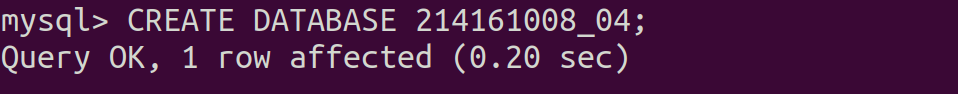
**Database Lab Take Home Assignment - 4**

**Roll No. 214161008**

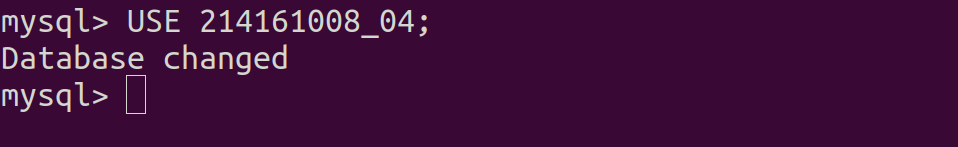
**DATABASE Creation**

**CREATE DATABASE 214161008\_04;**

****

**DATABASE SELECTION**

**USE 214161008\_04;**

****

1. **Construct a table with following details given below:**

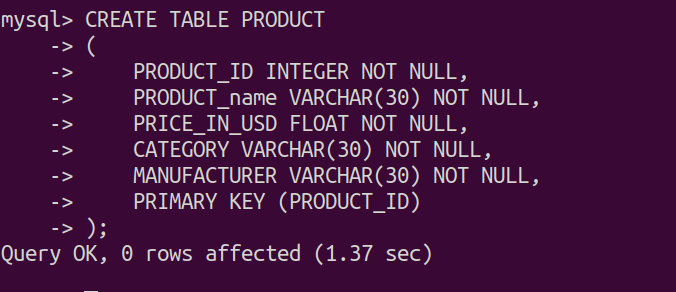
**Product (Pid, Pname, Price, Category, Manufacturer**

**Purchase (ProdId, buyerName, date, price)**

**Pid: is primary key for the product table.**

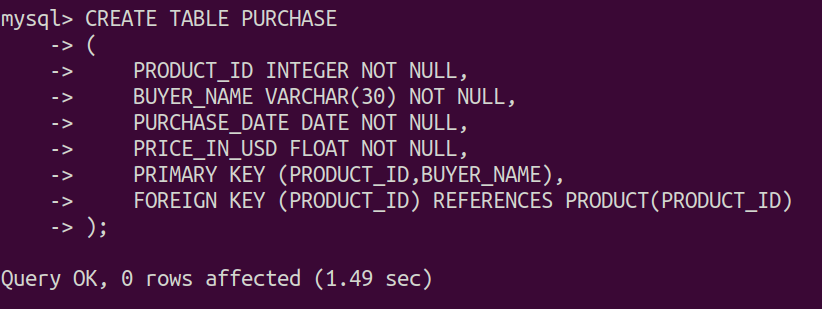
**PRODUCT Table Creation**

| **CREATE** **TABLE** PRODUCT (  PRODUCT\_ID INTEGER **NOT** NULL,  PRODUCT\_name VARCHAR(30) **NOT** NULL,  PRICE\_IN\_USD FLOAT **NOT** NULL,  **CATEGORY** VARCHAR(30) **NOT** NULL,  MANUFACTURER VARCHAR(30) **NOT** NULL,  PRIMARY **KEY** (PRODUCT\_ID) ); |
| --- |



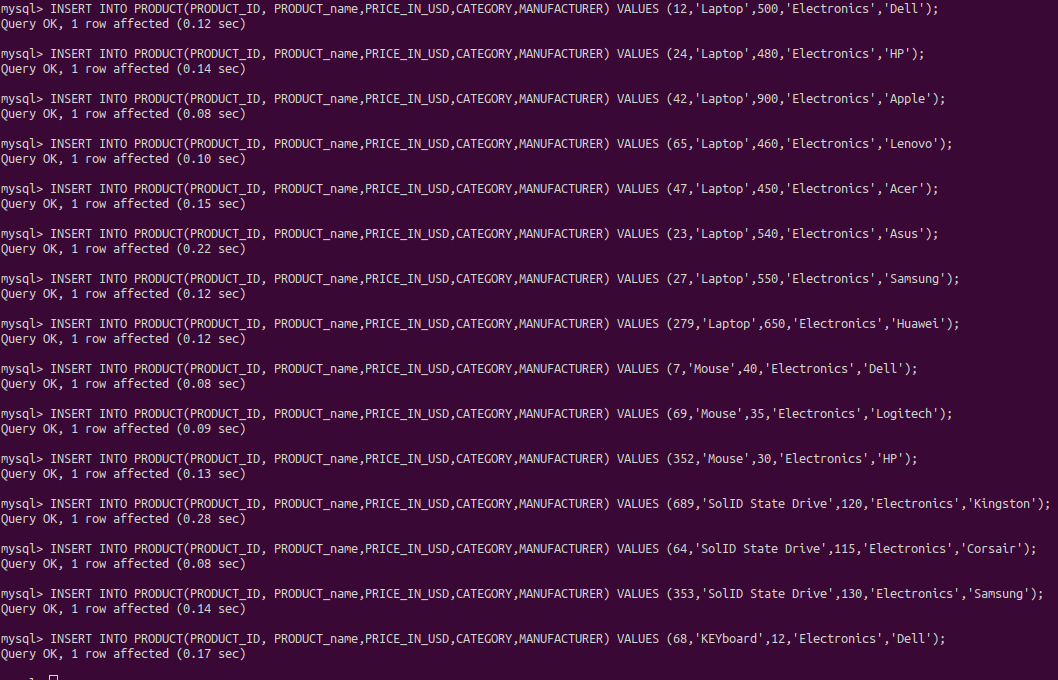
**PURCHASE Table Creation**

| **CREATE** **TABLE** PURCHASE (  PRODUCT\_ID INTEGER **NOT** NULL,  BUYER\_NAME VARCHAR(30) **NOT** NULL,  PURCHASE\_DATE DATE **NOT** NULL,  PRICE\_IN\_USD FLOAT **NOT** NULL,  PRIMARY **KEY** (PRODUCT\_ID,BUYER\_NAME),  FOREIGN **KEY** (PRODUCT\_ID) **REFERENCES** PRODUCT(PRODUCT\_ID) ); |
| --- |



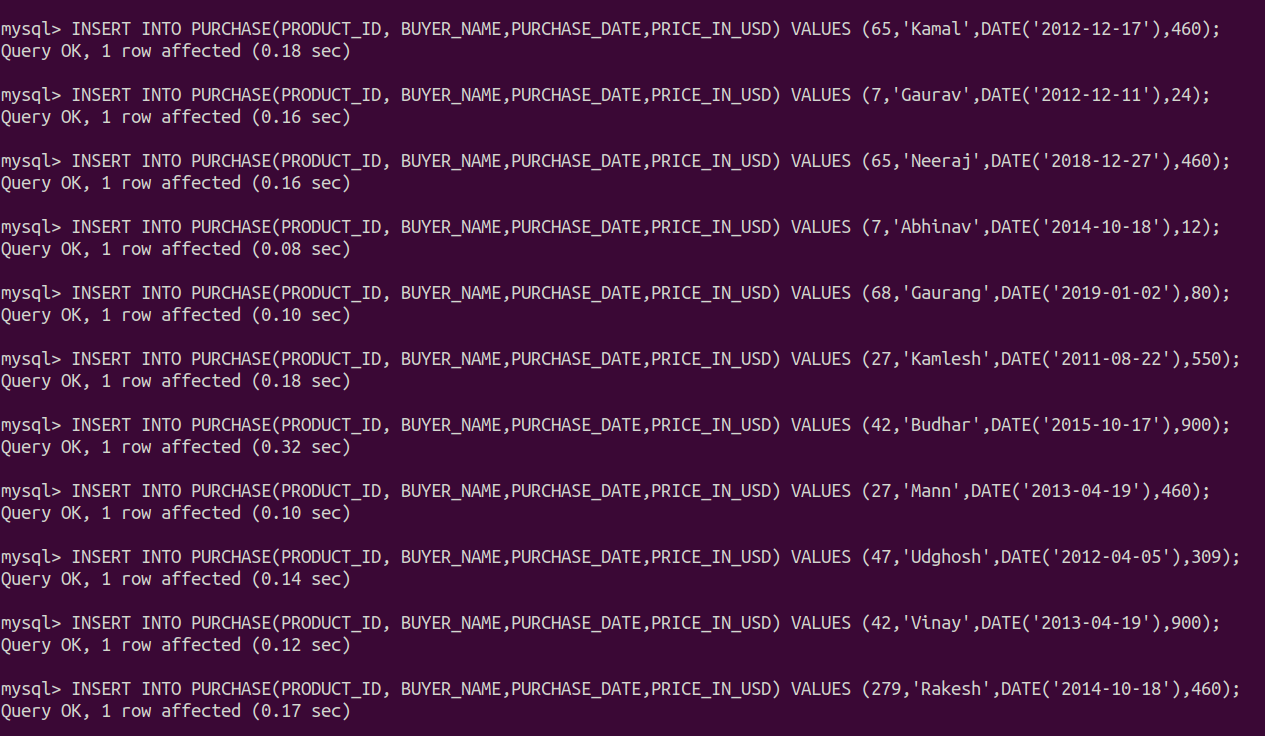
**INSERTION INTO PRODUCT Table**

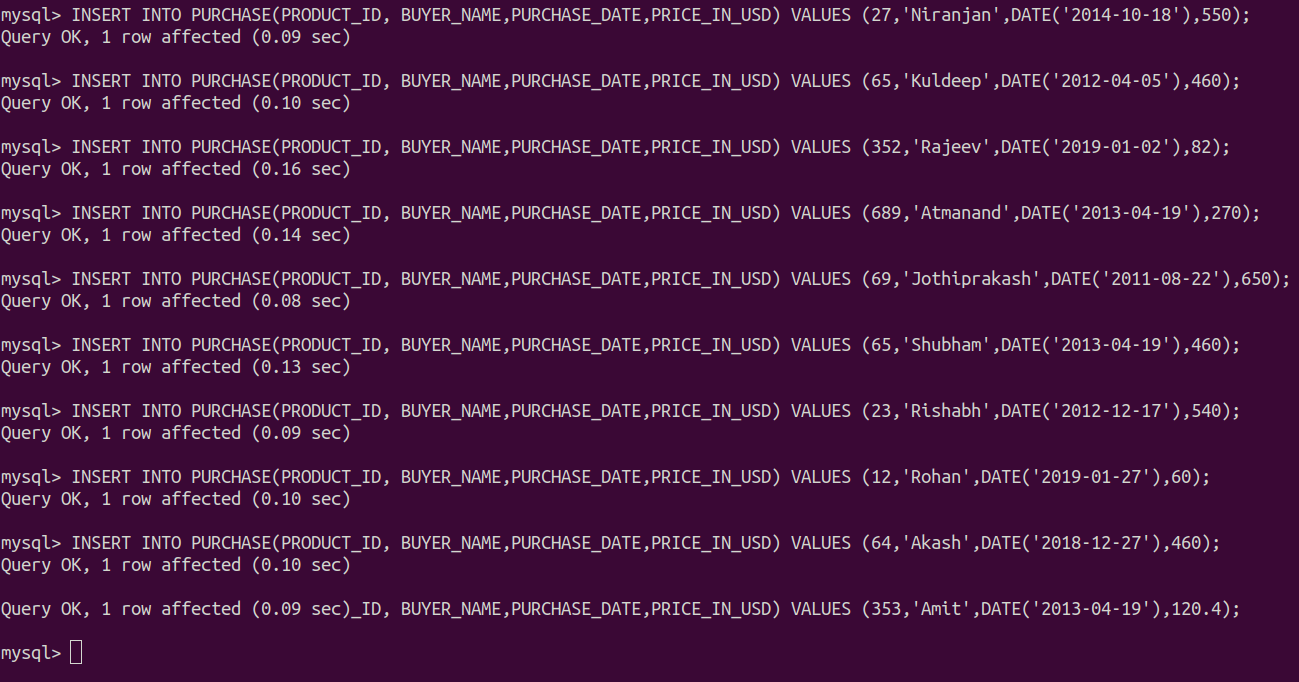
| **INSERT** **INTO** PRODUCT(PRODUCT\_ID, PRODUCT\_name,PRICE\_IN\_USD,**CATEGORY**,MANUFACTURER) **VALUES** (12,'Laptop',500,'Electronics','Dell'); **INSERT** **INTO** PRODUCT(PRODUCT\_ID, PRODUCT\_name,PRICE\_IN\_USD,**CATEGORY**,MANUFACTURER) **VALUES** (24,'Laptop',480,'Electronics','HP'); **INSERT** **INTO** PRODUCT(PRODUCT\_ID, PRODUCT\_name,PRICE\_IN\_USD,**CATEGORY**,MANUFACTURER) **VALUES** (42,'Laptop',900,'Electronics','Apple'); **INSERT** **INTO** PRODUCT(PRODUCT\_ID, PRODUCT\_name,PRICE\_IN\_USD,**CATEGORY**,MANUFACTURER) **VALUES** (65,'Laptop',460,'Electronics','Lenovo'); **INSERT** **INTO** PRODUCT(PRODUCT\_ID, PRODUCT\_name,PRICE\_IN\_USD,**CATEGORY**,MANUFACTURER) **VALUES** (47,'Laptop',450,'Electronics','Acer'); **INSERT** **INTO** PRODUCT(PRODUCT\_ID, PRODUCT\_name,PRICE\_IN\_USD,**CATEGORY**,MANUFACTURER) **VALUES** (23,'Laptop',540,'Electronics','Asus'); **INSERT** **INTO** PRODUCT(PRODUCT\_ID, PRODUCT\_name,PRICE\_IN\_USD,**CATEGORY**,MANUFACTURER) **VALUES** (27,'Laptop',550,'Electronics','Samsung'); **INSERT** **INTO** PRODUCT(PRODUCT\_ID, PRODUCT\_name,PRICE\_IN\_USD,**CATEGORY**,MANUFACTURER) **VALUES** (279,'Laptop',650,'Electronics','Huawei'); **INSERT** **INTO** PRODUCT(PRODUCT\_ID, PRODUCT\_name,PRICE\_IN\_USD,**CATEGORY**,MANUFACTURER) **VALUES** (7,'Mouse',40,'Electronics','Dell'); **INSERT** **INTO** PRODUCT(PRODUCT\_ID, PRODUCT\_name,PRICE\_IN\_USD,**CATEGORY**,MANUFACTURER) **VALUES** (69,'Mouse',35,'Electronics','Logitech'); **INSERT** **INTO** PRODUCT(PRODUCT\_ID, PRODUCT\_name,PRICE\_IN\_USD,**CATEGORY**,MANUFACTURER) **VALUES** (352,'Mouse',30,'Electronics','HP'); **INSERT** **INTO** PRODUCT(PRODUCT\_ID, PRODUCT\_name,PRICE\_IN\_USD,**CATEGORY**,MANUFACTURER) **VALUES** (689,'SolID State Drive',120,'Electronics','Kingston'); **INSERT** **INTO** PRODUCT(PRODUCT\_ID, PRODUCT\_name,PRICE\_IN\_USD,**CATEGORY**,MANUFACTURER) **VALUES** (64,'SolID State Drive',115,'Electronics','Corsair'); **INSERT** **INTO** PRODUCT(PRODUCT\_ID, PRODUCT\_name,PRICE\_IN\_USD,**CATEGORY**,MANUFACTURER) **VALUES** (353,'SolID State Drive',130,'Electronics','Samsung'); **INSERT** **INTO** PRODUCT(PRODUCT\_ID, PRODUCT\_name,PRICE\_IN\_USD,**CATEGORY**,MANUFACTURER) **VALUES** (68,'KEYboard',12,'Electronics','Dell'); |
| --- |



