MySQL DDL exercises

**Create table**

1. Create a table countries with columns country\_id, country\_name and region\_id

CREATE TABLE countries(

COUNTRY\_ID varchar(2),

COUNTRY\_NAME varchar(40),

REGION\_ID decimal(10,0)

);

**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**

**Create table**

2. Create a table countries with columns country\_id, country\_name and region\_id which already **exists**

CREATE TABLE IF NOT EXISTS countries(

COUNTRY\_ID varchar(2),

COUNTRY\_NAME varchar(40),

REGION\_ID decimal(10,0)

);

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

**Create table**

1. Create the structure of a table dup\_countries similar to countries

CREATE TABLE IF NOT EXISTS dup\_countries

LIKE countries;

**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**

**Create table**

1. Create a duplicate copy of countries table including structure and data by name dup\_countries

CREATE TABLE IF NOT EXISTS dup\_countries

AS SELECT \* FROM countries;

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

CREATE TABLE

1. SQL statement to create a table **countries** set a constraint NOT NULL

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

);

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

CREATE TABLE

1. 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**

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)

);

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

CREATE TABLE

1. 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.

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)

);

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

CREATE TABLE

1. 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 '--/--/----'.

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

);

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

CREATE TABLE

1. 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

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)

);

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

CREATE TABLE

1. Create a table named **jobs** including columns **job\_id**, **job\_title**, **min\_salary** and **max\_salary**, and make sure that:

- values is job\_id column will be unique

- the default value for job\_title is blank by default

- min\_salary is 8000 by default

- max\_salary is NULL will be entered automatically at the time of insertion if no value assigned for the specified columns

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

);

DESC jobs;

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

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

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

| JOB\_ID | varchar(10) | NO | PRI | | |

| JOB\_TITLE | varchar(35) | NO | | | |

| MIN\_SALARY | decimal(6,0) | YES | | 8000 | |

| MAX\_SALARY | decimal(6,0) | YES | | NULL | |

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

4 rows in set

CREATE TABLE

**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.

**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.

**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.

**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 exist 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 | |

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

**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 | |

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

**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](https://dev.mysql.com/doc/refman/5.0/en/example-foreign-keys.html)

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 | |

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

**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 | |

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

**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 | |

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

**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 | |

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

**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 | |

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