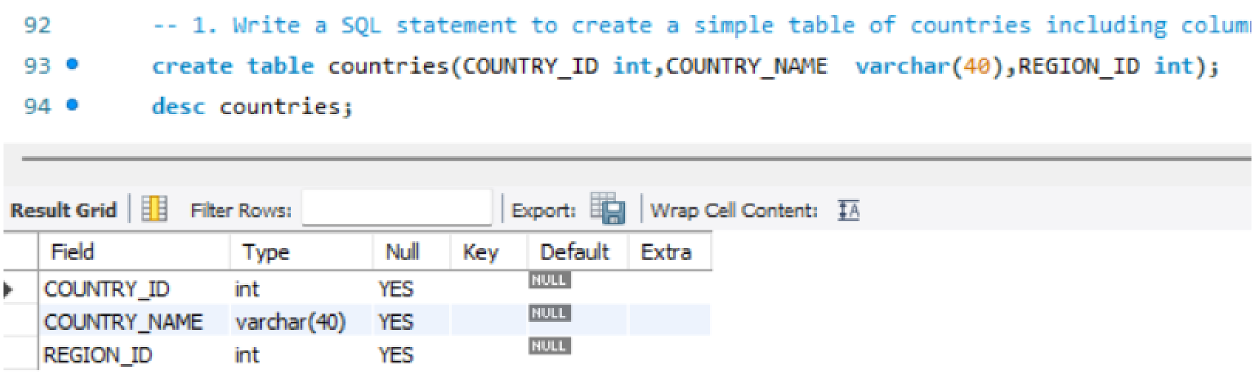
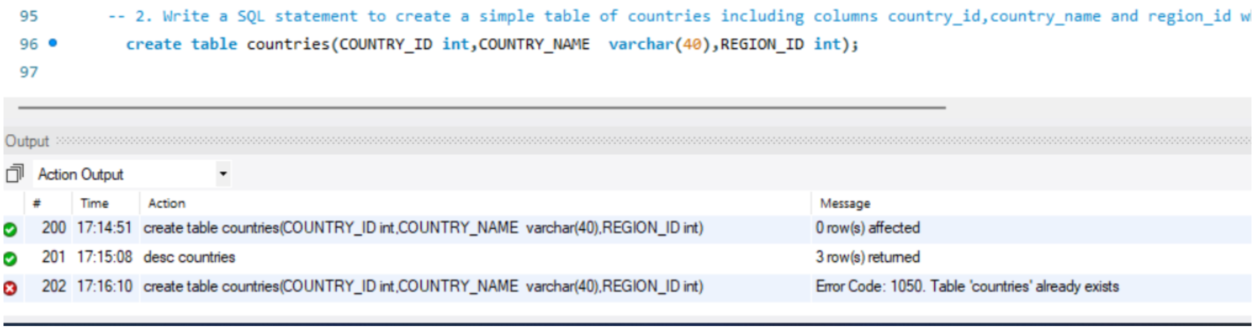
1. Write a SQL statement to create a simple table of countries including columns country_id,country_name and region_id.



```sql
    -- 1. Write a SQL statement to create a simple table of countries including colum
    create table countries(COUNTRY_ID int,COUNTRY_NAME  varchar(40),REGION_ID int);
    desc countries;
```

Field	Type	Null	Key	Default	Extra
COUNTRY_ID	int	YES		NULL	
COUNTRY_NAME	varchar(40)	YES		NULL	
REGION_ID	int	YES		NULL	

