Java Enterprise Edition

ANUDIP FOUNDATION

MySQL Create Table Time: 90 Min

1. Write a SQL statement to create a simple table countries including columns country_id,country_name and region_id.

Sample Solution:

```
CREATE TABLE countries(

COUNTRY_ID varchar(2),

COUNTRY_NAME varchar(40),

REGION_ID decimal(10,0)

);
```

Let execute the above code in MySQL 5.6 command prompt

Here is the structure of the table:

```
mysql> DESC countries;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| COUNTRY_ID   | varchar(2)    | YES  |     | NULL    |       |
| COUNTRY_NAME | varchar(40)   | YES  |     | NULL    |       |
| REGION_ID    | decimal(10,0) | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

2. Write a SQL statement to create a simple table countries including columns country_id,country_name and region_id which is already exists.

Sample Solution:

```
CREATE TABLE IF NOT EXISTS countries (

COUNTRY_ID varchar(2),

COUNTRY_NAME varchar(40),

REGION_ID decimal(10,0)

);
```

Let execute the above code in MySQL 5.6 command prompt

Here is the structure of the table:

```
mysql> DESC countries;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| COUNTRY_ID   | varchar(2)    | YES  |     | NULL    |       |
| COUNTRY_NAME | varchar(40)   | YES  |     | NULL    |       |
| REGION_ID    | decimal(10,0) | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
3 rows in set (0.13 sec)
```

3. Write a SQL statement to create the structure of a table dup_countries similar to countries.

Sample Solution:

```
CREATE TABLE IF NOT EXISTS dup_countries

LIKE countries;
```

Let execute the above code in MySQL 5.6 command prompt

Here is the structure of the table:

```
mysql> DESC dup_countries;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| COUNTRY_ID   | varchar(2)    | YES  |     | NULL    |       |
| COUNTRY_NAME | varchar(40)   | YES  |     | NULL    |       |
| REGION_ID    | decimal(10,0) | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
3 rows in set (0.03 sec)
```

4. Write a SQL statement to create a duplicate of countries table including structure and data by name dup_countries.

Sample Solution:

```
CREATE TABLE IF NOT EXISTS dup_countries

AS SELECT * FROM  countries;
```

Let execute the above code in MySQL 5.6 command prompt

Here is the structure of the table:

```
mysql> DESC dup_countries;
```

```
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| COUNTRY_ID   | varchar(2)    | YES  |     | NULL    |       |
| COUNTRY_NAME | varchar(40)   | YES  |     | NULL    |       |
| REGION_ID    | decimal(10,0) | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
3 rows in set (0.11 sec)
```

5. Write a SQL statement to create a table countries set a constraint NULL.

Sample Solution:

```
CREATE TABLE IF NOT EXISTS countries (

COUNTRY_ID varchar(2) NOT NULL,

COUNTRY_NAME varchar(40) NOT NULL,

REGION_ID decimal(10,0) NOT NULL

);
```

Let execute the above code in MySQL 5.6 command prompt

Here is the structure of the table:

```
mysql> desc countries;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| COUNTRY_ID   | varchar(2)    | NO   |     | NULL    |       |
| COUNTRY_NAME | varchar(40)   | NO   |     | NULL    |       |
| REGION_ID    | decimal(10,0) | NO   |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

6. 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.

Sample Solution:

```
CREATE TABLE IF NOT EXISTS jobs (

JOB_ID varchar(10) NOT NULL ,

JOB_TITLE varchar(35) NOT NULL,

MIN_SALARY decimal(6,0),
```

```
MAX_SALARY decimal(6,0)

CHECK(MAX_SALARY<=25000)

);
```

Let execute the above code in MySQL 5.6 command prompt

Here is the structure of the table:

```
mysql> DESC jobs;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| JOB_ID     | varchar(10) | NO   |     | NULL    |       |
| JOB_TITLE  | varchar(35) | NO   |     | NULL    |       |
| MIN_SALARY | decimal(6,0) | YES  |     | NULL    |       |
| MAX_SALARY | decimal(6,0) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
4 rows in set (0.16 sec)
```

7. 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.

