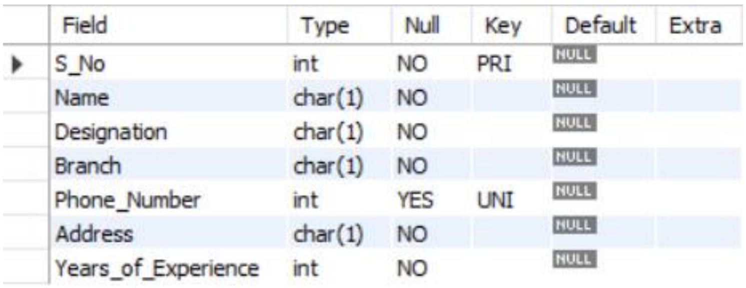
DDL COMMANDS

Create Command

Query:

```
use Exp;
create table Workers
(S_No int primary key,
Name char not null,
Designation char not null,
Branch char not null,
Phone_Number int unique,
Address char not null,
Years_of_Experience int not null);
```

Output:



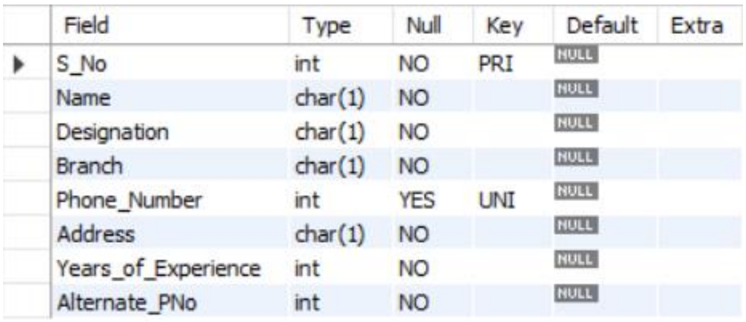
Field	Type	Null	Key	Default	Extra
S_No	int	NO	PRI	NULL	
Name	char(1)	NO		NULL	
Designation	char(1)	NO		NULL	
Branch	char(1)	NO		NULL	
Phone_Number	int	YES	UNI	NULL	
Address	char(1)	NO		NULL	
Years_of_Experience	int	NO		NULL	

Alter Command

Query:

```
use exp;
alter table workers
add (Alternate_PNo int not null);
```

Output:



Field	Type	Null	Key	Default	Extra
S_No	int	NO	PRI	NULL	
Name	char(1)	NO		NULL	
Designation	char(1)	NO		NULL	
Branch	char(1)	NO		NULL	
Phone_Number	int	YES	UNI	NULL	
Address	char(1)	NO		NULL	
Years_of_Experience	int	NO		NULL	
Alternate_PNo	int	NO		NULL	

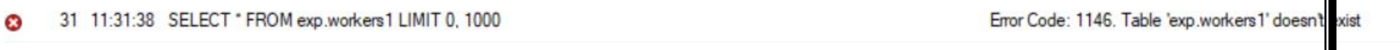
Drop Command

Query:

Use exp;

Drop table workers;

Output:



```
31  11:31:38  SELECT * FROM exp.workers1 LIMIT 0, 1000        Error Code: 1146. Table 'exp.workers1' doesn't exist
```

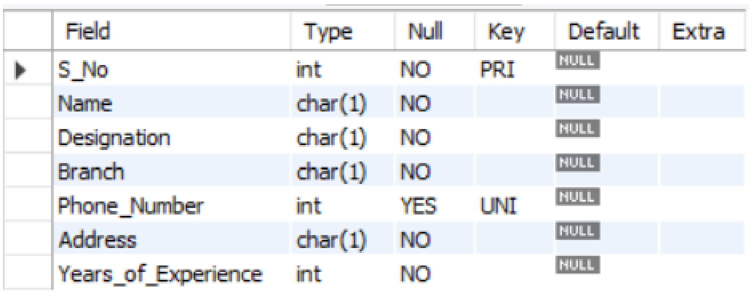
Truncate Command

Query:

use Exp;

truncate workers1;

Output:



Field	Type	Null	Key	Default	Extra
S_No	int	NO	PRI	NULL	
Name	char(1)	NO		NULL	
Designation	char(1)	NO		NULL	
Branch	char(1)	NO		NULL	
Phone_Number	int	YES	UNI	NULL	
Address	char(1)	NO		NULL	
Years_of_Experience	int	NO		NULL	

