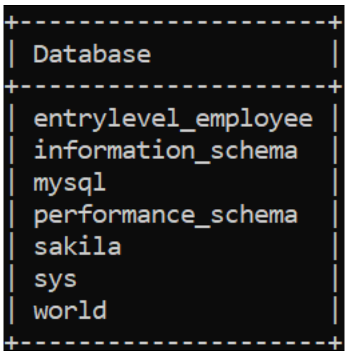
Exercise 1

Problem Statement:

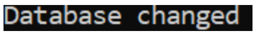
You have to create all tables whose structure is as mentioned below:

```
CREATE DATABASE entrylevel_employee;
SHOW databases;
```



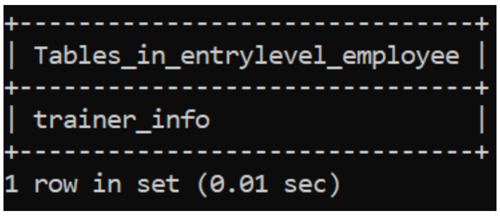
```
+---------------------+
| Database            |
+---------------------+
| entrylevel_employee |
| information_schema  |
| mysql               |
| performance_schema  |
| sakila              |
| sys                 |
| world               |
+---------------------+
```

```
USE entrylevel_employee;
```



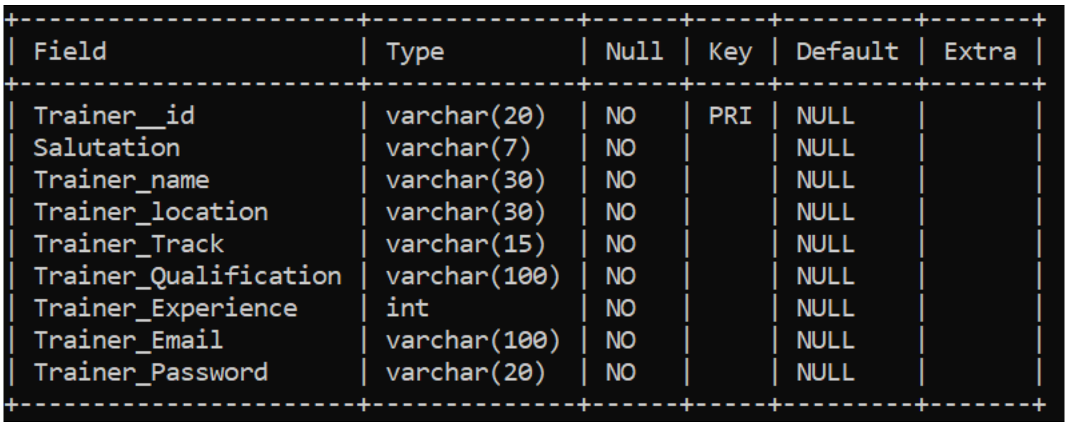
```
Database changed
```

```
CREATE table Trainer_info(
    -> Trainer__id VARCHAR(20) NOT NULL PRIMARY KEY,
    -> Salutation VARCHAR(7) NOT NULL,
    -> Trainer_name VARCHAR(30) NOT NULL,
    -> Trainer_location VARCHAR(30) NOT NULL,
    -> Trainer_Track VARCHAR(15) NOT NULL,
    -> Trainer_Qualification VARCHAR(100) NOT NULL,
    -> Trainer_Experience INT(11) NOT NULL,
    -> Trainer_Email VARCHAR(100) NOT NULL,
    -> Trainer_Password VARCHAR(20) NOT NULL
    -> );
SHOW tables;
```



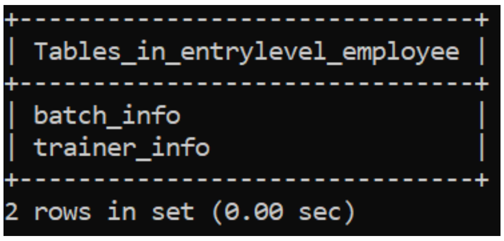
```
+-------------------------------+
| Tables_in_entrylevel_employee |
+-------------------------------+
| trainer_info                  |
+-------------------------------+
1 row in set (0.01 sec)
```

```
DESC Trainer_Info;
```



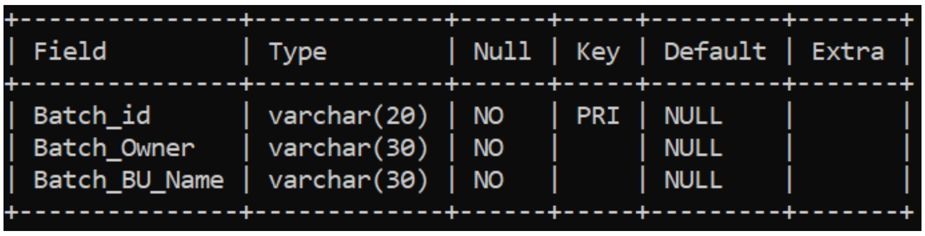
```
+-----------------------+--------------+------+-----+---------+-------+
| Field                 | Type         | Null | Key | Default | Extra |
+-----------------------+--------------+------+-----+---------+-------+
| Trainer__id           | varchar(20)  | NO   | PRI | NULL    |       |
| Salutation            | varchar(7)   | NO   |     | NULL    |       |
| Trainer_name          | varchar(30)  | NO   |     | NULL    |       |
| Trainer_location      | varchar(30)  | NO   |     | NULL    |       |
| Trainer_Track         | varchar(15)  | NO   |     | NULL    |       |
| Trainer_Qualification | varchar(100) | NO   |     | NULL    |       |
| Trainer_Experience    | int          | NO   |     | NULL    |       |
| Trainer_Email         | varchar(100) | NO   |     | NULL    |       |
| Trainer_Password      | varchar(20)  | NO   |     | NULL    |       |
+-----------------------+--------------+------+-----+---------+-------+
```

```
CREATE table Batch_info(
    -> Batch_id VARCHAR(20) NOT NULL PRIMARY KEY,
    -> Batch_Owner VARCHAR(30) NOT NULL,
    -> Batch_BU_Name VARCHAR(30) NOT NULL
    -> );
```



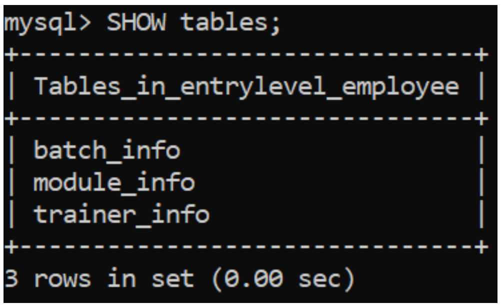
```
+------------------------------+
| Tables_in_entrylevel_employee |
+------------------------------+
| batch_info                   |
| trainer_info                 |
+------------------------------+
2 rows in set (0.00 sec)
```

DESC Batch_info;



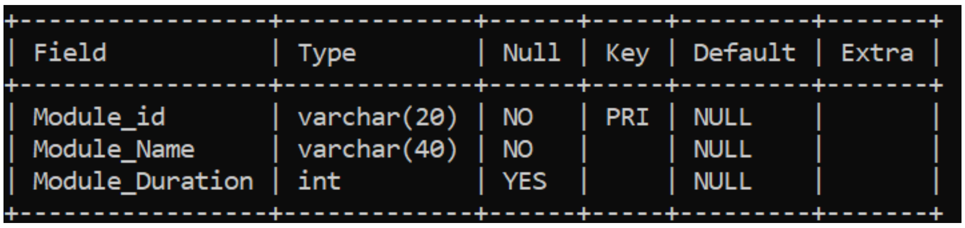
```
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| Batch_id      | varchar(20) | NO   | PRI | NULL    |       |
| Batch_Owner   | varchar(30) | NO   |     | NULL    |       |
| Batch_BU_Name | varchar(30) | NO   |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
```

```
CREATE table Module_Info(
    -> Module_id VARCHAR(20) NOT NULL PRIMARY KEY,
    -> Module_Name VARCHAR(40) NOT NULL,
    -> Module_Duration INT(11)
    -> );
```



```
mysql> SHOW tables;
+------------------------------+
| Tables_in_entrylevel_employee |
+------------------------------+
| batch_info                   |
| module_info                  |
| trainer_info                 |
+------------------------------+
3 rows in set (0.00 sec)
```

DESC Module_info;

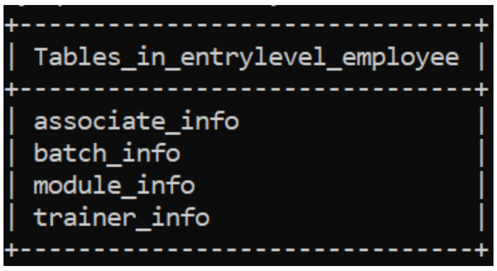


```
+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| Module_id       | varchar(20) | NO   | PRI | NULL    |       |
| Module_Name     | varchar(40) | NO   |     | NULL    |       |
| Module_Duration | int         | YES  |     | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
```

```
CREATE table Associate_Info(
    -> Associate_Id VARCHAR(20) NOT NULL PRIMARY KEY,
    -> Salutation VARCHAR(7) NOT NULL,
    -> Associate_Name VARCHAR(30) NOT NULL,
    -> Associate_Location VARCHAR(30) NOT NULL,
    -> Associate_Track VARCHAR(15) NOT NULL,
    -> Associate_Qualification VARCHAR(200) NOT NULL,
    -> Associate_Email VARCHAR(100) NOT NULL,
    -> Associate_Password VARCHAR(20) NOT NULL
    -> )
    -> ;
```

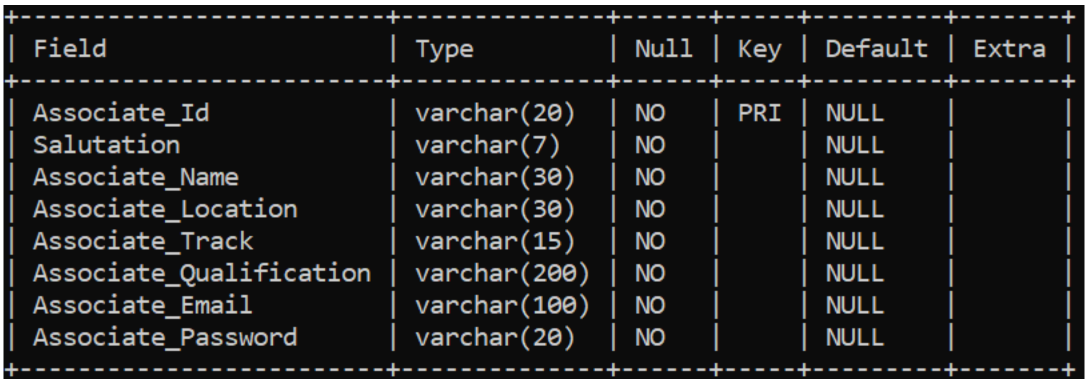
Krithika Devi Chandran

SHOW tables;



```
+-----------------------------+
| Tables_in_entrylevel_employee |
+-----------------------------+
| associate_info              |
| batch_info                  |
| module_info                 |
| trainer_info                |
+-----------------------------+
```

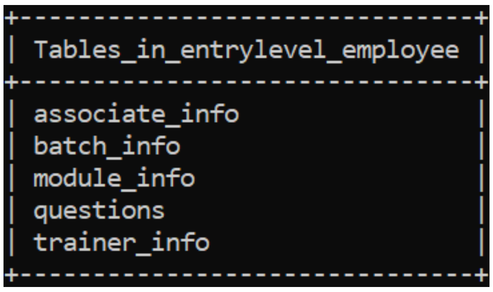
DESC Associate_Info;



```
+-------------------------+--------------+------+-----+---------+-------+
| Field                   | Type         | Null | Key | Default | Extra |
+-------------------------+--------------+------+-----+---------+-------+
| Associate_Id            | varchar(20)  | NO   | PRI | NULL    |       |
| Salutation              | varchar(7)   | NO   |     | NULL    |       |
| Associate_Name          | varchar(30)  | NO   |     | NULL    |       |
| Associate_Location      | varchar(30)  | NO   |     | NULL    |       |
| Associate_Track         | varchar(15)  | NO   |     | NULL    |       |
| Associate_Qualification | varchar(200) | NO   |     | NULL    |       |
| Associate_Email         | varchar(100) | NO   |     | NULL    |       |
| Associate_Password      | varchar(20)  | NO   |     | NULL    |       |
+-------------------------+--------------+------+-----+---------+-------+
```