Sample Solution:

```
CREATE TABLE IF NOT EXISTS countries (

COUNTRY_ID varchar(2),

COUNTRY_NAME varchar(40)

CHECK(COUNTRY_NAME IN('Italy','India','China')) ,

REGION_ID decimal(10,0)

);
```

Let execute the above code in MySQL 5.6 command prompt

Here is the structure of the table:

```
mysql> DESC countries;
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| COUNTRY_ID   | varchar(2)   | YES  |     | NULL    |       |
| COUNTRY_NAME | varchar(40)  | YES  |     | NULL    |       |
```

```
| REGION_ID   | decimal(10,0) | YES  |     | NULL    |       |
+-------------+---------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

8. Write a SQL statement to create a table named job_histry including columns employee_id, start_date, end_date, job_id and department_id and make sure that the value against column end_date will be entered at the time of insertion to the format like '--/--/----'.

Sample Solution:

```
CREATE TABLE IF NOT EXISTS job_history (

EMPLOYEE_ID decimal(6,0) NOT NULL,

START_DATE date NOT NULL,

END_DATE date NOT NULL

CHECK (END_DATE LIKE '--/--/----'),

JOB_ID varchar(10) NOT NULL,

DEPARTMENT_ID decimal(4,0) NOT NULL

);
```

Let execute the above code in MySQL 5.6 command prompt

Here is the structure of the table:

```
mysql> DESC job_history;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| EMPLOYEE_ID  | decimal(6,0) | NO   |     | NULL    |       |
| START_DATE   | date        | NO   |     | NULL    |       |
| END_DATE     | date        | NO   |     | NULL    |       |
| JOB_ID       | varchar(10) | NO   |     | NULL    |       |
| DEPARTMENT_ID | decimal(4,0) | NO   |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
5 rows in set (0.04 sec)
```

9. Write a SQL statement to create a table named countries including columns country_id,country_name and region_id and make sure that no duplicate data against column country_id will be allowed at the time of insertion.

Sample Solution:

```
CREATE TABLE IF NOT EXISTS countries (
COUNTRY_ID varchar(2) NOT NULL,
COUNTRY_NAME varchar(40) NOT NULL,
REGION_ID decimal(10,0) NOT NULL,
UNIQUE(COUNTRY_ID)
);
```

Let execute the above code in MySQL 5.6 command prompt

Here is the structure of the table:

```
mysql> DESC countries;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| COUNTRY_ID   | varchar(2)    | YES  |     | NULL    |       |
| COUNTRY_NAME | varchar(40)   | YES  |     | NULL    |       |
| REGION_ID    | decimal(10,0) | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

10. Write a SQL statement to create a table named jobs including columns job_id, job_title, min_salary and max_salary, and make sure that, the default value for job_title is blank and min_salary is 8000 and max_salary is NULL will be entered automatically at the time of insertion if no value assigned for the specified columns.

Sample Solution:

```
CREATE TABLE IF NOT EXISTS jobs (
JOB_ID varchar(10) NOT NULL UNIQUE,
JOB_TITLE varchar(35) NOT NULL DEFAULT ' ',
MIN_SALARY decimal(6,0) DEFAULT 8000,
MAX_SALARY decimal(6,0) DEFAULT NULL
);
```

Let execute the above code in MySQL 5.6 command prompt

Here is the structure of the table:

```
mysql> DESC jobs;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| JOB_ID     | varchar(10) | NO   | PRI | NULL    |       |
| JOB_TITLE  | varchar(35) | NO   |     |         |       |
| MIN_SALARY | decimal(6,0) | YES |     | 8000    |       |
| MAX_SALARY | decimal(6,0) | YES |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

11. Write a SQL statement to create a table named countries including columns country_id, country_name and region_id and make sure that the country_id column will be a key field which will not contain any duplicate data at the time of insertion.

Sample Solution:

```
CREATE TABLE IF NOT EXISTS countries (

COUNTRY_ID varchar(2) NOT NULL UNIQUE PRIMARY KEY,

COUNTRY_NAME varchar(40) NOT NULL,

REGION_ID decimal(10,0) NOT NULL

);
```

Copy
Let execute the above code in MySQL 5.6 command prompt

Here is the structure of the table:

```
mysql> DESC countries;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| COUNTRY_ID  | varchar(2)   | YES  |     | NULL    |       |
| COUNTRY_NAME | varchar(40) | YES  |     | NULL    |       |
| REGION_ID   | decimal(10,0) | YES |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

12. Write a SQL statement to create a table countries including columns country_id, country_name and region_id and make sure that the column country_id will be unique and store an auto incremented value.

