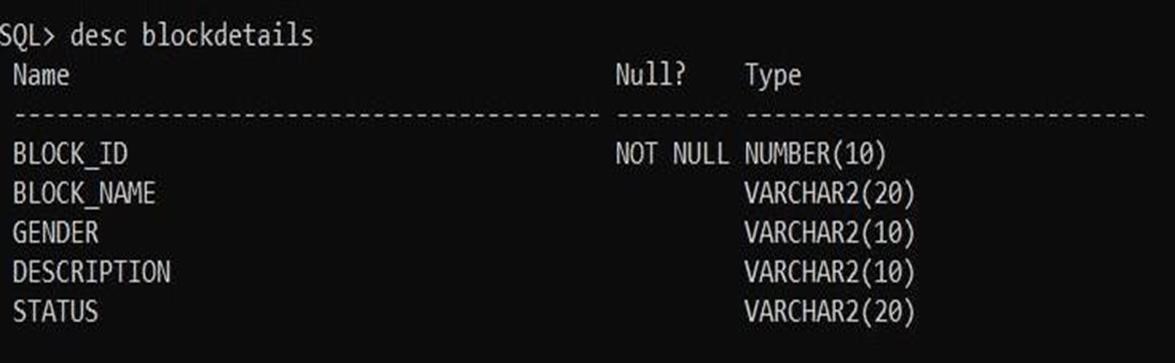
**COURSE TABLE**



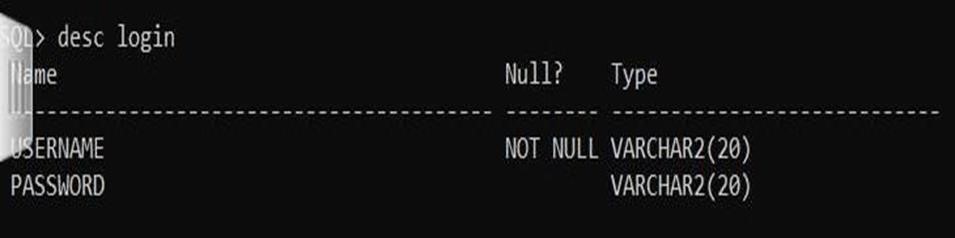
**FEE TABLE**



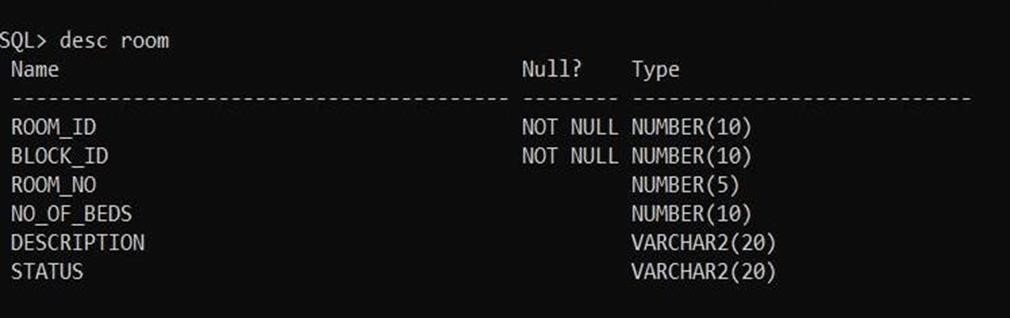
**BLOCKDETAILS TABLE:**



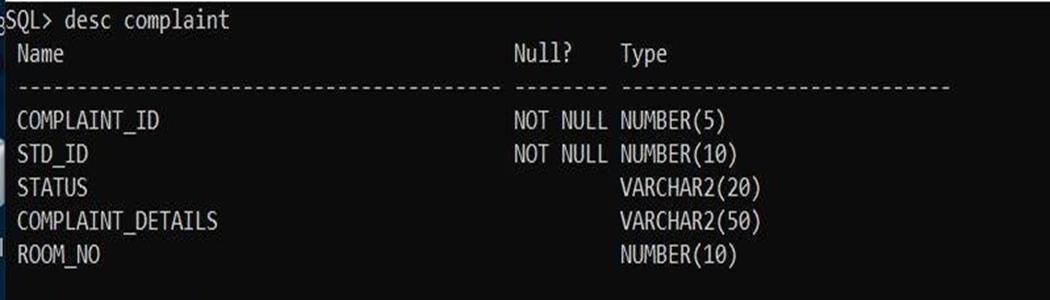
**LOGIN TABLE:**



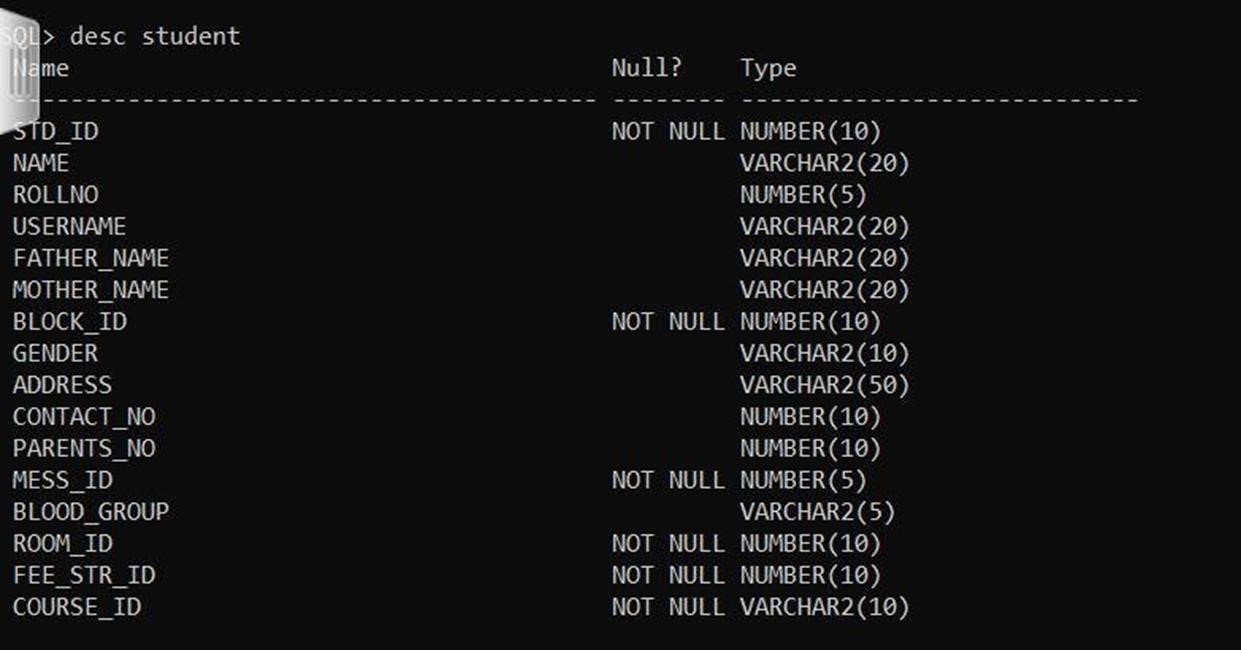