**INSERTION INTO PURCHASE Table**

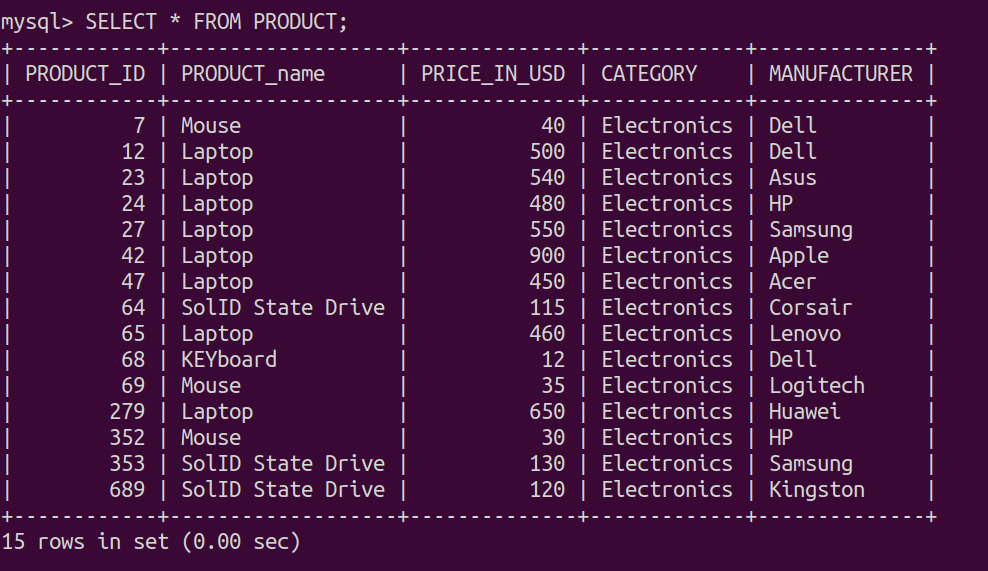
| **INSERT INTO PURCHASE(PRODUCT\_ID, BUYER\_NAME,PURCHASE\_DATE,PRICE\_IN\_USD) VALUES (65,'Kamal',DATE('2012-12-17'),460); INSERT INTO PURCHASE(PRODUCT\_ID, BUYER\_NAME,PURCHASE\_DATE,PRICE\_IN\_USD) VALUES (7,'Gaurav',DATE('2012-12-11'),24); INSERT INTO PURCHASE(PRODUCT\_ID, BUYER\_NAME,PURCHASE\_DATE,PRICE\_IN\_USD) VALUES (65,'Neeraj',DATE('2018-12-27'),460); INSERT INTO PURCHASE(PRODUCT\_ID, BUYER\_NAME,PURCHASE\_DATE,PRICE\_IN\_USD) VALUES (7,'Abhinav',DATE('2014-10-18'),12); INSERT INTO PURCHASE(PRODUCT\_ID, BUYER\_NAME,PURCHASE\_DATE,PRICE\_IN\_USD) VALUES (68,'Gaurang',DATE('2019-01-02'),80); INSERT INTO PURCHASE(PRODUCT\_ID, BUYER\_NAME,PURCHASE\_DATE,PRICE\_IN\_USD) VALUES (27,'Kamlesh',DATE('2011-08-22'),550); INSERT INTO PURCHASE(PRODUCT\_ID, BUYER\_NAME,PURCHASE\_DATE,PRICE\_IN\_USD) VALUES (42,'Budhar',DATE('2015-10-17'),900); INSERT INTO PURCHASE(PRODUCT\_ID, BUYER\_NAME,PURCHASE\_DATE,PRICE\_IN\_USD) VALUES (27,'Mann',DATE('2013-04-19'),460); INSERT INTO PURCHASE(PRODUCT\_ID, BUYER\_NAME,PURCHASE\_DATE,PRICE\_IN\_USD) VALUES (47,'Udghosh',DATE('2012-04-05'),309); INSERT INTO PURCHASE(PRODUCT\_ID, BUYER\_NAME,PURCHASE\_DATE,PRICE\_IN\_USD) VALUES (42,'Vinay',DATE('2013-04-19'),900); INSERT INTO PURCHASE(PRODUCT\_ID, BUYER\_NAME,PURCHASE\_DATE,PRICE\_IN\_USD) VALUES (279,'Rakesh',DATE('2014-10-18'),460); INSERT INTO PURCHASE(PRODUCT\_ID, BUYER\_NAME,PURCHASE\_DATE,PRICE\_IN\_USD) VALUES (27,'Niranjan',DATE('2014-10-18'),550); INSERT INTO PURCHASE(PRODUCT\_ID, BUYER\_NAME,PURCHASE\_DATE,PRICE\_IN\_USD) VALUES (65,'Kuldeep',DATE('2012-04-05'),460); INSERT INTO PURCHASE(PRODUCT\_ID, BUYER\_NAME,PURCHASE\_DATE,PRICE\_IN\_USD) VALUES (352,'Rajeev',DATE('2019-01-02'),82); INSERT INTO PURCHASE(PRODUCT\_ID, BUYER\_NAME,PURCHASE\_DATE,PRICE\_IN\_USD) VALUES (689,'Atmanand',DATE('2013-04-19'),270); INSERT INTO PURCHASE(PRODUCT\_ID, BUYER\_NAME,PURCHASE\_DATE,PRICE\_IN\_USD) VALUES (69,'Jothiprakash',DATE('2011-08-22'),650); INSERT INTO PURCHASE(PRODUCT\_ID, BUYER\_NAME,PURCHASE\_DATE,PRICE\_IN\_USD) VALUES (65,'Shubham',DATE('2013-04-19'),460); INSERT INTO PURCHASE(PRODUCT\_ID, BUYER\_NAME,PURCHASE\_DATE,PRICE\_IN\_USD) VALUES (23,'Rishabh',DATE('2012-12-17'),540); INSERT INTO PURCHASE(PRODUCT\_ID, BUYER\_NAME,PURCHASE\_DATE,PRICE\_IN\_USD) VALUES (12,'Rohan',DATE('2019-01-27'),60); INSERT INTO PURCHASE(PRODUCT\_ID, BUYER\_NAME,PURCHASE\_DATE,PRICE\_IN\_USD) VALUES (64,'Akash',DATE('2018-12-27'),460); INSERT INTO PURCHASE(PRODUCT\_ID, BUYER\_NAME,PURCHASE\_DATE,PRICE\_IN\_USD) VALUES (353,'Amit',DATE('2013-04-19'),120.4);** |
| --- |





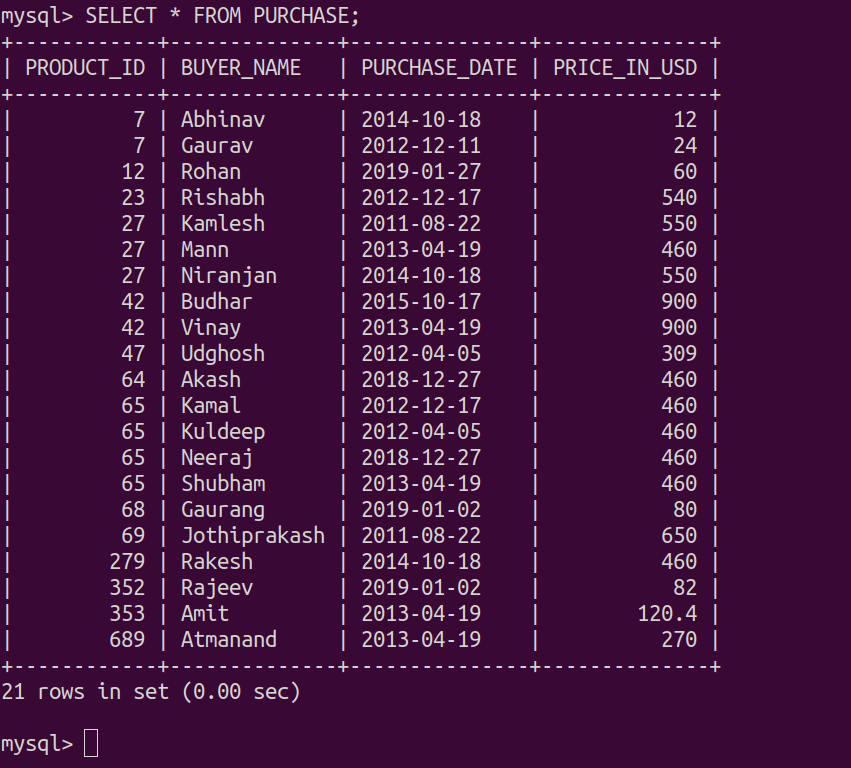
**DATA IN PRODUCT Table**

| **SELECT \* FROM PRODUCT;** |
| --- |

****

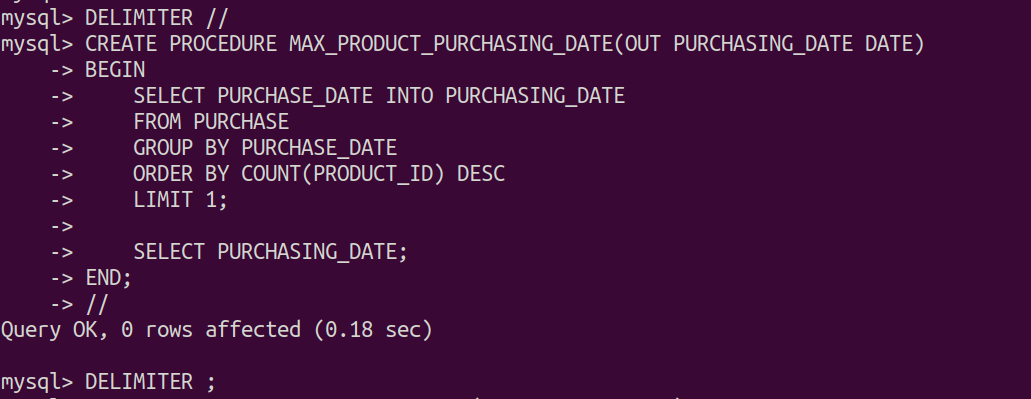
**DATA INTO PURCHASE Table**

| **SELECT \* FROM PURCHASE;** |
| --- |

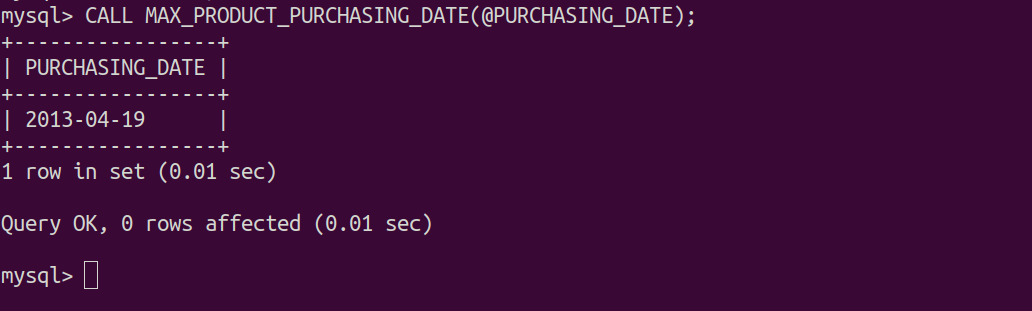
****

1. Write a stored procedures which outputs the date on which maximum number of products was purchased.

| DELIMITER // **CREATE** **PROCEDURE** MAX\_PRODUCT\_PURCHASING\_DATE(**OUT** PURCHASING\_DATE DATE) **BEGIN**  **SELECT** PURCHASE\_DATE **INTO** PURCHASING\_DATE  **FROM** PURCHASE  **GROUP** **BY** PURCHASE\_DATE  **ORDER** **BY** **COUNT**(PRODUCT\_ID) **DESC**  **LIMIT** 1;   **SELECT** PURCHASING\_DATE; **END**; // DELIMITER ; |
| --- |