Sample Solution:

```
CREATE TABLE IF NOT EXISTS countries (

COUNTRY_ID integer NOT NULL UNIQUE AUTO_INCREMENT PRIMARY KEY,
```

```
COUNTRY_NAME varchar(40) NOT NULL,

REGION_ID decimal(10,0) NOT NULL

);

DESC countries;
```

Copy
Let execute the above code in MySQL 5.6 command prompt

Here is the structure of the table:

```
mysql> DESC countries;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| COUNTRY_ID   | varchar(2)    | NO   | PRI |         |       |
| COUNTRY_NAME | varchar(40)   | YES  |     | NULL    |       |
| REGION_ID    | decimal(10,0) | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

13. Write a SQL statement to create a table countries including columns country_id, country_name and region_id and make sure that the combination of columns country_id and region_id will be unique.

Sample Solution:

```
CREATE TABLE IF NOT EXISTS countries (

COUNTRY_ID varchar(2) NOT NULL UNIQUE DEFAULT '',

COUNTRY_NAME varchar(40) DEFAULT NULL,

REGION_ID decimal(10,0) NOT NULL,

PRIMARY KEY (COUNTRY_ID,REGION_ID));
```

Let execute the above code in MySQL 5.6 command prompt

Here is the structure of the table:

```
mysql> DESC countries;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| COUNTRY_ID   | varchar(2)    | NO   | PRI |         |       |
| COUNTRY_NAME | varchar(40)   | YES  |     | NULL    |       |
| REGION_ID    | decimal(10,0) | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
```

```
3 rows in set (0.01 sec)
```

14. 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 and the foreign key column job_id contain only those values which are exists in the jobs table.

Here is the structure of the table jobs;

```
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| JOB_ID     | varchar(10)  | NO   | PRI |         |       |
| JOB_TITLE  | varchar(35)  | NO   |     | NULL    |       |
| MIN_SALARY | decimal(6,0) | YES  |     | NULL    |       |
| MAX_SALARY | decimal(6,0) | YES  |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
```

Sample Solution:

```
CREATE TABLE job_history (

EMPLOYEE_ID decimal(6,0) NOT NULL PRIMARY KEY,

START_DATE date NOT NULL,

END_DATE date NOT NULL,

JOB_ID varchar(10) NOT NULL,

DEPARTMENT_ID decimal(4,0) DEFAULT NULL,

FOREIGN KEY (job_id) REFERENCES jobs(job_id)

)ENGINE=InnoDB;
```

Copy
Let execute the above code in MySQL 5.6 command prompt

Here is the structure of the table:

```
mysql> DESC job_history;
+---------------+--------------+------+-----+---------+-------+
| Field         | Type         | Null | Key | Default | Extra |
+---------------+--------------+------+-----+---------+-------+
| EMPLOYEE_ID   | decimal(6,0) | NO   | PRI | NULL    |       |
| START_DATE    | date         | NO   |     | NULL    |       |
| END_DATE      | date         | NO   |     | NULL    |       |
| JOB_ID        | varchar(10)  | NO   | MUL | NULL    |       |
| DEPARTMENT_ID | decimal(4,0) | YES  |     | NULL    |       |
+---------------+--------------+------+-----+---------+-------+
```

```
5 rows in set (0.02 sec)
```

15. Write a SQL statement to create a table employees including columns employee_id, first_name, last_name, email, phone_number hire_date, job_id, salary, commission, manager_id and department_id and make sure that, the employee_id column does not contain any duplicate value at the time of insertion and the foreign key columns combined by department_id and manager_id columns contain only those unique combination values, which combinations are exists in the departments table.

Assume the structure of departments table below.

```
+-----------------+-------------+------+-----+---------+-----
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| DEPARTMENT_ID   | decimal(4,0) | NO   | PRI | 0       |       |
| DEPARTMENT_NAME | varchar(30)  | NO   |     | NULL    |       |
| MANAGER_ID      | decimal(6,0) | NO   | PRI | 0       |       |
| LOCATION_ID     | decimal(4,0) | YES  |     | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
```

Sample Solution:

```
CREATE TABLE IF NOT EXISTS employees (

EMPLOYEE_ID decimal(6,0) NOT NULL PRIMARY KEY,

FIRST_NAME varchar(20) DEFAULT NULL,

LAST_NAME varchar(25) NOT NULL,

EMAIL varchar(25) NOT NULL,

PHONE_NUMBER varchar(20) DEFAULT NULL,

HIRE_DATE date NOT NULL,

JOB_ID varchar(10) NOT NULL,

SALARY decimal(8,2) DEFAULT NULL,

COMMISSION_PCT decimal(2,2) DEFAULT NULL,

MANAGER_ID decimal(6,0) DEFAULT NULL,

DEPARTMENT_ID decimal(4,0) DEFAULT NULL,

FOREIGN KEY(DEPARTMENT_ID,MANAGER_ID)

