EXERCISE-1

1)EMPLOYEE

+-------+-----------+----------+------------+--------+

| empid | empname | address | doj | salary |

+-------+-----------+----------+------------+--------+

| 1 | Employee1 | India | 2005-12-12 | 10000 |

| 3 | Employee3 | USA | 2006-11-12 | 7000 |

| 2 | Employee2 | Pakistan | 2006-06-12 | 30000 |

+-------+-----------+----------+------------+--------+

3 rows in set (0.00 sec)

PROJECT

-----------+----------+-------------+

| projectNo | duration | projectName |

+-----------+----------+-------------+

| 1 | 100 | P1 |

| 2 | 320 | Second |

| 3 | 10 | Third |

+-----------+----------+-------------+

3 rows in set (0.00 sec)

WORKSON

+-------+-----------+

| empid | projectNo |

+-------+-----------+

| 1 | 3 |

| 2 | 1 |

| 3 | 2 |

+-------+-----------+

3 rows in set (0.00 sec)

1. Display the Employee details in the descending order based on name.

select \* from employee order by empname desc;

+-------+-----------+----------+------------+--------+

| empid | empname | address | doj | salary |

+-------+-----------+----------+------------+--------+

| 3 | Employee3 | USA | 2006-11-12 | 7000 |

| 2 | Employee2 | Pakistan | 2006-06-12 | 30000 |

| 1 | Employee1 | India | 2005-12-12 | 10000 |

+-------+-----------+----------+------------+--------+

3 rows in set (0.00 sec)

2. Display the project details if project id is given.

select \* from project where projectNo=1;

+-----------+----------+-------------+

| projectNo | duration | projectName |

+-----------+----------+-------------+

| 1 | 100 | P1 |

+-----------+----------+-------------+

1 row in set (0.00 sec)

3. Display the employee names starting with â€˜B'

select empname from employee where empname like 'B%';

Empty set (0.00 sec)

4. Display the employee ID's working in a particular project if projectNo is given. (for projectNo =2)

select empid from workson where projectNo=2;

+-------+

| empid |

+-------+

| 3 |

+-------+

1 row in set (0.00 sec)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2)

1. Insert 10 student details and 3 department details. Insert

details in the studdep table.

mysql> insert into student values(1,'A', 10,20,30,40,50,60,210);

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(2,'AB', 1,2,3,4,5,6,21);

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(3,'B', 0,2,3,4,5,6,20);

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(4,'X', 10,2,3,4,5,6,30);

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(5,'D', 10,20,3,4,5,6,48);

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(6,'L', 100,20,3,4,5,6,138);

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(7,'P', 20,20,3,4,5,6,58);

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(8,'o', 21,20,3,4,5,6,59);

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(9,'oO', 21,20,30,4,5,6,86);

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(10,'PL', 21,20,30,40,5,6,122);

Query OK, 1 row affected (0.00 sec)

mysql> insert into department values(1,'CSE','hod1');

Query OK, 1 row affected (0.00 sec)

mysql> insert into department values(2,'EEE','hod2');

Query OK, 1 row affected (0.00 sec)

mysql> insert into department values(3,'ECE','hod3');

Query OK, 1 row affected (0.00 sec)

mysql> insert into studDep values(1,1);

Query OK, 1 row affected (0.00 sec)

mysql> insert into studDep values(2,1);

Query OK, 1 row affected (0.00 sec)

mysql> insert into studDep values(3,2);

Query OK, 1 row affected (0.00 sec)

mysql> insert into studDep values(4,1);

Query OK, 1 row affected (0.00 sec)

mysql> insert into studDep values(5,3);

Query OK, 1 row affected (0.00 sec)

mysql> insert into studDep values(6,3);

Query OK, 1 row affected (0.00 sec)

mysql> insert into studDep values(7,2);

Query OK, 1 row affected (0.00 sec)

mysql> insert into studDep values(8,1);

Query OK, 1 row affected (0.00 sec)

mysql> insert into studDep values(9,2);

Query OK, 1 row affected (0.00 sec)

mysql> insert into studDep values(10,1);

Query OK, 1 row affected (0.00 sec)

mysql> select \* from student

-> ;

+--------+------+--------+--------+--------+--------+--------+--------+-------+

| rollNo | name | marks1 | marks2 | marks3 | marks4 | marks5 | marks6 | total |

+--------+------+--------+--------+--------+--------+--------+--------+-------+

| 1 | A | 10 | 20 | 30 | 40 | 50 | 60 | 210 |

| 2 | AB | 1 | 2 | 3 | 4 | 5 | 6 | 21 |

| 3 | B | 0 | 2 | 3 | 4 | 5 | 6 | 20 |

| 4 | X | 10 | 2 | 3 | 4 | 5 | 6 | 30 |

| 5 | D | 10 | 20 | 3 | 4 | 5 | 6 | 48 |

| 6 | L | 100 | 20 | 3 | 4 | 5 | 6 | 138 |

| 7 | P | 20 | 20 | 3 | 4 | 5 | 6 | 58 |

| 8 | o | 21 | 20 | 3 | 4 | 5 | 6 | 59 |

| 9 | oO | 21 | 20 | 30 | 4 | 5 | 6 | 86 |

| 10 | PL | 21 | 20 | 30 | 40 | 5 | 6 | 122 |

+--------+------+--------+--------+--------+--------+--------+--------+-------+

10 rows in set (0.00 sec)

mysql> select \* from department;

+--------+----------+---------+

| deptId | deptname | HODName |

+--------+----------+---------+

| 1 | CSE | hod1 |

| 2 | EEE | hod2 |

| 3 | ECE | hod3 |

+--------+----------+---------+

3 rows in set (0.00 sec)

mysql> select \* from studDep;

+--------+--------+

| rollNo | deptId |

+--------+--------+

| 1 | 1 |

| 2 | 1 |

| 3 | 2 |

| 4 | 1 |

| 5 | 3 |

| 6 | 3 |

| 7 | 2 |

| 8 | 1 |

| 9 | 2 |

| 10 | 1 |

+--------+--------+

10 rows in set (0.00 sec)

2. Display the Student details if deptid is given

for deptId=1

select student.rollNo, name, marks1,marks2,marks3,marks4,marks5,marks6,total from student,studDep where student.rollNo=studDep.rollNo and deptId=1;

+--------+------+--------+--------+--------+--------+--------+--------+-------+

| rollNo | name | marks1 | marks2 | marks3 | marks4 | marks5 | marks6 | total |

+--------+------+--------+--------+--------+--------+--------+--------+-------+

| 1 | A | 10 | 20 | 30 | 40 | 50 | 60 | 210 |

| 2 | AB | 1 | 2 | 3 | 4 | 5 | 6 | 21 |

| 4 | X | 10 | 2 | 3 | 4 | 5 | 6 | 30 |

| 8 | o | 21 | 20 | 3 | 4 | 5 | 6 | 59 |

| 10 | PL | 21 | 20 | 30 | 40 | 5 | 6 | 122 |

+--------+------+--------+--------+--------+--------+--------+--------+-------+

5 rows in set (0.00 sec)