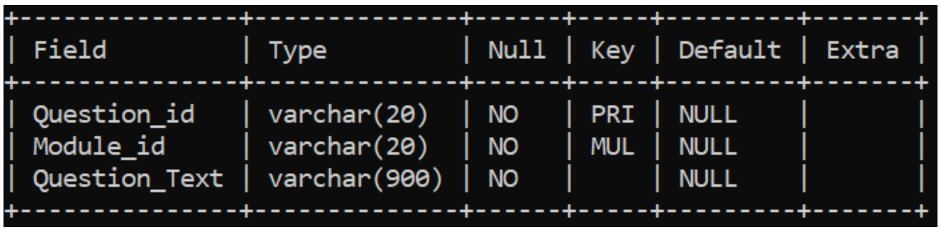
```
CREATE table Questions(
    -> Question_id VARCHAR(20) NOT NULL PRIMARY KEY,
    -> Module_id VARCHAR(20) NOT NULL,
    -> Question_Text VARCHAR(900) NOT NULL,
    -> FOREIGN KEY (Module_id) REFERENCES Module_Info(Module_id)
    -> );
```

SHOW tables;



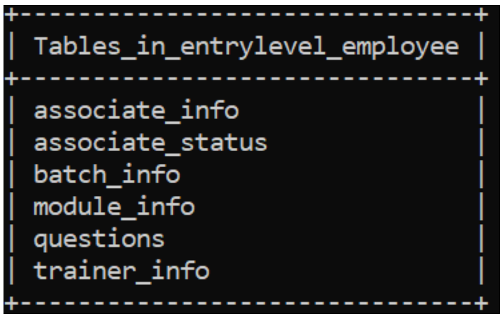
```
+-----------------------------+
| Tables_in_entrylevel_employee |
+-----------------------------+
| associate_info              |
| batch_info                  |
| module_info                 |
| questions                   |
| trainer_info                |
+-----------------------------+
```

DESC Questions;



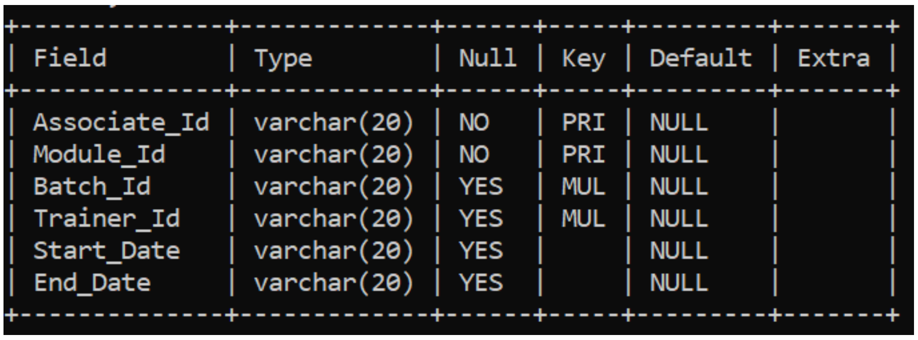
```
+---------------+--------------+------+-----+---------+-------+
| Field         | Type         | Null | Key | Default | Extra |
+---------------+--------------+------+-----+---------+-------+
| Question_id   | varchar(20)  | NO   | PRI | NULL    |       |
| Module_id     | varchar(20)  | NO   | MUL | NULL    |       |
| Question_Text | varchar(900) | NO   |     | NULL    |       |
+---------------+--------------+------+-----+---------+-------+
```

```
CREATE table Associate_Status(
    ->     Associate_Id VARCHAR(20),
    ->     Module_Id VARCHAR(20),
    ->     Batch_Id VARCHAR(20),
    ->     Trainer_Id VARCHAR(20),
    ->     Start_Date VARCHAR(20),
    ->     End_Date VARCHAR(20),
    ->     PRIMARY KEY(Associate_Id,Module_Id),
    ->    FOREIGN KEY (Batch_Id) REFERENCES Batch_Info(Batch_Id),
    ->    FOREIGN KEY (Trainer_Id) REFERENCES Trainer_Info(Trainer__Id)
    ->    );
SHOW tables;
```



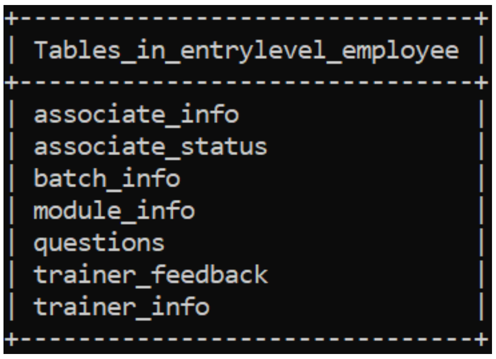
```
+-----------------------------+
| Tables_in_entrylevel_employee |
+-----------------------------+
| associate_info              |
| associate_status            |
| batch_info                  |
| module_info                 |
| questions                   |
| trainer_info                |
+-----------------------------+
```

DESC Associate_Status;



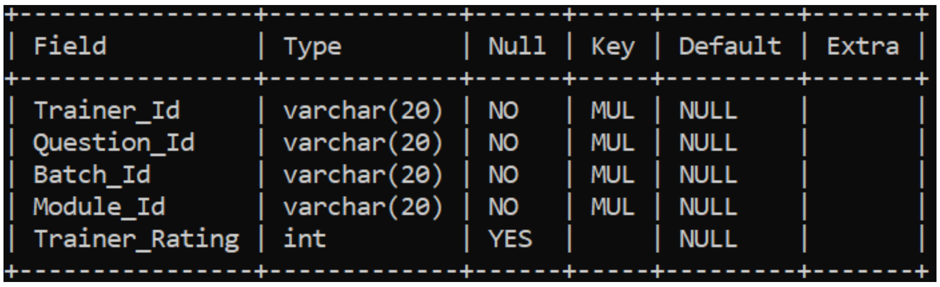
```
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| Associate_Id | varchar(20) | NO   | PRI | NULL    |       |
| Module_Id    | varchar(20) | NO   | PRI | NULL    |       |
| Batch_Id     | varchar(20) | YES  | MUL | NULL    |       |
| Trainer_Id   | varchar(20) | YES  | MUL | NULL    |       |
| Start_Date   | varchar(20) | YES  |     | NULL    |       |
| End_Date     | varchar(20) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
```

```
CREATE table Trainer_Feedback(
  -> Trainer_Id VARCHAR(20) NOT NULL,
  -> Question_Id VARCHAR(20) NOT NULL,
  -> Batch_Id VARCHAR(20) NOT NULL,
  -> Module_Id VARCHAR(20) NOT NULL,
  -> Trainer_Rating INT(11),
  -> FOREIGN KEY (Trainer_Id) REFERENCES Trainer_Info(Trainer__Id),
  -> FOREIGN KEY (Question_id) REFERENCES Questions(Question_id),
  -> FOREIGN KEY (Batch_Id) REFERENCES Batch_info(Batch_Id),
  -> FOREIGN KEY (Module_Id) REFERENCES Module_Info(Module_Id)
  -> );
SHOW tables;
```



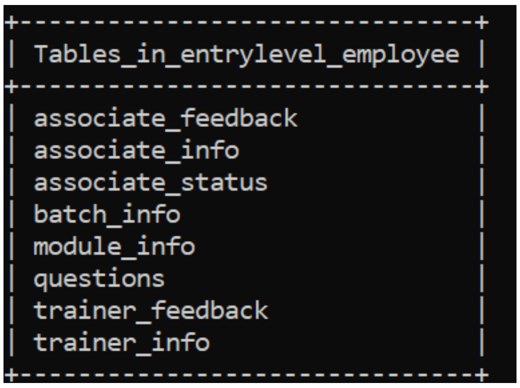
```
+-----------------------------+
| Tables_in_entrylevel_employee |
+-----------------------------+
| associate_info              |
| associate_status            |
| batch_info                  |
| module_info                 |
| questions                   |
| trainer_feedback            |
| trainer_info                |
+-----------------------------+
```

DESC Trainer_Feedback;



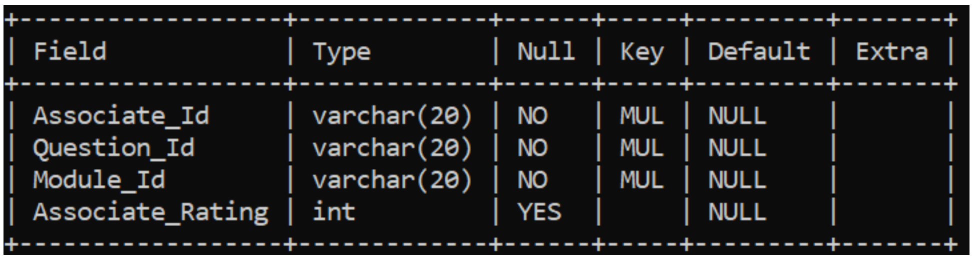
```
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| Trainer_Id     | varchar(20) | NO   | MUL | NULL    |       |
| Question_Id    | varchar(20) | NO   | MUL | NULL    |       |
| Batch_Id       | varchar(20) | NO   | MUL | NULL    |       |
| Module_Id      | varchar(20) | NO   | MUL | NULL    |       |
| Trainer_Rating | int         | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
```

```
CREATE table Associate_Feedback(
   -> Associate_Id VARCHAR(20) NOT NULL,
   -> Question_Id VARCHAR(20) NOT NULL,
   -> Module_Id VARCHAR(20) NOT NULL,
   -> Associate_Rating INT(11),
   -> FOREIGN KEY (Associate_Id) REFERENCES Associate_Info(Associate_Id),
   -> FOREIGN KEY (Question_id) REFERENCES Questions(Question_id),
   -> FOREIGN KEY (Module_Id) REFERENCES Module_Info(Module_Id)
   -> );
SHOW tables;
```



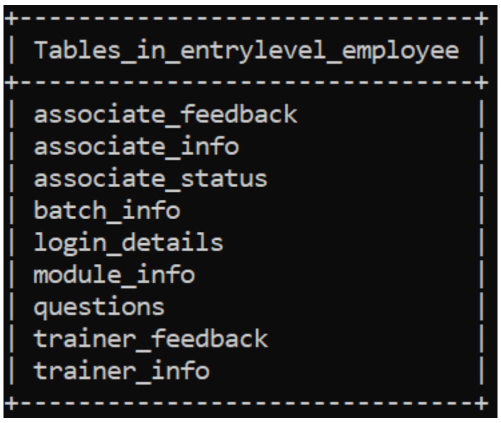
```
+------------------------------+
| Tables_in_entrylevel_employee |
+------------------------------+
| associate_feedback           |
| associate_info               |
| associate_status             |
| batch_info                   |
| module_info                  |
| questions                    |
| trainer_feedback             |
| trainer_info                 |
+------------------------------+
```

```
DESC Associate_Feedback;
```

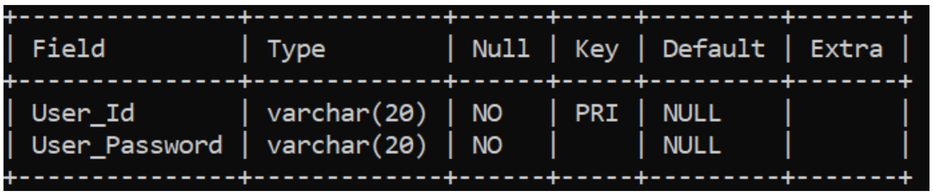


```
+------------------+-------------+------+-----+---------+-------+
| Field            | Type        | Null | Key | Default | Extra |
+------------------+-------------+------+-----+---------+-------+
| Associate_Id     | varchar(20) | NO   | MUL | NULL    |       |
| Question_Id      | varchar(20) | NO   | MUL | NULL    |       |
| Module_Id        | varchar(20) | NO   | MUL | NULL    |       |
| Associate_Rating | int         | YES  |     | NULL    |       |
+------------------+-------------+------+-----+---------+-------+
```

```
CREATE table Login_Details(
   -> User_Id VARCHAR(20) NOT NULL PRIMARY KEY,
   -> User_Password VARCHAR(20) NOT NULL
   -> );
SHOW tables;
```


```
+-----------------------------+
| Tables_in_entrylevel_employee |
+-----------------------------+
| associate_feedback          |
| associate_info              |
| associate_status            |
| batch_info                  |
| login_details               |
| module_info                 |
| questions                   |
| trainer_feedback            |
| trainer_info                |
+-----------------------------+
```

DESC Login_Details;

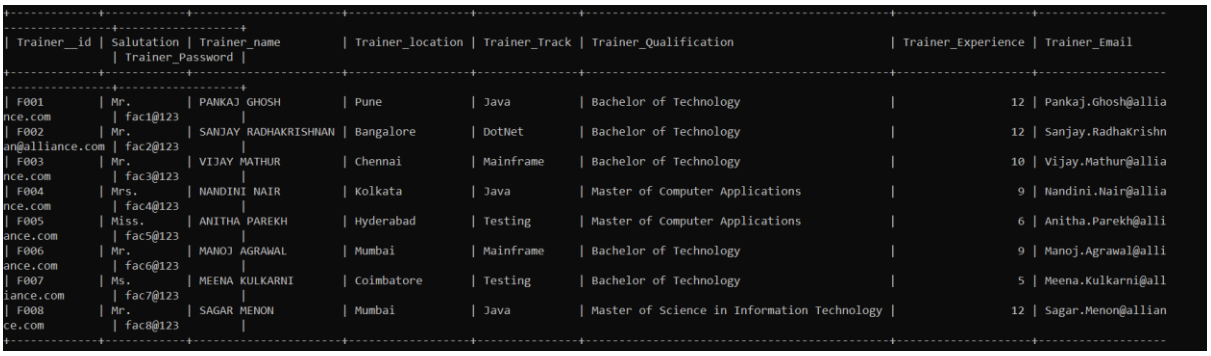

```
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| User_Id       | varchar(20) | NO   | PRI | NULL    |       |
| User_Password | varchar(20) | NO   |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
```