REFERENCES  departments(DEPARTMENT_ID,MANAGER_ID)

)ENGINE=InnoDB;
```

Copy
Let execute the above code in MySQL 5.6 command prompt

Here is the structure of the table:

```
mysql> DESC employees;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| EMPLOYEE_ID    | decimal(6,0) | NO   | PRI | NULL    |       |
| FIRST_NAME     | varchar(20)  | YES  |     | NULL    |       |
| LAST_NAME      | varchar(25)  | NO   |     | NULL    |       |
| EMAIL          | varchar(25)  | NO   |     | NULL    |       |
| PHONE_NUMBER   | varchar(20)  | YES  |     | NULL    |       |
| HIRE_DATE      | date         | NO   |     | NULL    |       |
| JOB_ID         | varchar(10)  | NO   |     | NULL    |       |
| SALARY         | decimal(8,2) | YES  |     | NULL    |       |
| COMMISSION_PCT | decimal(2,2) | YES  |     | NULL    |       |
| MANAGER_ID     | decimal(6,0) | YES  |     | NULL    |       |
| DEPARTMENT_ID  | decimal(4,0) | YES  | MUL | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
11 rows in set (0.03 sec)
```

16. Write a SQL statement to create a table employees including columns employee_id, first_name, last_name, email, phone_number hire_date, job_id, salary, commission, manager_id and department_id and make sure that, the employee_id column does not contain any duplicate value at the time of insertion, and the foreign key column department_id, reference by the column department_id of departments table, can contain only those values which are exists in the departments table and another foreign key column job_id, referenced by the column job_id of jobs table, can contain only those values which are exists in the jobs table. The InnoDB Engine have been used to create the tables.

"A foreign key constraint is not required merely to join two tables. For storage engines other than InnoDB, it is possible when defining a column to use a REFERENCES tbl_name(col_name) clause, which has no actual effect, and serves only as a memo or comment to you that the column which you are currently defining is intended to refer to a column in another table." - Reference dev.mysql.com

Assume that the structure of two tables departments and jobs.

```
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| DEPARTMENT_ID   | decimal(4,0) | NO   | PRI | 0       |       |
| DEPARTMENT_NAME | varchar(30)  | NO   |     | NULL    |       |
| MANAGER_ID      | decimal(6,0) | YES  |     | NULL    |       |

| LOCATION_ID     | decimal(4,0) | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
```

```
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| JOB_ID     | varchar(10)  | NO   | PRI |         |       |
| JOB_TITLE  | varchar(35)  | NO   |     | NULL    |       |
| MIN_SALARY | decimal(6,0) | YES  |     | NULL    |       |
| MAX_SALARY | decimal(6,0) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
```

Sample Solution:

```
CREATE TABLE IF NOT EXISTS employees (

EMPLOYEE_ID decimal(6,0) NOT NULL PRIMARY KEY,

FIRST_NAME varchar(20) DEFAULT NULL,

LAST_NAME varchar(25) NOT NULL,

EMAIL varchar(25) NOT NULL,

PHONE_NUMBER varchar(20) DEFAULT NULL,

HIRE_DATE date NOT NULL,

JOB_ID varchar(10) NOT NULL,

SALARY decimal(8,2) DEFAULT NULL,

COMMISSION_PCT decimal(2,2) DEFAULT NULL,

MANAGER_ID decimal(6,0) DEFAULT NULL,

DEPARTMENT_ID decimal(4,0) DEFAULT NULL,

FOREIGN KEY(DEPARTMENT_ID)

REFERENCES  departments(DEPARTMENT_ID),

FOREIGN KEY(JOB_ID)

REFERENCES  jobs(JOB_ID)

)ENGINE=InnoDB;
```

Let execute the above code in MySQL 5.6 command prompt

Here is the structure of the table:

```
mysql> DESC employees;
+---------------+-------------+------+-----+---------+-------+
```