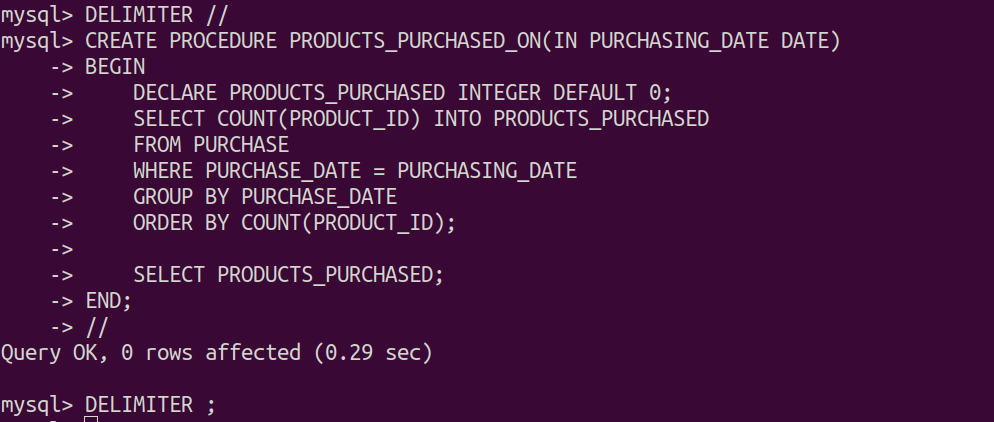


| **CALL** **MAX\_PRODUCT\_PURCHASING\_DATE**(@PURCHASING\_DATE); |
| --- |

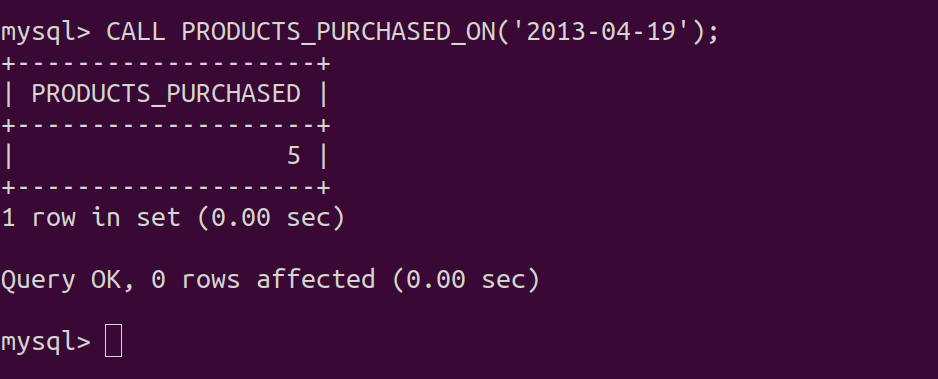


1. Write a stored procedure which returns the number of products by a given date.

| DELIMITER *//* **CREATE** **PROCEDURE** **PRODUCTS\_PURCHASED\_ON**(**IN** PURCHASING\_DATE DATE) **BEGIN**  **DECLARE** **PRODUCTS\_PURCHASED** **INTEGER** **DEFAULT** 0;  **SELECT** COUNT(PRODUCT\_ID) **INTO** PRODUCTS\_PURCHASED  **FROM** PURCHASE  **WHERE** PURCHASE\_DATE = PURCHASING\_DATE  **GROUP** **BY** PURCHASE\_DATE  **ORDER** **BY** COUNT(PRODUCT\_ID);   **SELECT** PRODUCTS\_PURCHASED; **END**; *//* DELIMITER ; |
| --- |

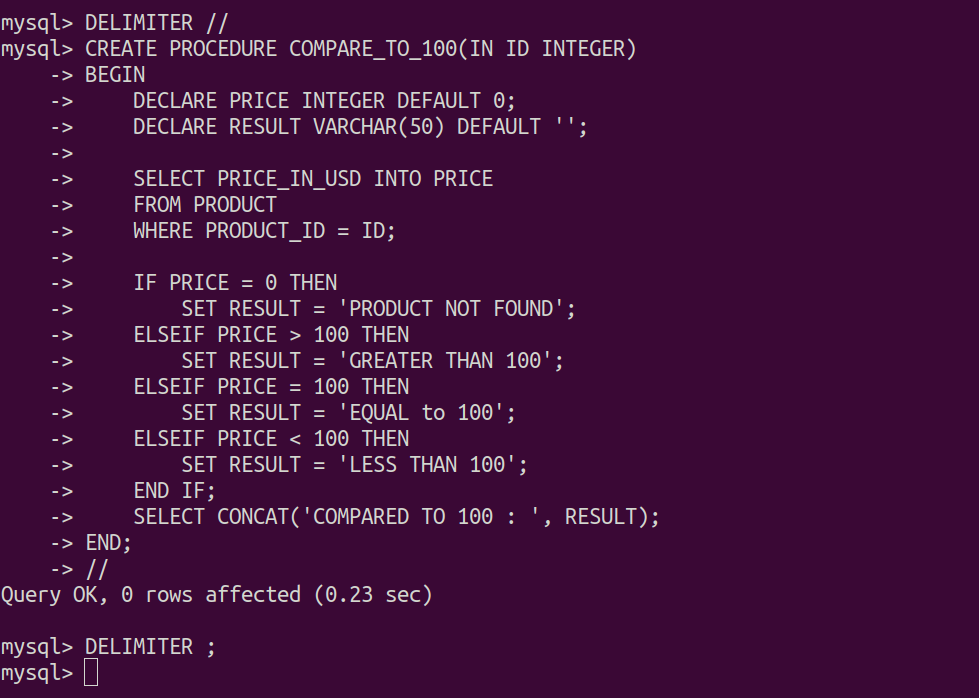


| CALL PRODUCTS\_PURCHASED\_ON('2013-04-19'); |
| --- |

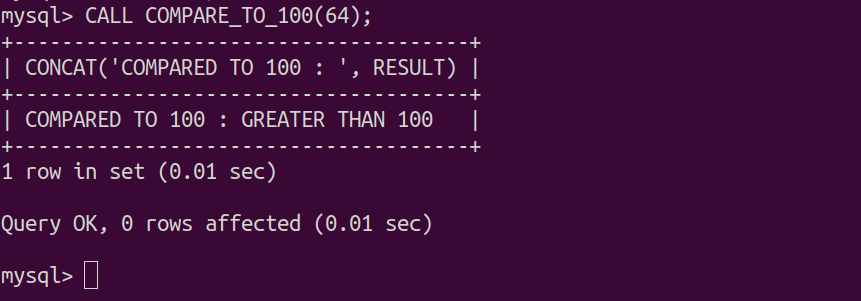


1. Write a store procedure which takes prodId of a product as input and check the price and print ‘GREATER THAN 100’, ‘EQUAL to 100’ or ‘LESS THAN 100’ after comparing the price

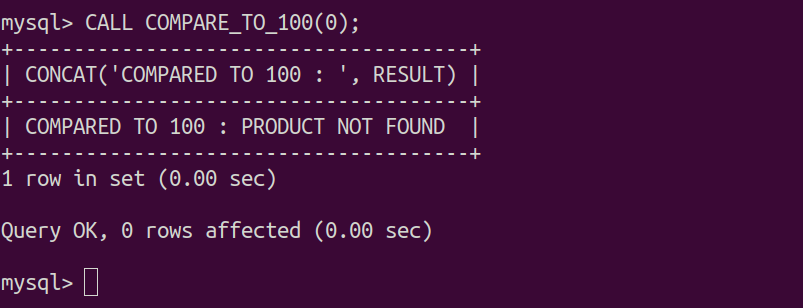
| **DELIMITER *//* CREATE PROCEDURE COMPARE\_TO\_100(IN ID INTEGER) BEGIN  DECLARE PRICE INTEGER DEFAULT 0;  DECLARE RESULT VARCHAR(50) DEFAULT '';   SELECT PRICE\_IN\_USD INTO PRICE  FROM PRODUCT  WHERE PRODUCT\_ID = ID;   IF PRICE = 0 THEN  SET RESULT = 'PRODUCT NOT FOUND';  ELSEIF PRICE > 100 THEN  SET RESULT = 'GREATER THAN 100';  ELSEIF PRICE = 100 THEN  SET RESULT = 'EQUAL to 100';  ELSEIF PRICE < 100 THEN  SET RESULT = 'LESS THAN 100';  END IF;  SELECT CONCAT('COMPARED TO 100 : ', RESULT); END; *//* DELIMITER ;** |
| --- |

****

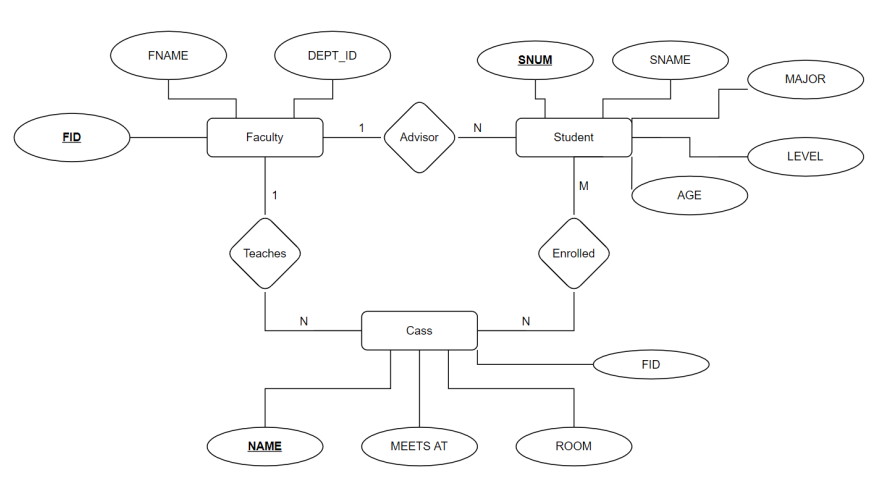
| **CALL COMPARE\_TO\_100*(64)*;** |
| --- |

****

| **CALL COMPARE\_TO\_100*(0)*;** |
| --- |

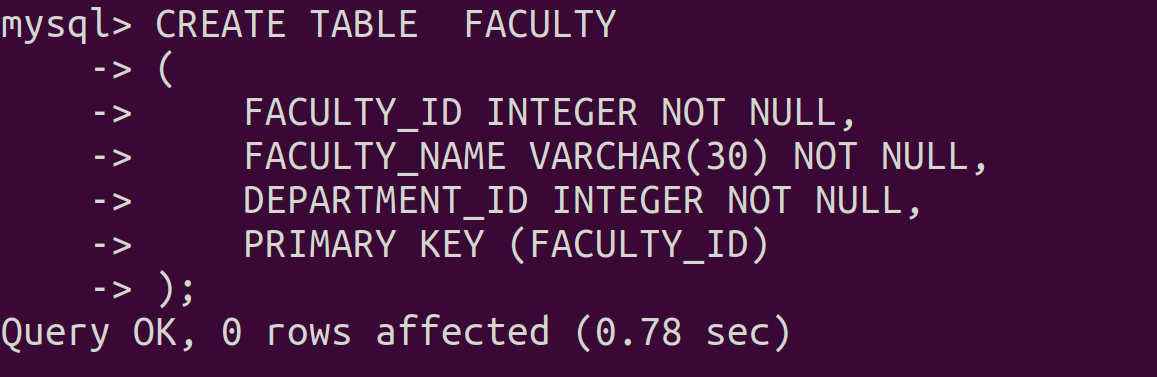
****

1. **Consider the following ER Diagram.**



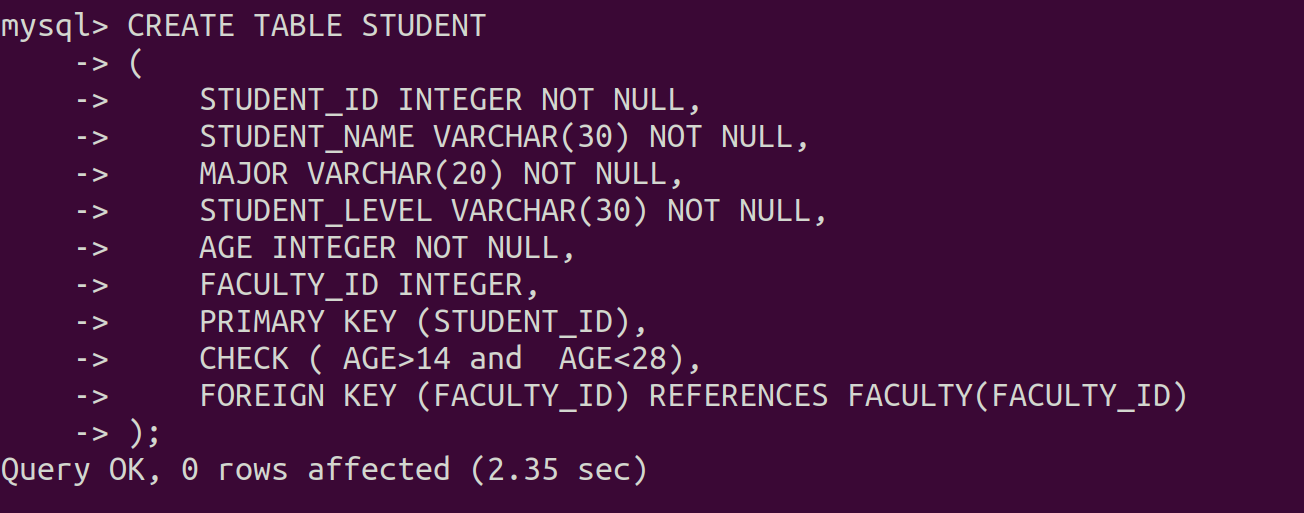
**FACULTY Table Creation**

| **CREATE TABLE FACULTY (  FACULTY\_ID INTEGER NOT NULL,  FACULTY\_NAME VARCHAR(30) NOT NULL,  DEPARTMENT\_ID INTEGER NOT NULL,  PRIMARY KEY (FACULTY\_ID) );** |
| --- |