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

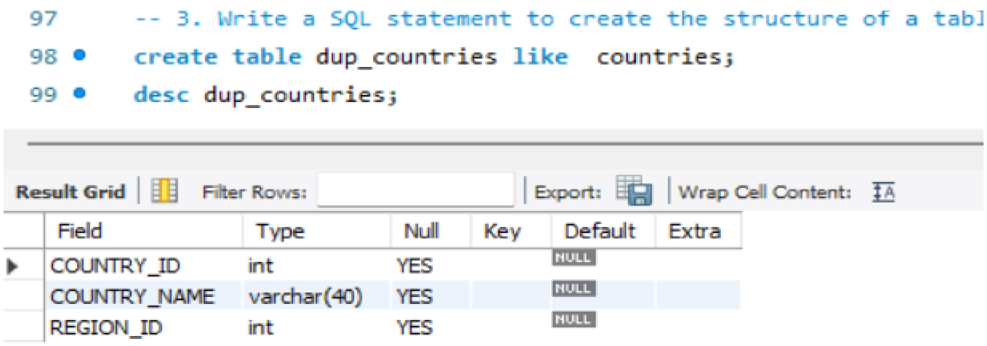


```sql
    -- 2. Write a SQL statement to create a simple table of countries including columns country_id,country_name and region_id w
      create table countries(COUNTRY_ID int,COUNTRY_NAME  varchar(40),REGION_ID int);

```

	#	Time	Action	Message
✓	200	17:14:51	create table countries(COUNTRY_ID int,COUNTRY_NAME varchar(40),REGION_ID int)	0 row(s) affected
✓	201	17:15:08	desc countries	3 row(s) returned
✗	202	17:16:10	create table countries(COUNTRY_ID int,COUNTRY_NAME varchar(40),REGION_ID int)	Error Code: 1050. Table 'countries' already exists

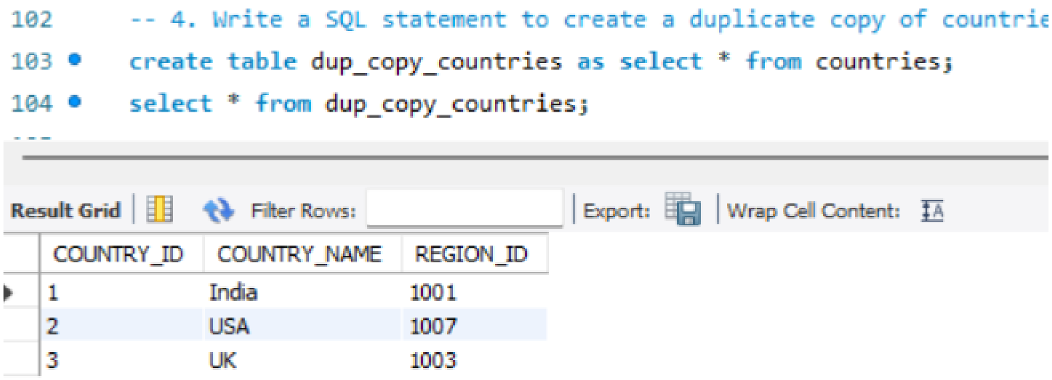
3. Write a SQL statement to create the structure of a table dup_countries similar to countries.



```sql
    -- 3. Write a SQL statement to create the structure of a tabl
    create table dup_countries like  countries;
    desc dup_countries;
```

Field	Type	Null	Key	Default	Extra
COUNTRY_ID	int	YES		NULL	
COUNTRY_NAME	varchar(40)	YES		NULL	
REGION_ID	int	YES		NULL	