3. Display the department details if rollno is given

for rollno=3

select department.deptId, deptName,HODName from department,studDep where rollNo=3 and department.deptId=studDep.deptId;

+--------+----------+---------+

| deptId | deptName | HODName |

+--------+----------+---------+

| 2 | EEE | hod2 |

+--------+----------+---------+

1 row in set (0.00 sec)

4. Display the student names who got total greater than 500

select \* from student where total>500;

Empty set (0.00 sec)

5. Display the HOD name of the CSE department

select HODName from department where deptName='CSE';

+---------+

| HODName |

+---------+

| hod1 |

+---------+

1 row in set (0.00 sec)

6. Display the student rollnos of the CSE department

select rollNo from studDep,department where deptName='CSE' and department.deptId=studDep.deptId;

+--------+

| rollNo |

+--------+

| 1 |

| 2 |

| 4 |

| 8 |

| 10 |

+--------+

5 rows in set (0.00 sec)

3)

SALESPERSON

+-----+--------+------------+---------+

| ssn | name | start\_year | dept\_no |

+-----+--------+------------+---------+

| 1 | First | 1996 | 1 |

| 2 | Second | 1999 | 2 |

| 3 | Abc | 2009 | 2 |

+-----+--------+------------+---------+

3 rows in set (0.00 sec)

TRIP

+------+------------+------------+----------------+-------------+---------+

| ssn | from\_city | to\_city | departure\_date | return\_date | trip\_id |

+------+------------+------------+----------------+-------------+---------+

| 1 | firstcity | secondcity | 2001-01-01 | 2001-02-02 | 1 |

| 2 | secondcity | thirdcity | 2006-01-07 | 2007-04-02 | 2 |

| 1 | secondcity | thirdcity | 2006-01-07 | 2007-04-02 | 3 |

| 1 | firstcity | thirdcity | 2015-01-01 | 2015-02-02 | 4 |

+------+------------+------------+----------------+-------------+---------+

4 rows in set (0.00 sec)

SALEREP\_EXPENSE

+---------+--------------+--------+

| trip\_id | expense\_type | amount |

+---------+--------------+--------+

| 1 | TRAVEL | 1000 |

| 1 | FOOD | 100 |

| 1 | STAY | 100 |

| 2 | TRAVEL | 1300 |

| 2 | STAY | 1300 |

+---------+--------------+--------+

5 rows in set (0.00 sec)

1. Give the details (all attributes of trip relation) for trips that

exceed Rs2000.

select trip.\* from salerep\_expense, trip where trip.trip\_id=salerep\_expense.trip\_id group by trip\_id having sum(amount)>2000;

+------+------------+-----------+----------------+-------------+---------+

| ssn | from\_city | to\_city | departure\_date | return\_date | trip\_id |

+------+------------+-----------+----------------+-------------+---------+

| 2 | secondcity | thirdcity | 2006-01-07 | 2007-04-02 | 2 |

+------+------------+-----------+----------------+-------------+---------+

1 row in set (0.00 sec)

2. Print the ssn of salesperson who took trips to chennai more than once. (let chennai=thirdcity)

select ssn from trip where to\_city='thirdcity' group by ssn having count(\*)>1;

+------+

| ssn |

+------+

| 1 |

+------+

1 row in set (0.00 sec)

3. Print the total trip expenses incurred by the salesperson with ssn = 1

select sum(amount) from salerep\_expense, trip, salesperson where trip.trip\_id=salerep\_expense.trip\_id and trip.ssn=salesperson.ssn and trip.ssn=1;

+-------------+

| sum(amount) |

+-------------+

| 1200 |

+-------------+

1 row in set (0.00 sec)

4. select \* from salesperson order by name asc;

+-----+--------+------------+---------+

| ssn | name | start\_year | dept\_no |

+-----+--------+------------+---------+

| 3 | Abc | 2009 | 2 |

| 1 | First | 1996 | 1 |

| 2 | Second | 1999 | 2 |

+-----+--------+------------+---------+

3 rows in set (0.00 sec)

EXERCISE-2

1.

create table president(lastName varchar(20),firstName varchar(20), state varchar(20),city varchar(20), birthDate date,deathDate date);

Query OK, 0 rows affected (0.05 sec)

1. Insert appropriate values into the table(min of 10 records).

mysql> insert into president values("william","first","XXYY","PQRS","1900-4-4","1980-5-6");

Query OK, 1 row affected (0.00 sec)

mysql> insert into president values("william","george","XXYY","LMNO","1750-9-8","1780-6-6");

Query OK, 1 row affected (0.00 sec)

mysql> insert into president values("uiop","zxcv","uiop","qwerty","1410-9-8","1467-1-6");

Query OK, 1 row affected (0.00 sec)

mysql> insert into president values("uiop","Second","uiop","qwerty","1970-2-8","");

Query OK, 1 row affected, 1 warning (0.00 sec)

mysql> insert into president values("william","second","XXYY","LMNO","1940-2-8","");

Query OK, 1 row affected, 1 warning (0.00 sec)

mysql> insert into president values("qwer","jklm","efgh","abcd","1800-8-8","");Query OK, 1 row affected, 1 warning (0.00 sec)

mysql> insert into president values("vwxy","rstu","xxxx","abcd","1890-8-8","1920-7-7");

Query OK, 1 row affected (0.00 sec)

mysql> insert into president values("bbbb","fghj","xxxx","abcd","1880-1-1","1920-7-8");

Query OK, 1 row affected (0.00 sec)

mysql> insert into president values("b","o","xxxx","abcd","1936-2-1","1990-5-8");

Query OK, 1 row affected (0.00 sec)

mysql> insert into president values("last","pre","xxcv","bcda","1736-9-1","1780-5-8");

Query OK, 1 row affected (0.00 sec)

2. Retrieve the records entered.

mysql> select \* from president;

+----------+-----------+-------+--------+------------+------------+

| lastName | firstName | state | city | birthDate | deathDate |

+----------+-----------+-------+--------+------------+------------+

| william | first | XXYY | PQRS | 1900-04-04 | 1980-05-06 |

| william | george | XXYY | LMNO | 1750-09-08 | 1780-06-06 |

| uiop | zxcv | uiop | qwerty | 1410-09-08 | 1467-01-06 |

| uiop | Second | uiop | qwerty | 1970-02-08 | 0000-00-00 |

| william | second | XXYY | LMNO | 1940-02-08 | 0000-00-00 |

| qwer | jklm | efgh | abcd | 1800-08-08 | 0000-00-00 |

| vwxy | rstu | xxxx | abcd | 1890-08-08 | 1920-07-07 |

| bbbb | fghj | xxxx | abcd | 1880-01-01 | 1920-07-08 |

| b | o | xxxx | abcd | 1936-02-01 | 1990-05-08 |

| last | pre | xxcv | bcda | 1736-09-01 | 1780-05-08 |

+----------+-----------+-------+--------+------------+------------+

10 rows in set (0.00 sec)

3. display the president details from state "XXXX".

select \* from president where state="XXXX";