****

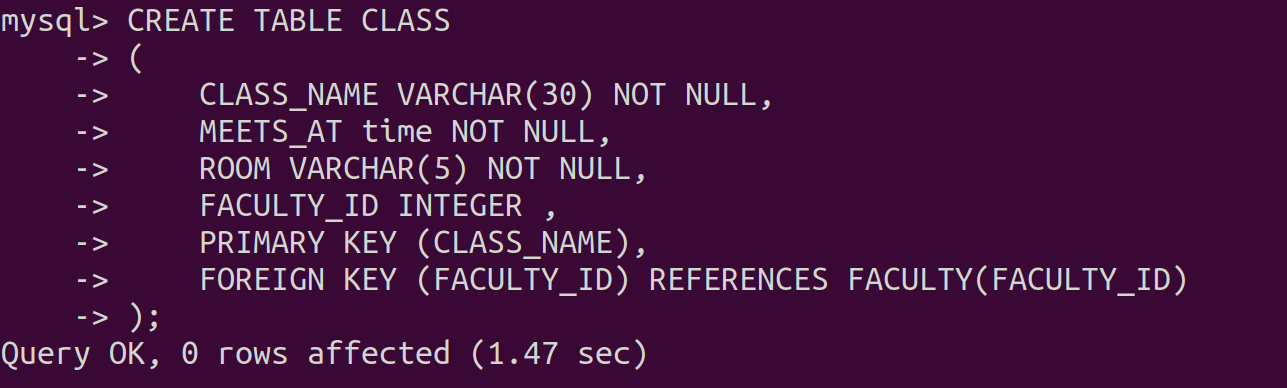
**STUDENT Table Creation**

| **CREATE TABLE STUDENT (  STUDENT\_ID INTEGER NOT NULL,  STUDENT\_NAME VARCHAR(30) NOT NULL,  MAJOR VARCHAR(20) NOT NULL,  STUDENT\_LEVEL VARCHAR(30) NOT NULL,  AGE INTEGER NOT NULL,  FACULTY\_ID INTEGER,  PRIMARY KEY (STUDENT\_ID),  CHECK ( AGE>14 and AGE<28),  FOREIGN KEY (FACULTY\_ID) REFERENCES FACULTY(FACULTY\_ID) );** |
| --- |

****

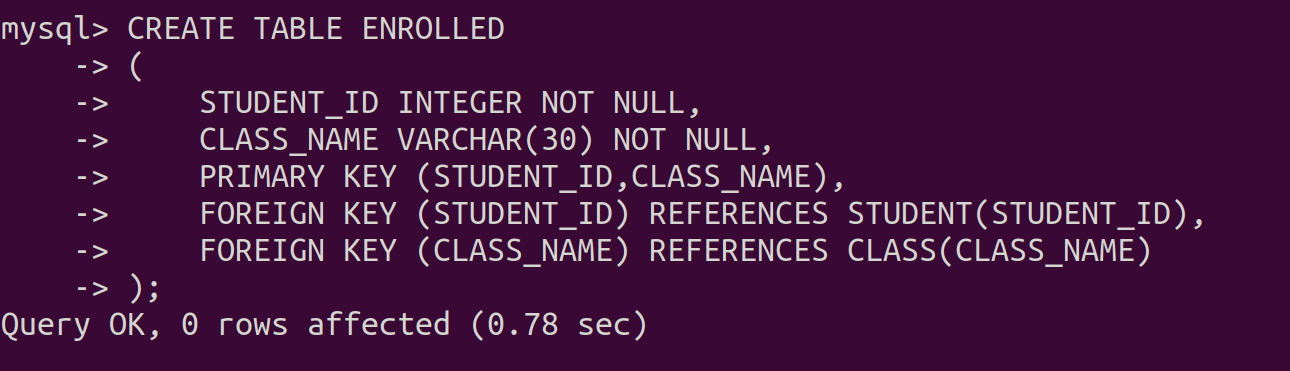
**CLASS Table Creation**

| **CREATE TABLE CLASS (  CLASS\_NAME VARCHAR(30) NOT NULL,  MEETS\_AT time NOT NULL,  ROOM VARCHAR(5) NOT NULL,  FACULTY\_ID INTEGER ,  PRIMARY KEY (CLASS\_NAME),  FOREIGN KEY (FACULTY\_ID) REFERENCES FACULTY(FACULTY\_ID) );** |
| --- |

****

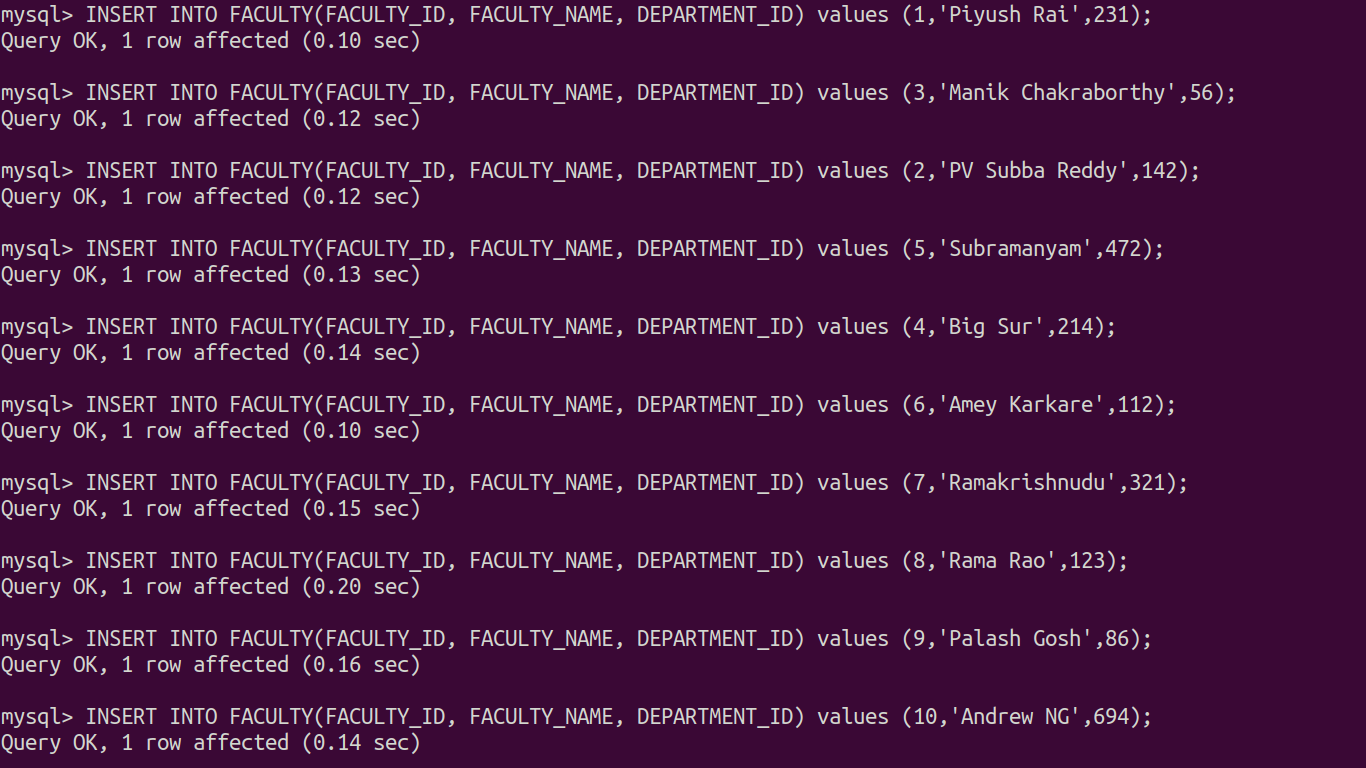
**ENROLLED Table Creation**

| **CREATE TABLE ENROLLED (  STUDENT\_ID INTEGER NOT NULL,  CLASS\_NAME VARCHAR(30) NOT NULL,  PRIMARY KEY (STUDENT\_ID,CLASS\_NAME),  FOREIGN KEY (STUDENT\_ID) REFERENCES STUDENT(STUDENT\_ID),  FOREIGN KEY (CLASS\_NAME) REFERENCES CLASS(CLASS\_NAME) );** |
| --- |

****

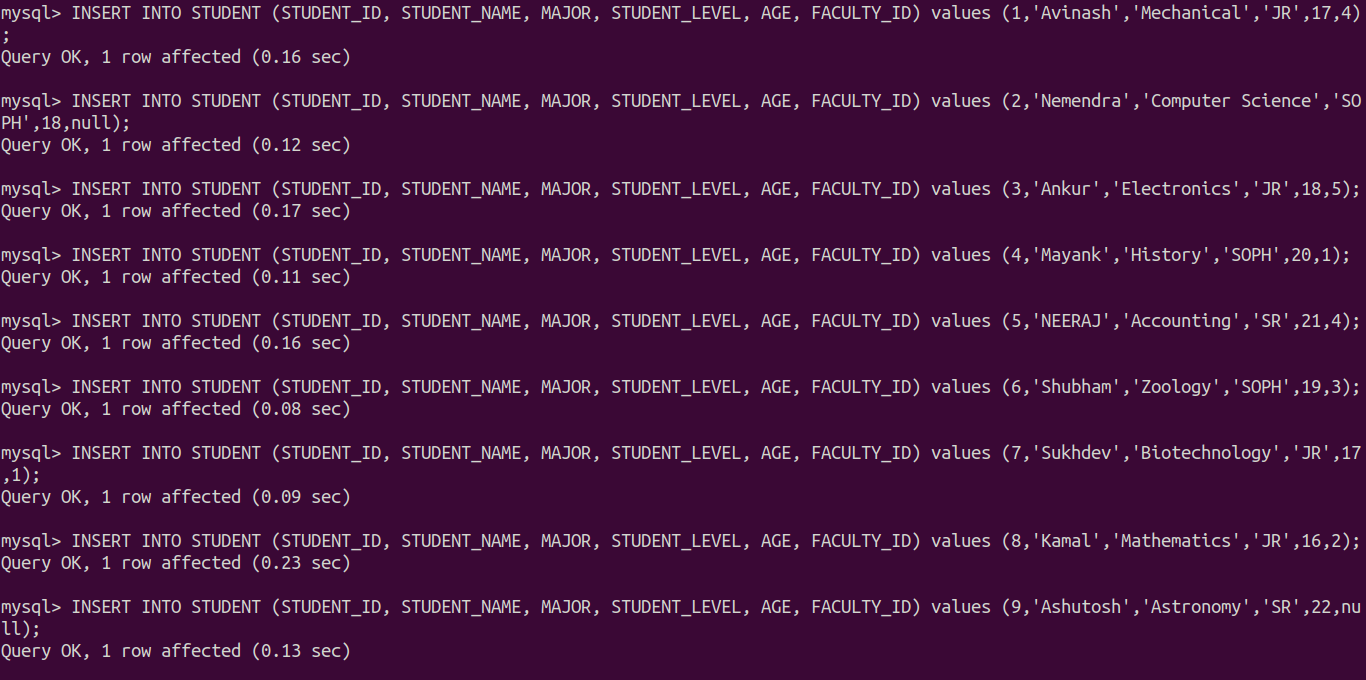
**INSERTION IN FACULTY Table**

| **INSERT INTO FACULTY(FACULTY\_ID, FACULTY\_NAME, DEPARTMENT\_ID) values (1,'Piyush Rai',231); INSERT INTO FACULTY(FACULTY\_ID, FACULTY\_NAME, DEPARTMENT\_ID) values (3,'Manik Chakraborthy',56); INSERT INTO FACULTY(FACULTY\_ID, FACULTY\_NAME, DEPARTMENT\_ID) values (2,'PV Subba Reddy',142); INSERT INTO FACULTY(FACULTY\_ID, FACULTY\_NAME, DEPARTMENT\_ID) values (5,'Subramanyam',472); INSERT INTO FACULTY(FACULTY\_ID, FACULTY\_NAME, DEPARTMENT\_ID) values (4,'Big Sur',214); INSERT INTO FACULTY(FACULTY\_ID, FACULTY\_NAME, DEPARTMENT\_ID) values (6,'Amey Karkare',112); INSERT INTO FACULTY(FACULTY\_ID, FACULTY\_NAME, DEPARTMENT\_ID) values (7,'Ramakrishnudu',321); INSERT INTO FACULTY(FACULTY\_ID, FACULTY\_NAME, DEPARTMENT\_ID) values (8,'Rama Rao',123); INSERT INTO FACULTY(FACULTY\_ID, FACULTY\_NAME, DEPARTMENT\_ID) values (9,'Palash Gosh',86); INSERT INTO FACULTY(FACULTY\_ID, FACULTY\_NAME, DEPARTMENT\_ID) values (10,'Andrew NG',694);** |
| --- |