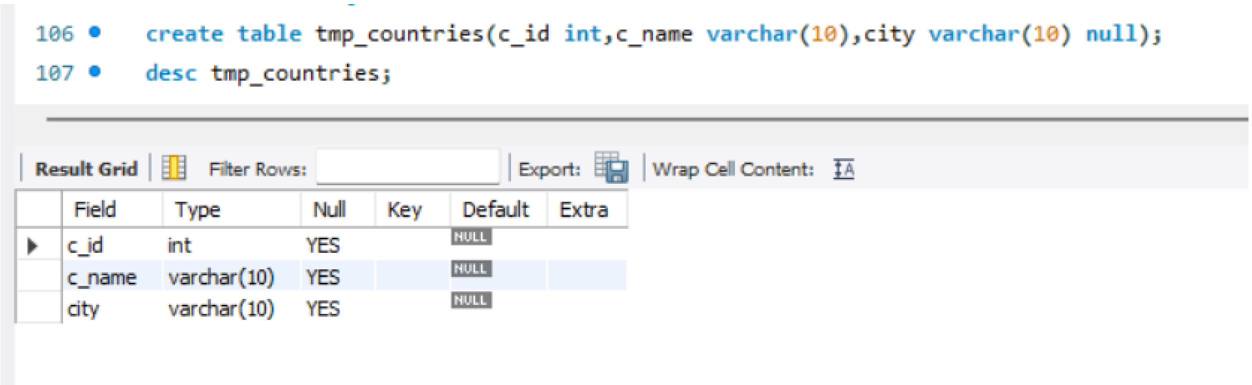
4. Write a SQL statement to create a duplicate copy of countries table including structure and data by name dup_countries.



```
-- 4. Write a SQL statement to create a duplicate copy of countrie
create table dup_copy_countries as select * from countries;
select * from dup_copy_countries;
```

COUNTRY_ID	COUNTRY_NAME	REGION_ID
1	India	1001
2	USA	1007
3	UK	1003

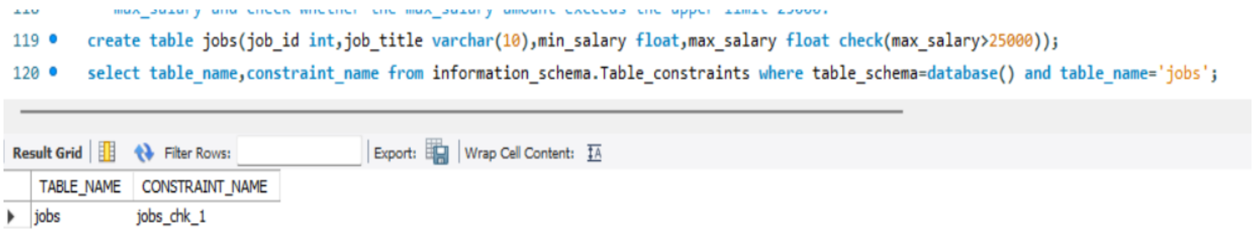
5. Write a SQL statement to create a table where countries set a constraint NULL.



```
create table tmp_countries(c_id int,c_name varchar(10),city varchar(10) null);
desc tmp_countries;
```

Field	Type	Null	Key	Default	Extra
c_id	int	YES		NULL	
c_name	varchar(10)	YES		NULL	
city	varchar(10)	YES		NULL	

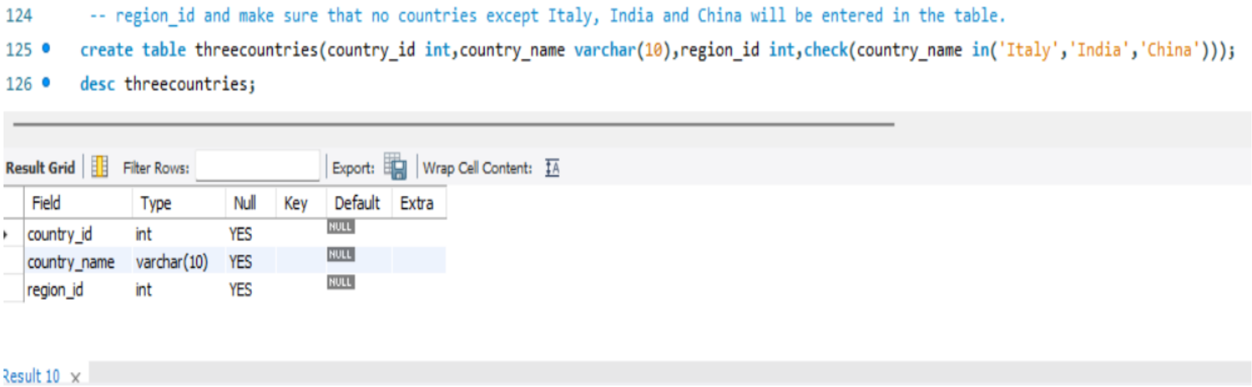
6. Write a SQL statement to create a table named jobs including columns job_id, job_title, min_salary, max_salary and check whether the max_salary amount exceeds the upper limit 25000.



```
max_salary and check whether the max_salary amount exceeds the upper limit 25000.
create table jobs(job_id int,job_title varchar(10),min_salary float,max_salary float check(max_salary>25000));
select table_name,constraint_name from information_schema.Table_constraints where table_schema=database() and table_name='jobs';
```