+----------+-----------+-------+------+------------+------------+

| lastName | firstName | state | city | birthDate | deathDate |

+----------+-----------+-------+------+------------+------------+

| vwxy | rstu | xxxx | abcd | 1890-08-08 | 1920-07-07 |

| bbbb | fghj | xxxx | abcd | 1880-01-01 | 1920-07-08 |

| b | o | xxxx | abcd | 1936-02-01 | 1990-05-08 |

+----------+-----------+-------+------+------------+------------+

3 rows in set (0.00 sec)

4. Display state, first name, last name from the table.

select state,firstName,lastName from president;

+-------+-----------+----------+

| state | firstName | lastName |

+-------+-----------+----------+

| XXYY | first | william |

| XXYY | george | william |

| uiop | zxcv | uiop |

| uiop | Second | uiop |

| XXYY | second | william |

| efgh | jklm | qwer |

| xxxx | rstu | vwxy |

| xxxx | fghj | bbbb |

| xxxx | o | b |

| xxcv | pre | last |

+-------+-----------+----------+

10 rows in set (0.00 sec)

5. delete the record of the president whose first name is "george"

delete from president where firstName="george";

Query OK, 1 row affected (0.00 sec)

select \* from president;

+----------+-----------+-------+--------+------------+------------+

| lastName | firstName | state | city | birthDate | deathDate |

+----------+-----------+-------+--------+------------+------------+

| william | first | XXYY | PQRS | 1900-04-04 | 1980-05-06 |

| uiop | zxcv | uiop | qwerty | 1410-09-08 | 1467-01-06 |

| uiop | Second | uiop | qwerty | 1970-02-08 | 0000-00-00 |

| william | second | XXYY | LMNO | 1940-02-08 | 0000-00-00 |

| qwer | jklm | efgh | abcd | 1800-08-08 | 0000-00-00 |

| vwxy | rstu | xxxx | abcd | 1890-08-08 | 1920-07-07 |

| bbbb | fghj | xxxx | abcd | 1880-01-01 | 1920-07-08 |

| b | o | xxxx | abcd | 1936-02-01 | 1990-05-08 |

| last | pre | xxcv | bcda | 1736-09-01 | 1780-05-08 |

+----------+-----------+-------+--------+------------+------------+

9 rows in set (0.00 sec)

6. Change the state from "XXYY" to "YYYY" for the president whose last name is "william"

update president set state="YYYY" where state="XXXX";

Query OK, 3 rows affected (0.00 sec)

Rows matched: 3 Changed: 3 Warnings: 0

mysql> select \* from president;

+----------+-----------+-------+--------+------------+------------+

| lastName | firstName | state | city | birthDate | deathDate |

+----------+-----------+-------+--------+------------+------------+

| william | first | XXYY | PQRS | 1900-04-04 | 1980-05-06 |

| uiop | zxcv | uiop | qwerty | 1410-09-08 | 1467-01-06 |

| uiop | Second | uiop | qwerty | 1970-02-08 | 0000-00-00 |

| william | second | XXYY | LMNO | 1940-02-08 | 0000-00-00 |

| qwer | jklm | efgh | abcd | 1800-08-08 | 0000-00-00 |

| vwxy | rstu | YYYY | abcd | 1890-08-08 | 1920-07-07 |

| bbbb | fghj | YYYY | abcd | 1880-01-01 | 1920-07-08 |

| b | o | YYYY | abcd | 1936-02-01 | 1990-05-08 |

| last | pre | xxcv | bcda | 1736-09-01 | 1780-05-08 |

+----------+-----------+-------+--------+------------+------------+

9 rows in set (0.00 sec)

7. Display the last name and birth date of president who are still alive.

select lastName,birthDate from president where deathDate="0000-00-00";

+----------+------------+

| lastName | birthDate |

+----------+------------+

| uiop | 1970-02-08 |

| william | 1940-02-08 |

| qwer | 1800-08-08 |

+----------+------------+

3 rows in set (0.00 sec)

8. Display the presidents who are born in 18th century.

select \* from president where birthDate between "1700-1-1" and "1799-12-31";

+----------+-----------+-------+------+------------+------------+

| lastName | firstName | state | city | birthDate | deathDate |

+----------+-----------+-------+------+------------+------------+

| last | pre | xxcv | bcda | 1736-09-01 | 1780-05-08 |

+----------+-----------+-------+------+------------+------------+

1 row in set (0.00 sec)

9. Display the president who was born first.

select \* from president order by birthDate asc limit 1;

+----------+-----------+-------+--------+------------+------------+

| lastName | firstName | state | city | birthDate | deathDate |

+----------+-----------+-------+--------+------------+------------+

| uiop | zxcv | uiop | qwerty | 1410-09-08 | 1467-01-06 |

+----------+-----------+-------+--------+------------+------------+

1 row in set (0.00 sec)

10. Display the names of first five state(in descending order) in which the greatest

number of president have been born.

select state from president group by state order by count(\*) desc limit 5;

+-------+

| state |

+-------+

| YYYY |

| XXYY |

| uiop |

| xxcv |

| efgh |

+-------+

5 rows in set (0.00 sec)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2.

create table person(driverId int(5), name varchar(20), address varchar(20),primary key(driverId));

Query OK, 0 rows affected (0.05 sec)

mysql> create table car(carNo int(5),model varchar(10), year int(4),primary key(carNo));

Query OK, 0 rows affected (0.08 sec)

mysql> create table accident(reportNo int(5), location varchar(10), date date, primary key(reportNo));

Query OK, 0 rows affected (0.06 sec)

mysql> create table owns(driverId int(5), carNo int(5), foreign key(driverId) references person, foreign key(carNo) references car);

Query OK, 0 rows affected (0.07 sec)

mysql> create table participated(driverId int(5), carNo int(5), reportNo int(5), damageAmount int(5), foreign key(driverId) references person, foreign key(carNo) references car, foreign key(reportNo) references accident);

Query OK, 0 rows affected (0.07 sec)

mysql> insert into person values(1,"John Smith","xxyy");

Query OK, 1 row affected (0.00 sec)

mysql> insert into car values(2000,"Model1",1985);

Query OK, 1 row affected (0.00 sec)

mysql> insert into accident values(2196,"xxyy",1989);

Query OK, 1 row affected, 1 warning (0.00 sec)

mysql> insert into owns values(1,2000);

Query OK, 1 row affected (0.00 sec)

mysql> insert into participated values(1,2000,2196,2304);

Query OK, 1 row affected (0.00 sec)

mysql> insert into person values(2,"Second","qwer");

Query OK, 1 row affected (0.00 sec)

mysql> insert into car values(2010,"Next",1988);

Query OK, 1 row affected (0.00 sec)

mysql> insert into accident values(2197,"qwer",1989);

Query OK, 1 row affected, 1 warning (0.00 sec)

mysql> insert into owns values(2,2010);

Query OK, 1 row affected (0.00 sec)

mysql> insert into participated values(2,2010,2197,2000);

Query OK, 1 row affected (0.00 sec)

mysql> insert into person values(3,"Third","uiop");

Query OK, 1 row affected (0.00 sec)

mysql> insert into car values(2020,"after",1978);

Query OK, 1 row affected (0.00 sec)

mysql> insert into accident values(2198,"uiop",1986);

Query OK, 1 row affected, 1 warning (0.00 sec)

mysql> insert into owns values(3,2020);