DML COMMANDS

Insert Command

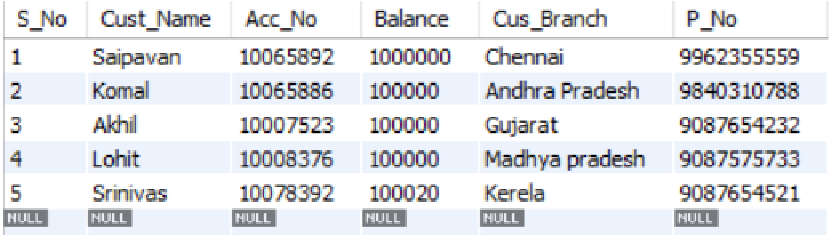
Query:

```
use exp;
insert into bank
values
(1,'Saipavan',10065892,10000000,'Chennai',9962355559),
(2,'Komal',10065886,100000,'Andhra Pradesh',9840310788),
(3,'Akhil',10007523,100000,'Gujarat',9087654232),
(4,'Lohit',10008376,100000,'Madhya pradesh',9087575733),
(5,'Srinivas',10078392,100020,'Kerela',9087654521);
```

Select Command

Query:

```
Use exp;
Select * from bank;
```

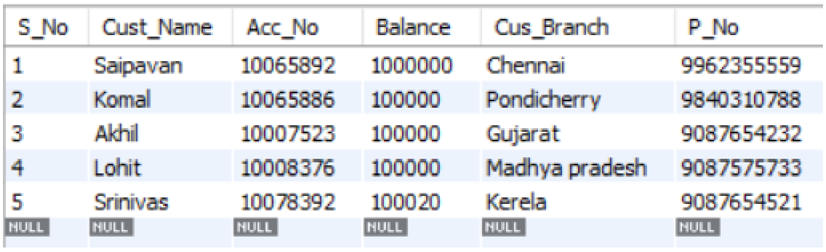


S_No	Cust_Name	Acc_No	Balance	Cus_Branch	P_No
1	Saipavan	10065892	1000000	Chennai	9962355559
2	Komal	10065886	100000	Andhra Pradesh	9840310788
3	Akhil	10007523	100000	Gujarat	9087654232
4	Lohit	10008376	100000	Madhya pradesh	9087575733
5	Srinivas	10078392	100020	Kerela	9087654521
NULL	NULL	NULL	NULL	NULL	NULL

Update Command

Query:

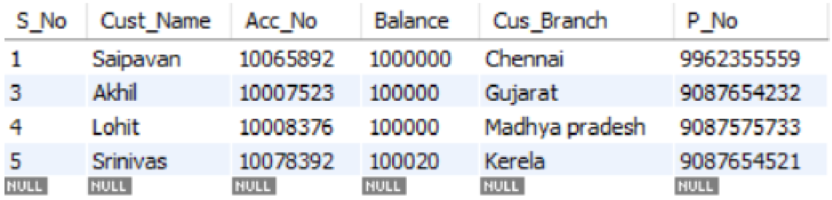
```
use exp;
update bank
set
cus_branch='Pondicherry' where S_No=2;
```



S_No	Cust_Name	Acc_No	Balance	Cus_Branch	P_No
1	Saipavan	10065892	1000000	Chennai	9962355559
2	Komal	10065886	100000	Pondicherry	9840310788
3	Akhil	10007523	100000	Gujarat	9087654232
4	Lohit	10008376	100000	Madhya pradesh	9087575733
5	Srinivas	10078392	100020	Kerela	9087654521
NULL	NULL	NULL	NULL	NULL	NULL

Delete Command

```
use exp;
delete from bank where Cust_Name='Komal';
```



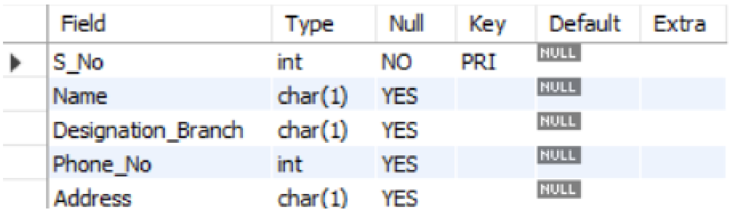
S_No	Cust_Name	Acc_No	Balance	Cus_Branch	P_No
1	Saipavan	10065892	1000000	Chennai	9962355559
3	Akhil	10007523	100000	Gujarat	9087654232
4	Lohit	10008376	100000	Madhya pradesh	9087575733
5	Srinivas	10078392	100020	Kerela	9087654521
NULL	NULL	NULL	NULL	NULL	NULL