****

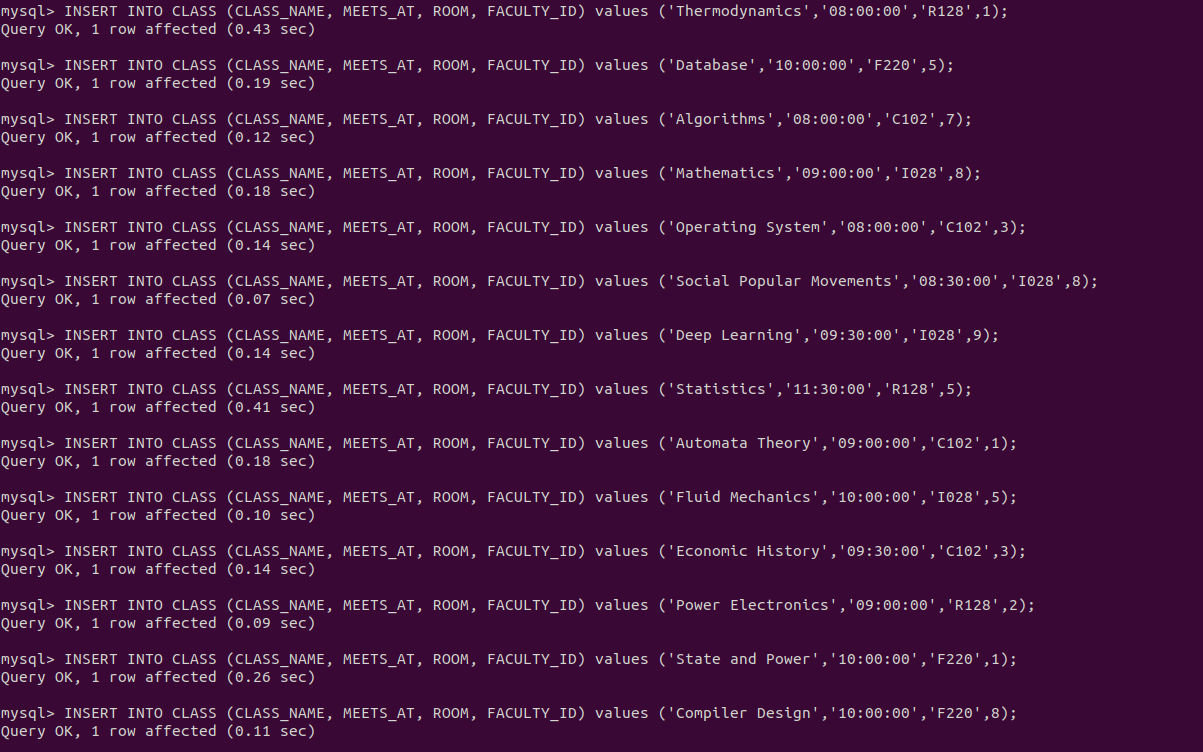
**INSERTION IN STUDENT Table**

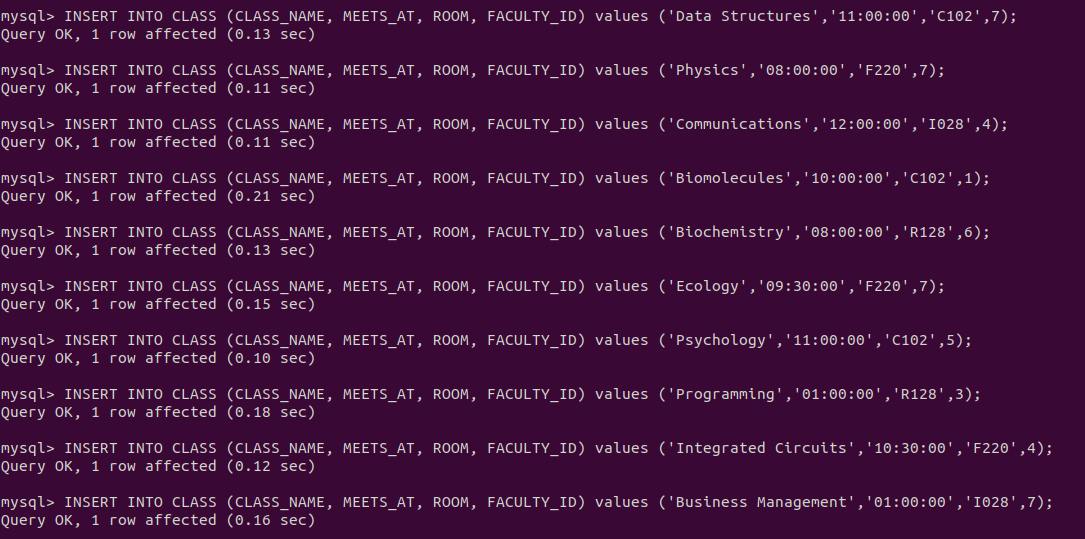
| **INSERT INTO STUDENT (STUDENT\_ID, STUDENT\_NAME, MAJOR, STUDENT\_LEVEL, AGE, FACULTY\_ID) values (1,'Avinash','Mechanical','JR',17,4); INSERT INTO STUDENT (STUDENT\_ID, STUDENT\_NAME, MAJOR, STUDENT\_LEVEL, AGE, FACULTY\_ID) values (2,'Nemendra','Computer Science','SOPH',18,null); INSERT INTO STUDENT (STUDENT\_ID, STUDENT\_NAME, MAJOR, STUDENT\_LEVEL, AGE, FACULTY\_ID) values (3,'Ankur','Electronics','JR',18,5); INSERT INTO STUDENT (STUDENT\_ID, STUDENT\_NAME, MAJOR, STUDENT\_LEVEL, AGE, FACULTY\_ID) values (4,'Mayank','History','SOPH',20,1); INSERT INTO STUDENT (STUDENT\_ID, STUDENT\_NAME, MAJOR, STUDENT\_LEVEL, AGE, FACULTY\_ID) values (5,'NEERAJ','Accounting','SR',21,4); INSERT INTO STUDENT (STUDENT\_ID, STUDENT\_NAME, MAJOR, STUDENT\_LEVEL, AGE, FACULTY\_ID) values (6,'Shubham','Zoology','SOPH',19,3); INSERT INTO STUDENT (STUDENT\_ID, STUDENT\_NAME, MAJOR, STUDENT\_LEVEL, AGE, FACULTY\_ID) values (7,'Sukhdev','Biotechnology','JR',17,1); INSERT INTO STUDENT (STUDENT\_ID, STUDENT\_NAME, MAJOR, STUDENT\_LEVEL, AGE, FACULTY\_ID) values (8,'Kamal','Mathematics','JR',16,2); INSERT INTO STUDENT (STUDENT\_ID, STUDENT\_NAME, MAJOR, STUDENT\_LEVEL, AGE, FACULTY\_ID) values (9,'Ashutosh','Astronomy','SR',22,null);** |
| --- |

****

**INSERTION IN CLASS Table**

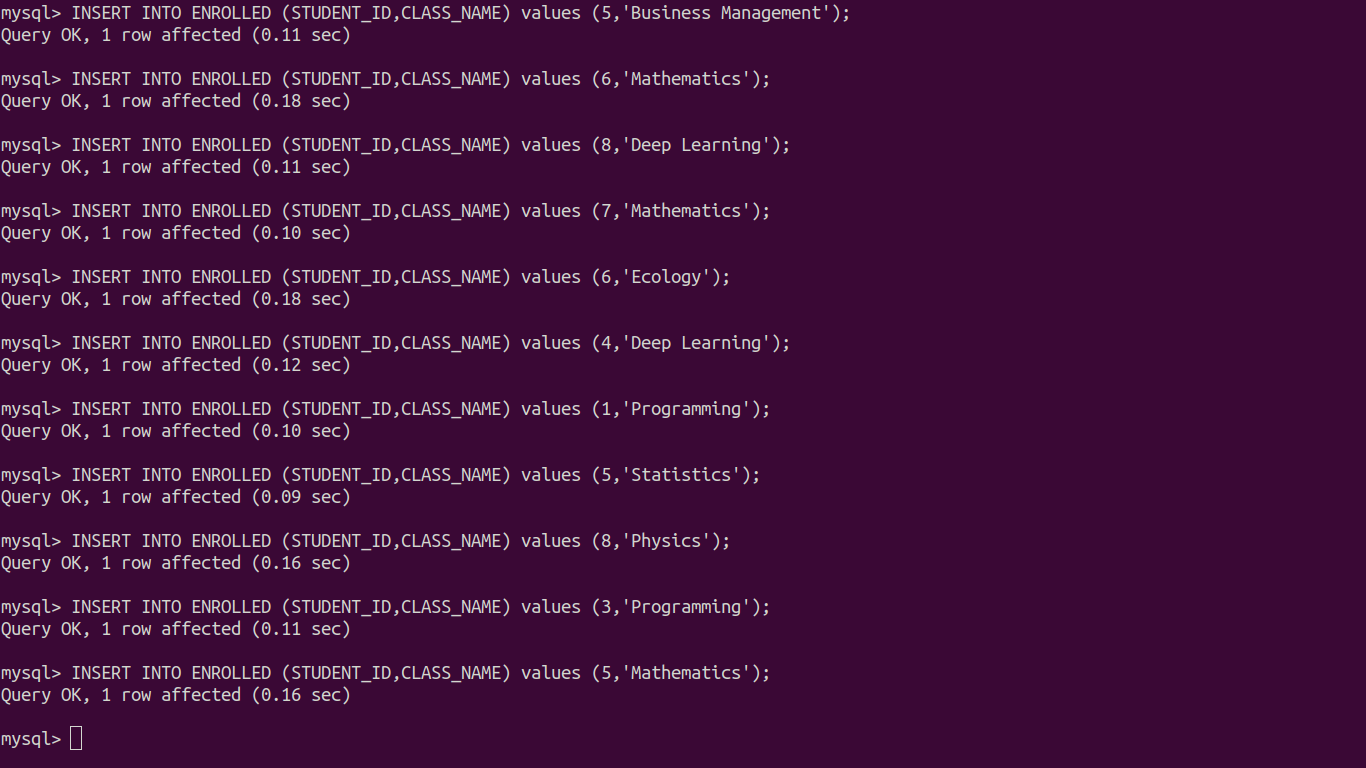
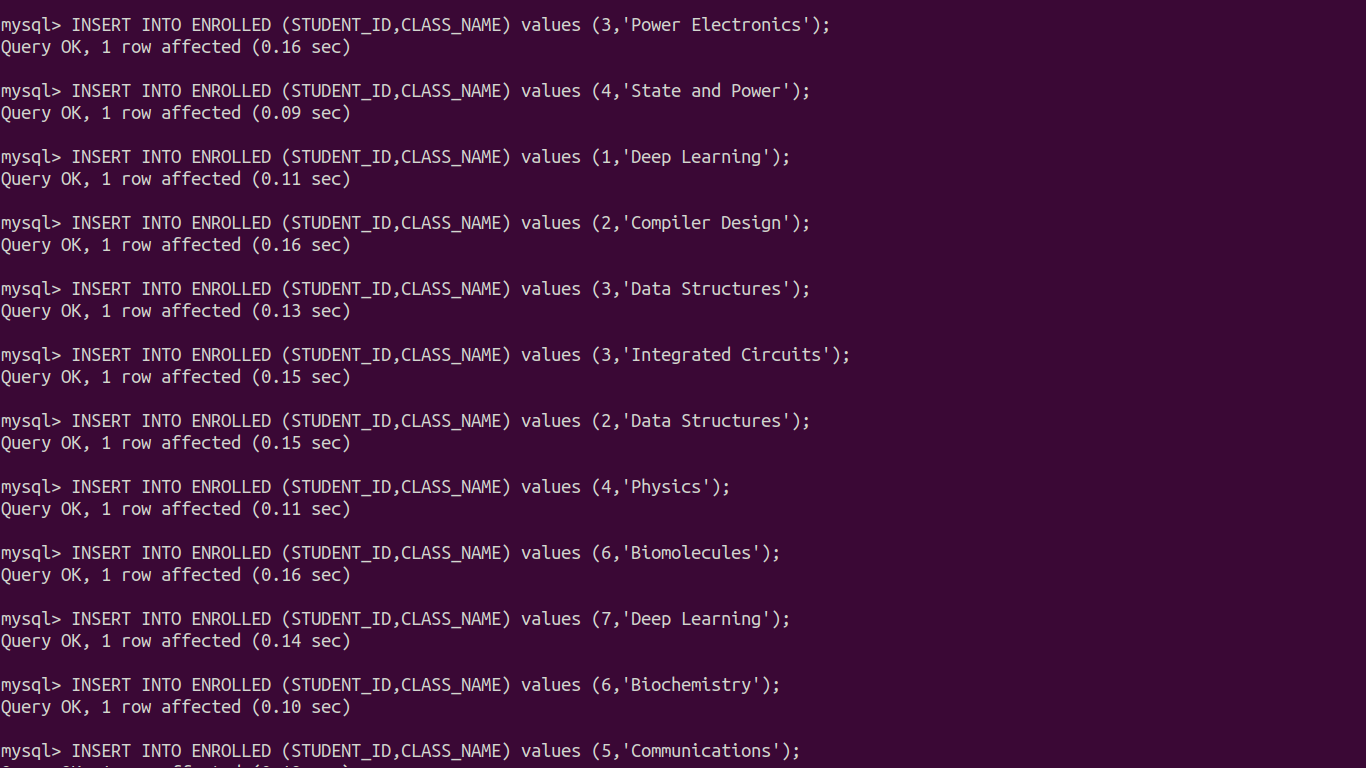
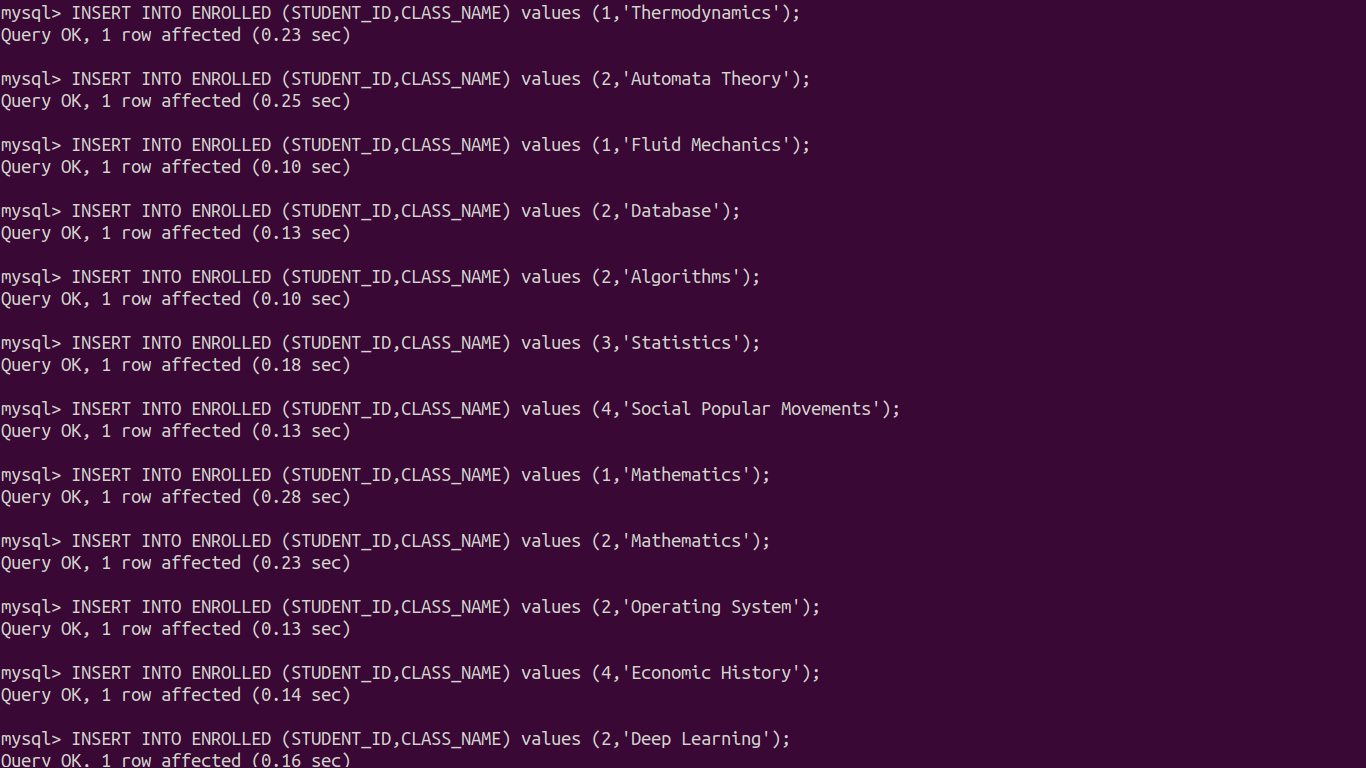
| **INSERT INTO CLASS (CLASS\_NAME, MEETS\_AT, ROOM, FACULTY\_ID) values ('Thermodynamics','08:00:00','R128',1); INSERT INTO CLASS (CLASS\_NAME, MEETS\_AT, ROOM, FACULTY\_ID) values ('Database','10:00:00','F220',5); INSERT INTO CLASS (CLASS\_NAME, MEETS\_AT, ROOM, FACULTY\_ID) values ('Algorithms','08:00:00','C102',7); INSERT INTO CLASS (CLASS\_NAME, MEETS\_AT, ROOM, FACULTY\_ID) values ('Mathematics','09:00:00','I028',8); INSERT INTO CLASS (CLASS\_NAME, MEETS\_AT, ROOM, FACULTY\_ID) values ('Operating System','08:00:00','C102',3); INSERT INTO CLASS (CLASS\_NAME, MEETS\_AT, ROOM, FACULTY\_ID) values ('Social Popular Movements','08:30:00','I028',8); INSERT INTO CLASS (CLASS\_NAME, MEETS\_AT, ROOM, FACULTY\_ID) values ('Deep Learning','09:30:00','I028',9); INSERT INTO CLASS (CLASS\_NAME, MEETS\_AT, ROOM, FACULTY\_ID) values ('Statistics','11:30:00','R128',5); INSERT INTO CLASS (CLASS\_NAME, MEETS\_AT, ROOM, FACULTY\_ID) values ('Automata Theory','09:00:00','C102',1); INSERT INTO CLASS (CLASS\_NAME, MEETS\_AT, ROOM, FACULTY\_ID) values ('Fluid Mechanics','10:00:00','I028',5); INSERT INTO CLASS (CLASS\_NAME, MEETS\_AT, ROOM, FACULTY\_ID) values ('Economic History','09:30:00','C102',3); INSERT INTO CLASS (CLASS\_NAME, MEETS\_AT, ROOM, FACULTY\_ID) values ('Power Electronics','09:00:00','R128',2); INSERT INTO CLASS (CLASS\_NAME, MEETS\_AT, ROOM, FACULTY\_ID) values ('State and Power','10:00:00','F220',1); INSERT INTO CLASS (CLASS\_NAME, MEETS\_AT, ROOM, FACULTY\_ID) values ('Compiler Design','10:00:00','F220',8); INSERT INTO CLASS (CLASS\_NAME, MEETS\_AT, ROOM, FACULTY\_ID) values ('Data Structures','11:00:00','C102',7); INSERT INTO CLASS (CLASS\_NAME, MEETS\_AT, ROOM, FACULTY\_ID) values ('Physics','08:00:00','F220',7); INSERT INTO CLASS (CLASS\_NAME, MEETS\_AT, ROOM, FACULTY\_ID) values ('Communications','12:00:00','I028',4); INSERT INTO CLASS (CLASS\_NAME, MEETS\_AT, ROOM, FACULTY\_ID) values ('Biomolecules','10:00:00','C102',1); INSERT INTO CLASS (CLASS\_NAME, MEETS\_AT, ROOM, FACULTY\_ID) values ('Biochemistry','08:00:00','R128',6); INSERT INTO CLASS (CLASS\_NAME, MEETS\_AT, ROOM, FACULTY\_ID) values ('Ecology','09:30:00','F220',7); INSERT INTO CLASS (CLASS\_NAME, MEETS\_AT, ROOM, FACULTY\_ID) values ('Psychology','11:00:00','C102',5); INSERT INTO CLASS (CLASS\_NAME, MEETS\_AT, ROOM, FACULTY\_ID) values ('Programming','01:00:00','R128',3); INSERT INTO CLASS (CLASS\_NAME, MEETS\_AT, ROOM, FACULTY\_ID) values ('Integrated Circuits','10:30:00','F220',4); INSERT INTO CLASS (CLASS\_NAME, MEETS\_AT, ROOM, FACULTY\_ID) values ('Business Management','01:00:00','I028',7);** |
| --- |