Query OK, 1 row affected (0.00 sec)

mysql> insert into participated values(3,2020,2198,3200);

Query OK, 1 row affected (0.00 sec)

mysql> select \* from person;

+----------+------------+---------+

| driverId | name | address |

+----------+------------+---------+

| 1 | John Smith | xxyy |

| 2 | Second | qwer |

| 3 | Third | uiop |

+----------+------------+---------+

3 rows in set (0.00 sec)

mysql> select \* from car;

+-------+--------+------+

| carNo | model | year |

+-------+--------+------+

| 2000 | Model1 | 1985 |

| 2010 | Next | 1988 |

| 2020 | after | 1978 |

+-------+--------+------+

3 rows in set (0.00 sec)

mysql> select \* from accident;

+----------+----------+------------+

| reportNo | location | date |

+----------+----------+------------+

| 2196 | xxyy | 1989-02-03 |

| 2197 | qwer | 1989-05-03 |

| 2198 | uiop | 1986-03-03 |

+----------+----------+------------+

3 rows in set (0.00 sec)

mysql> select \* from owns";

"> ";

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '";

"' at line 1

mysql> select \* from owns;

+----------+-------+

| driverId | carNo |

+----------+-------+

| 1 | 2000 |

| 2 | 2010 |

| 3 | 2020 |

+----------+-------+

3 rows in set (0.00 sec)

mysql> select \* from participated;

+----------+-------+----------+--------------+

| driverId | carNo | reportNo | damageAmount |

+----------+-------+----------+--------------+

| 1 | 2000 | 2196 | 2304 |

| 2 | 2010 | 2197 | 2000 |

| 3 | 2020 | 2198 | 3200 |

+----------+-------+----------+--------------+

3 rows in set (0.00 sec)

1. Find the total no. of people who owned cars that were involved in accidents in 1989.

select count(distinct driverId) from accident,participated where accident.reportNo=participated.reportNo and date between "1989-1-1" and "1989-12-31";

+--------------------------+

| count(distinct driverId) |

+--------------------------+

| 2 |

+--------------------------+

1 row in set (0.00 sec)

2. Find the no. of accidents in which the cars belonging to â€˜John Smithâ€™ involved.

select count(\*) from person,participated where person.driverId=participated.driverId and person.name="John Smith";

+----------+

| count(\*) |

+----------+

| 1 |

+----------+

1 row in set (0.00 sec)

3. Add a new accident to the database. Assume any values of required attributes.

insert into accident values(2199,"qqqq","2000-12-12");

Query OK, 1 row affected (0.00 sec)

mysql> select \* from accident;

+----------+----------+------------+

| reportNo | location | date |

+----------+----------+------------+

| 2196 | xxyy | 1989-02-03 |

| 2197 | qwer | 1989-05-03 |

| 2198 | uiop | 1986-03-03 |

| 2199 | qqqq | 2000-12-12 |

+----------+----------+------------+

4 rows in set (0.00 sec)

4. Update the damage amount for the car with the carno â€˜TMA2000â€™ in the accident with

report no â€˜AR2197â€™ to Rs.3000/-

update participated set damageAmount=3000 where reportNo=2197;

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> select \* from participated;

+----------+-------+----------+--------------+

| driverId | carNo | reportNo | damageAmount |

+----------+-------+----------+--------------+

| 1 | 2000 | 2196 | 2304 |

| 2 | 2010 | 2197 | 3000 |

| 3 | 2020 | 2198 | 3200 |

+----------+-------+----------+--------------+

3 rows in set (0.00 sec)

EXERCISE-3

Department:

select \* from department;

+-----+-----------+-------------+

| Dno | Dname | Managername |

+-----+-----------+-------------+

| 1 | admin | qwer |

| 2 | finance | tyuio |

| 3 | sales | abcde |

| 4 | marketing | abcde |

| 5 | abcd | bnmv |

+-----+-----------+-------------+

5 rows in set (0.00 sec)

Emp:

select \* from emp;

+-------+-------+---------+--------+------------+---------------+--------+----------+---------+--------+

| Empno | Ename | address | Sex | dob | dateOfJoining | Deptno | Division | desig | salary |

+-------+-------+---------+--------+------------+---------------+--------+----------+---------+--------+

| 1 | emp1 | asdf | Male | 1990-03-03 | 2010-04-04 | 2 | ne | member | 2000 |

| 2 | emp2 | ghjj | Female | 1998-02-03 | 2015-04-09 | 2 | se | head | 4000 |

| 3 | abcde | iiu | Female | 1988-09-03 | 2005-09-09 | 3 | se | manager | 4000 |

| 4 | tyuio | iiu | Male | 1988-09-03 | 2005-09-09 | 2 | tt | manager | 6000 |

+-------+-------+---------+--------+------------+---------------+--------+----------+---------+--------+

4 rows in set (0.00 sec)

1. List the names of the employes with their division whose salary is not in the range 3000 to 5000.

select Ename,division from emp where salary <3000 or salary>5000;

+-------+----------+

| Ename | division |

+-------+----------+

| emp1 | ne |

| tyuio | tt |

+-------+----------+

2 rows in set (0.00 sec)

2. list the names of employee and their salary whose department name may be in the

group of admin, finance and sales.

select Ename,salary from emp,department where emp.Deptno=department.Dno and Dname in ("admin","finance","sales");

+-------+--------+

| Ename | salary |

+-------+--------+

| emp1 | 2000 |

| emp2 | 4000 |

| abcde | 4000 |

| tyuio | 6000 |

+-------+--------+

4 rows in set (0.00 sec)

3. list the names of employee with dept. names in dept. sales and marketing using

union operator.

select Ename,dname from emp,department where dname="sales" and department.dno=emp.Deptno union select Ename,dname from emp,department where dname="marketing" and department.dno=emp.Deptno;

+-------+-------+

| Ename | dname |

+-------+-------+

| abcde | sales |

+-------+-------+

1 row in set (0.00 sec)

4. list the names of employee common to division “ne” or “se”

select ename from emp where division in ("ne","se");

+-------+

| ename |

+-------+

| emp1 |

| emp2 |

| abcde |

+-------+

3 rows in set (0.00 sec)

5. list the name of the employee who is drawing the highest salary.

select ename from emp where salary=(select max(salary) from emp);+-------+

| ename |

+-------+

| tyuio |

+-------+

1 row in set (0.00 sec)

6. list the jobs with average salary

select desig,avg(salary) from emp group by desig;

+---------+-------------+

| desig | avg(salary) |

+---------+-------------+

| head | 4000.0000 |

| manager | 5000.0000 |

| member | 2000.0000 |

+---------+-------------+

3 rows in set (0.00 sec)

7. list the names of employee who earn lowest salary in each department

select ename,Deptno,min(salary) from emp group by Deptno;

+-------+--------+-------------+

| ename | Deptno | min(salary) |

+-------+--------+-------------+

| emp1 | 2 | 2000 |

| abcde | 3 | 4000 |

+-------+--------+-------------+

2 rows in set (0.00 sec)

8. list the employee details who earn salary greater than the average salary for their

department

select \* from emp e where salary>(select avg(salary) from emp e2 where e.deptno=e2.deptno);