# Exercise 2

Problem Statement:

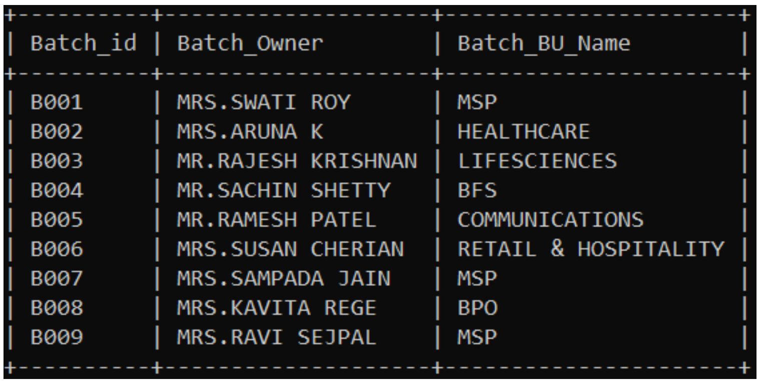
Insert below details into table:

```
INSERT INTO
Trainer_Info(Trainer__Id,Salutation,Trainer_Name,Trainer_Location,Trainer_Track,Trainer_Qualifica
tion,Trainer_Experience,Trainer_Email,Trainer_Password)
   -> values('F001','Mr.','PANKAJ GHOSH','Pune','Java','Bachelor of
Technology',12,'Pankaj.Ghosh@alliance.com','fac1@123'),
   -> ('F002','Mr.','SANJAY RADHAKRISHNAN','Bangalore','DotNet','Bachelor of
Technology',12,'Sanjay.RadhaKrishnan@alliance.com','fac2@123'),
   -> ('F003','Mr.','VIJAY MATHUR','Chennai','Mainframe','Bachelor of
Technology',10,'Vijay.Mathur@alliance.com','fac3@123'),
   -> ('F004','Mrs.','NANDINI NAIR','Kolkata','Java','Master of Computer
Applications',9,'Nandini.Nair@alliance.com','fac4@123'),
   -> ('F005','Miss.','ANITHA PAREKH','Hyderabad','Testing','Master of Computer
Applications',6,'Anitha.Parekh@alliance.com','fac5@123'),
   -> ('F006','Mr.','MANOJ AGRAWAL','Mumbai','Mainframe','Bachelor of
Technology',9,'Manoj.Agrawal@alliance.com','fac6@123'),
   -> ('F007','Ms.','MEENA KULKARNI','Coimbatore','Testing','Bachelor of
Technology',5,'Meena.Kulkarni@alliance.com','fac7@123'),
   -> ('F008','Mr.','SAGAR MENON','Mumbai','Java','Master of Science in Information
Technology',12,'Sagar.Menon@alliance.com','fac8@123');
SELECT * FROM Trainer_Info;
```



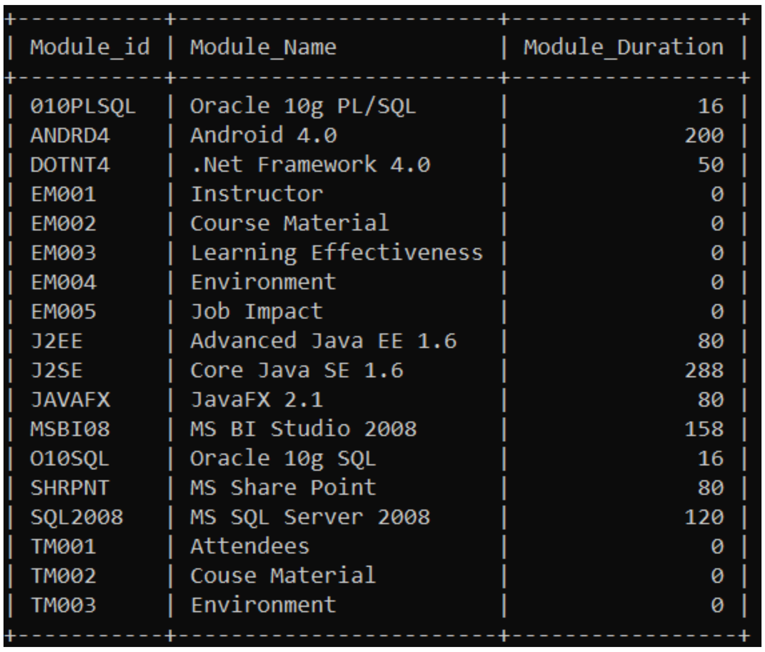
```
+-------------+------------+-----------------------+-------------------+----------------+---------------------------------------------+--------------------+------------------
----------------+------------------+
| Trainer__id | Salutation | Trainer_name          | Trainer_location  | Trainer_Track  | Trainer_Qualification                       | Trainer_Experience | Trainer_Email
                | Trainer_Password |
+-------------+------------+-----------------------+-------------------+----------------+---------------------------------------------+--------------------+------------------
----------------+------------------+
| F001        | Mr.        | PANKAJ GHOSH          | Pune              | Java           | Bachelor of Technology                      |                 12 | Pankaj.Ghosh@allia
nce.com         | fac1@123         |
| F002        | Mr.        | SANJAY RADHAKRISHNAN  | Bangalore         | DotNet         | Bachelor of Technology                      |                 12 | Sanjay.RadhaKrishn
an@alliance.com | fac2@123         |
| F003        | Mr.        | VIJAY MATHUR          | Chennai           | Mainframe      | Bachelor of Technology                      |                 10 | Vijay.Mathur@allia
nce.com         | fac3@123         |
| F004        | Mrs.       | NANDINI NAIR          | Kolkata           | Java           | Master of Computer Applications             |                  9 | Nandini.Nair@allia
nce.com         | fac4@123         |
| F005        | Miss.      | ANITHA PAREKH         | Hyderabad         | Testing        | Master of Computer Applications             |                  6 | Anitha.Parekh@alli
ance.com        | fac5@123         |
| F006        | Mr.        | MANOJ AGRAWAL         | Mumbai            | Mainframe      | Bachelor of Technology                      |                  9 | Manoj.Agrawal@alli
ance.com        | fac6@123         |
| F007        | Ms.        | MEENA KULKARNI        | Coimbatore        | Testing        | Bachelor of Technology                      |                  5 | Meena.Kulkarni@all
iance.com       | fac7@123         |
| F008        | Mr.        | SAGAR MENON           | Mumbai            | Java           | Master of Science in Information Technology |                 12 | Sagar.Menon@allian
ce.com          | fac8@123         |
+-------------+------------+-----------------------+-------------------+----------------+---------------------------------------------+--------------------+------------------
```

```
INSERT INTO Batch_info(Batch_Id,Batch_Owner,Batch_BU_Name)
    -> values('B001','MRS.SWATI ROY','MSP'),
    -> ('B002','MRS.ARUNA K','HEALTHCARE'),
    -> ('B003','MR.RAJESH KRISHNAN','LIFESCIENCES'),
    -> ('B004','MR.SACHIN SHETTY','BFS'),
    -> ('B005','MR.RAMESH PATEL','COMMUNICATIONS'),
    -> ('B006','MRS.SUSAN CHERIAN','RETAIL & HOSPITALITY'),
    -> ('B007','MRS.SAMPADA JAIN','MSP'),
    -> ('B008','MRS.KAVITA REGE','BPO'),
    -> ('B009','MRS.RAVI SEJPAL','MSP');
SELECT * FROM Batch_info;
```



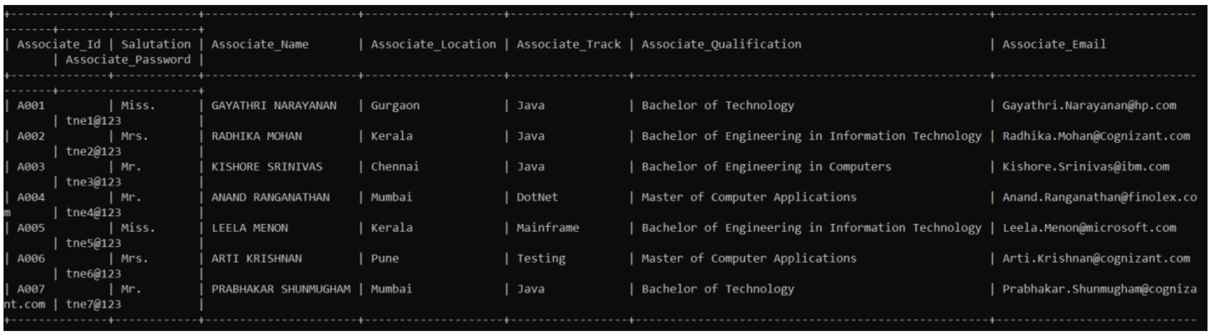
```
+----------+--------------------+----------------------+
| Batch_id | Batch_Owner        | Batch_BU_Name        |
+----------+--------------------+----------------------+
| B001     | MRS.SWATI ROY      | MSP                  |
| B002     | MRS.ARUNA K        | HEALTHCARE           |
| B003     | MR.RAJESH KRISHNAN | LIFESCIENCES         |
| B004     | MR.SACHIN SHETTY   | BFS                  |
| B005     | MR.RAMESH PATEL    | COMMUNICATIONS       |
| B006     | MRS.SUSAN CHERIAN  | RETAIL & HOSPITALITY |
| B007     | MRS.SAMPADA JAIN   | MSP                  |
| B008     | MRS.KAVITA REGE    | BPO                  |
| B009     | MRS.RAVI SEJPAL    | MSP                  |
+----------+--------------------+----------------------+
```

```
INSERT INTO Module_Info(Module_Id,Module_Name,Module_Duration)
    -> values('O10SQL','Oracle 10g SQL',16),
    -> ('010PLSQL','Oracle 10g PL/SQL',16),
    -> ('J2SE','Core Java SE 1.6',288),
    -> ('J2EE','Advanced Java EE 1.6',80),
    -> ('JAVAFX','JavaFX 2.1',80),
    -> ('DOTNT4','.Net Framework 4.0',50),
    -> ('SQL2008','MS SQL Server 2008',120),
    -> ('MSBI08','MS BI Studio 2008',158),
    -> ('SHRPNT','MS Share Point',80),
    -> ('ANDRD4','Android 4.0',200),
    -> ('EM001','Instructor',0),
    -> ('EM002','Course Material',0),
    -> ('EM003','Learning Effectiveness',0),
    -> ('EM004','Environment',0),
    -> ('EM005','Job Impact',0),
    -> ('TM001','Attendees',0),
    -> ('TM002','Couse Material',0),
    -> ('TM003','Environment',0);
SELECT * FROM Module_Info;
```