****

****

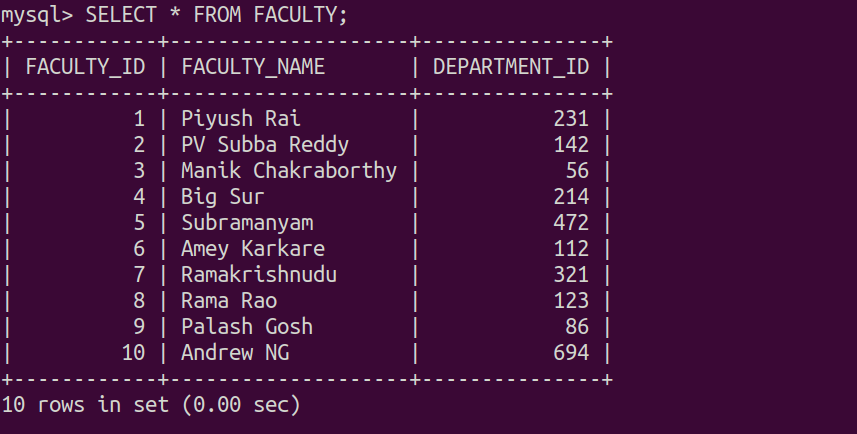
**INSERTION IN ENROLLED Table**

| **INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (1,'Thermodynamics'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (2,'Automata Theory'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (1,'Fluid Mechanics'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (2,'Database'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (2,'Algorithms'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (3,'Statistics'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (4,'Social Popular Movements'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (1,'Mathematics'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (2,'Mathematics'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (2,'Operating System'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (4,'Economic History'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (2,'Deep Learning'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (3,'Power Electronics'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (4,'State and Power'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (1,'Deep Learning'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (2,'Compiler Design'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (3,'Data Structures'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (3,'Integrated Circuits'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (2,'Data Structures'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (4,'Physics'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (6,'Biomolecules'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (7,'Deep Learning'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (6,'Biochemistry'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (5,'Communications'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (5,'Business Management'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (6,'Mathematics'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (8,'Deep Learning'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (7,'Mathematics'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (6,'Ecology'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (4,'Deep Learning'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (1,'Programming'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (5,'Statistics'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (8,'Physics'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (3,'Programming'); INSERT INTO ENROLLED (STUDENT\_ID,CLASS\_NAME) values (5,'Mathematics');** |
| --- |