+-------+-------+---------+------+------------+---------------+--------+----------+---------+--------+

| Empno | Ename | address | Sex | dob | dateOfJoining | Deptno | Division | desig | salary |

+-------+-------+---------+------+------------+---------------+--------+----------+---------+--------+

| 4 | tyuio | iiu | Male | 1988-09-03 | 2005-09-09 | 2 | tt | manager | 6000 |

+-------+-------+---------+------+------------+---------------+--------+----------+---------+--------+

1 row in set (0.00 sec)

9. list the employee details who do not manage any one.

select \* from emp where ename not in (select managername from department);

+-------+-------+---------+--------+------------+---------------+--------+----------+--------+--------+

| Empno | Ename | address | Sex | dob | dateOfJoining | Deptno | Division | desig | salary |

+-------+-------+---------+--------+------------+---------------+--------+----------+--------+--------+

| 1 | emp1 | asdf | Male | 1990-03-03 | 2010-04-04 | 2 | ne | member | 2000 |

| 2 | emp2 | ghjj | Female | 1998-02-03 | 2015-04-09 | 2 | se | head | 4000 |

+-------+-------+---------+--------+------------+---------------+--------+----------+--------+--------+

2 rows in set (0.00 sec)

10. list the employee details whose salary is greater than the any one of the manager

select e.\* from emp e,emp e2,department d where e.salary>e2.salary and e2.deptno=d.dno and d.managername=e2.ename;

+-------+-------+---------+------+------------+---------------+--------+----------+---------+--------+

| Empno | Ename | address | Sex | dob | dateOfJoining | Deptno | Division | desig | salary |

+-------+-------+---------+------+------------+---------------+--------+----------+---------+--------+

| 4 | tyuio | iiu | Male | 1988-09-03 | 2005-09-09 | 2 | tt | manager | 6000 |

+-------+-------+---------+------+------------+---------------+--------+----------+---------+--------+

1 row in set (0.00 sec)

11. list the department that has no employees.

select \* from department where department.dno not in (select deptno from emp);

+-----+-----------+-------------+

| Dno | Dname | Managername |

+-----+-----------+-------------+

| 1 | admin | qwer |

| 4 | marketing | abcde |

| 5 | abcd | bnmv |

+-----+-----------+-------------+

3 rows in set (0.00 sec)

12. list the details of employee who is drawing more than the highest paid manager.

select \* from emp where salary>(select max(salary) from emp,department where emp.deptno=department.dno and managername=ename);

Empty set (0.00 sec)

13. list the names of employee with their age.

select ename,extract(year from now())-extract(year from dob) from emp;+-------+-------------------------------------------------+

| ename | extract(year from now())-extract(year from dob) |

+-------+-------------------------------------------------+

| emp1 | 25 |

| emp2 | 17 |

| abcde | 27 |

| tyuio | 27 |

+-------+-------------------------------------------------+

4 rows in set (0.00 sec)

14. list the names of employee whose experience is greater than 10 years.

select ename from emp where extract(year from now())-extract(year from dateOfJoining)>10;

Empty set (0.00 sec)

15. create a view from this table and also create a nested view

create view view2 as select \* from emp;

Query OK, 0 rows affected (0.03 sec)

mysql> select \* from view2;

+-------+-------+---------+--------+------------+---------------+--------+----------+---------+--------+

| Empno | Ename | address | Sex | dob | dateOfJoining | Deptno | Division | desig | salary |

+-------+-------+---------+--------+------------+---------------+--------+----------+---------+--------+

| 1 | emp1 | asdf | Male | 1990-03-03 | 2010-04-04 | 2 | ne | member | 2000 |

| 2 | emp2 | ghjj | Female | 1998-02-03 | 2015-04-09 | 2 | se | head | 4000 |

| 3 | abcde | iiu | Female | 1988-09-03 | 2005-09-09 | 3 | se | manager | 4000 |

| 4 | tyuio | iiu | Male | 1988-09-03 | 2005-09-09 | 2 | tt | manager | 6000 |

+-------+-------+---------+--------+------------+---------------+--------+----------+---------+--------+

4 rows in set (0.00 sec)

NESTED VIEW:

create view view3 as select \* from view2;

Query OK, 0 rows affected (0.04 sec)

mysql> select \* from view3;

+-------+-------+---------+--------+------------+---------------+--------+----------+---------+--------+

| Empno | Ename | address | Sex | dob | dateOfJoining | Deptno | Division | desig | salary |

+-------+-------+---------+--------+------------+---------------+--------+----------+---------+--------+

| 1 | emp1 | asdf | Male | 1990-03-03 | 2010-04-04 | 2 | ne | member | 2000 |

| 2 | emp2 | ghjj | Female | 1998-02-03 | 2015-04-09 | 2 | se | head | 4000 |

| 3 | abcde | iiu | Female | 1988-09-03 | 2005-09-09 | 3 | se | manager | 4000 |

| 4 | tyuio | iiu | Male | 1988-09-03 | 2005-09-09 | 2 | tt | manager | 6000 |

+-------+-------+---------+--------+------------+---------------+--------+----------+---------+--------+

4 rows in set (0.00 sec)

EXERCISE-4

select \* from employee;

+-------+------+-------+-----+------------+----------+--------+--------+----------+------+

| fname | mint | lname | ssn | bdate | address | sex | salary | superssn | dno |

+-------+------+-------+-----+------------+----------+--------+--------+----------+------+

| Abc | R | Smith | 101 | 1994-03-04 | Z street | Male | 10200 | 110 | 1 |

| pqr | Y | Smith | 102 | 1995-09-04 | Y street | Female | 17800 | 110 | 2 |

| pqr | O | xyz | 103 | 1990-10-07 | Y street | Female | 14000 | 108 | 2 |

| iou | O | asd | 104 | 1992-10-10 | L street | Male | 20000 | 111 | 3 |

| Abc | O | Nbm | 110 | 1982-12-10 | L street | Male | 20000 | 110 | 1 |

| asd | D | oq | 111 | 1981-12-10 | L street | Male | 10000 | 111 | 3 |

+-------+------+-------+-----+------------+----------+--------+--------+----------+------+

6 rows in set (0.00 sec)

select \* from department;

+----------+-----+--------+--------------+-----------+

| dname | dno | mgrssn | mgrstartdate | dlocation |

+----------+-----+--------+--------------+-----------+

| research | 1 | 110 | 2001-08-07 | Loc1 |

| dev | 2 | 108 | 2001-02-01 | Loc2 |

| admin | 3 | 111 | 2001-07-07 | Loc3 |

+----------+-----+--------+--------------+-----------+

3 rows in set (0.00 sec)

select \* from project;

+------------+---------+-----------+------+

| pname | pnumber | plocation | dnum |

+------------+---------+-----------+------+

| FirstProj | 1 | stafford | 2 |

| SecondProj | 2 | Loc3 | 3 |

| ThirdProj | 3 | stafford | 1 |

+------------+---------+-----------+------+

3 rows in set (0.00 sec)

select \* from workson;