DDL COMMANDS WITH CONSTRAINTS

Add Primary Key Command

Query:

```
alter table employee
add primary key(S_No);
```

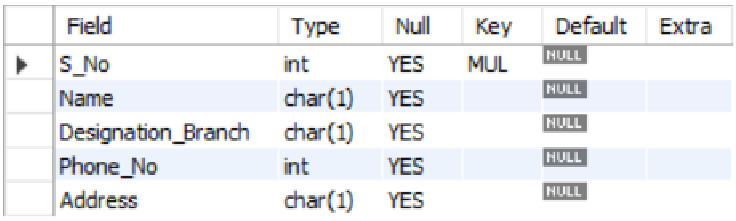


Field	Type	Null	Key	Default	Extra
S_No	int	NO	PRI	NULL	
Name	char(1)	YES		NULL	
Designation_Branch	char(1)	YES		NULL	
Phone_No	int	YES		NULL	
Address	char(1)	YES		NULL	

Add Foreign Key Command

Query:

```
alter table workers
add constraint link
foreign key(S_No)
references employee(S_No);
```

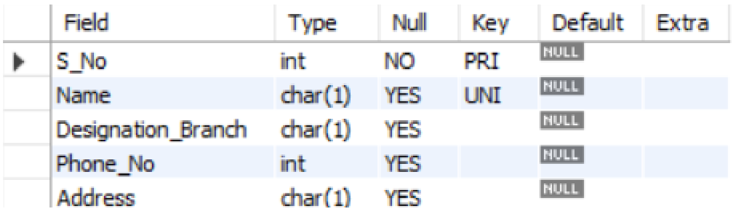


Field	Type	Null	Key	Default	Extra
S_No	int	YES	MUL	NULL	
Name	char(1)	YES		NULL	
Designation_Branch	char(1)	YES		NULL	
Phone_No	int	YES		NULL	
Address	char(1)	YES		NULL	

Add Unique Command

Query:

```
alter table employee
add unique (name);
```

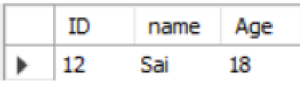


Field	Type	Null	Key	Default	Extra
S_No	int	NO	PRI	NULL	
Name	char(1)	YES	UNI	NULL	
Designation_Branch	char(1)	YES		NULL	
Phone_No	int	YES		NULL	
Address	char(1)	YES		NULL	

Check Command

Query:

```
use exp;
CREATE TABLE Bank2 (
            ID int not null,
   Name char,
Age int,
CHECK (Age>=18));
```



ID	name	Age
12	Sai	18

DCL and TCL Commands

Savepoint Command

```
use exp;
insert into Book
values
(1,'Wings of fire', 'Sai pavan', 1000, 'Surat'),
(2,'Moon', 'Komal', 1500, 'Prashanth'),
(3,'Charndrayan','Akhil',2000,'Ramesh'),
(4,'The sun','Mithil',3000,'Surya'),
(5,'The Mars','Puma',2700,'Srinu');
SAVEPOINT new;
```

Rollback command

```
use exp;
update Book
set
Author='Sai teja' where Author='Sai pavan';
rollback;
```

Grant command

```
Use exp;
Grant select,update on bank to sai;
```

Revoke command

```
Use exp,
Revoke insert,select on bank from sai;
```

Commit command

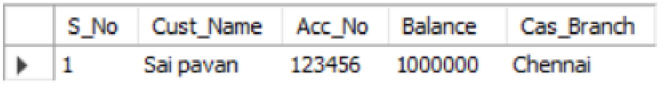
```
Use exp;
Insert into Book
Values
(5,'The Mars','Puma',2700,'Srinu');
Commit;
```

DDL COMMANDS WITH CONSTRAINTS

Select with Where Clause Command

Query:

```
use exp;
select * from bank where Acc_No=123456;
```

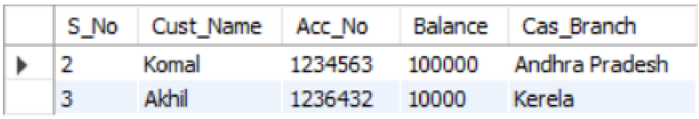


	S_No	Cust_Name	Acc_No	Balance	Cas_Branch
▶	1	Sai pavan	123456	1000000	Chennai

Select with Between Clause Command

Query:

```
use exp;
select * from bank where Balance between 10000 and 100000;
```