****

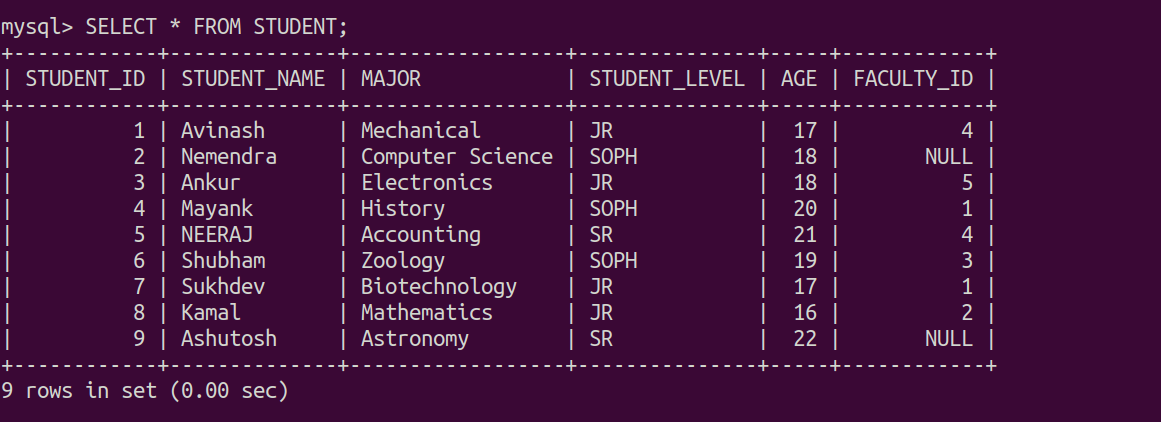
**DATA IN FACULTY Table**

| **SELECT \* FROM FACULTY;** |
| --- |

****

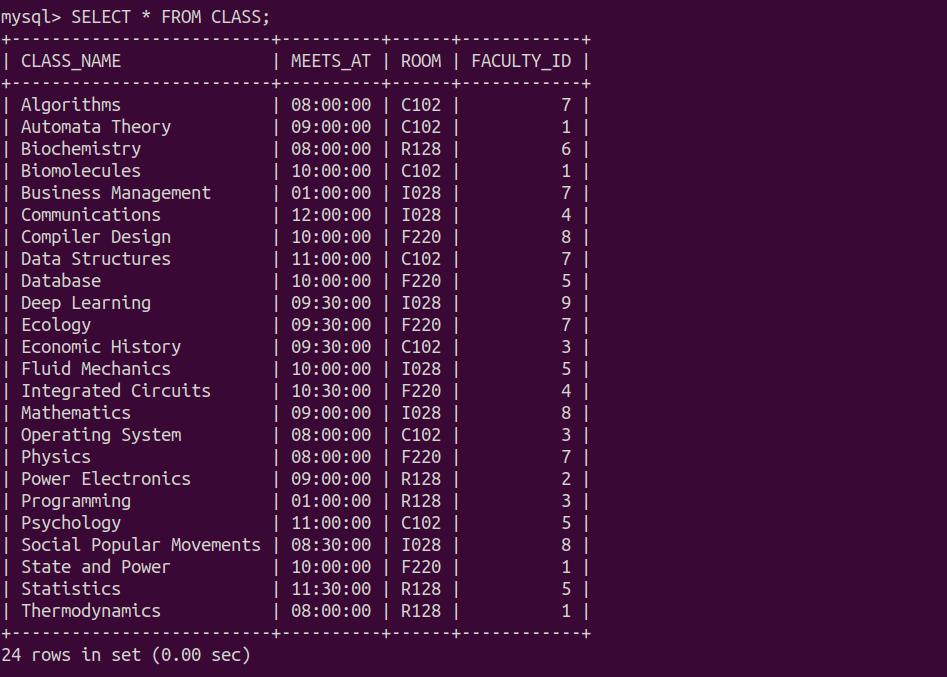
**DATA IN STUDENT Table**

| **SELECT \* FROM STUDENT;** |
| --- |

****

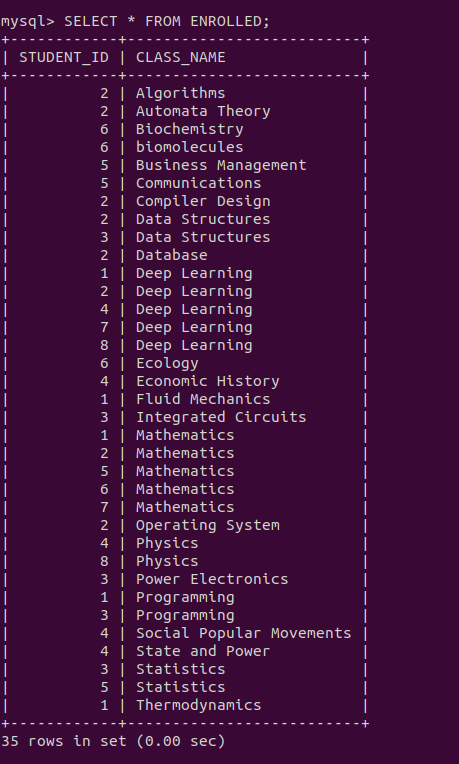
**DATA IN CLASS Table**

| **SELECT \* FROM CLASS;** |
| --- |

****

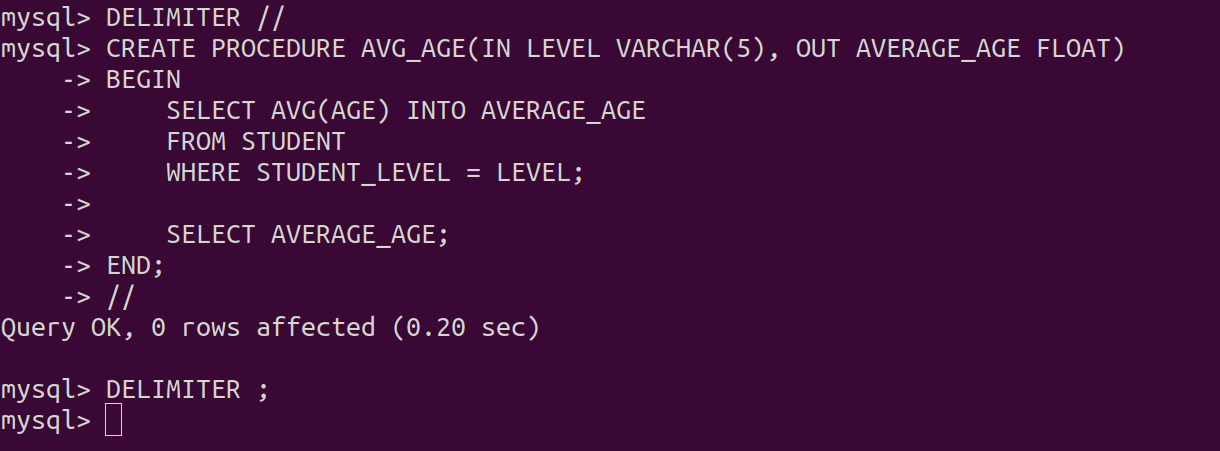
**DATA IN ENROLLED Table**

| **SELECT \* FROM ENROLLED;** |
| --- |

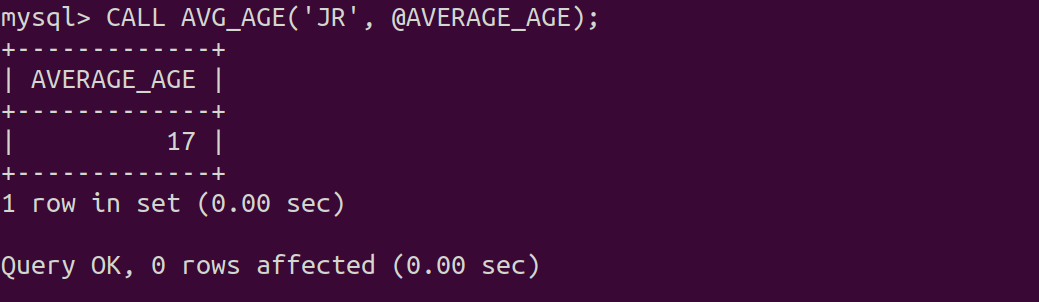
****

1. Write a stored procedure which takes student\_level as input and return the average age of the students of the particular level.

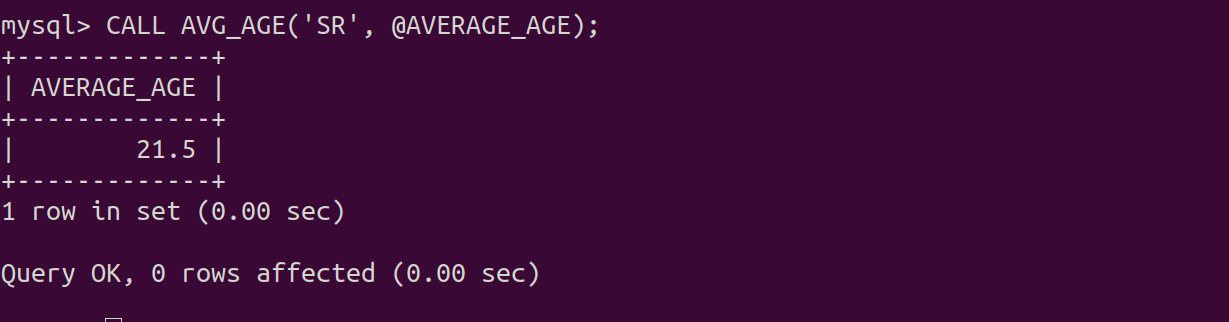
| DELIMITER // **CREATE** **PROCEDURE** AVG\_AGE(**IN** **LEVEL** VARCHAR(5), **OUT** AVERAGE\_AGE FLOAT) **BEGIN**  **SELECT** **AVG**(AGE) **INTO** AVERAGE\_AGE  **FROM** STUDENT  **WHERE** STUDENT\_LEVEL = **LEVEL**;   **SELECT** AVERAGE\_AGE; **END**; // DELIMITER ; |
| --- |



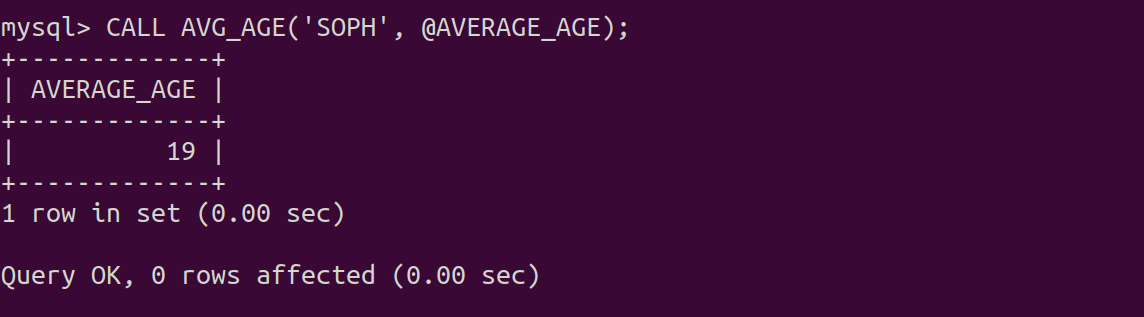
| **CALL** AVG\_AGE('JR', @AVERAGE\_AGE); |
| --- |



| **CALL** AVG\_AGE('SR', @AVERAGE\_AGE); |
| --- |

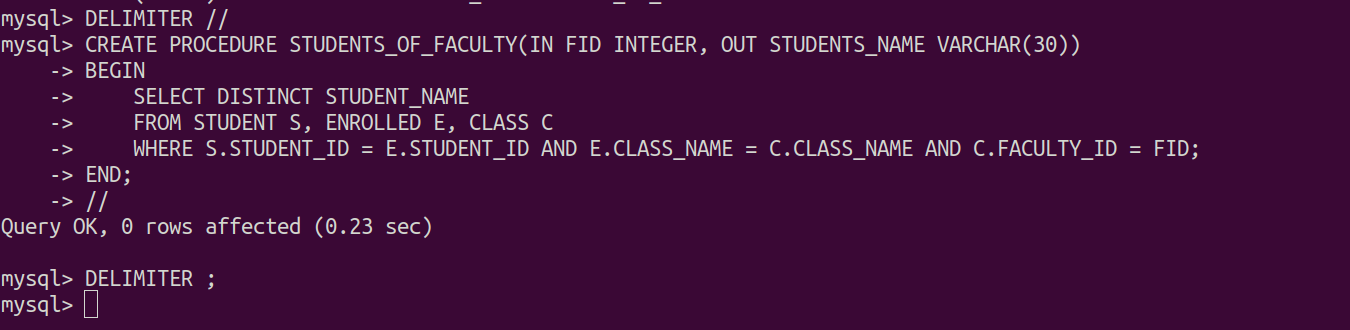


| **CALL** AVG\_AGE('SOPH', @AVERAGE\_AGE); |
| --- |

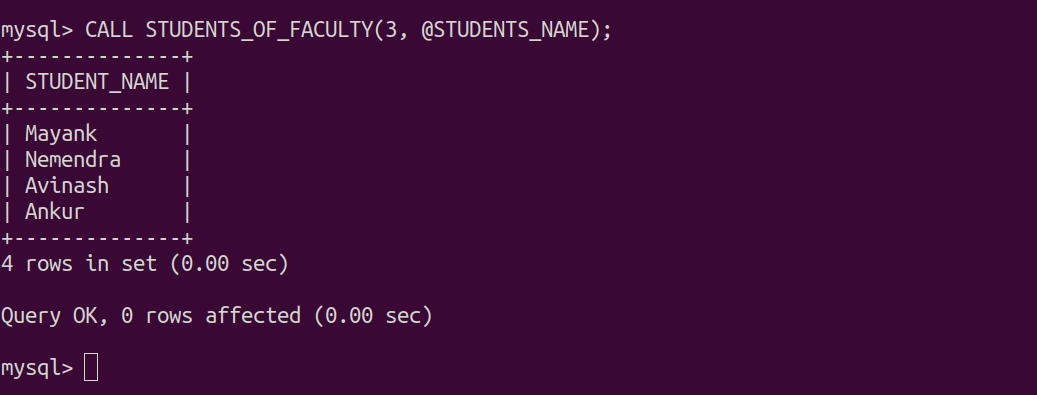


1. Write a stored procedure which takes faculty\_id as input and return the list of all the students those are taught by faculty faculty\_id.

| DELIMITER *//* **CREATE** **PROCEDURE** **STUDENTS\_OF\_FACULTY**(**IN** FID INTEGER, **OUT** STUDENTS\_NAME VARCHAR(30)) **BEGIN**  **SELECT** **DISTINCT** **STUDENT\_NAME**  **FROM** **STUDENT** **S**, **ENROLLED** **E**, **CLASS** **C**  **WHERE** **S**.**STUDENT\_ID** = **E**.**STUDENT\_ID** **AND** **E**.**CLASS\_NAME** = **C**.**CLASS\_NAME** **AND** **C**.**FACULTY\_ID** = **FID**; **END**; *//* DELIMITER ; |
| --- |

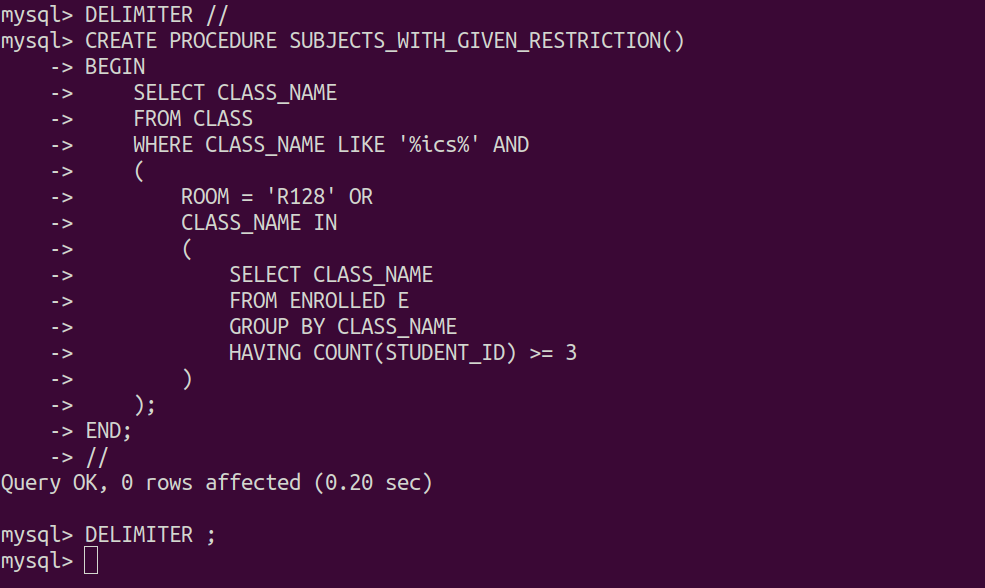


| **CALL** STUDENTS\_OF\_FACULTY(3); |
| --- |

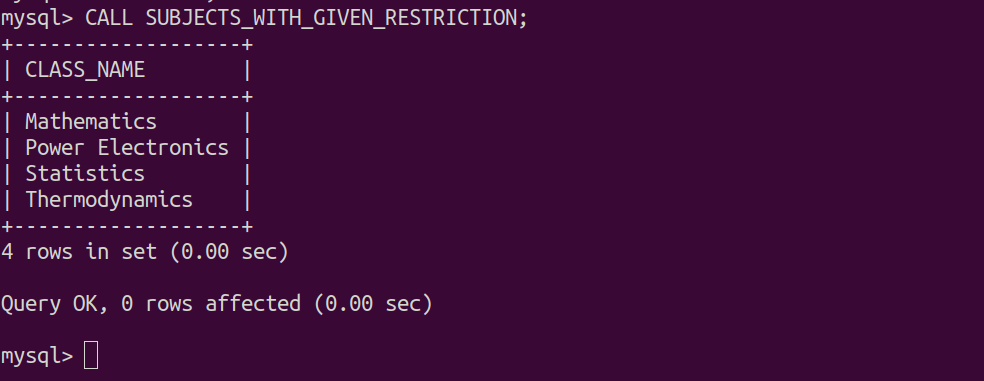


1. Write a stored procedure which return all the class having keyword “ics” in its name and those either meet in room R128 or having 3 or more students enrolled.