+------+------+-------+

| essn | pno | hours |

+------+------+-------+

| 101 | 3 | 10 |

| 102 | 3 | 11 |

| 103 | 2 | 21 |

| 104 | 1 | 25 |

+------+------+-------+

4 rows in set (0.00 sec)

select \* from dependent;

+------+---------------+--------+------------+--------------+

| essn | dependentname | sex | bdate | relationship |

+------+---------------+--------+------------+--------------+

| 110 | Abc R Smith | Male | 1994-03-04 | Junior |

| 110 | Pqr Y Smith | Female | 1995-09-04 | Assistant |

| 111 | iou O asd | Male | 1992-10-10 | Assistant |

| 108 | pqr O xyz | Female | 1990-10-07 | Junior |

+------+---------------+--------+------------+--------------+

4 rows in set (0.00 sec)

1. Retrieve the birth date and address of the employees who work for â€˜Researchâ€™ department.

select bdate, address from employee,department where employee.dno=department.dno and dname="research";

+------------+----------+

| bdate | address |

+------------+----------+

| 1994-03-04 | Z street |

| 1982-12-10 | L street |

+------------+----------+

2 rows in set (0.00 sec)

2. For every project located in Stafford list the project number, the controlling department and

dept. managers last name, address and birth date.

select pnumber,dname,lname from employee,department,project where project.dnum=department.dno and plocation="stafford" and department.dno=employee.dno and mgrssn=ssn;

+---------+----------+-------+

| pnumber | dname | lname |

+---------+----------+-------+

| 3 | research | Nbm |

+---------+----------+-------+

1 row in set (0.00 sec)

3. For each employee, retrieve the employees first name and last name and first and last name of

his immediate supervisor.

select distinct e1.fname,e1.lname,e2.fname,e2.lname from employee e1,employee e2 where e1.superssn=e2.ssn and e1.ssn != e2.ssn;

+-------+-------+-------+-------+

| fname | lname | fname | lname |

+-------+-------+-------+-------+

| Abc | Smith | Abc | Nbm |

| pqr | Smith | Abc | Nbm |

| iou | asd | asd | oq |

+-------+-------+-------+-------+

3 rows in set (0.00 sec)

4. Get all informations about employees working in â€˜Researchâ€™ department including the

department information.

select employee.\*, department.\* from employee,department where department.dno=employee.dno and dname="research";

+-------+------+-------+-----+------------+----------+------+--------+----------+------+----------+-----+--------+--------------+-----------+

| fname | mint | lname | ssn | bdate | address | sex | salary | superssn | dno | dname | dno | mgrssn | mgrstartdate | dlocation |

+-------+------+-------+-----+------------+----------+------+--------+----------+------+----------+-----+--------+--------------+-----------+

| Abc | R | Smith | 101 | 1994-03-04 | Z street | Male | 10200 | 110 | 1 | research | 1 | 110 | 2000-01-01 | Loc1 |

| Abc | O | Nbm | 110 | 1982-12-10 | L street | Male | 20000 | 110 | 1 | research | 1 | 110 | 2000-01-01 | Loc1 |

+-------+------+-------+-----+------------+----------+------+--------+----------+------+----------+-----+--------+--------------+-----------+

2 rows in set (0.00 sec)

5. Make a list of all project numbers for project that involve an employee whose last name is

â€˜Smithâ€™ as a worker or as a manager of the department that controls the project.

select distinct pno from workson,employee,department,project where lname="Smith" and (essn=ssn or mgrssn=ssn and dnum=department.dno);

+------+

| pno |

+------+

| 3 |

+------+

1 row in set (0.00 sec)

6. Retrieve the names of employees who have no dependents.

select distinct fname,mint,lname from employee,dependent where ssn not in (select essn from dependent);

+-------+------+-------+

| fname | mint | lname |

+-------+------+-------+

| Abc | R | Smith |

| pqr | Y | Smith |

| pqr | O | xyz |

| iou | O | asd |

+-------+------+-------+

4 rows in set (0.00 sec)

7. Retrieve the name of each employee who has a dependent with the same first name and same

sex as the employee.

select fname,mint,lname from employee,dependent where ssn=essn and fname=substring(dependentname,1,locate(" ",dependentname,1)) and employee.sex=dependent.sex;

+-------+------+-------+

| fname | mint | lname |

+-------+------+-------+

| Abc | O | Nbm |

+-------+------+-------+

1 row in set (0.00 sec)

8. Find the maximum and minimum salary of all employees

select max(salary),min(salary) from employee;

+-------------+-------------+

| max(salary) | min(salary) |

+-------------+-------------+

| 20000 | 10000 |

+-------------+-------------+

1 row in set (0.00 sec)

9. Count the no of distinct salary values in the database.

select count(distinct salary) from employee;

+------------------------+

| count(distinct salary) |

+------------------------+

| 5 |

+------------------------+

1 row in set (0.00 sec)

10. Create index for employee table based on fname

create index id1 on employee(fname);

Query OK, 6 rows affected (0.07 sec)

Records: 6 Duplicates: 0 Warnings: 0

11. Create a view with fname and salary

create view view1 as select fname,salary from employee;

Query OK, 0 rows affected (0.04 sec)

mysql> select \* from view1;

+-------+--------+

| fname | salary |

+-------+--------+

| Abc | 10200 |

| pqr | 17800 |

| pqr | 14000 |

| iou | 20000 |

| Abc | 20000 |

| asd | 10000 |

+-------+--------+

6 rows in set (0.00 sec)

12. Create a view with fname, deptid, projectnumber

create view view2 as select pno, fname,dno from employee,workson where essn=ssn;

Query OK, 0 rows affected (0.02 sec)

mysql> select \* from view2;

+------+-------+------+

| pno | fname | dno |

+------+-------+------+

| 3 | Abc | 1 |

| 3 | pqr | 2 |

| 2 | pqr | 2 |

| 1 | iou | 3 |

+------+-------+------+

4 rows in set (0.00 sec)

13. List the total salary paid to the employees

select sum(salary) from employee;

+-------------+

| sum(salary) |

+-------------+

| 92000 |

+-------------+

1 row in set (0.00 sec)

14. List the maximum, minimum and average salary in the company.

select min(salary),max(salary),avg(salary) from employee;

+-------------+-------------+-------------+

| min(salary) | max(salary) | avg(salary) |

+-------------+-------------+-------------+

| 10000 | 20000 | 15333.3333 |

+-------------+-------------+-------------+

1 row in set (0.00 sec)

15. List employee names for those who have joined between 30-6-2001 and 30-12-2001.

select fname,mint,lname from employee,department where mgrssn=ssn and mgrstartdate between "2001-6-6" and "2001-12-30";

+-------+------+-------+

| fname | mint | lname |

+-------+------+-------+

| Abc | O | Nbm |

| asd | D | oq |

+-------+------+-------+

2 rows in set (0.00 sec)

EXERCISE-5

select \* from branch;

+------------+----------+--------+

| branchname | location | assets |

+------------+----------+--------+

| First | X | A |

| Second | Y | B |

| Third | X | B |

| Fourth | Y | A |

+------------+----------+--------+

4 rows in set (0.00 sec)

select \* from customer;