**ROOM TABLE:**



**COMPLAINT** **TABLE:**

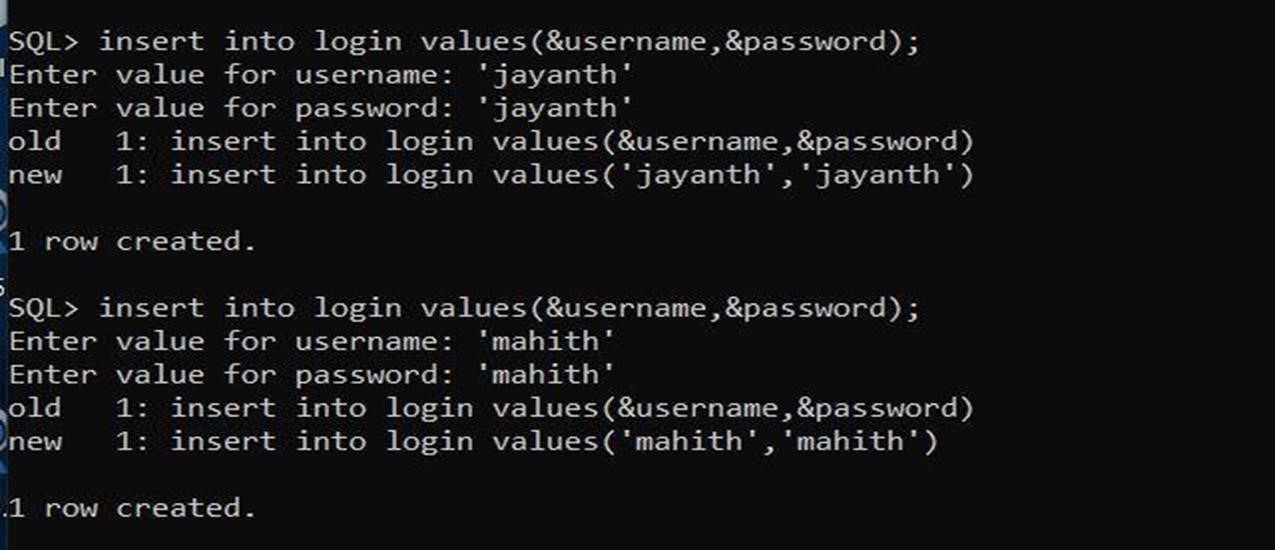


**STUDENT** **TABLE:**



**INTERACTIVE DATA ENTRY**

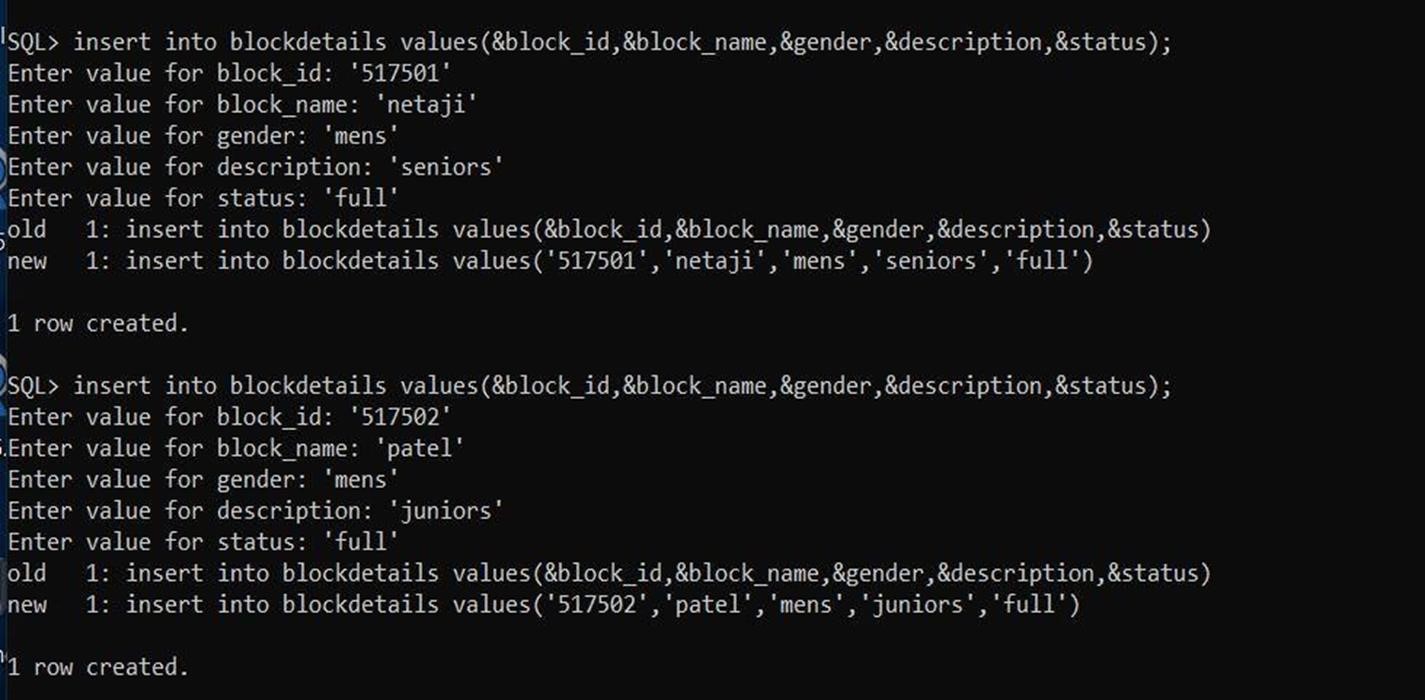
## Entering values into login table incorporating integrity constraints