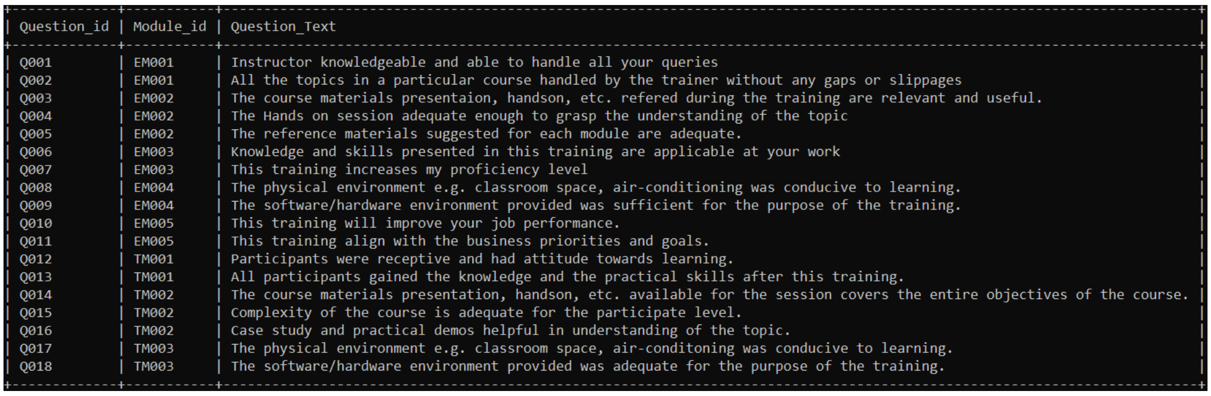
```
+-----------+-----------------------+-----------------+
| Module_id | Module_Name           | Module_Duration |
+-----------+-----------------------+-----------------+
| 010PLSQL  | Oracle 10g PL/SQL     |              16 |
| ANDRD4    | Android 4.0           |             200 |
| DOTNT4    | .Net Framework 4.0    |              50 |
| EM001     | Instructor            |               0 |
| EM002     | Course Material       |               0 |
| EM003     | Learning Effectiveness |              0 |
| EM004     | Environment           |               0 |
| EM005     | Job Impact            |               0 |
| J2EE      | Advanced Java EE 1.6  |              80 |
| J2SE      | Core Java SE 1.6      |             288 |
| JAVAFX    | JavaFX 2.1            |              80 |
| MSBI08    | MS BI Studio 2008     |             158 |
| O10SQL    | Oracle 10g SQL        |              16 |
| SHRPNT    | MS Share Point        |              80 |
| SQL2008   | MS SQL Server 2008    |             120 |
| TM001     | Attendees             |               0 |
| TM002     | Couse Material        |               0 |
| TM003     | Environment           |               0 |
+-----------+-----------------------+-----------------+
```

```
INSERT INTO
Associate_Info(Associate_Id,Salutation,Associate_Name,Associate_Location,Associate_Track,Associa
te_Qualification,Associate_Email,Associate_Password)
    ->  values('A001','Miss.','GAYATHRI NARAYANAN','Gurgaon','Java','Bachelor of
Technology','Gayathri.Narayanan@hp.com','tne1@123'),
    -> ('A002','Mrs.','RADHIKA MOHAN','Kerala','Java','Bachelor of Engineering in Information
Technology','Radhika.Mohan@Cognizant.com','tne2@123'),
    -> ('A003','Mr.','KISHORE SRINIVAS','Chennai','Java','Bachelor of Engineering in
Computers','Kishore.Srinivas@ibm.com','tne3@123'),
    -> ('A004','Mr.','ANAND RANGANATHAN','Mumbai','DotNet','Master of Computer
Applications','Anand.Ranganathan@finolex.com','tne4@123'),
    -> ('A005','Miss.','LEELA MENON','Kerala','Mainframe','Bachelor of Engineering in Information
Technology','Leela.Menon@microsoft.com','tne5@123'),
    -> ('A006','Mrs.','ARTI KRISHNAN','Pune','Testing','Master of Computer
Applications','Arti.Krishnan@cognizant.com','tne6@123'),
    -> ('A007','Mr.','PRABHAKAR SHUNMUGHAM','Mumbai','Java','Bachelor of
Technology','Prabhakar.Shunmugham@cognizant.com','tne7@123');
SELECT * FROM Associate_Info;
```



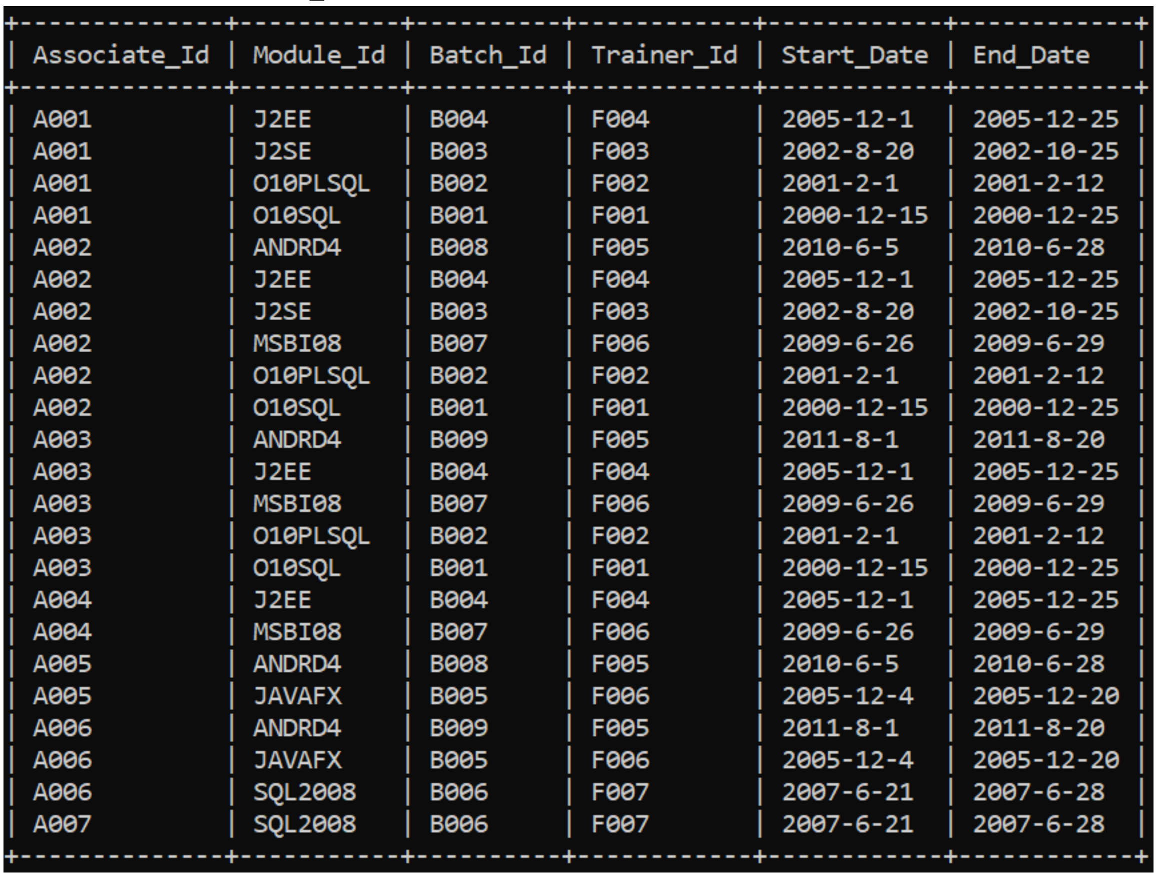
```
+--------------+------------+---------------------+--------------------+-----------------+----------------------------------------------+-----------------------------
-------+--------------------+
| Associate_Id | Salutation | Associate_Name      | Associate_Location | Associate_Track | Associate_Qualification                      | Associate_Email
       | Associate_Password |
+--------------+------------+---------------------+--------------------+-----------------+----------------------------------------------+-----------------------------
-------+--------------------+
| A001         | Miss.      | GAYATHRI NARAYANAN  | Gurgaon            | Java            | Bachelor of Technology                       | Gayathri.Narayanan@hp.com
       | tne1@123           |
| A002         | Mrs.       | RADHIKA MOHAN       | Kerala             | Java            | Bachelor of Engineering in Information Technology | Radhika.Mohan@Cognizant.com
       | tne2@123           |
| A003         | Mr.        | KISHORE SRINIVAS    | Chennai            | Java            | Bachelor of Engineering in Computers         | Kishore.Srinivas@ibm.com
       | tne3@123           |
| A004         | Mr.        | ANAND RANGANATHAN   | Mumbai             | DotNet          | Master of Computer Applications              | Anand.Ranganathan@finolex.co
m      | tne4@123           |
| A005         | Miss.      | LEELA MENON         | Kerala             | Mainframe       | Bachelor of Engineering in Information Technology | Leela.Menon@microsoft.com
       | tne5@123           |
| A006         | Mrs.       | ARTI KRISHNAN       | Pune               | Testing         | Master of Computer Applications              | Arti.Krishnan@cognizant.com
       | tne6@123           |
| A007         | Mr.        | PRABHAKAR SHUNMUGHAM | Mumbai            | Java            | Bachelor of Technology                       | Prabhakar.Shunmugham@cogniza
nt.com | tne7@123           |
+--------------+------------+---------------------+--------------------+-----------------+----------------------------------------------+-----------------------------
```

```
INSERT INTO Questions(Question_Id,Module_Id,Question_Text)
    -> values('Q001','EM001','Instructor knowledgeable and able to handle all your queries'),
    -> ('Q002','EM001','All the topics in a particular course handled by the trainer without any gaps or slippages'),
    -> ('Q003','EM002','The course materials presentaion, handson, etc. refered during the training are relevant and useful.'),
    -> ('Q004','EM002','The Hands on session adequate enough to grasp the understanding of the topic'),
    -> ('Q005','EM002','The reference materials suggested for each module are adequate.'),
    -> ('Q006','EM003','Knowledge and skills presented in this training are applicable at your work'),
    -> ('Q007','EM003','This training increases my proficiency level'),
    -> ('Q008','EM004','The physical environment e.g. classroom space, air-conditioning was conducive to learning.'),
    -> ('Q009','EM004','The software/hardware environment provided was sufficient for the purpose of the training.'),
    -> ('Q010','EM005','This training will improve your job performance.'),
    -> ('Q011','EM005','This training align with the business priorities and goals.'),
    -> ('Q012','TM001','Participants were receptive and had attitude towards learning.'),
    -> ('Q013','TM001','All participants gained the knowledge and the practical skills after this training.'),
    -> ('Q014','TM002','The course materials presentation, handson, etc. available for the session covers the entire objectives of the course.'),
    -> ('Q015','TM002','Complexity of the course is adequate for the participate level.'),
    -> ('Q016','TM002','Case study and practical demos helpful in understanding of the topic.'),
    -> ('Q017','TM003','The physical environment e.g. classroom space, air-conditoning was conducive to learning.'),
    -> ('Q018','TM003','The software/hardware environment provided was adequate for the purpose of the training.');
SELECT * FROM Questions;
```



```
+-------------+-----------+------------------------------------------------------------------------------------------------------------------------+
| Question_id | Module_id | Question_Text                                                                                                          |
+-------------+-----------+------------------------------------------------------------------------------------------------------------------------+
| Q001        | EM001     | Instructor knowledgeable and able to handle all your queries                                                           |
| Q002        | EM001     | All the topics in a particular course handled by the trainer without any gaps or slippages                             |
| Q003        | EM002     | The course materials presentaion, handson, etc. refered during the training are relevant and useful.                  |
| Q004        | EM002     | The Hands on session adequate enough to grasp the understanding of the topic                                           |
| Q005        | EM002     | The reference materials suggested for each module are adequate.                                                        |
| Q006        | EM003     | Knowledge and skills presented in this training are applicable at your work                                            |
| Q007        | EM003     | This training increases my proficiency level                                                                           |
| Q008        | EM004     | The physical environment e.g. classroom space, air-conditioning was conducive to learning.                             |
| Q009        | EM004     | The software/hardware environment provided was sufficient for the purpose of the training.                             |
| Q010        | EM005     | This training will improve your job performance.                                                                       |
| Q011        | EM005     | This training align with the business priorities and goals.                                                            |
| Q012        | TM001     | Participants were receptive and had attitude towards learning.                                                         |
| Q013        | TM001     | All participants gained the knowledge and the practical skills after this training.                                    |
| Q014        | TM002     | The course materials presentation, handson, etc. available for the session covers the entire objectives of the course. |
| Q015        | TM002     | Complexity of the course is adequate for the participate level.                                                        |
| Q016        | TM002     | Case study and practical demos helpful in understanding of the topic.                                                  |
| Q017        | TM003     | The physical environment e.g. classroom space, air-conditoning was conducive to learning.                              |
| Q018        | TM003     | The software/hardware environment provided was adequate for the purpose of the training.                               |
+-------------+-----------+------------------------------------------------------------------------------------------------------------------------+
```

```
INSERT INTO Associate_Status(Associate_Id,Module_Id,Batch_Id,Trainer_Id,Start_Date,End_Date)
    ->     values('A001','O10SQL','B001','F001','2000-12-15','2000-12-25'),
    ->     ('A002','O10SQL','B001','F001','2000-12-15','2000-12-25'),
    ->     ('A003','O10SQL','B001','F001','2000-12-15','2000-12-25'),
    ->     ('A001','O10PLSQL','B002','F002','2001-2-1','2001-2-12'),
    ->     ('A002','O10PLSQL','B002','F002','2001-2-1','2001-2-12'),
    ->     ('A003','O10PLSQL','B002','F002','2001-2-1','2001-2-12'),
    ->     ('A001','J2SE','B003','F003','2002-8-20','2002-10-25'),
    ->     ('A002','J2SE','B003','F003','2002-8-20','2002-10-25'),
    ->     ('A001','J2EE','B004','F004','2005-12-1','2005-12-25'),
    ->     ('A002','J2EE','B004','F004','2005-12-1','2005-12-25'),
    ->    ('A003','J2EE','B004','F004','2005-12-1','2005-12-25'),
    ->    ('A004','J2EE','B004','F004','2005-12-1','2005-12-25'),
    ->    ('A005','JAVAFX','B005','F006','2005-12-4','2005-12-20'),
    ->    ('A006','JAVAFX','B005','F006','2005-12-4','2005-12-20'),
    ->    ('A006','SQL2008','B006','F007','2007-6-21','2007-6-28'),
    ->    ('A007','SQL2008','B006','F007','2007-6-21','2007-6-28'),
    ->    ('A002','MSBI08','B007','F006','2009-6-26','2009-6-29'),
    ->    ('A003','MSBI08','B007','F006','2009-6-26','2009-6-29'),
    ->    ('A004','MSBI08','B007','F006','2009-6-26','2009-6-29'),
    ->    ('A002','ANDRD4','B008','F005','2010-6-5','2010-6-28'),
    ->    ('A005','ANDRD4','B008','F005','2010-6-5','2010-6-28'),
    ->    ('A003','ANDRD4','B009','F005','2011-8-1','2011-8-20'),
    ->    ('A006','ANDRD4','B009','F005','2011-8-1','2011-8-20');
SELECT * FROM Associate_Status;
```



