**1.** Write a SQL statement to create a simple table countries including columns country\_id,country\_name and region\_id.

create table countries (

country\_id integer,

country\_name varchar(20),

region\_id integer);

**2.** Write a SQL statement to create a simple table countries including columns country\_id,country\_name and region\_id which is already exists.

create table if not exists countries (

country\_id integer,

country\_name varchar(20),

region\_id integer);

**3.** Write a SQL statement to create the structure of a table dup\_countries similar to countries.

create table dup\_countries like countries;

**4.** Write a SQL statement to create a duplicate copy of countries table including structure and data by name dup\_countries.

create table dup\_countries as

select \*

from countries;

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

create table countries (

country\_id integer not NULL,

country\_name varchar(2) not NULL,

region\_id integer not NULL);

**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.

create table jobs (

job\_id varchar(20),

jop\_title varchar(40),

min\_salary decimal,

max\_salary decimal check (max\_salary <= 25000)

);

**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.

create table countries (

country\_id integer,

country\_name varchar(20) check (country\_name = "Italy" || country\_name ="India" (|| country\_name = "China"),

region\_id integer

);

**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 '--/--/----'

create table job\_histry (

employee\_id integer,

start\_date date,

end\_date date check (end\_date like "--/--/----"),

job\_id varchar(20),

department\_id varchar(20)

);

**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.

create table countries (

country\_id integer,

country\_name varchar(20),

region\_id integer,

unique(country\_id)

);

**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.

create table jobs (

job\_id varchar(20),

job\_title varchar(40) default " ",

min\_salary decimal default 8000,

max\_salary decinal default NULL

);

**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.

create table countries (

country\_id integer primary key,

country\_name varchar(20),

region\_id integer

);

**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.

create table countries (

country\_id integer unique auto\_increment primary key,

country\_name varchar(20),

regrion\_id integer

);

**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.

create table countries (

country\_id integer,

country\_name varchar(20),

region\_id integer,

primary key (country\_id, region\_id)

);

**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 | |

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

create table job\_history (

employee\_id varchar(10),

start\_date date,

end\_date date,

job\_id varchar(10),

department\_id varchar(10),

primary key (employee\_id),

foreign key (job\_id) references jobs(job\_id)

);

**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 | |

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

create table employees (

employee\_id varchar(10) primary key,

first\_name varchar(20),

last\_name varchar(20),

email varchar(20),

phone\_number varchar(20),

hire\_date date,

job\_id varchar(10),

salary decimal,

commission varchar(40),

manager\_id decimal(6, 0) not null,

department\_id decimal(4, 0) not null,

foreign key (department\_id, manager\_id) references departments(department\_id, manager\_id)

);

**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 | |

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

CREATE TABLE 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)

);

**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 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)

);

**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 | |

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

CREATE TABLE 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

);

**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 | |

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

CREATE TABLE 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

);

**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 | |

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

CREATE TABLE 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

);