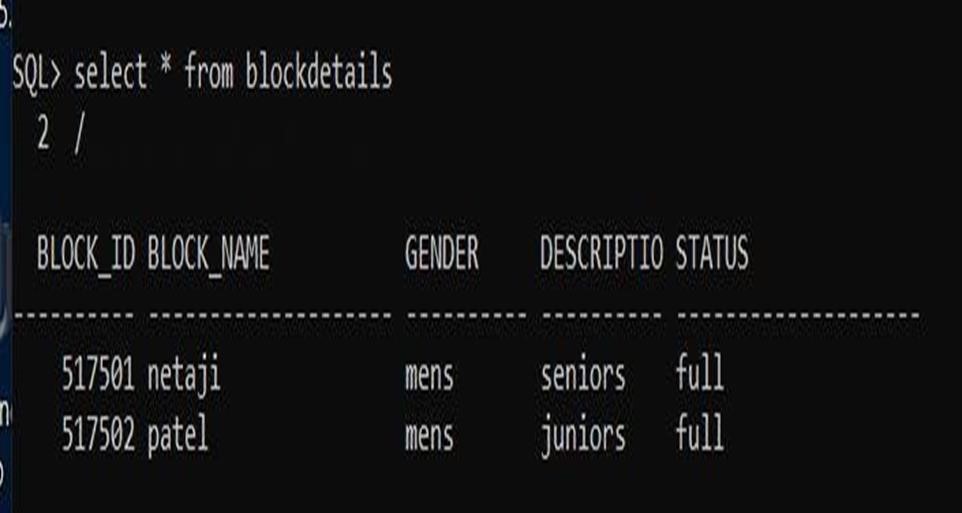
## Getting info in login table



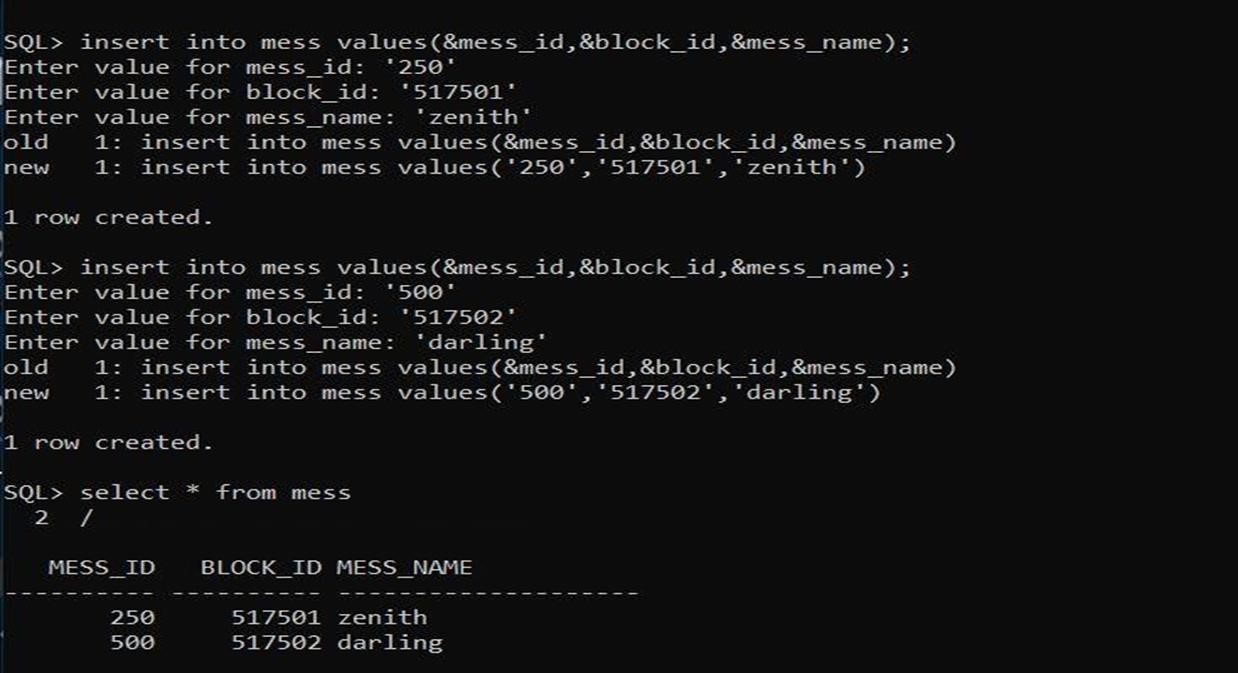
## Entering values into block details table incorporating integrity constraints