TABLE_NAME	CONSTRAINT_NAME
jobs	jobs_chk_1

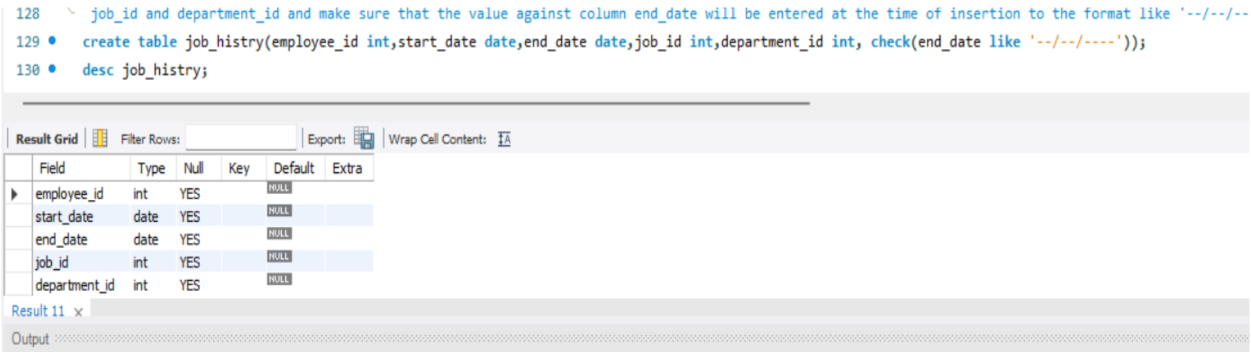
6. Write a SQL statement to create a table named countries including columns country_id, country_name and region_id and make sure that no countries except Italy, India and China will be entered in the table.



```
 -- region_id and make sure that no countries except Italy, India and China will be entered in the table.
create table threecountries(country_id int,country_name varchar(10),region_id int,check(country_name in('Italy','India','China')));
desc threecountries;
```

Field	Type	Null	Key	Default	Extra
country_id	int	YES		NULL	
country_name	varchar(10)	YES		NULL	
region_id	int	YES		NULL	