```
| Field         | Type         | Null | Key | Default | Extra |
+---------------+--------------+------+-----+---------+-------+
| EMPLOYEE_ID    | decimal(6,0) | NO   | PRI | NULL    |       |
| FIRST_NAME     | varchar(20)  | YES  |     | NULL    |       |
| LAST_NAME      | varchar(25)  | NO   |     | NULL    |       |
| EMAIL          | varchar(25)  | NO   |     | NULL    |       |
| PHONE_NUMBER   | varchar(20)  | YES  |     | NULL    |       |
| HIRE_DATE      | date         | NO   |     | NULL    |       |
| JOB_ID         | varchar(10)  | NO   |     | NULL    |       |
| SALARY         | decimal(8,2) | YES  |     | NULL    |       |
| COMMISSION_PCT | decimal(2,2) | YES  |     | NULL    |       |
| MANAGER_ID     | decimal(6,0) | YES  |     | NULL    |       |
| DEPARTMENT_ID  | decimal(4,0) | YES  | MUL | NULL    |       |
+---------------+--------------+------+-----+---------+-------+
11 rows in set (0.01 sec)
```

17. Write a SQL statement to create a table employees including columns employee_id, first_name, last_name, job_id, salary and make sure that, the employee_id column does not contain any duplicate value at the time of insertion, and the foreign key column job_id, referenced by the column job_id of jobs table, can contain only those values which are exists in the jobs table. The InnoDB Engine have been used to create the tables. The specialty of the statement is that, The ON UPDATE CASCADE action allows you to perform cross-table update and ON DELETE RESTRICT action reject the deletion. The default action is ON DELETE RESTRICT.

Assume that the structure of the table jobs and InnoDB Engine have been used to create the table jobs.

```
CREATE TABLE IF NOT EXISTS jobs (
JOB_ID integer NOT NULL UNIQUE PRIMARY KEY,
JOB_TITLE varchar(35) NOT NULL DEFAULT ' ',
MIN_SALARY decimal(6,0) DEFAULT 8000,
MAX_SALARY decimal(6,0) DEFAULT NULL
)ENGINE=InnoDB;
```

```
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| JOB_ID     | int(11)     | NO   | PRI | NULL    |       |
| JOB_TITLE  | varchar(35) | NO   |     |         |       |
| MIN_SALARY | decimal(6,0) | YES  |     | 8000    |       |
| MAX_SALARY | decimal(6,0) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
```

```
CREATE TABLE IF NOT EXISTS employees (

EMPLOYEE_ID decimal(6,0) NOT NULL PRIMARY KEY,

FIRST_NAME varchar(20) DEFAULT NULL,

LAST_NAME varchar(25) NOT NULL,
```

```
EMAIL varchar(25) NOT NULL,
PHONE_NUMBER varchar(20) DEFAULT NULL,
HIRE_DATE date NOT NULL,
JOB_ID varchar(10) NOT NULL,
SALARY decimal(8,2) DEFAULT NULL,
COMMISSION_PCT decimal(2,2) DEFAULT NULL,
MANAGER_ID decimal(6,0) DEFAULT NULL,
DEPARTMENT_ID decimal(4,0) DEFAULT NULL,
FOREIGN KEY(DEPARTMENT_ID)
REFERENCES  departments(DEPARTMENT_ID),
FOREIGN KEY(JOB_ID)
REFERENCES  jobs(JOB_ID)
)ENGINE=InnoDB;
```

Let execute the above code in MySQL 5.6 command prompt

Here is the structure of the table:

```
mysql> DESC employees;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| EMPLOYEE_ID   | decimal(6,0) | NO   | PRI | NULL    |       |
| FIRST_NAME    | varchar(20)  | YES  |     | NULL    |       |
| LAST_NAME     | varchar(25)  | NO   |     | NULL    |       |
| EMAIL         | varchar(25)  | NO   |     | NULL    |       |
| PHONE_NUMBER  | varchar(20)  | YES  |     | NULL    |       |
| HIRE_DATE     | date         | NO   |     | NULL    |       |
| JOB_ID        | varchar(10)  | NO   |     | NULL    |       |
| SALARY        | decimal(8,2) | YES  |     | NULL    |       |
| COMMISSION_PCT | decimal(2,2) | YES  |     | NULL    |       |
| MANAGER_ID    | decimal(6,0) | YES  |     | NULL    |       |
| DEPARTMENT_ID | decimal(4,0) | YES  | MUL | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
11 rows in set (0.01 sec)
```

18. Write a SQL statement to create a table employees including columns employee_id, first_name, last_name, job_id, salary and make sure that, the employee_id column does not contain any duplicate

value at the time of insertion, and the foreign key column job_id, referenced by the column job_id of jobs table, can contain only those values which are exists in the jobs table. The InnoDB Engine have been used to create the tables. The specialty of the statement is that, The ON DELETE CASCADE that lets you allow to delete records in the employees(child) table that refer to a record in the jobs(parent) table when the record in the parent table is deleted and the ON UPDATE RESTRICT actions reject any updates.

Assume that the structure of the table jobs and InnoDB Engine have been used to create the table jobs.