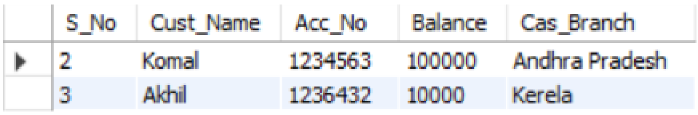


	S_No	Cust_Name	Acc_No	Balance	Cas_Branch
▶	2	Komal	1234563	100000	Andhra Pradesh
	3	Akhil	1236432	10000	Kerela

Select with In Command

Query:

```
use exp;
select * from bank where Balance in (10000,100000);
```

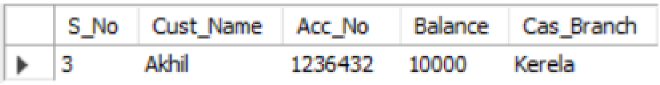


	S_No	Cust_Name	Acc_No	Balance	Cas_Branch
▶	2	Komal	1234563	100000	Andhra Pradesh
	3	Akhil	1236432	10000	Kerela

Select with Pattern matching Command

Query:

```
use exp;
select * from bank where Balance=10000;
```



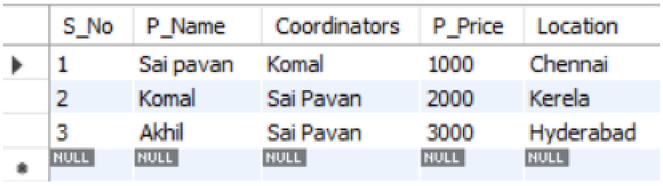
	S_No	Cust_Name	Acc_No	Balance	Cas_Branch
▶	3	Akhil	1236432	10000	Kerela

SELECT WITH VARIOUS CLAUSES

Select with Group by Command

Query:

```
use exp;
select * from project group by P_Price;
```

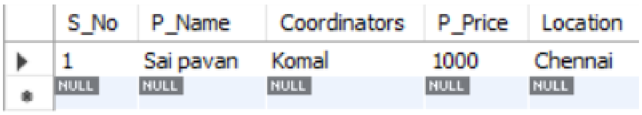


	S_No	P_Name	Coordinators	P_Price	Location
▶	1	Sai pavan	Komal	1000	Chennai
	2	Komal	Sai Pavan	2000	Kerela
	3	Akhil	Sai Pavan	3000	Hyderabad
*	NULL	NULL	NULL	NULL	NULL

Select with Having Command

Query:

```
use exp;
select * from project having P_Price=1000;
```

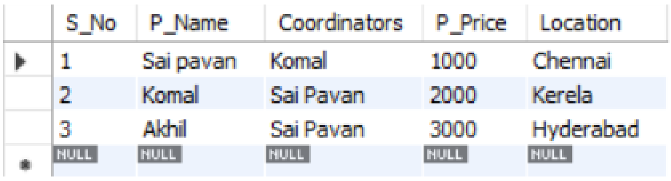


	S_No	P_Name	Coordinators	P_Price	Location
▶	1	Sai pavan	Komal	1000	Chennai
*	NULL	NULL	NULL	NULL	NULL

Select with Order By Command

Query:

```
use exp;
select * from project order by P_Price;
```

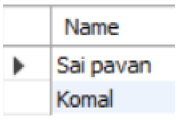


	S_No	P_Name	Coordinators	P_Price	Location
▶	1	Sai pavan	Komal	1000	Chennai
	2	Komal	Sai Pavan	2000	Kerela
	3	Akhil	Sai Pavan	3000	Hyderabad
*	NULL	NULL	NULL	NULL	NULL

Subquery

Query

```
use exp;
select Name from employee where Designation='Manager' and Branch='Chennai';
```



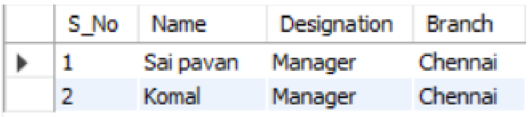
Name
Sai pavan
Komal

Indexing & View

Query

```
create view tab as
select S_No,Name,Designation,Branch from employee;
```

```
insert into tab
values
(1,’Sai pavan’,’Manager’,’Chennai’),
(2,’Komal’,’Manager’,’Chennai’);
```