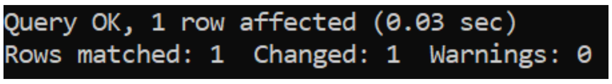
```
+--------------+-----------+----------+------------+------------+------------+
| Associate_Id | Module_Id | Batch_Id | Trainer_Id | Start_Date | End_Date   |
+--------------+-----------+----------+------------+------------+------------+
| A001         | J2EE      | B004     | F004       | 2005-12-1  | 2005-12-25 |
| A001         | J2SE      | B003     | F003       | 2002-8-20  | 2002-10-25 |
| A001         | O10PLSQL  | B002     | F002       | 2001-2-1   | 2001-2-12  |
| A001         | O10SQL    | B001     | F001       | 2000-12-15 | 2000-12-25 |
| A002         | ANDRD4    | B008     | F005       | 2010-6-5   | 2010-6-28  |
| A002         | J2EE      | B004     | F004       | 2005-12-1  | 2005-12-25 |
| A002         | J2SE      | B003     | F003       | 2002-8-20  | 2002-10-25 |
| A002         | MSBI08    | B007     | F006       | 2009-6-26  | 2009-6-29  |
| A002         | O10PLSQL  | B002     | F002       | 2001-2-1   | 2001-2-12  |
| A002         | O10SQL    | B001     | F001       | 2000-12-15 | 2000-12-25 |
| A003         | ANDRD4    | B009     | F005       | 2011-8-1   | 2011-8-20  |
| A003         | J2EE      | B004     | F004       | 2005-12-1  | 2005-12-25 |
| A003         | MSBI08    | B007     | F006       | 2009-6-26  | 2009-6-29  |
| A003         | O10PLSQL  | B002     | F002       | 2001-2-1   | 2001-2-12  |
| A003         | O10SQL    | B001     | F001       | 2000-12-15 | 2000-12-25 |
| A004         | J2EE      | B004     | F004       | 2005-12-1  | 2005-12-25 |
| A004         | MSBI08    | B007     | F006       | 2009-6-26  | 2009-6-29  |
| A005         | ANDRD4    | B008     | F005       | 2010-6-5   | 2010-6-28  |
| A005         | JAVAFX    | B005     | F006       | 2005-12-4  | 2005-12-20 |
| A006         | ANDRD4    | B009     | F005       | 2011-8-1   | 2011-8-20  |
| A006         | JAVAFX    | B005     | F006       | 2005-12-4  | 2005-12-20 |
| A006         | SQL2008   | B006     | F007       | 2007-6-21  | 2007-6-28  |
| A007         | SQL2008   | B006     | F007       | 2007-6-21  | 2007-6-28  |
+--------------+-----------+----------+------------+------------+------------+
```

# Exercise 3

Problem Statement:

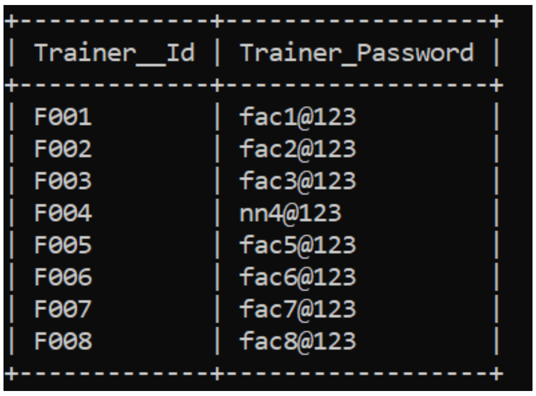
Change the password of trainer F004 from fac4@123 to nn4@123

```
UPDATE Trainer_Info SET Trainer_Password = 'nn4@123'
WHERE Trainer__ID='F004';
```



```
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
SELECT Trainer__Id, Trainer_Password FROM Trainer_Info;
```



```
+-------------+------------------+
| Trainer__Id | Trainer_Password |
+-------------+------------------+
| F001        | fac1@123         |
| F002        | fac2@123         |
| F003        | fac3@123         |
| F004        | nn4@123          |
| F005        | fac5@123         |
| F006        | fac6@123         |
| F007        | fac7@123         |
| F008        | fac8@123         |
+-------------+------------------+
```

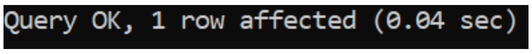
# Exercise 4

Problem Statement:

Remove following record form associate_status table

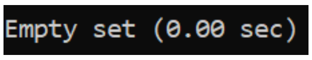
A003,J2EE,B004,F004,2005-12-1,2005-12-25

DELETE FROM Associate_Status
WHERE Associate_ID='A003' AND Module_Id='J2EE' AND Batch_Id='B004' AND Trainer_Id='F004'
AND Start_Date='2005-12-1' AND End_Date='2005-12-25';



```
Query OK, 1 row affected (0.04 sec)
```

SELECT Associate_ID, Module_Id FROM Associate_Status
WHERE Associate_ID='A003' AND Module_Id='J2EE';



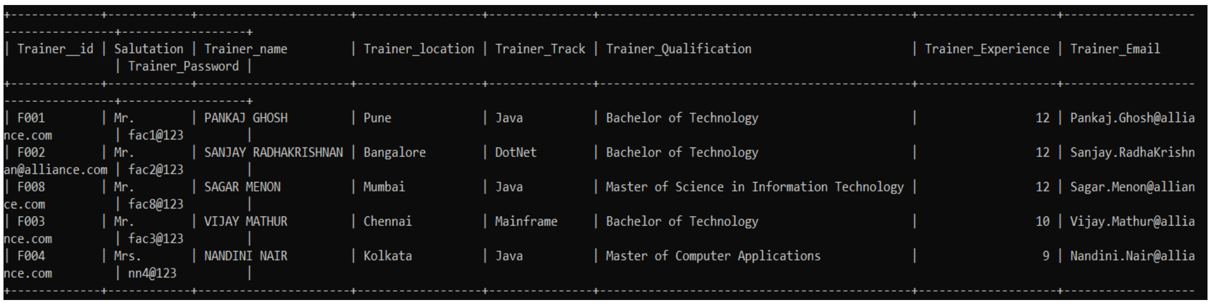
```
Empty set (0.00 sec)
```

# Exercise 5

Problem Statement:

Fetch first five trainers who have maximum years of experience & display there details.

SELECT * FROM Trainer_Info
ORDER BY Trainer_Experience DESC
LIMIT 5;



```
+-------------+------------+----------------------+------------------+---------------+------------------------------------------+--------------------+------------------
-------------+------------------+
| Trainer__id | Salutation | Trainer_name         | Trainer_location | Trainer_Track | Trainer_Qualification                    | Trainer_Experience | Trainer_Email
             | Trainer_Password |
+-------------+------------+----------------------+------------------+---------------+------------------------------------------+--------------------+------------------
-------------+------------------+
| F001        | Mr.        | PANKAJ GHOSH         | Pune             | Java          | Bachelor of Technology                   |                 12 | Pankaj.Ghosh@allia
nce.com      | fac1@123         |
| F002        | Mr.        | SANJAY RADHAKRISHNAN | Bangalore        | DotNet        | Bachelor of Technology                   |                 12 | Sanjay.RadhaKrishn
an@alliance.com | fac2@123      |
| F008        | Mr.        | SAGAR MENON          | Mumbai           | Java          | Master of Science in Information Technology |              12 | Sagar.Menon@allian
ce.com       | fac8@123         |
| F003        | Mr.        | VIJAY MATHUR         | Chennai          | Mainframe     | Bachelor of Technology                   |                 10 | Vijay.Mathur@allia
nce.com      | fac3@123         |
| F004        | Mrs.       | NANDINI NAIR         | Kolkata          | Java          | Master of Computer Applications          |                  9 | Nandini.Nair@allia
nce.com      | nn4@123          |
+-------------+------------+----------------------+------------------+---------------+------------------------------------------+--------------------+------------------
```

# Exercise 6

Problem Statement:

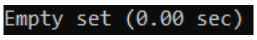
Begin transaction & insert below to records into Login_Details

'U001' Admin1@123

'U002' Admin2@123

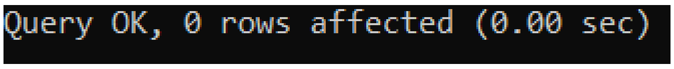
Perform rollback operation & verify whether any records are inserted in table or not.

SELECT * FROM Login_Details;



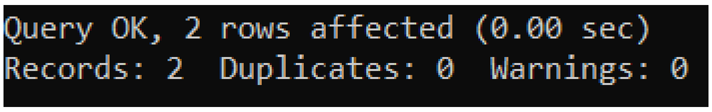
```
Empty set (0.00 sec)
```

START TRANSACTION;



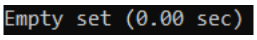
```
Query OK, 0 rows affected (0.00 sec)
```

```
INSERT INTO (User_Id, User_Password)
   -> values('U001','Admin1@123'),
   -> ('U002','Admin2@123');
```



```
Query OK, 2 rows affected (0.00 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

SELECT * FROM Login_Details;



```
Empty set (0.00 sec)
```

# EXERCISE 7

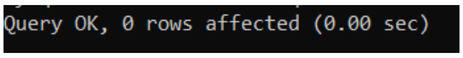
Problem Statement:

Create a dummy user in database. Grant create & select table privilege to him/her.

Repeat the above all queries using login credentials of newly created user.

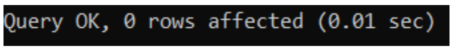
Revoke the privilege assigned to this newly created user.

CREATE USER 'CKrithika@localhost' IDENTIFIED BY 'password';



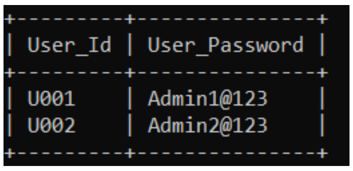
```
Query OK, 0 rows affected (0.00 sec)
```

GRANT CREATE, SELECT ON sqlassessment TO 'CKrithika@localhost';



```
Query OK, 0 rows affected (0.01 sec)
```

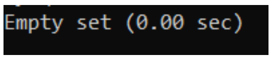
```
START TRANSACTION;
INSERT INTO Login_Details(User_Id, User_Password)
   -> values('U001','Admin1@123'),
   -> ('U002','Admin2@123');
SELECT * FROM Login_Details;
```



```
+---------+---------------+
| User_Id | User_Password |
+---------+---------------+
| U001    | Admin1@123    |
| U002    | Admin2@123    |
+---------+---------------+
```

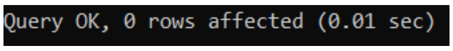
ROLLBACK;

SELECT * FROM Login_Details;



```
Empty set (0.00 sec)
```

REVOKE CREATE, SELECT ON sqlassessment FROM 'CKrithika@localhost';



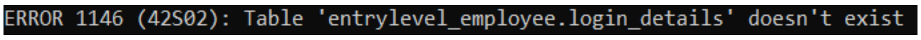
```
Query OK, 0 rows affected (0.01 sec)
```

# EXERCISE 8

Problem Statement:

Remove table Login_Details from database.

Drop table Login_Details;
DESC Login_Details;

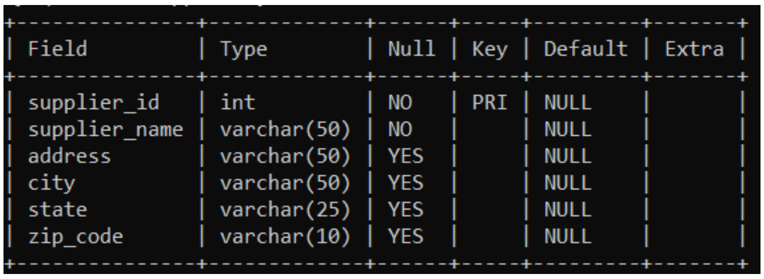


```
ERROR 1146 (42S02): Table 'entrylevel_employee.login_details' doesn't exist
```

# EXERCISE 9

Create a table called suppliers that stores supplier ID, name, and address information.