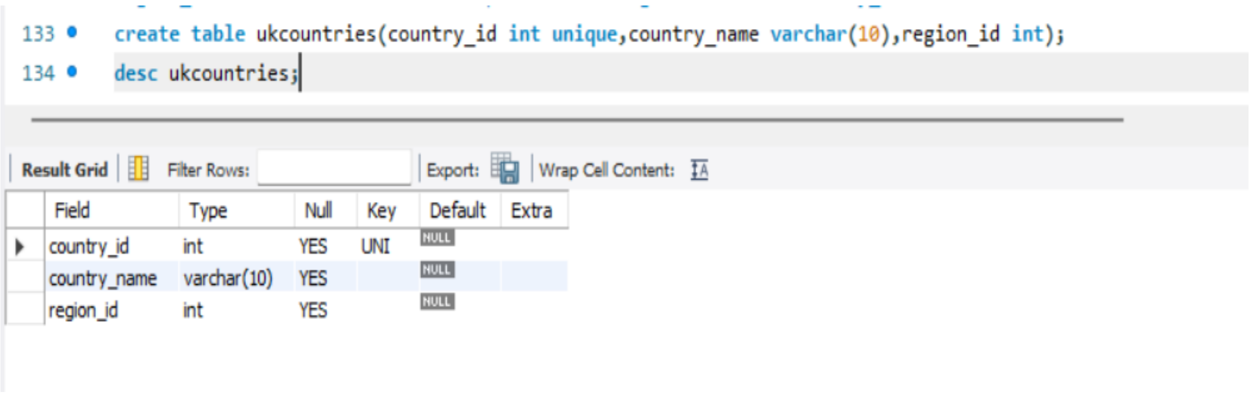
7. Write a SQL statement to create a table named job_histry including columns employee_id, start_date, end_date, job_id and department_id and make sure that the value against column end_date will be entered at the time of insertion to the format like '--/--/----'.



```
 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 int,start_date date,end_date date,job_id int,department_id int, check(end_date like '--/--/----'));
desc job_histry;
```

Field	Type	Null	Key	Default	Extra
employee_id	int	YES		NULL	
start_date	date	YES		NULL	
end_date	date	YES		NULL	
job_id	int	YES		NULL	
department_id	int	YES		NULL	

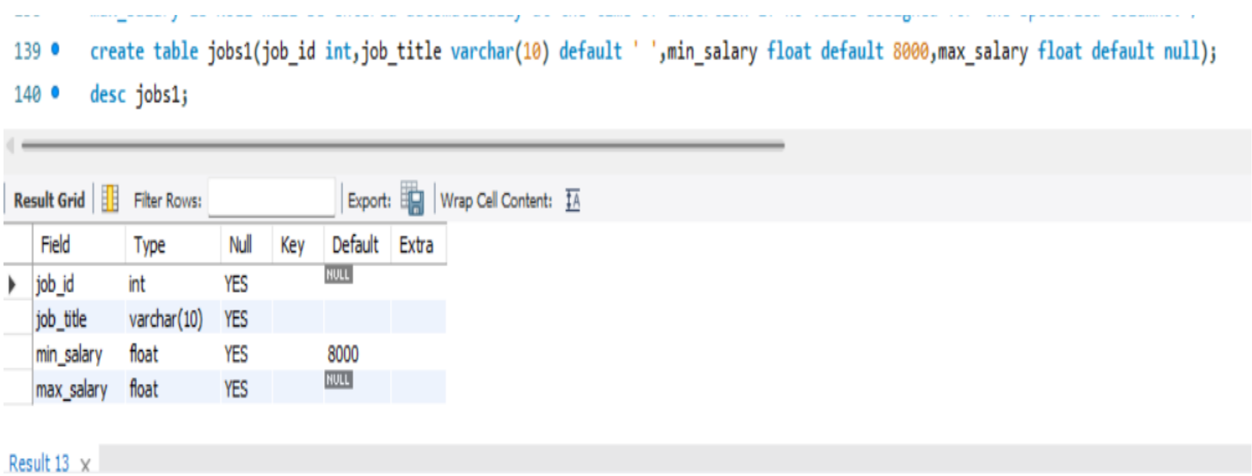
8. Write a SQL statement to create a table named countries including columns country_id,country_name and region_id and make sure that no duplicate data against column country_id will be allowed at the time of insertion.



```
create table ukcountries(country_id int unique,country_name varchar(10),region_id int);
desc ukcountries;
```

Field	Type	Null	Key	Default	Extra
country_id	int	YES	UNI	NULL	
country_name	varchar(10)	YES		NULL	
region_id	int	YES		NULL	