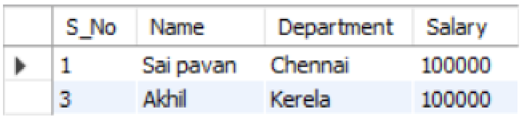


	S_No	Name	Designation	Branch
▶	1	Sai pavan	Manager	Chennai
	2	Komal	Manager	Chennai

EXP 9

Query:

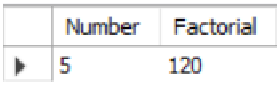
```
DELIMITER &&
CREATE PROCEDURE new()
BEGIN
        select * from employee where Salary>20000;
END &&
DELIMITER ;
```



	S_No	Name	Department	Salary
▶	1	Sai pavan	Chennai	100000
	3	Akhil	Kerela	100000

Query:

```
Delimiter //
CREATE PROCEDURE fact(IN x INT)
BEGIN
DECLARE result INT;
DECLARE i INT;
SET result = 1;
SET i = 1;
WHILE i <= x DO
SET result = result * i;
SET i = i + 1;
END WHILE;
SELECT x AS Number, result as Factorial;
END//
```

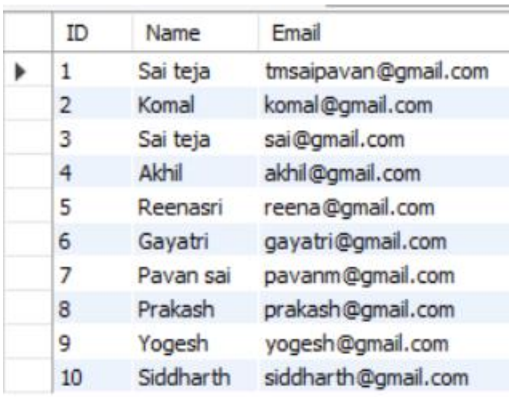


	Number	Factorial
▶	5	120

EXP 10

Query

```
DELIMITER &&
CREATE PROCEDURE display()
BEGIN
select * from employee;
END &&
DELIMITER ;

call display();
```

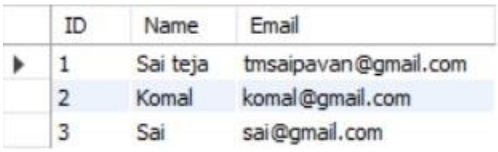


ID	Name	Email
1	Sai teja	tmsaipavan@gmail.com
2	Komal	komal@gmail.com
3	Sai teja	sai@gmail.com
4	Akhil	akhil@gmail.com
5	Reenasri	reena@gmail.com
6	Gayatri	gayatri@gmail.com
7	Pavan sai	pavanm@gmail.com
8	Prakash	prakash@gmail.com
9	Yogesh	yogesh@gmail.com
10	Siddharth	siddharth@gmail.com

EXP 11

Query:

```
DELIMITER //
CREATE TRIGGER checks
before update on employee for each row
insert into employee_audit select * from employee
end//

update employee
set
Name='Sai teja' where ID=3;
```

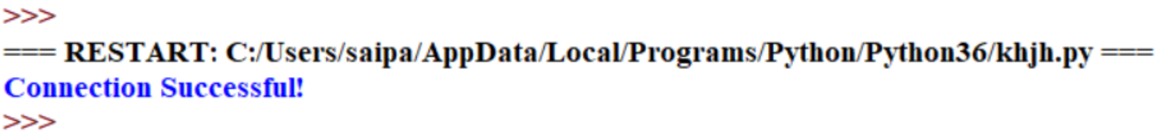


ID	Name	Email
1	Sai teja	tmsaipavan@gmail.com
2	Komal	komal@gmail.com
3	Sai	sai@gmail.com

EXP 12

Query

```
import mysql.connector as mc
mycon =
mc.connect(user='root',host='localhost',passwd='saipavan55',database='exp',auth_plugin='mysql_native_password')
if mycon.is_connected():
   print("Connection Successful!")
else:
   print("Connetion Failed!")
```

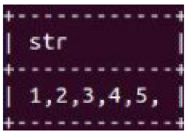


```
>>>
=== RESTART: C:/Users/saipa/AppData/Local/Programs/Python/Python36/khjh.py ===
Connection Successful!
>>>
```