+--------+----------+----------+------+

| custid | custname | address1 | city |

+--------+----------+----------+------+

| 4 | lmn | U | O |

| 2 | pqr | X | P |

| 3 | def | Y | Q |

| 1 | abc | X | P |

+--------+----------+----------+------+

4 rows in set (0.00 sec)

select \* from account;

+-------+------------+---------+

| accno | startdate | balance |

+-------+------------+---------+

| 1 | 2000-02-02 | 500 |

| 2 | 2010-05-02 | 1500 |

| 3 | 1990-07-07 | 3500 |

| 4 | 1996-09-07 | 3600 |

| 5 | 1992-09-07 | 9000 |

| 6 | 1992-01-02 | 10000 |

+-------+------------+---------+

6 rows in set (0.00 sec)

mysql> select \* from depositor;

+-------+--------+------------+

| accno | custid | branchname |

+-------+--------+------------+

| 1 | 2 | First |

| 2 | 1 | Fourth |

| 3 | 2 | Third |

| 4 | 2 | Third |

| 5 | 3 | Second |

| 6 | 4 | First |

+-------+--------+------------+

6 rows in set (0.00 sec)

select \* from loan;

+--------+--------+

| loanid | amount |

+--------+--------+

| 1 | 300 |

| 2 | 500 |

| 3 | 1000 |

| 4 | 2000 |

| 5 | 5000 |

| 6 | 5500 |

+--------+--------+

6 rows in set (0.00 sec)

mysql> select \* from borrower;

+--------+--------+------------+

| loanno | custid | branchname |

+--------+--------+------------+

| 1 | 4 | First |

| 2 | 2 | Third |

| 3 | 2 | Second |

| 4 | 2 | Fourth |

| 5 | 1 | Fourth |

| 6 | 3 | First |

+--------+--------+------------+

6 rows in set (0.00 sec)

Write a function

a.to return the asset if branch name is given

create function returnAsset(bname varchar(20)) returns varchar(20)

-> deterministic

-> begin

-> declare a varchar(20);

-> select assets into a from branch where branchname=bname;

-> return a;

-> end

-> $$

select returnAsset("First") from branch limit 1;

+----------------------+

| returnAsset("First") |

+----------------------+

| A |

+----------------------+

1 row in set (0.00 sec)

b.

to return the no of accounts in a particular branch

create function countacc2(bname varchar(20)) returns int(3)

deterministic

begin

declare c int(3);

select count(\*) into c from depositor where branchname=bname group by branchname;

return c;

end

$$

select countacc2("First") from depositor limit 1;

+--------------------+

| countacc2("First") |

+--------------------+

| 2 |

+--------------------+

1 row in set (0.00 sec)

c.

to return the account details of a customer

create function details (id int(5)) returns int(5)

deterministic

begin

declare acc int(5);

select account.accno into acc from account,depositor where account.accno=depositor.accno and custid=id;

return acc;

end$$

Query OK, 0 rows affected (0.00 sec)

mysql> select \* from account where accno=details(3)$$

+-------+------------+---------+

| accno | startdate | balance |

+-------+------------+---------+

| 5 | 1992-09-07 | 9000 |

+-------+------------+---------+

1 row in set (0.00 sec)

Write a Procedure

a.

to find the customer name who got more than 2 accounts

create procedure Y(out n varchar(20))

begin

select custname from customer where custid in (select custid from depositor group by custid having count(\*)>2) into n;

end

$$

call Y(@ans);

select @ans;

+------+

| @ans |

+------+

| pqr |

+------+

1 row in set (0.00 sec)

b. to find the joint account holder details

create procedure joint3()

begin

select \* from customer where custid in (select custid from depositor where accno in(select d2.accno from depositor d2 where d2.custid!=depositor.custid));

end$$

call joint3();

-> $$

+--------+----------+----------+------+

| custid | custname | address1 | city |

+--------+----------+----------+------+

| 2 | pqr | X | P |

| 3 | def | Y | Q |

+--------+----------+----------+------+

2 rows in set (0.00 sec)

c.to find the total amounts deposited in a branch

create procedure tot(in bname varchar(20), out sum int(6))

begin

select sum(balance) from account where accno in (select accno from depositor where branchname = bname) into sum;

end

$$

call tot("First",@sum);

select @sum;

+-------+

| @sum |

+-------+

| 10500 |

+-------+

1 row in set (0.00 sec)

select \* from salesman;

+------------+--------------+---------+--------+--------+

| salesmanNo | salesmanName | address | sex | salary |

+------------+--------------+---------+--------+--------+

| 1 | Abc | Q | Male | 5000 |

| 2 | cba | W | Female | 6000 |

| 3 | xya | W | Male | 10000 |

| 4 | rrt | T | Female | 3000 |

+------------+--------------+---------+--------+--------+

4 rows in set (0.00 sec)

select \* from transaction;

+------------+-------------+------+

| salesmanNo | dateOfSales | qty |

+------------+-------------+------+

| 1 | 2000-04-04 | 20 |

| 1 | 2000-08-04 | 10 |

| 2 | 1990-07-06 | 30 |

| 2 | 1960-08-06 | 20 |

| 3 | 1990-07-06 | 2 |

| 4 | 2015-08-08 | 33 |

+------------+-------------+------+

6 rows in set (0.00 sec)

mysql> select \* from commission;

+------------+------------+-----------+

| salesmanNo | commission | targetQty |

+------------+------------+-----------+

| 1 | 3000 | 20 |

| 2 | 1000 | 30 |

| 3 | 5500 | 40 |

| 4 | 500 | 20 |

+------------+------------+-----------+

4 rows in set (0.00 sec)

1. create UDF to display the salesman details for the given salesmanno

create function display2(sno int(5)) returns varchar(40)

deterministic

begin

declare ans varchar(40);

select concat(salesmanName," ", address," ", sex ," ", salary) into ans from salesman where salesmanNo=sno;

return ans;

end $$

select display2(2) from salesman limit 1;

+-------------------+

| display2(2) |

+-------------------+

| cba W Female 6000 |

+-------------------+

1 row in set (0.00 sec)

2. create UDF to display the commission details for the given salesmanno

create function displayComm(sno int(5)) returns varchar(40)

deterministic

begin

declare ans varchar(40);

select concat(commission," ", targetQty) into ans from commission where salesmanNo=sno;

return ans;

end $$

select displayComm(3);

-> $$

+----------------+

| displayComm(3) |

+----------------+

| 5500 40 |

+----------------+

1 row in set (0.00 sec)

3. create UDF to display salesman no, name, who is getting maximum commission

create function b(com int(5)) returns int(5)

-> deterministic

-> begin

-> declare s int(5);

-> select salesmanNo into s from commission where commission=com;

-> return s;

-> end

-> $$

create function d(no int(5)) returns varchar(30)

deterministic

begin

declare s varchar(30);

select concat(salesmanNo," ",salesmanName) into s from salesman where salesmanNo=no;

return s;

end

$$

select d(b(max(commission))) from commission;

-----------------------+

| d(b(max(commission))) |

+-----------------------+

| 3 xya |

+-----------------------+

1 row in set (0.00 sec)

ite a Procedure

a.