```
CREATE TABLE suppliers(
   -> supplier_id INT(10) NOT NULL PRIMARY KEY,
   -> supplier_name VARCHAR(50) NOT NULL,
   -> address VARCHAR(50),
   -> city VARCHAR(50),
   -> state VARCHAR(25),
   -> zip_code VARCHAR(10)
   -> );
DESC suppliers;
```

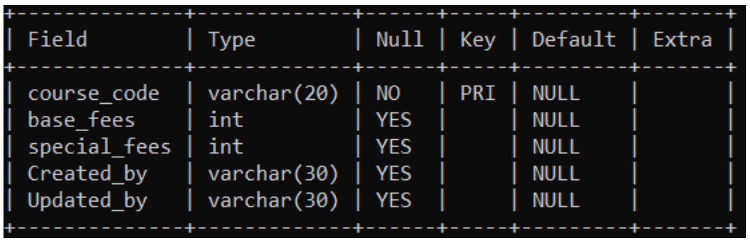


```
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| supplier_id   | int         | NO   | PRI | NULL    |       |
| supplier_name | varchar(50) | NO   |     | NULL    |       |
| address       | varchar(50) | YES  |     | NULL    |       |
| city          | varchar(50) | YES  |     | NULL    |       |
| state         | varchar(25) | YES  |     | NULL    |       |
| zip_code      | varchar(10) | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
```

# EXERCISE 10

Problem Statement: **Display all the unique courses between course fees and course fees_history**

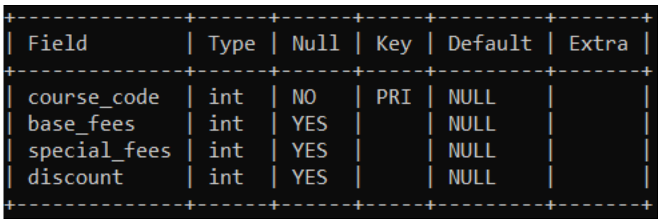
```
CREATE TABLE course(
   -> course_code VARCHAR(20) NOT NULL PRIMARY KEY,
   -> base_fees INT(11),
   -> special_fees INT(11),
   -> Created_by VARCHAR(30),
   -> Updated_by VARCHAR(30)
   -> );
DESC course;
```



```
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| course_code  | varchar(20) | NO   | PRI | NULL    |       |
| base_fees    | int         | YES  |     | NULL    |       |
| special_fees | int         | YES  |     | NULL    |       |
| Created_by   | varchar(30) | YES  |     | NULL    |       |
| Updated_by   | varchar(30) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
```

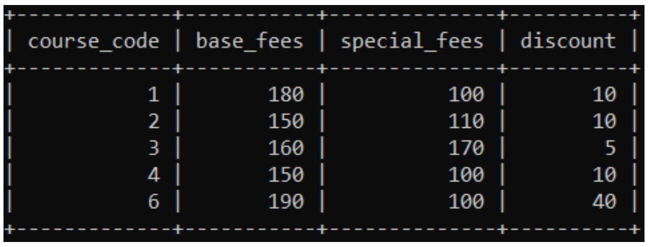
```
CREATE TABLE Course_Fees(
   -> course_code INT(5) NOT NULL PRIMARY KEY,
   -> base_fees INT(11),
   -> special_fees INT(11),
   -> discount INT(11)
   -> );
```

DESC Course_Fees;



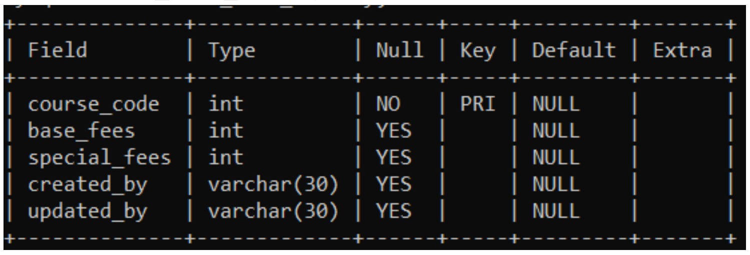
```
+--------------+------+------+-----+---------+-------+
| Field        | Type | Null | Key | Default | Extra |
+--------------+------+------+-----+---------+-------+
| course_code  | int  | NO   | PRI | NULL    |       |
| base_fees    | int  | YES  |     | NULL    |       |
| special_fees | int  | YES  |     | NULL    |       |
| discount     | int  | YES  |     | NULL    |       |
+--------------+------+------+-----+---------+-------+
```

```
INSERT INTO Course_Fees (course_code,base_fees,special_fees,discount)
    -> values(1,180,100,10),
    -> (2,150,110,10),
    -> (3,160,170,5),
    -> (4,150,100,10),
    -> (6,190,100,40);
```



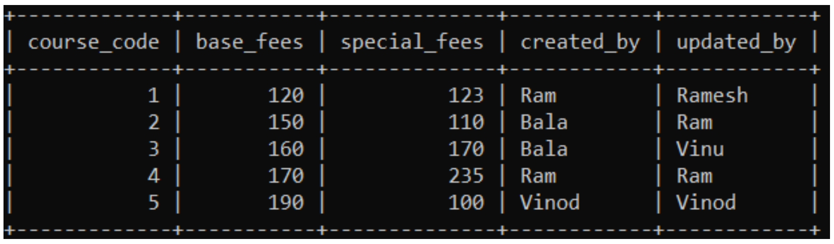
```
+-------------+-----------+--------------+----------+
| course_code | base_fees | special_fees | discount |
+-------------+-----------+--------------+----------+
|           1 |       180 |          100 |       10 |
|           2 |       150 |          110 |       10 |
|           3 |       160 |          170 |        5 |
|           4 |       150 |          100 |       10 |
|           6 |       190 |          100 |       40 |
+-------------+-----------+--------------+----------+
```

```
CREATE TABLE Course_Fees_History(
    -> course_code INT(5) NOT NULL PRIMARY KEY,
    -> base_fees INT(11),
    -> special_fees INT(11),
    -> created_by VARCHAR(30),
    -> updated_by VARCHAR(30)
    -> );
DESC Course_Fees_History;
```



```
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| course_code  | int         | NO   | PRI | NULL    |       |
| base_fees    | int         | YES  |     | NULL    |       |
| special_fees | int         | YES  |     | NULL    |       |
| created_by   | varchar(30) | YES  |     | NULL    |       |
| updated_by   | varchar(30) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
```

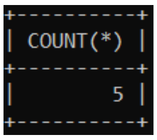
```
INSERT INTO Course_Fees_History(Course_code, base_fees, special_fees, created_by, updated_by)
    -> values(1,120,123,'Ram','Ramesh'),
    -> (2,150,110,'Bala','Ram'),
    -> (3,160,170,'Bala','Vinu'),
    -> (4,170,235,'Ram','Ram'),
    -> (5,190,100,'Vinod','Vinod');
SELECT * FROM Course_Fees_History;
```



```
+-------------+-----------+--------------+------------+------------+
| course_code | base_fees | special_fees | created_by | updated_by |
+-------------+-----------+--------------+------------+------------+
|           1 |       120 |          123 | Ram        | Ramesh     |
|           2 |       150 |          110 | Bala       | Ram        |
|           3 |       160 |          170 | Bala       | Vinu       |
|           4 |       170 |          235 | Ram        | Ram        |
|           5 |       190 |          100 | Vinod      | Vinod      |
+-------------+-----------+--------------+------------+------------+
```

# EXERCISE 11
Use the following columns to check for uniqueness of Course_Code, BASE_FEES and SPECIAL_FEES of the courses in both the COURSE_FEES and COURSE_FEES_HISTORY.

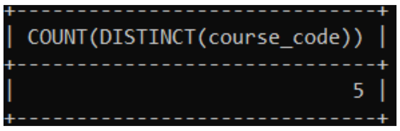
SELECT COUNT(*) FROM course_fees;



```
+----------+
| COUNT(*) |
+----------+
|        5 |
+----------+
```

#count on number of rows in course_fees

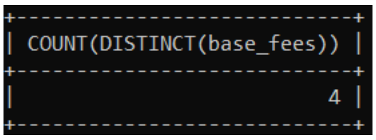
SELECT COUNT(DISTINCT(course_code)) FROM Course_Fees;



```
+------------------------------+
| COUNT(DISTINCT(course_code)) |
+------------------------------+
|                            5 |
+------------------------------+
```

# since this count is equal to total count, course_code is unique in course_fees table.

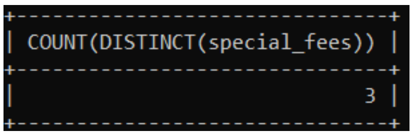
SELECT COUNT(DISTINCT(base_fees)) FROM Course_Fees;



```
+----------------------------+
| COUNT(DISTINCT(base_fees)) |
+----------------------------+
|                          4 |
+----------------------------+
```

# since this count is not equal to total count, base_fees is not unique in course_fees table.

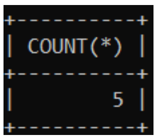
SELECT COUNT(DISTINCT(special_fees)) FROM Course_Fees;



```
+-------------------------------+
| COUNT(DISTINCT(special_fees)) |
+-------------------------------+
|                             3 |
+-------------------------------+
```

# since this count is not equal to total count, special_fees is not unique in course_fees table.

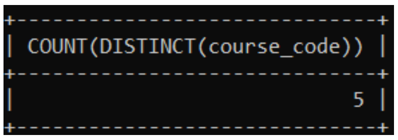
SELECT COUNT(*) FROM Course_Fees_History;



```
+----------+
| COUNT(*) |
+----------+
|        5 |
+----------+
```

#count on number of rows in Course_Fees_History

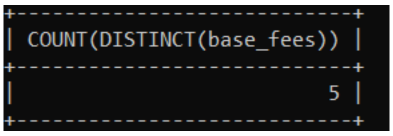
SELECT COUNT(DISTINCT(course_code)) FROM Course_Fees_History;



```
+------------------------------+
| COUNT(DISTINCT(course_code)) |
+------------------------------+
|                            5 |
+------------------------------+
```

# since this count is equal to total count, course_code is unique in Course_Fees_History table.

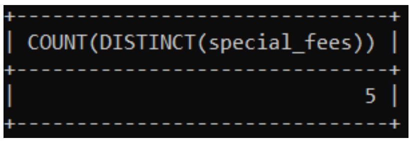
SELECT COUNT(DISTINCT(base_fees)) FROM Course_Fees_History;



```
+----------------------------+
| COUNT(DISTINCT(base_fees)) |
+----------------------------+
|                          5 |
+----------------------------+
```

# since this count is equal to total count, base_fees is unique in Course_Fees_History table.

SELECT COUNT(DISTINCT(special_fees)) FROM Course_Fees_History;



```
+-------------------------------+
| COUNT(DISTINCT(special_fees)) |
+-------------------------------+
|                             5 |
+-------------------------------+
```

# since this count is equal to total count, special_fees is unique in Course_Fees_History table.

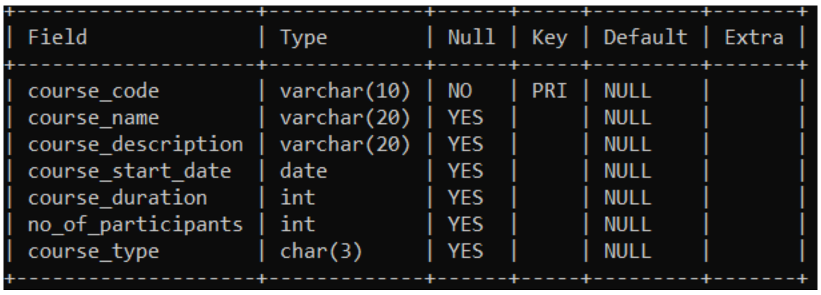
# EXERCISE 12
Pre-requisite : Use the Course_Info and Course_Fees table.

- **Insert 2 records in course_fees table with base fees as null.**
- **Insert 2 records in course_fees table with base fees as 300 and 175.**

Problem Statement:

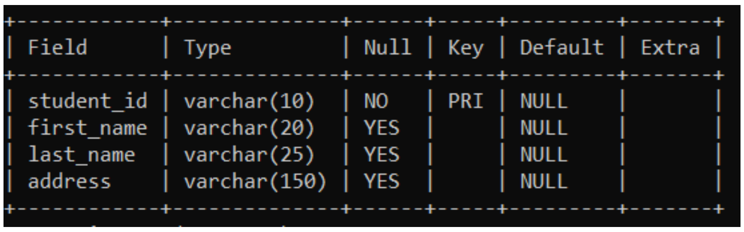
Display the minimum and maximum base fees of the courses.