EXP 13

Query :

```
CREATE PROCEDURE test()
 BEGIN
 DECLARE x INT;
 DECLARE str VARCHAR(255);
set x = 1;
set str = '';
WHILE x <= 5 DO
SET str = CONCAT(str, x, ',');
set x = x + 1
END WHILE;
select str;
end
//
```

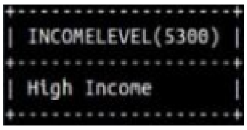


```
+------------+
| str        |
+------------+
| 1,2,3,4,5, |
+------------+
```

EXP 14

Query :

```
DELIMITER //
CREATE FUNCTION IncomeLevel (monthly_value INT)
returns varchar(20)
BEGIN
            DECLARE income_level varchar(20);
   CASE monthly_value
    WHEN 4000 THEN
     SET income_level = 'Low Income';
             WHEN 5000 THEN
     SET income_level = 'Avg Income';
             ELSE
     SET income_level = 'High Income';
            END CASE;
   RETURN income_level;
end; //
```

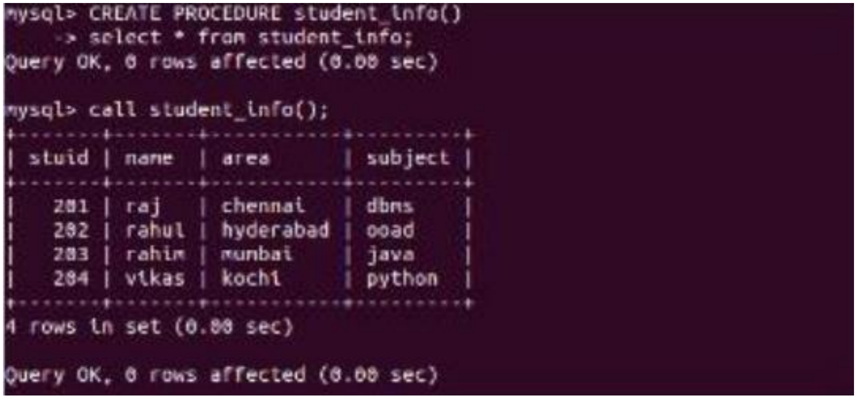


```
+-------------------+
| INCOMELEVEL(5300) |
+-------------------+
| High Income       |
+-------------------+
```

EXP 15

Query

CREATE PROCEDURE student_info()
SELECT * FROM student_info();

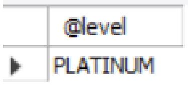


```
mysql> CREATE PROCEDURE student_info()
    -> select * from student_info;
Query OK, 0 rows affected (0.00 sec)

mysql> call student_info();
+-------+-------+-----------+---------+
| stuid | name  | area      | subject |
+-------+-------+-----------+---------+
|   201 | raj   | chennai   | dbms    |
|   202 | rahul | hyderabad | ooad    |
|   203 | rahim | mumbai    | java    |
|   204 | vikas | kochi     | python  |
+-------+-------+-----------+---------+
4 rows in set (0.00 sec)

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

EXP 16

Query :

```
DELIMITER $$
CREATE PROCEDURE GetCustomerLevel(
   IN  pCustomerNumber INT,
   OUT pCustomerLevel  VARCHAR(20))
BEGIN
   DECLARE credit DECIMAL DEFAULT 0;

   SELECT creditLimit
   INTO credit
   FROM customers
   WHERE customerNumber = pCustomerNumber;

   IF credit > 50000 THEN
      SET pCustomerLevel = 'PLATINUM';
   ELSEIF credit <= 50000 AND credit > 10000 THEN
      SET pCustomerLevel = 'GOLD';
   ELSE
      SET pCustomerLevel = 'SILVER';
   END IF;
END $$
DELIMITER ;

call GetCustomerLevel(211,@level);
select @level
```

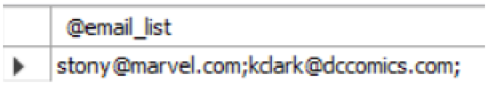


```
@level
PLATINUM
```

EXP 17

Query :

```
DELIMITER $$
CREATE PROCEDURE build_email_list (INOUT email_list varchar(4000))
BEGIN
DECLARE v_finished INTEGER DEFAULT 0;
DECLARE v_email varchar(100) DEFAULT "";
DEClARE email_cursor CURSOR FOR
SELECT email FROM employees;
DECLARE CONTINUE HANDLER FOR
NOT FOUND SET v_finished = 1;
OPEN email_cursor;
get_email: LOOP
FETCH email_cursor INTO v_email;
IF v_finished = 1 THEN
LEAVE get_email;
END IF;
SET email_list = CONCAT(v_email,";",email_list);
END LOOP get_email;
CLOSE email_cursor;
END$$
DELIMITER ;