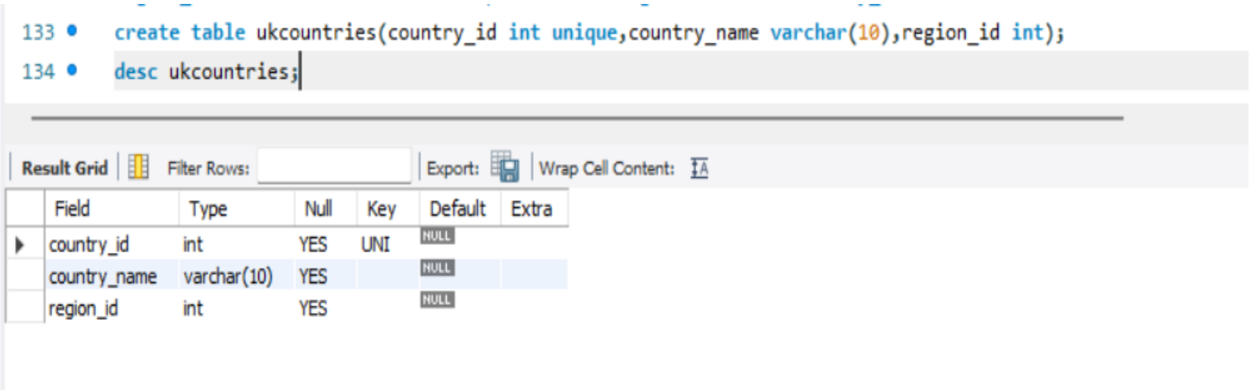
9. Write a SQL statement to create a table named jobs including columns job_id, job_title, min_salary and max_salary, and make sure that, the default value for job_title is blank and min_salary is 8000 and max_salary is NULL will be entered automatically at the time of insertion if no value assigned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



```
create table jobs1(job_id int,job_title varchar(10) default ' ',min_salary float default 8000,max_salary float default null);
desc jobs1;
```

Field	Type	Null	Key	Default	Extra
job_id	int	YES		NULL	
job_title	varchar(10)	YES			
min_salary	float	YES		8000	
max_salary	float	YES		NULL	

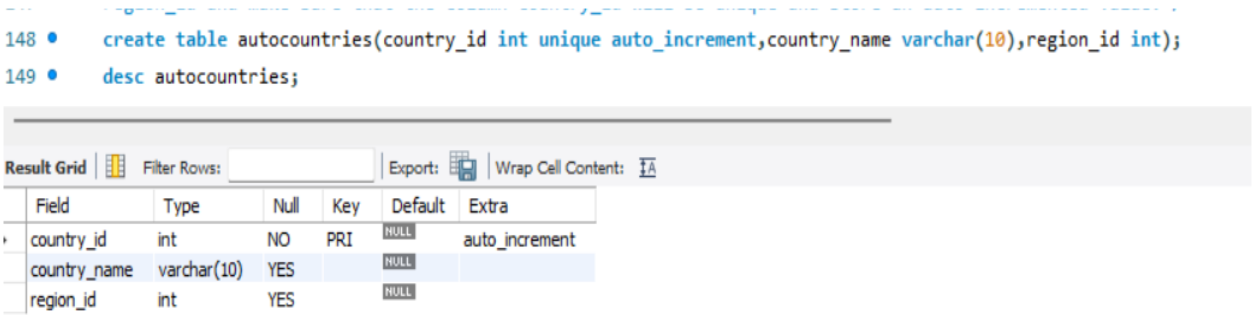
11. Write a SQL statement to create a table named countries including columns country_id, country_name and region_id and make sure that the country_id column will be a key field which will not contain any duplicate data at the time of insertion.



```
create table ukcountries(country_id int unique,country_name varchar(10),region_id int);
desc ukcountries;
```

Field	Type	Null	Key	Default	Extra
country_id	int	YES	UNI	NULL	
country_name	varchar(10)	YES		NULL	
region_id	int	YES		NULL	