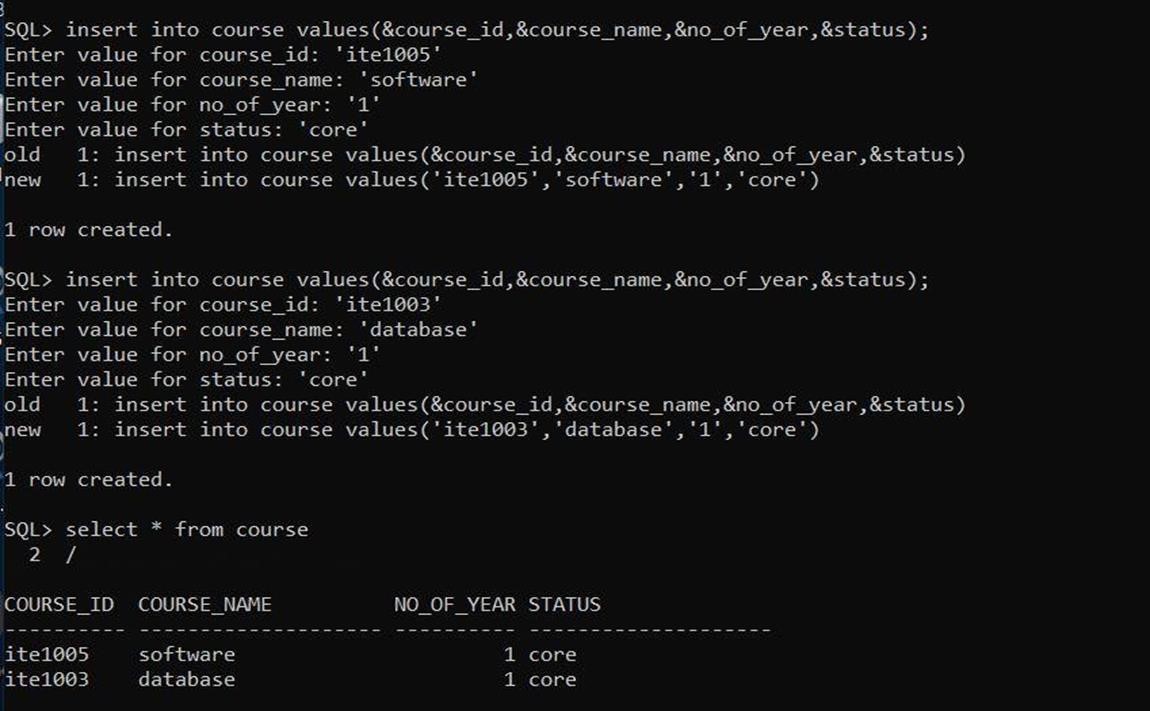
## Getting info in block details table



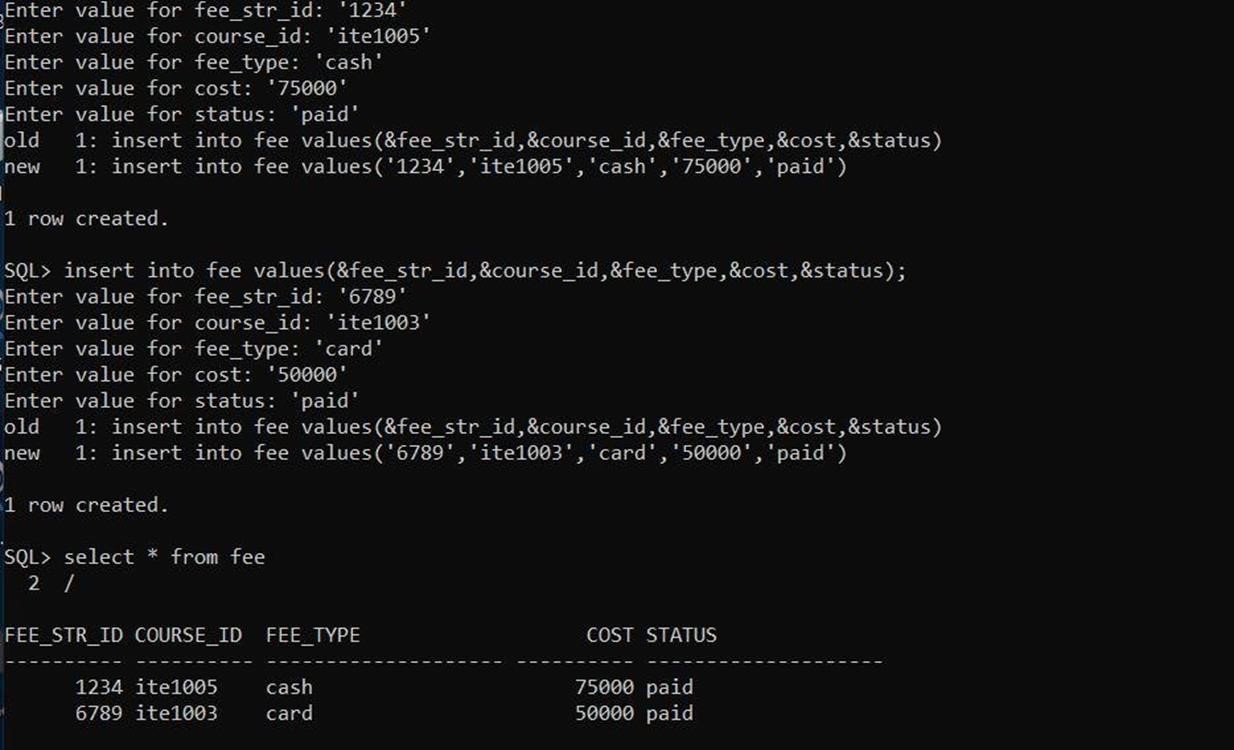
## Inserting and getting info from mess table



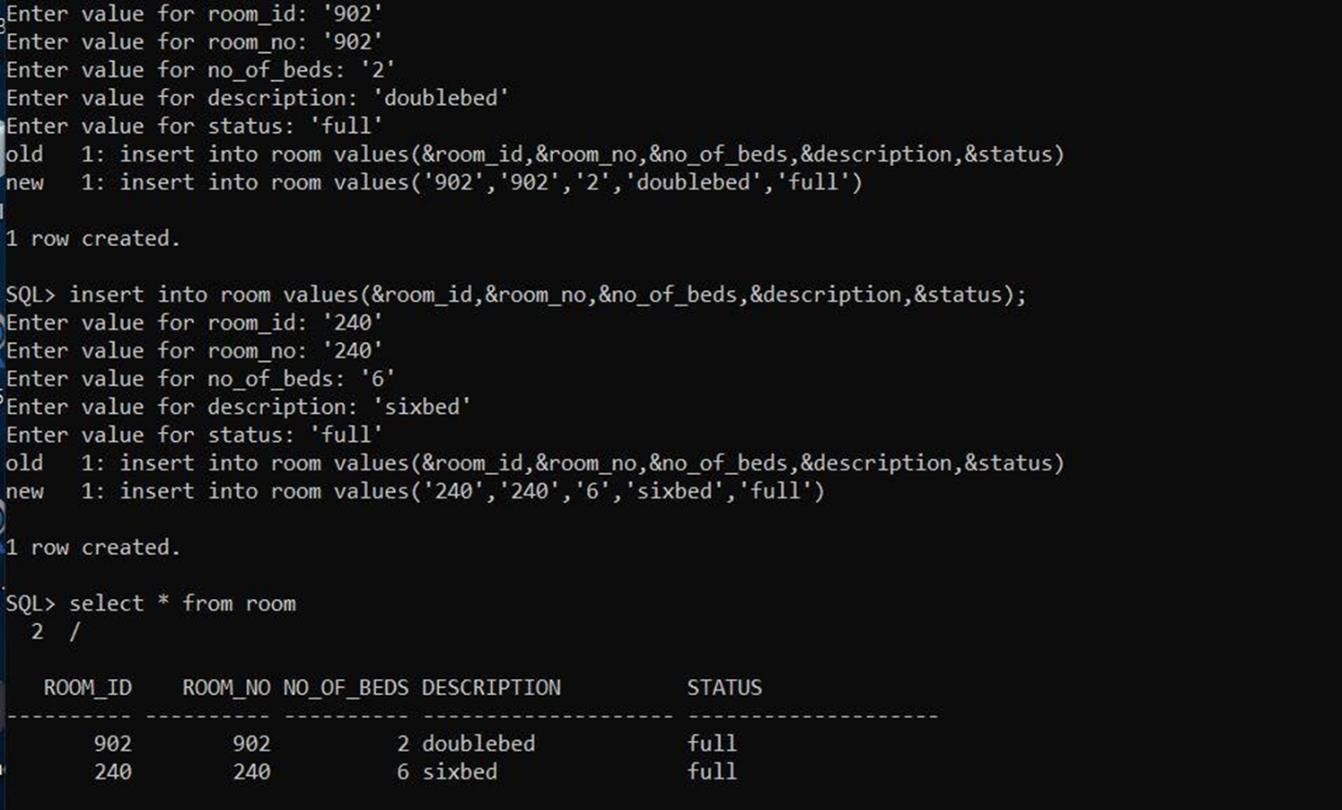
## Inserting and getting info from course table