SET @email_list = "";
CALL build_email_list(@email_list);
SELECT @email_list;
```

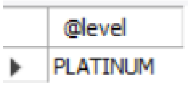


	@email_list
▶	stony@marvel.com;kclark@dccomics.com;

EXP 18

Query :

```
DELIMITER $$
CREATE PROCEDURE GetCustomerLevel(
   IN  pCustomerNumber INT,
   OUT pCustomerLevel  VARCHAR(20))
BEGIN
   DECLARE credit DECIMAL DEFAULT 0;

   SELECT creditLimit
   INTO credit
   FROM customers
   WHERE customerNumber = pCustomerNumber;

   IF credit > 50000 THEN
      SET pCustomerLevel = 'PLATINUM';
   ELSEIF credit <= 50000 AND credit > 10000 THEN
      SET pCustomerLevel = 'GOLD';
   ELSE
      SET pCustomerLevel = 'SILVER';
   END IF;
END $$
DELIMITER ;

call GetCustomerLevel(211,@level);
select @level
```

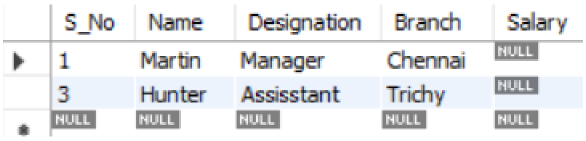


@level
PLATINUM

EXP 19

Query

```
CREATE TABLE workers
(S_No int primary key,
Name varchar(20),
Designation varchar(20),
Branch varchar(20));

alter table workers
add Salary int;

alter table workers
modify Name varchar(10);

desc workers;

create table trainee like workers;
insert into trainee select * from workers;
drop table workers;
```

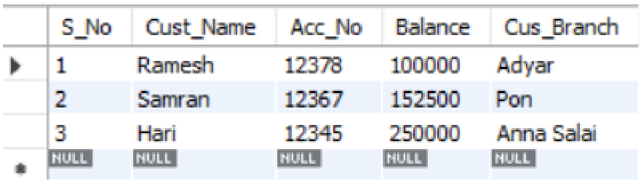


	S_No	Name	Designation	Branch	Salary
▶	1	Martin	Manager	Chennai	NULL
	3	Hunter	Assisstant	Trichy	NULL
*	NULL	NULL	NULL	NULL	NULL

EXP 20

Query

```
CREATE TABLE bank
(S_No int primary key,
Cust_Name varchar(20),
Acc_No int,
Balance int,
Cus_Branch varchar(20));

select * from bank where Cust_Name='Ramesh';

Select * from bank where Balance>15000;

select * from bank where Balance between 1000 and 100000;

update bank
set
Cus_Branch='Pon' where Cust_Name='Samran';

select max(Balance) from bank;

drop table bank;
```

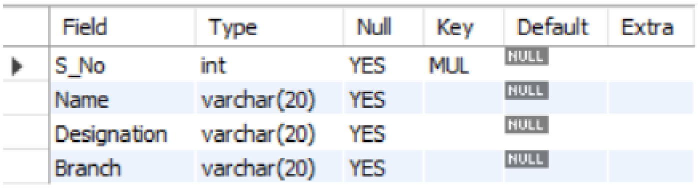


	S_No	Cust_Name	Acc_No	Balance	Cus_Branch
▶	1	Ramesh	12378	100000	Adyar
	2	Samran	12367	152500	Pon
	3	Hari	12345	250000	Anna Salai
*	NULL	NULL	NULL	NULL	NULL

EXP 21

Query

```
CREATE TABLE workers
(S_No int,
Name varchar(20),
Designation varchar(20),
Branch varchar(20));

CREATE TABLE employee
(S_No int,
Name varchar(20),
Designation varchar(20),
Branch varchar(20));