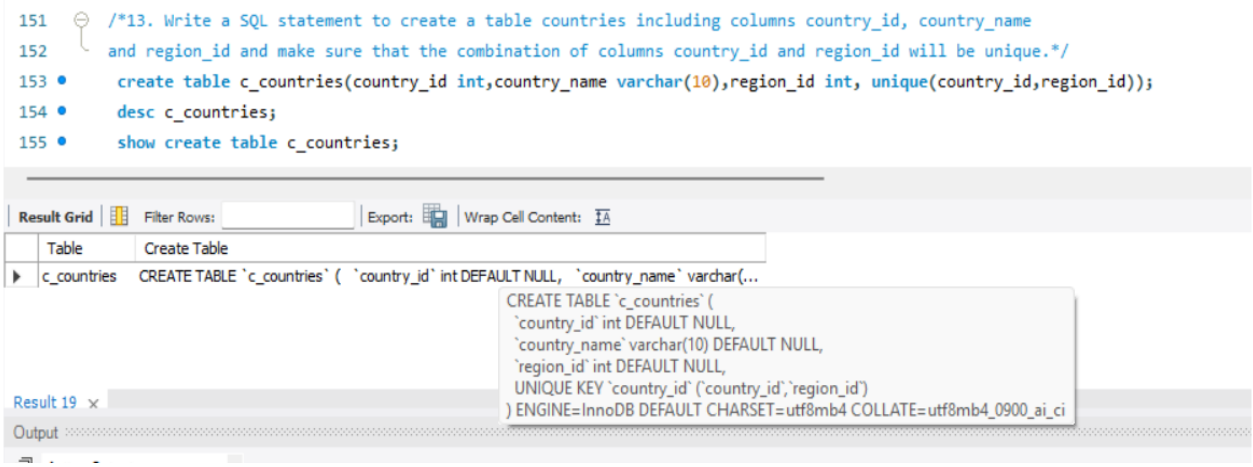
12. Write a SQL statement to create a table countries including columns country_id, country_name and region_id and make sure that the column country_id will be unique and store an auto incremented value.



```
create table autocountries(country_id int unique auto_increment,country_name varchar(10),region_id int);
desc autocountries;
```

Field	Type	Null	Key	Default	Extra
country_id	int	NO	PRI	NULL	auto_increment
country_name	varchar(10)	YES		NULL	
region_id	int	YES		NULL	

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.



```
/*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 c_countries(country_id int,country_name varchar(10),region_id int, unique(country_id,region_id));
 desc c_countries;
 show create table c_countries;
```

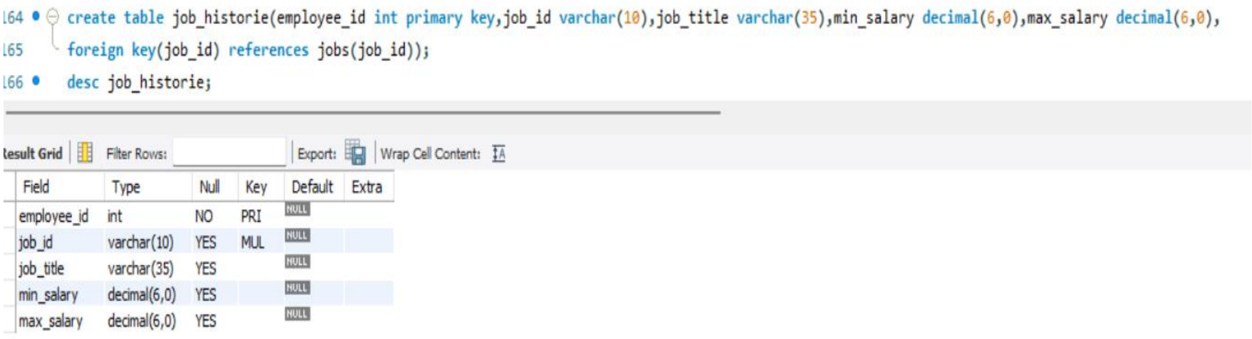
Table	Create Table
c_countries	CREATE TABLE `c_countries` (`country_id` int DEFAULT NULL, `country_name` varchar(...

```
CREATE TABLE `c_countries` (
 `country_id` int DEFAULT NULL,
 `country_name` varchar(10) DEFAULT NULL,
 `region_id` int DEFAULT NULL,
 UNIQUE KEY `country_id` (`country_id`,`region_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

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    |       |
+------------+--------------+------+-----+---------+-------+
```