| DELIMITER *//* **CREATE** **PROCEDURE** **SUBJECTS\_WITH\_GIVEN\_RESTRICTION**() **BEGIN**  **SELECT** **CLASS\_NAME**  **FROM** **CLASS**  **WHERE** **CLASS\_NAME** **LIKE** '%**ics**%' **AND**  (  ROOM = 'R128' **OR**  CLASS\_NAME **IN**  (  **SELECT** CLASS\_NAME  **FROM** ENROLLED E  **GROUP** **BY** CLASS\_NAME  HAVING COUNT(STUDENT\_ID) >= 3  )  ); **END**; *//* DELIMITER ; |
| --- |

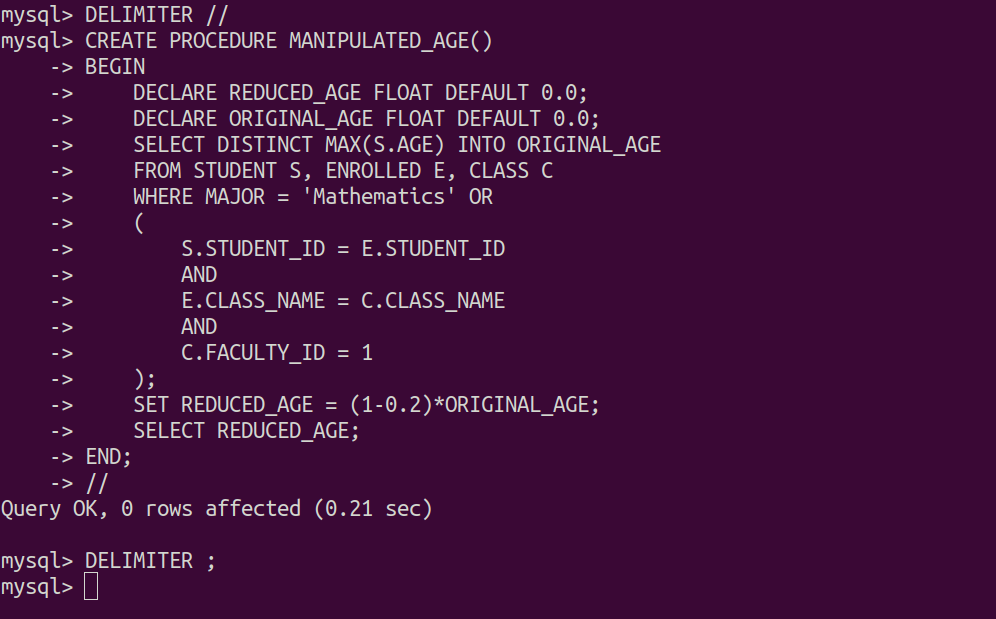


| **CALL** SUBJECTS\_WITH\_GIVEN\_RESTRICTION; |
| --- |

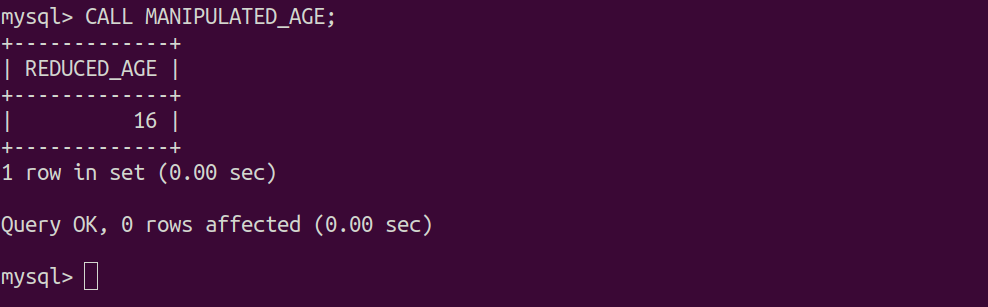


1. Write a stored procedure which return the reduced age of oldest student by 20% who is either a mathematics major or enrolled in a course taught by Teacher id 1.

| DELIMITER *//* **CREATE** **PROCEDURE** **MANIPULATED\_AGE**() **BEGIN**  **DECLARE** **REDUCED\_AGE** **FLOAT** **DEFAULT** 0.0;  DECLARE ORIGINAL\_AGE FLOAT **DEFAULT** 0.0;  **SELECT** **DISTINCT** MAX(S.AGE) **INTO** ORIGINAL\_AGE  **FROM** STUDENT S, ENROLLED E, **CLASS** C  **WHERE** MAJOR = 'Mathematics' **OR**  (  S.STUDENT\_ID = E.STUDENT\_ID  **AND**  E.CLASS\_NAME = C.CLASS\_NAME  **AND**  C.FACULTY\_ID = 1  );  **SET** REDUCED\_AGE = (1-0.2)\*ORIGINAL\_AGE;  **SELECT** REDUCED\_AGE; **END**; *//* DELIMITER ; |
| --- |

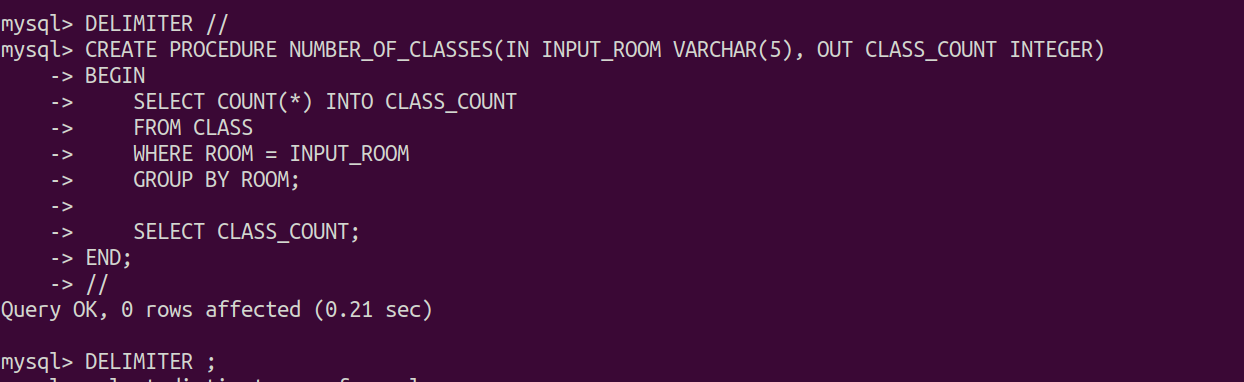


| **CALL** MANIPULATED\_AGE; |
| --- |

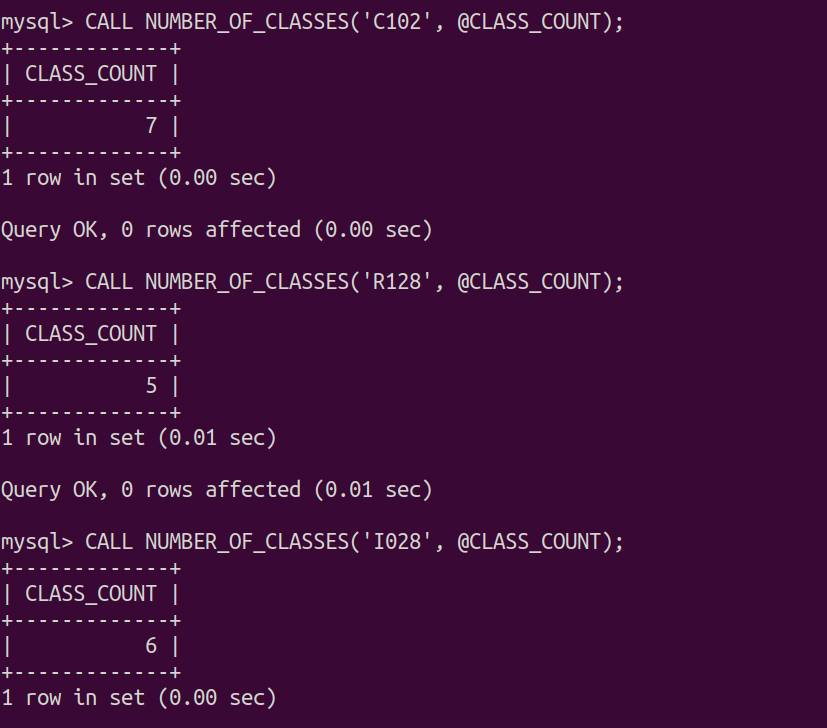


1. Write a stored procedure having IN and OUT parameter which take the room number as an IN parameter and write the count of all the classes held in a particular room into the OUT parameter.

| DELIMITER // **CREATE** **PROCEDURE** NUMBER\_OF\_CLASSES(**IN** INPUT\_ROOM VARCHAR(5), **OUT** CLASS\_COUNT INTEGER) **BEGIN**  **SELECT** **COUNT**(\*) **INTO** CLASS\_COUNT  **FROM** **CLASS**  **WHERE** ROOM = INPUT\_ROOM  **GROUP** **BY** ROOM;   **SELECT** CLASS\_COUNT; **END**; // DELIMITER ; |
| --- |

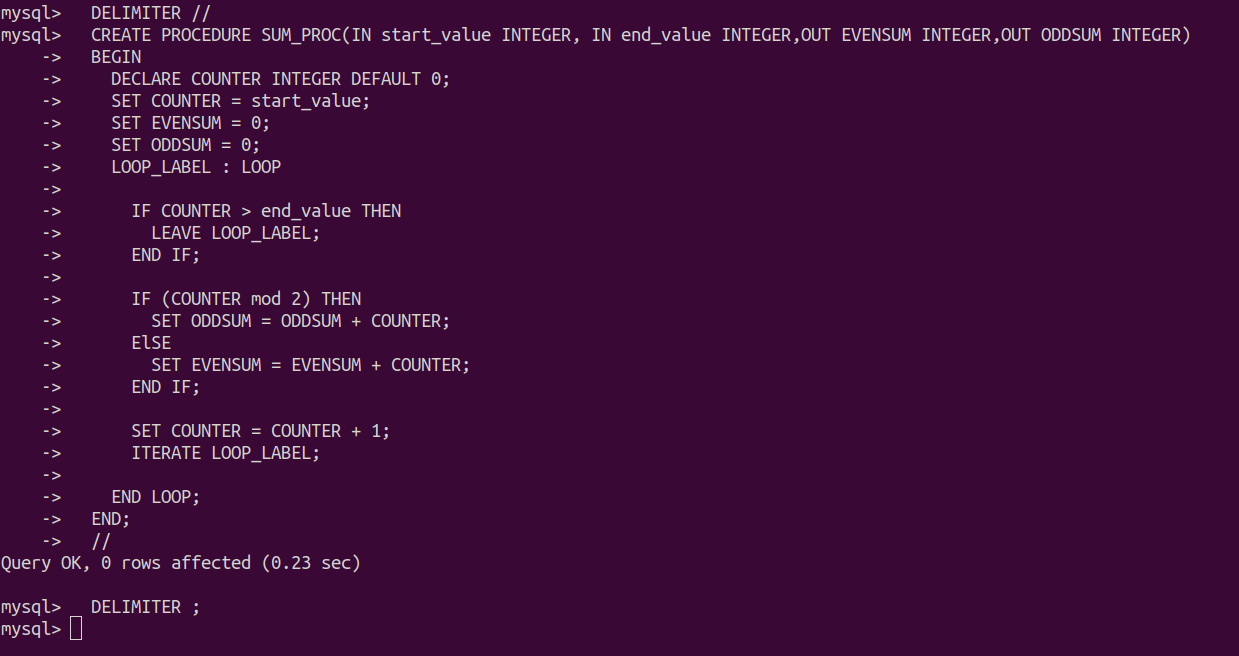


| **CALL** **NUMBER\_OF\_CLASSES**('C102', @CLASS\_COUNT); **CALL** **NUMBER\_OF\_CLASSES**('R128', @CLASS\_COUNT); **CALL** **NUMBER\_OF\_CLASSES**('I028', @CLASS\_COUNT); |
| --- |

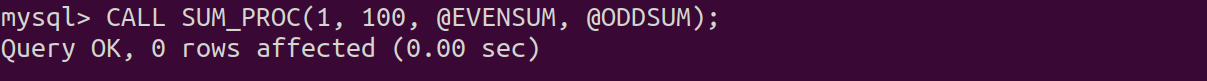


1. **Implement the following scenario using LOOPs in stored procedures.Procedure should take 2 IN parameter start and end and 2 OUT parameter. evensum and oddsum and write the total sum of odd integers into the oddsum parameter and total sum of even integers into the even sum parameter.**

| **DELIMITER *//* CREATE PROCEDURE SUM\_PROC(IN start\_value INTEGER, IN end\_value INTEGER,OUT EVENSUM INTEGER,OUT ODDSUM INTEGER) BEGIN DECLARE COUNTER INTEGER DEFAULT 0;  SET COUNTER = start\_value;  SET EVENSUM = 0;  SET ODDSUM = 0;  LOOP\_LABEL : LOOP   IF COUNTER > end\_value THEN  LEAVE LOOP\_LABEL;  END IF;    IF (COUNTER mod 2) THEN  SET ODDSUM = ODDSUM + COUNTER;  ElSE  SET EVENSUM = EVENSUM + COUNTER;  END IF;   SET COUNTER = COUNTER + 1;  ITERATE LOOP\_LABEL;   END LOOP; END; *//* DELIMITER ;** |
| --- |

****

| **CALL SUM\_PROC(1, 100, @EVENSUM, @ODDSUM);** |
| --- |

****

| **SELECT @EVENSUM*;*** |
| --- |

****

| **SELECT @ODDSUM*;*** |
| --- |

****