```
CREATE TABLE IF NOT EXISTS jobs (
JOB_ID integer NOT NULL UNIQUE PRIMARY KEY,
JOB_TITLE varchar(35) NOT NULL DEFAULT ' ',
MIN_SALARY decimal(6,0) DEFAULT 8000,
MAX_SALARY decimal(6,0) DEFAULT NULL
)ENGINE=InnoDB;
```

```
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| JOB_ID     | int(11)      | NO   | PRI | NULL    |       |
| JOB_TITLE  | varchar(35)  | NO   |     |         |       |
| MIN_SALARY | decimal(6,0) | YES  |     | 8000    |       |
| MAX_SALARY | decimal(6,0) | YES  |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
```

Sample Solution:

```
CREATE TABLE IF NOT EXISTS employees (

EMPLOYEE_ID decimal(6,0) NOT NULL PRIMARY KEY,

FIRST_NAME varchar(20) DEFAULT NULL,

LAST_NAME varchar(25) NOT NULL,

JOB_ID INTEGER NOT NULL,

SALARY decimal(8,2) DEFAULT NULL,

FOREIGN KEY(JOB_ID)

REFERENCES  jobs(JOB_ID)

ON DELETE CASCADE ON UPDATE RESTRICT

)ENGINE=InnoDB;
```

Copy
Let execute the above code in MySQL 5.6 command prompt

Here is the structure of the table:

```
mysql> DESC employees;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| EMPLOYEE_ID   | decimal(6,0) | NO   | PRI | NULL    |       |
| FIRST_NAME    | varchar(20)  | YES  |     | NULL    |       |
| LAST_NAME     | varchar(25)  | NO   |     | NULL    |       |
| EMAIL         | varchar(25)  | NO   |     | NULL    |       |
| PHONE_NUMBER  | varchar(20)  | YES  |     | NULL    |       |
| HIRE_DATE     | date         | NO   |     | NULL    |       |
| JOB_ID        | varchar(10)  | NO   |     | NULL    |       |
| SALARY        | decimal(8,2) | YES  |     | NULL    |       |
| COMMISSION_PCT | decimal(2,2) | YES  |     | NULL    |       |
| MANAGER_ID    | decimal(6,0) | YES  |     | NULL    |       |
| DEPARTMENT_ID | decimal(4,0) | YES  | MUL | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
11 rows in set (0.09 sec)
```

19. Write a SQL statement to create a table employees including columns employee_id, first_name, last_name, job_id, salary and make sure that, the employee_id column does not contain any duplicate value at the time of insertion, and the foreign key column job_id, referenced by the column job_id of jobs table, can contain only those values which are exists in the jobs table. The InnoDB Engine have been used to create the tables. The specialty of the statement is that, The ON DELETE SET NULL action will set the foreign key column values in the child table(employees) to NULL when the record in the parent table(jobs) is deleted, with a condition that the foreign key column in the child table must accept NULL values and the ON UPDATE SET NULL action resets the values in the rows in the child table(employees) to NULL values when the rows in the parent table(jobs) are updated.

Assume that the structure of two table jobs and InnoDB Engine have been used to create the table jobs.

```
CREATE TABLE IF NOT EXISTS jobs (
JOB_ID integer NOT NULL UNIQUE PRIMARY KEY,
JOB_TITLE varchar(35) NOT NULL DEFAULT ' ',
MIN_SALARY decimal(6,0) DEFAULT 8000,
MAX_SALARY decimal(6,0) DEFAULT NULL
)ENGINE=InnoDB;
```

```
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| JOB_ID     | int(11)     | NO   | PRI | NULL    |       |
| JOB_TITLE  | varchar(35) | NO   |     |         |       |
| MIN_SALARY | decimal(6,0) | YES  |     | 8000    |       |
| MAX_SALARY | decimal(6,0) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
```

Sample Solution:

```
CREATE TABLE IF NOT EXISTS employees (

EMPLOYEE_ID decimal(6,0) NOT NULL PRIMARY KEY,

FIRST_NAME varchar(20) DEFAULT NULL,

LAST_NAME varchar(25) NOT NULL,

JOB_ID INTEGER,

SALARY decimal(8,2) DEFAULT NULL,

FOREIGN KEY(JOB_ID)

REFERENCES  jobs(JOB_ID)

ON DELETE SET NULL

ON UPDATE SET NULL

)ENGINE=InnoDB;
```

Copy
Let execute the above code in MySQL 5.6 command prompt

Here is the structure of the table:

```
mysql> DESC employees;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| EMPLOYEE_ID    | decimal(6,0) | NO   | PRI | NULL    |       |
| FIRST_NAME     | varchar(20)  | YES  |     | NULL    |       |
| LAST_NAME      | varchar(25)  | NO   |     | NULL    |       |
| EMAIL          | varchar(25)  | NO   |     | NULL    |       |
| PHONE_NUMBER   | varchar(20)  | YES  |     | NULL    |       |
| HIRE_DATE      | date         | NO   |     | NULL    |       |
| JOB_ID         | varchar(10)  | NO   |     | NULL    |       |
| SALARY         | decimal(8,2) | YES  |     | NULL    |       |
| COMMISSION_PCT | decimal(2,2) | YES  |     | NULL    |       |
| MANAGER_ID     | decimal(6,0) | YES  |     | NULL    |       |
| DEPARTMENT_ID  | decimal(4,0) | YES  | MUL | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
11 rows in set (0.01 sec)
```

20. Write a SQL statement to create a table employees including columns employee_id, first_name, last_name, job_id, salary and make sure that, the employee_id column does not contain any duplicate value at the time of insertion, and the foreign key column job_id, referenced by the column job_id of jobs table, can contain only those values which are exists in the jobs table. The InnoDB Engine have

been used to create the tables. The specialty of the statement is that, The ON DELETE NO ACTION and the ON UPDATE NO ACTION actions will reject the deletion and any updates.

Assume that the structure of two table jobs and InnoDB Engine have been used to create the table jobs.

```
CREATE TABLE IF NOT EXISTS jobs (
JOB_ID integer NOT NULL UNIQUE PRIMARY KEY,
JOB_TITLE varchar(35) NOT NULL DEFAULT ' ',
MIN_SALARY decimal(6,0) DEFAULT 8000,
MAX_SALARY decimal(6,0) DEFAULT NULL
)ENGINE=InnoDB;
```

```
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| JOB_ID     | int(11)     | NO   | PRI | NULL    |       |
| JOB_TITLE  | varchar(35) | NO   |     |         |       |
| MIN_SALARY | decimal(6,0) | YES  |     | 8000    |       |
| MAX_SALARY | decimal(6,0) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
```

Sample Solution:

```
CREATE TABLE IF NOT EXISTS employees (

EMPLOYEE_ID decimal(6,0) NOT NULL PRIMARY KEY,

FIRST_NAME varchar(20) DEFAULT NULL,

LAST_NAME varchar(25) NOT NULL,

JOB_ID INTEGER NOT NULL,

SALARY decimal(8,2) DEFAULT NULL,

FOREIGN KEY(JOB_ID)

REFERENCES  jobs(JOB_ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

)ENGINE=InnoDB;
```

Copy
Let execute the above code in MySQL 5.6 command prompt

Here is the structure of the table:

```
mysql> DESC employees;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| EMPLOYEE_ID    | decimal(6,0) | NO   | PRI | NULL    |       |
| FIRST_NAME     | varchar(20)  | YES  |     | NULL    |       |
| LAST_NAME      | varchar(25)  | NO   |     | NULL    |       |
| EMAIL          | varchar(25)  | NO   |     | NULL    |       |
| PHONE_NUMBER   | varchar(20)  | YES  |     | NULL    |       |
| HIRE_DATE      | date         | NO   |     | NULL    |       |
| JOB_ID         | varchar(10)  | NO   |     | NULL    |       |
| SALARY         | decimal(8,2) | YES  |     | NULL    |       |
| COMMISSION_PCT | decimal(2,2) | YES  |     | NULL    |       |
| MANAGER_ID     | decimal(6,0) | YES  |     | NULL    |       |
| DEPARTMENT_ID  | decimal(4,0) | YES  | MUL | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
11 rows in set (0.01 sec)
```