1.Create Table

a.Write a SQL statement to create a table named jobs including columns job\_id, job\_title, min\_salary, max\_salary and check whether the max\_salary amount exceeding the upper limit 25000.

Query to create table;

mysql> create table jobs(job\_id int, job\_title varchar(50), min\_salary int, max\_salary int,primary key(job\_id));

Query OK, 0 rows affected (0.03 sec)

mysql> desc jobs;

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

| Field | Type | Null | Key | Default | Extra |

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

| job\_id | int | NO | PRI | NULL | |

| job\_title | varchar(50) | YES | | NULL | |

| min\_salary | int | YES | | NULL | |

| max\_salary | int | YES | | NULL | |

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

4 rows in set (0.01 sec)

b. Write a SQL statement to create a table named countries including columns country\_id,

country\_name and region\_id and make sure that no countries except Italy, India and China

will be entered in the table.

Query:

mysql> create table countries(country\_id int ,country\_name varchar(50),name enum('Italy','India','China'),region\_id varchar(50),primary key(country\_id));

Query OK, 0 rows affected (0.03 sec)

mysql> desc countries;

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

| Field | Type | Null | Key | Default | Extra |

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

| country\_id | int | NO | PRI | NULL | |

| country\_name | varchar(50) | YES | | NULL | |

| name | enum('Italy','India','China') | YES | | NULL | |

| region\_id | varchar(50) | YES | | NULL | |

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

4 rows in set (0.01 sec)

c.Write a SQL statement to create a table job\_history including columns employee\_id, start\_date,

end\_date, job\_id and department\_id and make sure that, the employee\_id column does not contain any duplicate value

at the time of insertion .

Query of the job\_history:

mysql> create table job\_history (employee\_id int(3) primary key,start\_date date,end\_date date,job\_id int(3),department\_id int(3));

Query OK, 0 rows affected, 3 warnings (0.06 sec)

mysql> desc job\_history;

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

| Field | Type | Null | Key | Default | Extra |

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

| employee\_id | int | NO | PRI | NULL | |

| start\_date | date | YES | | NULL | |

| end\_date | date | YES | | NULL | |

| job\_id | int | YES | MUL | NULL | |

| department\_id | int | YES | | NULL | |

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

5 rows in set (0.01 sec)

mysql> desc job\_History;

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

| Field | Type | Null | Key | Default | Extra |

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

| employee\_id | varchar(10) | NO | PRI | NULL | |

| start\_date | date | YES | | NULL | |

| end\_date | date | YES | | NULL | |

| job\_id | varchar(10) | YES | | NULL | |

| department\_id | varchar(10) | YES | | NULL | |

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

5 rows in set (0.01 sec)

2.Alter Table

a.Write a SQL statement to add a foreign key constraint named fk\_job\_id on job\_id column of

job\_history table referencing to the primary key job\_id of jobs table.

Query :

mysql> alter table job\_history add constraint fk\_job\_id foreign key(job\_id) references jobs(job\_id);

Query OK, 0 rows affected (0.10 sec)

Records: 0 Duplicates: 0 Warnings: 0

b.Write a SQL statement to rename the table countries to country\_new.

Query: to change the table name;

mysql> alter table countries rename to country\_new;

Query OK, 0 rows affected (0.03 sec)

c.Write a SQL statement to change the name of the column country\_id to con\_id,

keeping the data type and size same.

query: to change the column name;

mysql> alter table country\_new rename column country\_id to con\_id;

Query OK, 0 rows affected (0.04 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql> desc country\_new;

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

| Field | Type | Null | Key | Default | Extra |

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

| con\_id | int | NO | PRI | NULL | |

| country\_name | varchar(50) | YES | | NULL | |

| name | enum('Italy','India','China') | YES | | NULL | |

| region\_id | varchar(50) | YES | | NULL | |

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

4 rows in set (0.01 sec)

3.DML Statement

a.Write a SQL statement to insert a record with your own value into the table countries against each columns.

query to insert;

mysql> insert into country\_new values(1000,'India','India',101);

Query OK, 1 row affected (0.01 sec)

mysql> insert into country\_new values(1001,'Hong kong','Italy',102);

Query OK, 1 row affected (0.00 sec)

mysql> insert into country\_new values(1002,'Switzerland','China',102);

Query OK, 1 row affected (0.00 sec)

mysql> insert into country\_new values(1003,'south korea','China',104);

Query OK, 1 row affected (0.01 sec)

after insert table

query

mysql> select \* from country\_new;

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

| con\_id | country\_name | name | region\_id |

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

| 1000 | India | India | 101 |

| 1001 | Hong kong | Italy | 102 |

| 1002 | Switzerland | China | 102 |

| 1003 | south korea | China | 104 |

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

4 rows in set (0.00 sec)

b.Write a SQL statement to insert 5 rows by a single insert statement into jobs table.

query:

mysql> insert into job\_history values(100,' Marketing Representative',25000,35000),(101,' Public Relations Representative',20000,30000),(102,'Human Resources Representative ',45000,65000),(103,'Sales Representative ',98000,76000),(104,'Administration Vice President ',67000,98000);

Query OK, 5 rows affected (0.01 sec)

Records: 5 Duplicates: 0 Warnings: 0

mysql> select \* from jobs;

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

| job\_id | job\_title | min\_salary | max\_salary |

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

| 100 | Marketing Representative | 25000 | 35000 |

| 101 | Public Relations Representative | 20000 | 30000 |

| 102 | Human Resources Representative | 45000 | 65000 |

| 103 | Sales Representative | 98000 | 76000 |

| 104 | Administration Vice President | 67000 | 98000 |

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

5 rows in set (0.00 sec)

c.Write a SQL statement to change Min\_salary to 8000 and max\_Salary to 40000 for programmer.

query

before query table

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

| job\_id | job\_title | min\_salary | max\_salary |

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

| 100 | Marketing Representative | 25000 | 35000 |

| 101 | Public Relations Representative | 20000 | 30000 |

| 102 | Human Resources Representative | 45000 | 65000 |

| 103 | Sales Representative | 98000 | 76000 |

| 104 | Administration Vice President | 67000 | 98000 |

| 105 | programmer | 25000 | 35000 |

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

6 rows in set (0.00 sec)

Query to update min\_salary:

mysql> update jobs set min\_salary = '8000' WHERE job\_id = 105 ;

Query OK, 1 row affected (0.01 sec)

Rows matched: 1 Changed: 1 Warnings: 0

Query to update max\_salary;

mysql> update jobs set max\_salary='40000' where job\_id=105;

Query OK, 1 row affected (0.01 sec)

Rows matched: 1 Changed: 1 Warnings: 0

after query

mysql> select \* from jobs;

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

| job\_id | job\_title | min\_salary | max\_salary |

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

| 100 | Marketing Representative | 25000 | 35000 |

| 101 | Public Relations Representative | 20000 | 30000 |

| 102 | Human Resources Representative | 45000 | 65000 |

| 103 | Sales Representative | 98000 | 76000 |

| 104 | Administration Vice President | 67000 | 98000 |

| 105 | programmer | 8000 | 40000 |

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

6 rows in set (0.00 sec)