ALTER TABLE workers
add constraint link
foreign key(S_No)
references employee(S_No);
```

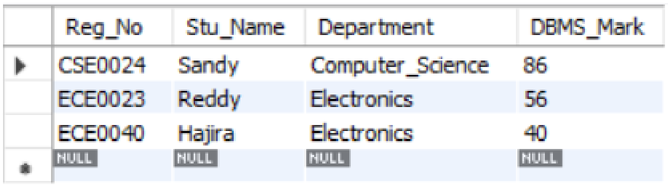


Field	Type	Null	Key	Default	Extra
S_No	int	YES	MUL	NULL	
Name	varchar(20)	YES		NULL	
Designation	varchar(20)	YES		NULL	
Branch	varchar(20)	YES		NULL	

EXP 22

Query

```
CREATE TABLE student
(Reg_No int primary key,
Stu_Name varchar(20),
Department varchar(20),
DBMS_Mark varchar(20));

create view sainew as
select * from student;

insert into sainew select * from student;

select min(DBMS_Mark) from student;

select * from student order by stu_Name;

select Stu_Name from student where Department='Electronics' group by Department;

select * from student where Department='Electronics';

select * from student where Stu_Name like "%dy";

select * from student where DBMS_Mark>50;

drop student;
```

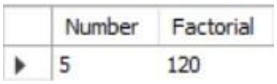


Reg_No	Stu_Name	Department	DBMS_Mark
CSE0024	Sandy	Computer_Science	86
ECE0023	Reddy	Electronics	56
ECE0040	Hajira	Electronics	40
NULL	NULL	NULL	NULL

EXP 23

Query:

```
Delimiter //
CREATE PROCEDURE fact(IN x INT)
BEGIN
DECLARE result INT;
DECLARE i INT;
SET result = 1;
SET  i  = 1;
WHILE i <= x DO
SET result = result * i;
SET i = i + 1;
END WHILE;
SELECT x AS Number, result as Factorial;
END//
```

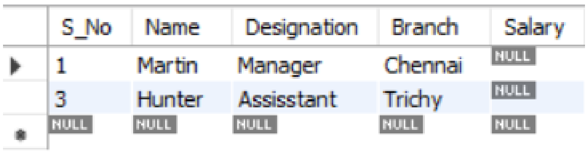


	Number	Factorial
▶	5	120

EXP 24

Query

```
CREATE TABLE trainee
(S_No int primary key,
Name varchar(20),
Designation varchar(20),
Branch varchar(20));

Alter table trainee
Add Salary int;

Alter table trainee
Modify designation varchar(10);

Desc trainee;

create table networkers like trainee;
insert into networkers select * from trainee;

delete from trainee where Name='Sopna';

drop trainee;
```

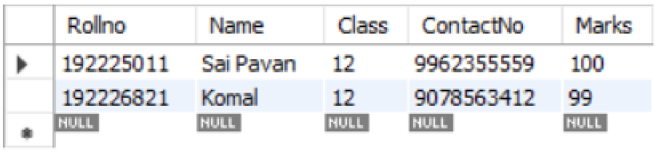


	S_No	Name	Designation	Branch	Salary
▶	1	Martin	Manager	Chennai	NULL
	3	Hunter	Assisstant	Trichy	NULL
*	NULL	NULL	NULL	NULL	NULL

EXP 25

Query

```
print("========WELCOME========")
print("SCHOOL DATABASE SYSTEM")
print("=========================")

import mysql.connector as mc
mycon = mc.connect(user='root',host='localhost',passwd='saipavan55',database='schooldb',
auth_plugin='mysql_native_password')
if mycon.is_connected():
    print("Connection To Server Is Successful!!!")
else:
    print("Connection To Server Is Failed!!!")
mycursor = mycon.cursor()

while True:
    print("Enter the Student details")
    roll = int(input("Enter Student Roll Number : "))
    print()
    name = input("Enter Student Name : ")
    print()
    clas = int(input("Enter Student Class: "))
    print()
    contact = int(input("Enter Student Contact Number : "))
    print()
    marks = int(input("Enter Student Marks : "))
    print()
    query = "INSERT INTO db VALUES(%s,'%s','%s',%s,%s)"%(roll,name,clas,contact,marks)
    mycursor.execute(query)
    mycon.commit()
    mycon.close()
    print("Successfully Registered!!!")
    print()
    con = input("Do you want to continue (y/n) : ")
    if con.lower()=='n':
        print("Thank you!")
        break
```



	Rollno	Name	Class	ContactNo	Marks
▶	192225011	Sai Pavan	12	9962355559	100
	192226821	Komal	12	9078563412	99
*	NULL	NULL	NULL	NULL	NULL