```
CREATE TABLE Course_info(
    course_code VARCHAR(10) NOT NULL PRIMARY KEY,
  course_name VARCHAR(20),
  course_description VARCHAR(20),
course_start_date DATE,
    course_duration INT(11),
no_of_participants INT(11),
   course_type CHAR(3)
   );
DESC Course_info;
```



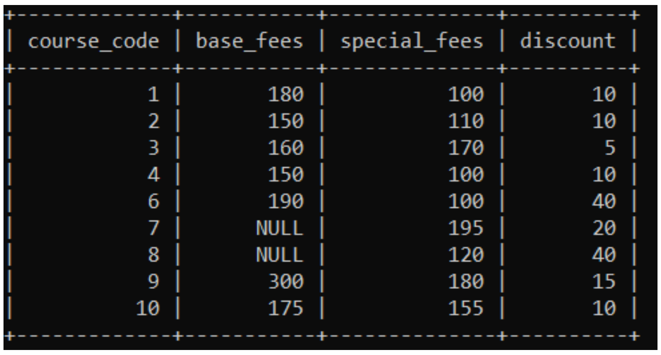
```
+--------------------+-------------+------+-----+---------+-------+
| Field              | Type        | Null | Key | Default | Extra |
+--------------------+-------------+------+-----+---------+-------+
| course_code        | varchar(10) | NO   | PRI | NULL    |       |
| course_name        | varchar(20) | YES  |     | NULL    |       |
| course_description | varchar(20) | YES  |     | NULL    |       |
| course_start_date  | date        | YES  |     | NULL    |       |
| course_duration    | int         | YES  |     | NULL    |       |
| no_of_participants | int         | YES  |     | NULL    |       |
| course_type        | char(3)     | YES  |     | NULL    |       |
+--------------------+-------------+------+-----+---------+-------+
```

```
CREATE TABLE Student_info(
    -> student_id VARCHAR(10),
    -> first_name VARCHAR(20),
    -> last_name VARCHAR(25),
    -> address VARCHAR(150),
    -> PRIMARY KEY (student_id)
    -> );
DESC student_info;
```



```
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| student_id | varchar(10)  | NO   | PRI | NULL    |       |
| first_name | varchar(20)  | YES  |     | NULL    |       |
| last_name  | varchar(25)  | YES  |     | NULL    |       |
| address    | varchar(150) | YES  |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
```

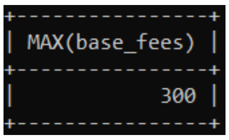
```
INSERT INTO Course_Fees (course_code,base_fees,special_fees,discount)
    -> values(7,NULL,195,20),
    -> (8,NULL,120,40)
    -> (9,300,180,15),
    -> (10,175,155,10);
SELECT * FROM Course_fees;
```



```
+-------------+-----------+--------------+----------+
| course_code | base_fees | special_fees | discount |
+-------------+-----------+--------------+----------+
|           1 |       180 |          100 |       10 |
|           2 |       150 |          110 |       10 |
|           3 |       160 |          170 |        5 |
|           4 |       150 |          100 |       10 |
|           6 |       190 |          100 |       40 |
|           7 |      NULL |          195 |       20 |
|           8 |      NULL |          120 |       40 |
|           9 |       300 |          180 |       15 |
|          10 |       175 |          155 |       10 |
+-------------+-----------+--------------+----------+
```

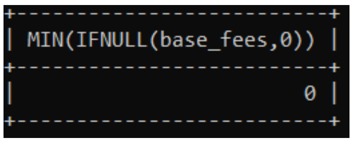
Krithika Devi Chandran

SELECT MAX(base_fees) FROM Course_fees;



```
+----------------+
| MAX(base_fees) |
+----------------+
|            300 |
+----------------+
```

SELECT MIN(IFNULL(base_fees,0)) FROM course_fees;



```
+--------------------------+
| MIN(IFNULL(base_fees,0)) |
+--------------------------+
|                        0 |
+--------------------------+
```

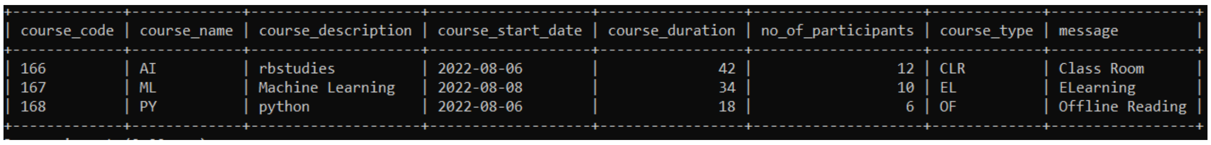
# EXERCISE 17
Pre-requisite: Use the Course_Info and Course_Fees table.

- **Insert 2 records in course_fees   table with base fees as null.**
- **Insert 2 records in course_fees   table with base fees as 300 and 175.**
- **Insert 3 records in course_info table each course with course type CLR,EL, OF**

<u>Problem Statement:</u>

Write a query which will display the course type and the appropriate message as mentioned below.

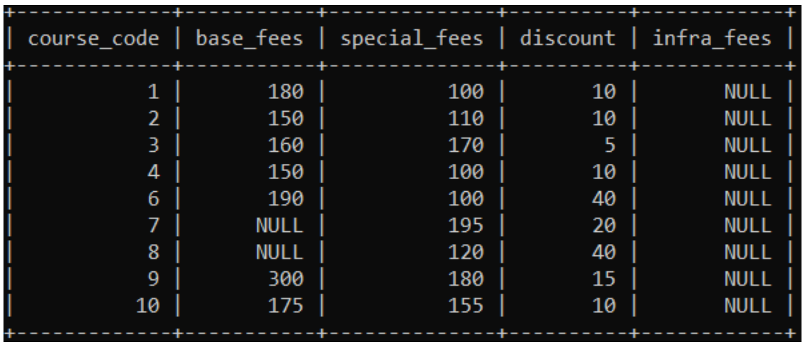
INSERT INTO course_info(course_code,course_name,course_description,course_start_date,course_duration,no_of_participants,course_type,message)
    -> values('166','AI','rbstudies','2022-08-06',42,12,'CLR','Class Room'),
    -> ('167','ML','Machine Learning','2022-08-08',34,10,'EL','ELearning'),
    -> ('168','PY','python','2022-08-06',18,6,'OF','Offline Reading');
SELECT * FROM course_info



```
+-------------+-------------+--------------------+-------------------+-----------------+--------------------+-------------+-----------------+
| course_code | course_name | course_description | course_start_date | course_duration | no_of_participants | course_type | message         |
+-------------+-------------+--------------------+-------------------+-----------------+--------------------+-------------+-----------------+
| 166         | AI          | rbstudies          | 2022-08-06        |              42 |                 12 | CLR         | Class Room      |
| 167         | ML          | Machine Learning   | 2022-08-08        |              34 |                 10 | EL          | ELearning       |
| 168         | PY          | python             | 2022-08-06        |              18 |                  6 | OF          | Offline Reading |
+-------------+-------------+--------------------+-------------------+-----------------+--------------------+-------------+-----------------+
```

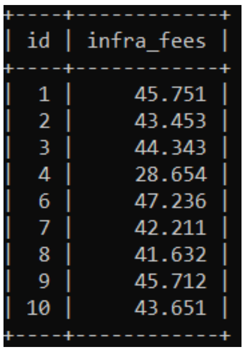
# EXERCISE 13
Display the average infra fees of the courses.

CREATE TABLE newcol(
    -> id INT NOT NULL PRIMARY KEY,
    -> infra_fees DECIMAL(5,3));
DESC newcol;
ALTER TABLE course_fees ADD infra_fees DECIMAL(5,3);
SELECT * FROM course_fees;



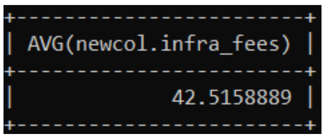
```
+-------------+-----------+--------------+----------+------------+
| course_code | base_fees | special_fees | discount | infra_fees |
+-------------+-----------+--------------+----------+------------+
|           1 |       180 |          100 |       10 |       NULL |
|           2 |       150 |          110 |       10 |       NULL |
|           3 |       160 |          170 |        5 |       NULL |
|           4 |       150 |          100 |       10 |       NULL |
|           6 |       190 |          100 |       40 |       NULL |
|           7 |      NULL |          195 |       20 |       NULL |
|           8 |      NULL |          120 |       40 |       NULL |
|           9 |       300 |          180 |       15 |       NULL |
|          10 |       175 |          155 |       10 |       NULL |
+-------------+-----------+--------------+----------+------------+
```

SELECT * FROM newcol;



```
+----+------------+
| id | infra_fees |
+----+------------+
|  1 |     45.751 |
|  2 |     43.453 |
|  3 |     44.343 |
|  4 |     28.654 |
|  6 |     47.236 |
|  7 |     42.211 |
|  8 |     41.632 |
|  9 |     45.712 |
| 10 |     43.651 |
+----+------------+
```

SELECT AVG(newcol.infra_fees) FROM course_fees INNER JOIN newcol ON course_fees.course_code=newcol.id;

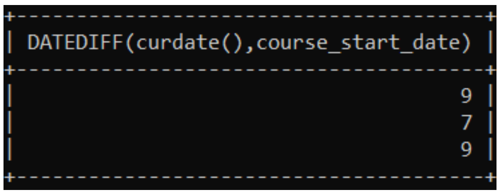


```
+------------------------+
| AVG(newcol.infra_fees) |
+------------------------+
|             42.5158889 |
+------------------------+
```

# EXERCISE 14
Problem Statement:

Develop a query which will display the course name and the number of days between the current date and course start date in Course_Info table

SELECT DATEDIFF(curdate(),course_start_date) FROM course_info;



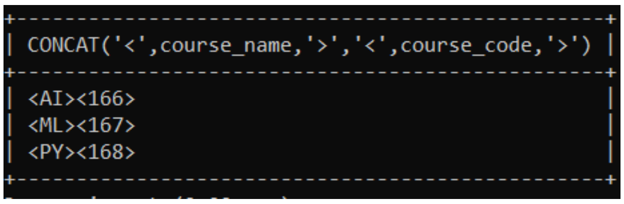
```
+-------------------------------------+
| DATEDIFF(curdate(),course_start_date) |
+-------------------------------------+
|                                   9 |
|                                   7 |
|                                   9 |
+-------------------------------------+
```

# EXERCISE 15
Problem Statement:

Develop a query which will concatenate the Course Name and Course Code in the following format and display all the courses in the course_info table.

"< Course Name><Course Code>"

SELECT CONCAT('<',course_name,'>','<',course_code,'>') FROM course_info;

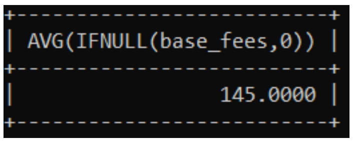


```
+----------------------------------------------+
| CONCAT('<',course_name,'>','<',course_code,'>') |
+----------------------------------------------+
| <AI><166>                                    |
| <ML><167>                                    |
| <PY><168>                                    |
+----------------------------------------------+
```

# EXERCISE 16

Develop a query calculate average of all the base fees, any records whose base fee is null needs to be considered as zero.

SELECT AVG(IFNULL(base_fees,0)) FROM course_fees;



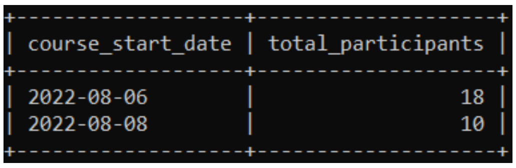
```
+-------------------------+
| AVG(IFNULL(base_fees,0)) |
+-------------------------+
|                145.0000 |
+-------------------------+
```

Ref Exercise 17 above in between Exercise 12 & 13

# EXERCISE 18

Problem Statement: Develop a query which would retrieve the total number of students enrolled for courses on a specific date grouped by course start date and display course start date and total number of students.

```
SELECT course_start_date, SUM(no_of_participants) AS total_participants
    -> FROM course_info
    -> GROUP BY course_start_date;
```

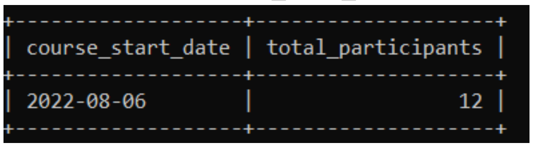


```
+-------------------+--------------------+
| course_start_date | total_participants |
+-------------------+--------------------+
| 2022-08-06        |                 18 |
| 2022-08-08        |                 10 |
+-------------------+--------------------+
```

# EXERCISE 19

Problem Statement: Develop a query which would retrieve the total number of students enrolled for courses where course_type="CLR" grouped by course start date and display course start date and total number of students.

```
SELECT course_start_date, SUM(no_of_participants) AS total_participants
    -> FROM course_info
    -> WHERE course_type='CLR'
    -> GROUP BY course_start_date;
```

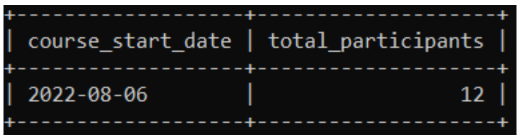


```
+-------------------+--------------------+
| course_start_date | total_participants |
+-------------------+--------------------+
| 2022-08-06        |                 12 |
+-------------------+--------------------+
```

# EXERCISE 20

Problem Statement: Develop a query which would retrieve the total number of students enrolled for courses where course_type="CLR" grouped by course start date and display course start date and total number of students where the total number of students > 10.