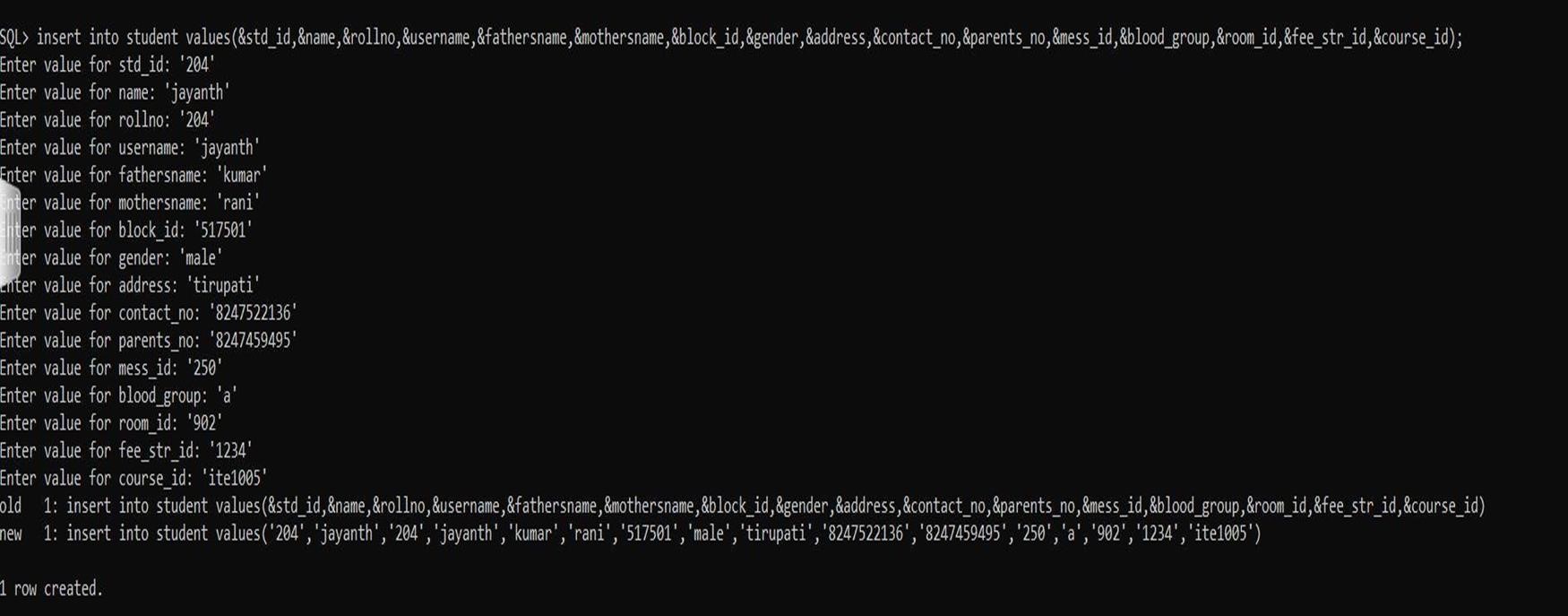
## Inserting and getting info from fee table



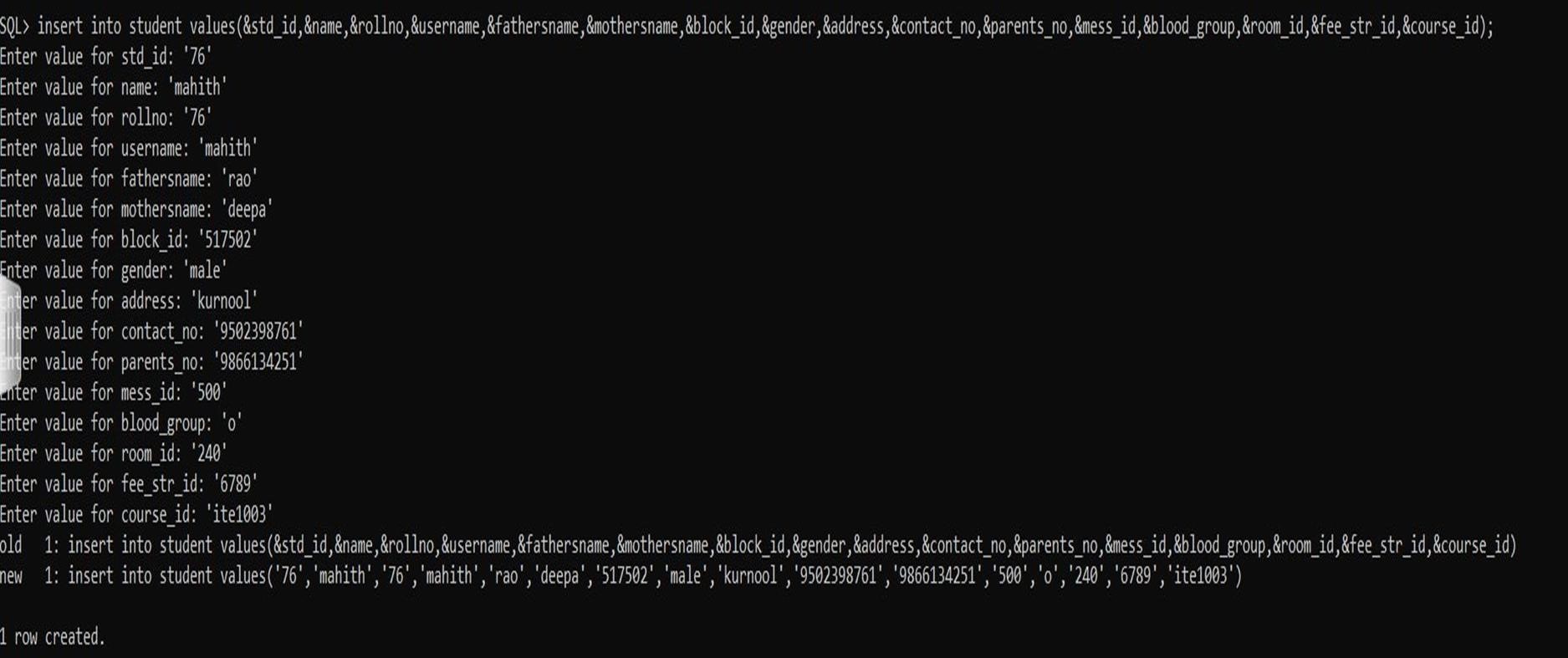
## Inserting and getting info from room table