to display the commission details for a particular salesman

create procedure t(in s int(5),out comm int(6),out targ int(3)) begin select commission from commission where salesmanNo=s into comm;

-> select targetQty from commission where salesmanNo=s into targ;

-> end

-> $$

call t(1,@com,@tar);

-> $$

Query OK, 0 rows affected (0.00 sec)

mysql> select @com,@tar;

-> $$

+------+------+

| @com | @tar |

+------+------+

| 3000 | 20 |

+------+------+

1 row in set (0.00 sec)

b.

to find the salesman name who got commission more than 2000

create procedure r()

begin

select salesmanName from salesman where salesmanNo in (select salesmanNo from commission where commission>2000);

end

$$

c.

to display the transaction details for a particular salesman

create procedure p(in s int(5)) begin select \* from transaction where salesmanNo=s; end$$

Query OK, 0 rows affected (0.00 sec)

mysql> call p(2)$$

+------------+-------------+------+

| salesmanNo | dateOfSales | qty |

+------------+-------------+------+

| 2 | 1990-07-06 | 30 |

| 2 | 1960-08-06 | 20 |

+------------+-------------+------+

2 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

EXERCISE-6

1. Create a trigger which will calculate the number of rows we have inserted till now

create table tb(num int, co int);

Query OK, 0 rows affected (0.05 sec)

mysql> set @count=0;

Query OK, 0 rows affected (0.00 sec)

create trigger trig1

before insert on tb

for each row

begin set @count=@count+1;

set NEW.co=@count;

end//

insert into tb values(0,0);

Query OK, 1 row affected (0.00 sec)

mysql> insert into tb values(0,0);

Query OK, 1 row affected (0.00 sec)

mysql> insert into tb values(0,0);

Query OK, 1 row affected (0.00 sec)

mysql> insert into tb values(0,0);

Query OK, 1 row affected (0.00 sec)

mysql> select \* from tb;

+------+------+

| num | co |

+------+------+

| 0 | 1 |

| 0 | 2 |

| 0 | 3 |

| 0 | 4 |

+------+------+

4 rows in set (0.00 sec)

2. Create a trigger that displays a message prior to an insert operation on the

payment

table

create table payments(buyer varchar(20), amount int);

create trigger displaymsg

before insert on payments

for each row

begin

set @x='Row inserted';

end //

insert into payments values('X',100);

insert into payments values('Y',200);

insert into payments values('Z',300);

select @x,payments.\* from payments;

+--------------+-------+--------+

| @x | buyer | amount |

+--------------+-------+--------+

| Row inserted | X | 100 |

| Row inserted | Y | 200 |

| Row inserted | Z | 300 |

+--------------+-------+--------+

3 rows in set (0.00 sec)

3. Create a trigger that whenever an insert, update, or delete operation occurs on the

sales\_by\_store table, a row is added to the salesauditlog table recording the date, user, and action

create trigger o before insert on S for each row begin insert into salesauditlog values(current\_date,CURRENT\_USER,'insert'); end //

Query OK, 0 rows affected (0.07 sec)

mysql> create trigger o1 before update on S for each row begin insert into salesauditlog values(current\_date,CURRENT\_USER,'update'); end //

Query OK, 0 rows affected (0.05 sec)

mysql> create trigger o2 before delete on S for each row begin insert into salesauditlog values(current\_date,CURRENT\_USER,'delete'); end //

Query OK, 0 rows affected (0.05 sec)

mysql> insert into S values ("A",200);

-> //

Query OK, 1 row affected (0.00 sec)

mysql> delimiter ;

mysql> insert into S values ("B",220);

Query OK, 1 row affected (0.00 sec)

mysql> insert into S values ("C",120);

Query OK, 1 row affected (0.00 sec)

mysql> update S set store="D" where store = "C";

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> delete from S where sales=120;

Query OK, 1 row affected (0.00 sec)

mysql> select \* from salesauditlog;

+------------+----------------+--------+

| date | user | action |

+------------+----------------+--------+

| 2015-09-15 | root@localhost | insert |

| 2015-09-15 | root@localhost | insert |

| 2015-09-15 | root@localhost | insert |

| 2015-09-15 | root@localhost | update |

| 2015-09-15 | root@localhost | delete |

+------------+----------------+--------+

5 rows in set (0.00 sec)

EXERCISE-7

select \* from order1;

+----------+------------+----------+

| ordernum | item\_price | quantity |

+----------+------------+----------+

| 1 | 20 | 9 |

| 2 | 200 | 15 |

| 3 | 100 | 5 |

| 4 | 1000 | 7 |

| 5 | 100 | 20 |

| 6 | 100 | 2 |

+----------+------------+----------+

6 rows in set (0.00 sec)

1. Create a procedure for order table(Ordernum, item\_price, quantity), given an ordernum

procedure has to give its total amount. calculate 10% tax for the amount greater than 1000.

create procedure p1(in num int,out amt int)

begin

select item\_price\*quantity+0.1\*(item\_price\*quantity)\*(item\_price\*quantity>1000) into amt from order1 where ordernum=num;

end

//

call p1(1,@amount);

Query OK, 0 rows affected (0.00 sec)

mysql> select @amount;

+---------+

| @amount |

+---------+

| 180 |

+---------+

1 row in set (0.00 sec)

mysql> call p1(5,@amount);

Query OK, 0 rows affected (0.00 sec)

mysql> select @amount;

+---------+

| @amount |

+---------+

| 2200 |

+---------+

1 row in set (0.00 sec)

2. write a function that will display item\_price for a given ordernum.

create function f1(ordernum int)

returns int

deterministic

begin

declare price int;

select item\_price into price from order1 where order1.ordernum=ordernum;

return price;

end//

select f1(2) from order1 limit 1;

+-------+

| f1(2) |

+-------+

| 200 |

+-------+

1 row in set (0.00 sec)

3. Write a function that will display the ordernum having highest price

create function f2()

returns int deterministic

begin declare num int;

select ordernum into num from order1 where item\_price=(select max(item\_price) from order1) limit 1;

return num;

end//

select f2();

+------+

| f2() |

+------+

| 4 |

+------+

1 row in set (0.00 sec)

4. Table pass(ID, Status, addedon)-

status takes values such as (Active, Expires), Write a trigger and a

procedure to set status to expires if it is more than 7 days ago( Addedon is field carries timestamp

value).

create trigger t1

before insert

on pass

for each row

begin if(datediff(current\_date,NEW.addedon)>=7)

then set NEW.status="Expires";

end if;

end

//

insert into pass values(1,"Active","2015-9-18");

insert into pass values(2,"Active","2015-9-12");

insert into pass values(3,"Active","2015-9-13");

insert into pass values(4,"Active","2015-9-22");

select \* from pass;

+------+---------+------------+

| id | status | addedon |

+------+---------+------------+

| 4 | Active | 2015-09-22 |

| 3 | Expires | 2015-09-13 |

| 2 | Expires | 2015-09-12 |

| 1 | Active | 2015-09-18 |

+------+---------+------------+

4 rows in set (0.00 sec)

PROCEDURE:

create procedure p2()

begin

update pass set status="Expires" where datediff(current\_date,addedon)>=7;

end

//