```
SELECT course_start_date, SUM(no_of_participants) AS total_participants
   -> FROM course_info
   -> WHERE course_type='CLR'
   -> GROUP BY course_start_date
   -> HAVING SUM(no_of_participants)>10;
```

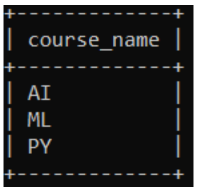


```
+-------------------+--------------------+
| course_start_date | total_participants |
+-------------------+--------------------+
| 2022-08-06        |                 12 |
+-------------------+--------------------+
```

# EXERCISE 21

Develop a query which displays all the courses in increasing order of course duration.

```
SELECT course_name  FROM course_info ORDER BY course_duration DESC;
```

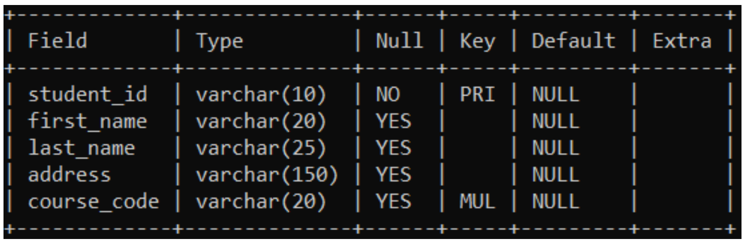


```
+-------------+
| course_name |
+-------------+
| AI          |
| ML          |
| PY          |
+-------------+
```

# EXERCISE 22

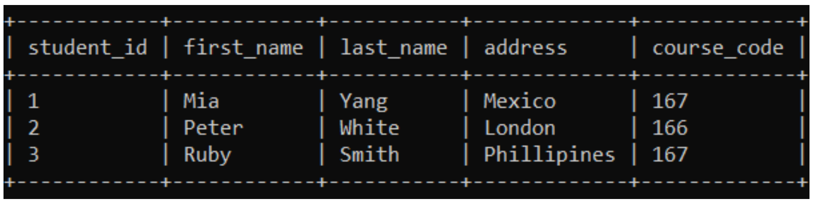
Write a query to fetch student ID, first name, last name, and course code for students who have enrolled for course having course_code as 167. Student_Info and student_courses to be queried.

```
ALTER TABLE student_info ADD course_code VARCHAR(20);
ALTER TABLE student_info ADD CONSTRAINT FOREIGN KEY (course_code) REFERENCES
course_info(course_code);
SELECT * FROM student_info;
DESC student_info;
```



```
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| student_id  | varchar(10)  | NO   | PRI | NULL    |       |
| first_name  | varchar(20)  | YES  |     | NULL    |       |
| last_name   | varchar(25)  | YES  |     | NULL    |       |
| address     | varchar(150) | YES  |     | NULL    |       |
| course_code | varchar(20)  | YES  | MUL | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
```

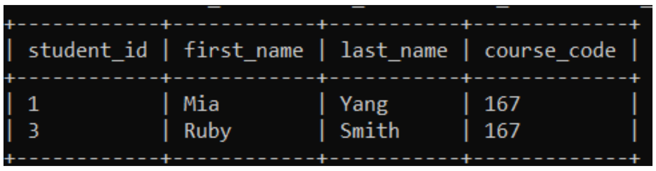
```
INSERT INTO student_info (student_id, first_name, last_name, address, course_code)
   -> values('1','Mia','Yang','Mexico',167),
   -> ('2','Peter','White','London',166),
   -> ('3','Ruby','Smith','Phillipines',167);
```



```
+------------+------------+-----------+-------------+-------------+
| student_id | first_name | last_name | address     | course_code |
+------------+------------+-----------+-------------+-------------+
| 1          | Mia        | Yang      | Mexico      | 167         |
| 2          | Peter      | White     | London      | 166         |
| 3          | Ruby       | Smith     | Phillipines | 167         |
+------------+------------+-----------+-------------+-------------+
```

Krithika Devi Chandran

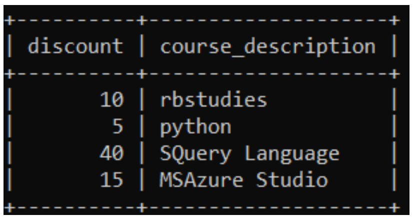
```
SELECT
student_info.student_id,student_info.first_name,student_info.last_name,student_info.course_code
FROM student_info
   -> INNER JOIN course_info
   -> ON student_info.course_code=course_info.course_code WHERE
student_info.course_code='167';
```



```
+------------+------------+-----------+-------------+
| student_id | first_name | last_name | course_code |
+------------+------------+-----------+-------------+
| 1          | Mia        | Yang      | 167         |
| 3          | Ruby       | Smith     | 167         |
+------------+------------+-----------+-------------+
```

# EXERCISE 23
Write a query to display the discount offered on the courses along with course descriptions.

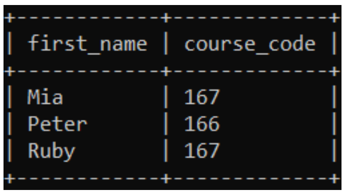
```
SELECT * FROM course_fees;
ALTER TABLE course_fees DELETE infra_fees;
INSERT INTO course_fees (course_code, base_fees, special_fees, discount)
   -> values(166,170,160,10),
   -> (168,162,140,5);
INSERT INTO course_info values
   -> ('6','SQL','SQuery Language','2022-07-22',64,20,'EL','ELearning'),
   -> ('9','MSAS', 'MSAzure Studio','2022-08-02',246,25,'CLR','Class Room');
SELECT course_fees.discount, course_info.course_description
   -> FROM course_fees
   -> INNER JOIN course_info
   -> ON course_fees.course_code = course_info.course_code;
```



```
+----------+--------------------+
| discount | course_description |
+----------+--------------------+
|       10 | rbstudies          |
|        5 | python             |
|       40 | SQuery Language    |
|       15 | MSAzure Studio     |
+----------+--------------------+
```

# EXERCISE 24
Problem Statement: Write a query to fetch first names of the students along with the course codes of the courses they have enrolled in.

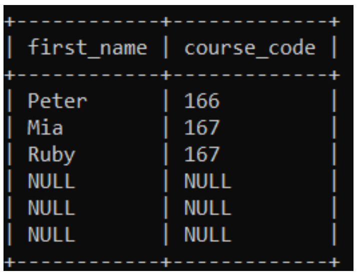
```
SELECT student_info.first_name,student_info.course_code
   -> FROM student_info LEFT JOIN course_info
   -> ON student_info.course_code = course_info.course_code;
```



```
+------------+-------------+
| first_name | course_code |
+------------+-------------+
| Mia        | 167         |
| Peter      | 166         |
| Ruby       | 167         |
+------------+-------------+
```

```
SELECT student_info.first_name,student_info.course_code
   -> FROM student_info RIGHT JOIN course_info
```

-> ON student_info.course_code = course_info.course_code;



```
+------------+-------------+
| first_name | course_code |
+------------+-------------+
| Peter      | 166         |
| Mia        | 167         |
| Ruby       | 167         |
| NULL       | NULL        |
| NULL       | NULL        |
| NULL       | NULL        |
+------------+-------------+
```

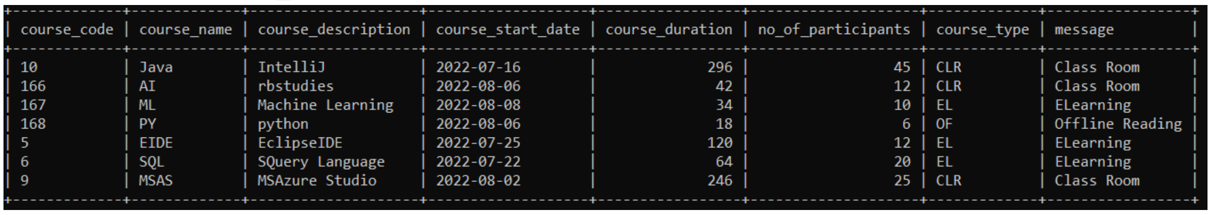
# EXERCISE 25

Pre-Requisite:

- **Insert the following records**
- **Add two new courses in course_info table**
- **Add the course fees for the two courses in course_Fees with fees amount < 1500**
- **Enroll two students to the newly added courses**

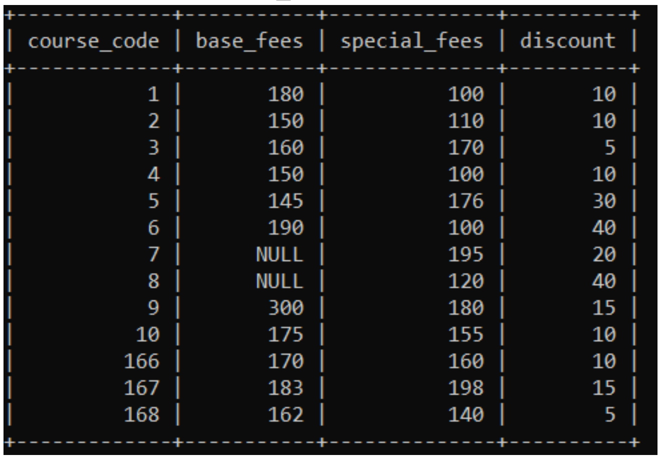
Problem Statement: Write a query which fetches the student id for students who have enrolled for at least one course whose fees is less than 1500.

```
SELECT * FROM course_info;
INSERT INTO course_info VALUES(
   -> '10','Java','IntelliJ','2022-07-16',296, 45, 'CLR','Class Room'),
   -> ('5','EIDE','EclipseIDE','2022-07-25',120,12,'EL','ELearning');
SELECT * FROM course_info;
```



```
+-------------+-------------+--------------------+-------------------+-----------------+--------------------+-------------+-----------------+
| course_code | course_name | course_description | course_start_date | course_duration | no_of_participants | course_type | message         |
+-------------+-------------+--------------------+-------------------+-----------------+--------------------+-------------+-----------------+
| 10          | Java        | IntelliJ           | 2022-07-16        |             296 |                 45 | CLR         | Class Room      |
| 166         | AI          | rbstudies          | 2022-08-06        |              42 |                 12 | CLR         | Class Room      |
| 167         | ML          | Machine Learning   | 2022-08-08        |              34 |                 10 | EL          | ELearning       |
| 168         | PY          | python             | 2022-08-06        |              18 |                  6 | OF          | Offline Reading |
| 5           | EIDE        | EclipseIDE         | 2022-07-25        |             120 |                 12 | EL          | ELearning       |
| 6           | SQL         | SQuery Language    | 2022-07-22        |              64 |                 20 | EL          | ELearning       |
| 9           | MSAS        | MSAzure Studio     | 2022-08-02        |             246 |                 25 | CLR         | Class Room      |
+-------------+-------------+--------------------+-------------------+-----------------+--------------------+-------------+-----------------+
```

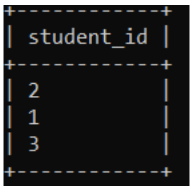
```
SELECT * FROM course_fees;
INSERT INTO course_fees VALUES(
   -> 167,183,198,15),
   -> (5,145,176,30);
SELECT * FROM course_fees;
```



```
+-------------+-----------+--------------+----------+
| course_code | base_fees | special_fees | discount |
+-------------+-----------+--------------+----------+
|           1 |       180 |          100 |       10 |
|           2 |       150 |          110 |       10 |
|           3 |       160 |          170 |        5 |
|           4 |       150 |          100 |       10 |
|           5 |       145 |          176 |       30 |
|           6 |       190 |          100 |       40 |
|           7 |      NULL |          195 |       20 |
|           8 |      NULL |          120 |       40 |
|           9 |       300 |          180 |       15 |
|          10 |       175 |          155 |       10 |
|         166 |       170 |          160 |       10 |
|         167 |       183 |          198 |       15 |
|         168 |       162 |          140 |        5 |
+-------------+-----------+--------------+----------+
```

SELECT * FROM student_info;

```
DESC student_info;
SELECT DISTINCT student_info.student_id FROM student_info
   -> INNER JOIN course_fees
   -> ON student_info.course_code=course_fees.course_code
   -> WHERE (base_fees+special_fees)*(1-discount/100)<1500;
```

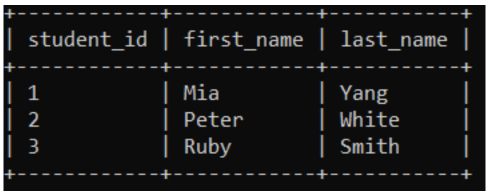


```
+------------+
| student_id |
+------------+
| 2          |
| 1          |
| 3          |
+------------+
```

# EXERCISE 26

Write a query which fetches the student id and student name for students who have enrolled for at least one course whose fees is less than 1500.

```
SELECT DISTINCT student_info.student_id,student_info.first_name,student_info.last_name
   -> FROM student_info
   -> INNER JOIN course_fees ON student_info.course_code=course_fees.course_code
   -> WHERE (base_fees+special_fees)*(1-discount/100)<1500;
```



```
+------------+------------+-----------+
| student_id | first_name | last_name |
+------------+------------+-----------+
| 1          | Mia        | Yang      |
| 2          | Peter      | White     |
| 3          | Ruby       | Smith     |
+------------+------------+-----------+
```