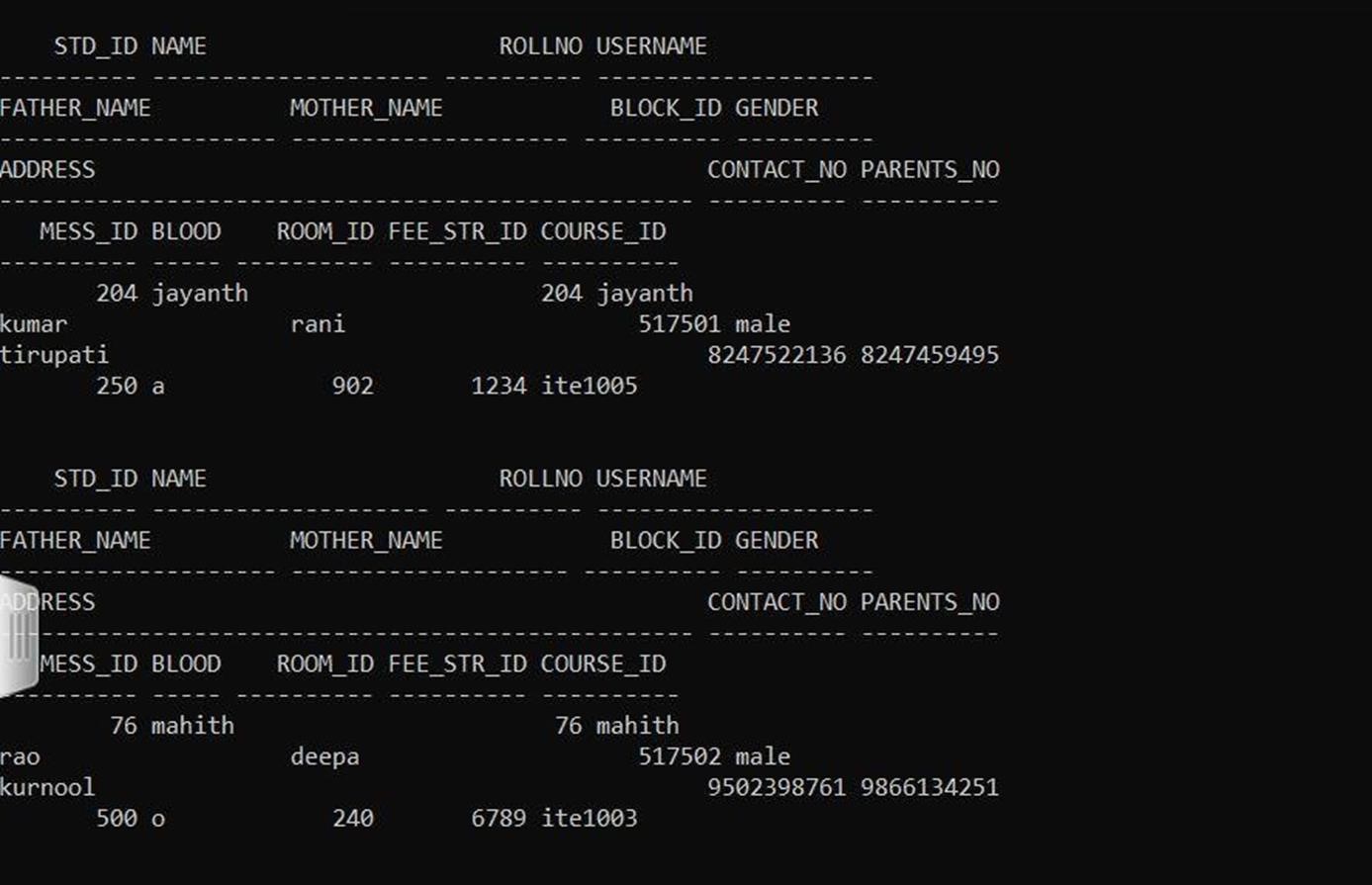
## Inserting and getting info from student table



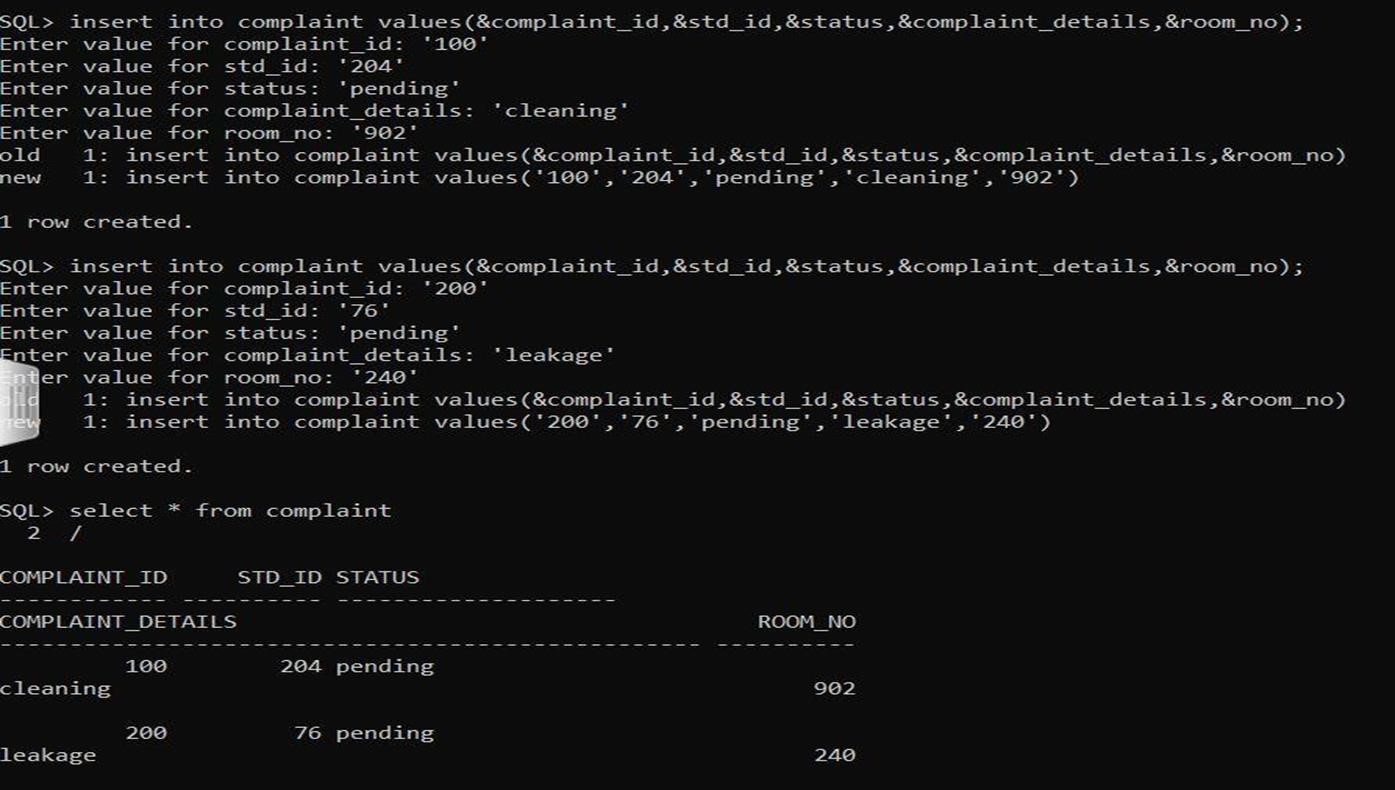
## Inserting and getting info from student table cont



## Inserting and getting info from student table content

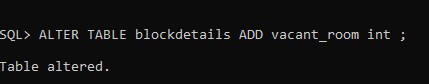


## Inserting and getting info from complaints table

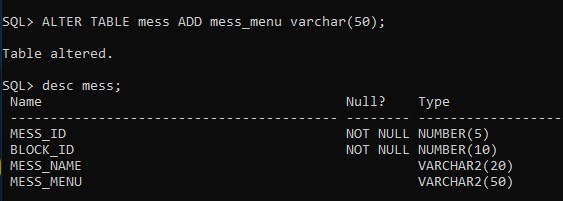


**QUERIES RELATED TO INSERT:**

## ➔INSERT NO OF VACANT ROOMS

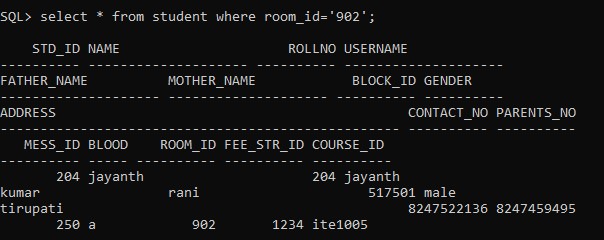


## ➔INSERT MESS MENU INTO MESS

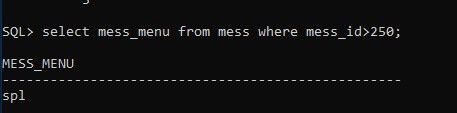


**QUERIES RELATED TO RETRIVEL OF DATA:**

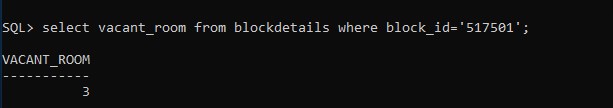
## ➔TO GET THE DATA OF STUDENT OF PARTICULAR ROOMID



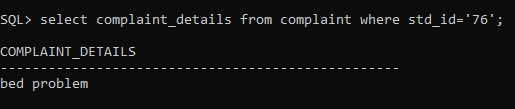
## ➔TO GET THE DETAILS OF PARTICULAR MESS MENU



## ➔TO KNOW THE NUMBER OF VACANT ROOMS PRESENT IN A PARTICULAR BLOCK

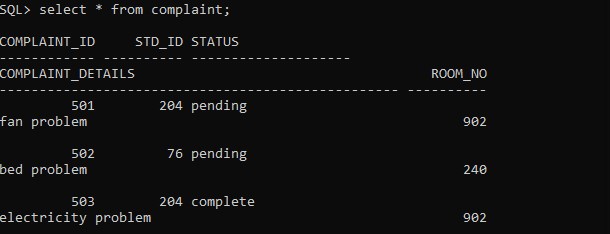


## ➔TO SEE THE COMPLAINT OF A PARTICULAR STUDENT WRT STDID

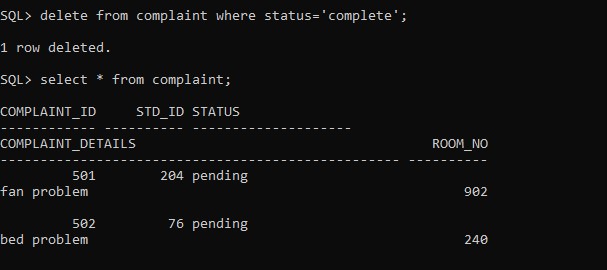


**QUERIES RELATED TO DELETION OF DATA**

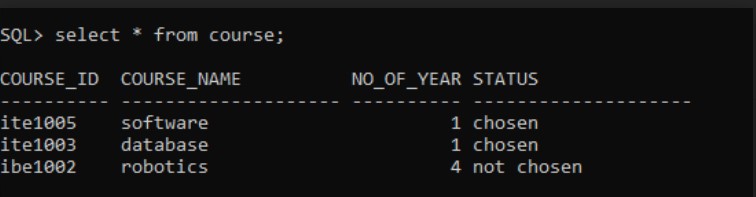
## ➔TO DELETE THE COMPLAINT WHOSE STATUS IS COMPLETED



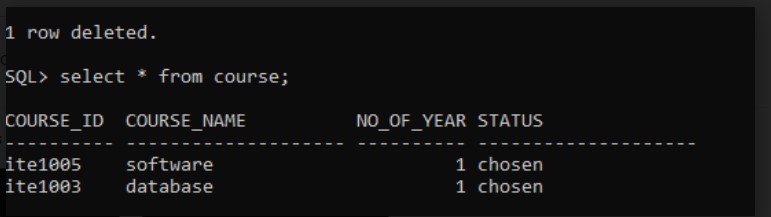
## AFTER DELETION OF COMPLAINT WITH STATUS COMPLETED



## ➔TO DELETE A PARTICULAR COURSE WHICH HAS NOT BEEN SELECTED BY ANY STUDENT BEFORE DELETION

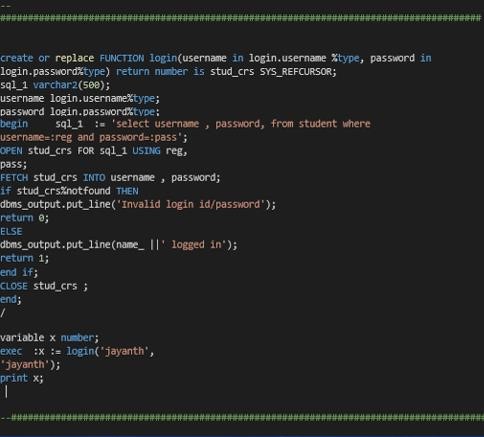


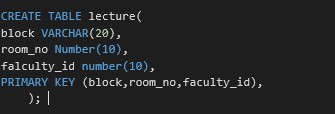
**AFTER DELETING COURSE WHICH IS NOT SELECTED**



**PL SQL FUNCTION:**

**Login:** to authenticate students with their login regno/password

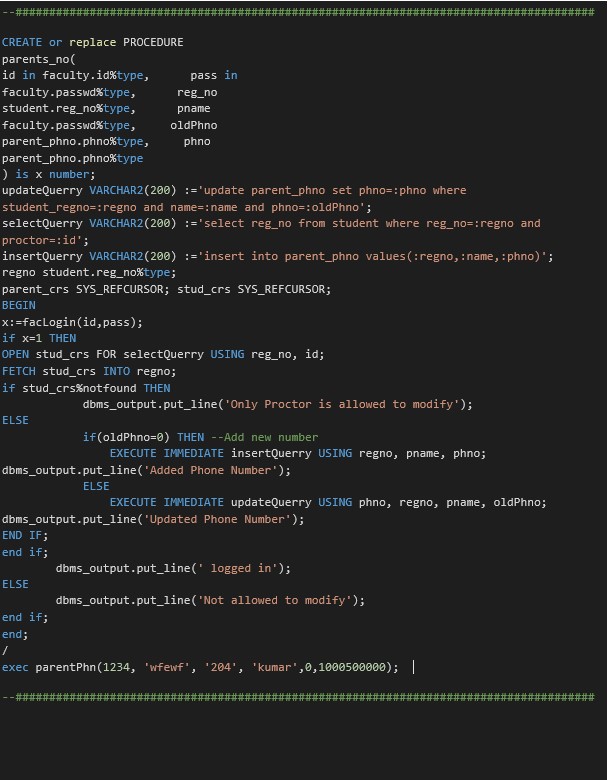




**PL SQL PROCEDURE**

**parentPhn:** to add/modify student’s parent’s phno. This validates and verify

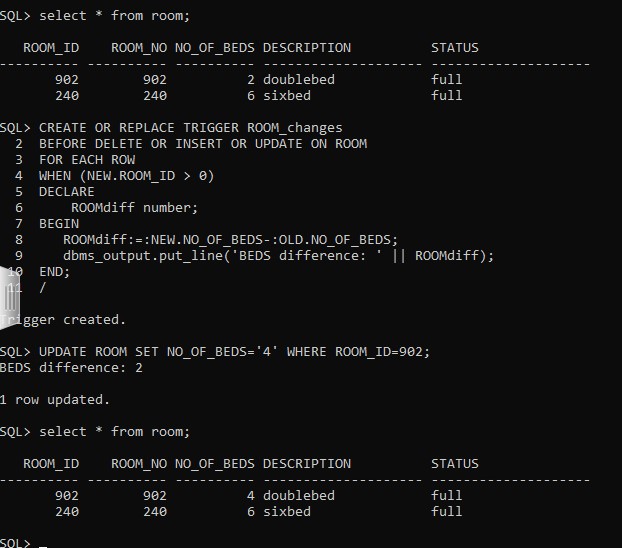
that the changes can be done by the faculty only.



**TRIGGERS:**

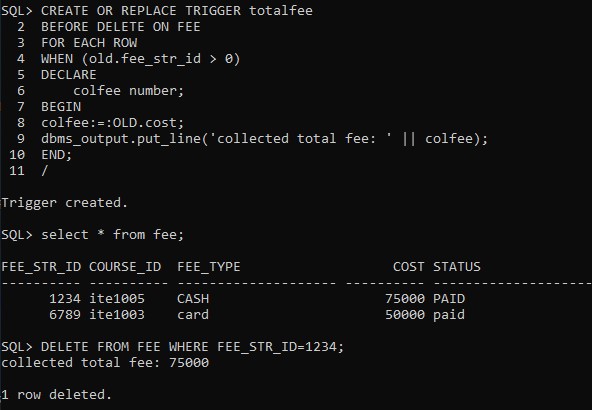
**TO FIND THE DIFFERENCE BETWEEN NUMBER OF BEDS**

**AFTER UPDATING VALUE**



**TO KNOW THE FEES THAT IS DELETED WRT**

**PARTICULAR FEE\_ID**



**CONCLUSION:**

The project is based on the requirement specification of the user and the analysis of the existing system, with flexibility for future enhancement.

The expanded functionality of today’s software requires an appropriate approach towards software development. This hostel management software is designed for people who want to manage various activities in the hostel.

For the past few years the numbers of educational institutions are increasing rapidly. Thereby the numbers of hostels are also increasing for the accommodation of the students studying in this institution.

And hence there is a lot of strain on the person who are running the hostel and 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 manually.

Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system which is more user friendly and more GUI oriented.