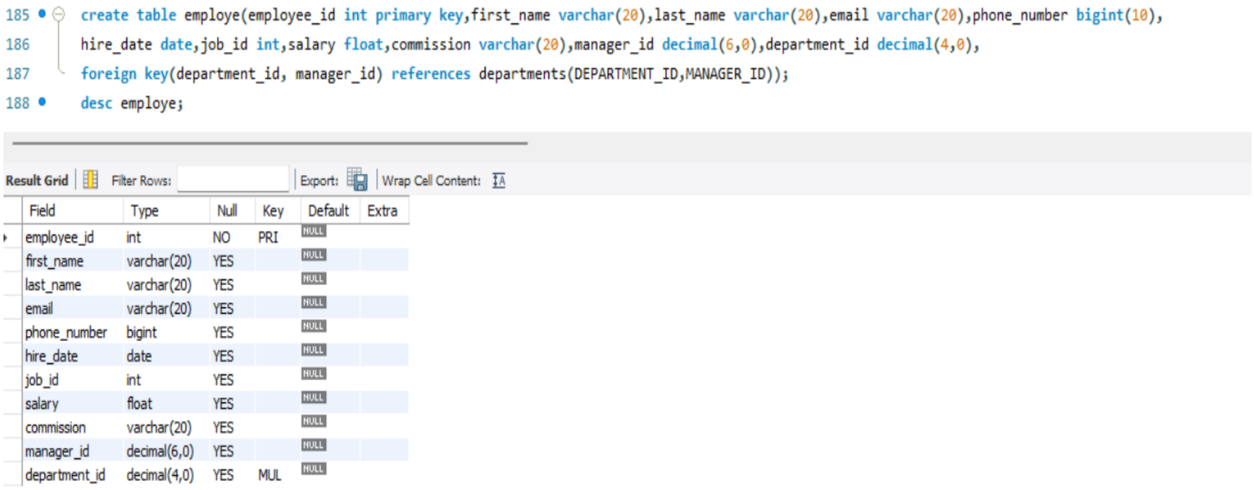
```
create table job_historie(employee_id int primary key,job_id varchar(10),job_title varchar(35),min_salary decimal(6,0),max_salary decimal(6,0),
foreign key(job_id) references jobs(job_id));
desc job_historie;
```

Field	Type	Null	Key	Default	Extra
employee_id	int	NO	PRI	NULL	
job_id	varchar(10)	YES	MUL	NULL	
job_title	varchar(35)	YES		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    |       |
+-----------------+--------------+------+-----+---------+-------+
```



```
create table employe(employee_id int primary key,first_name varchar(20),last_name varchar(20),email varchar(20),phone_number bigint(10),
hire_date date,job_id int,salary float,commission varchar(20),manager_id decimal(6,0),department_id decimal(4,0),
foreign key(department_id, manager_id) references departments(DEPARTMENT_ID,MANAGER_ID));
desc employe;
```

Field	Type	Null	Key	Default	Extra
employee_id	int	NO	PRI	NULL	
first_name	varchar(20)	YES		NULL	
last_name	varchar(20)	YES		NULL	
email	varchar(20)	YES		NULL	
phone_number	bigint	YES		NULL	
hire_date	date	YES		NULL	
job_id	int	YES		NULL	
salary	float	YES		NULL	
commission	varchar(20)	YES		NULL	
manager_id	decimal(6,0)	YES		NULL	
department_id	decimal(4,0)	YES	MUL	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.

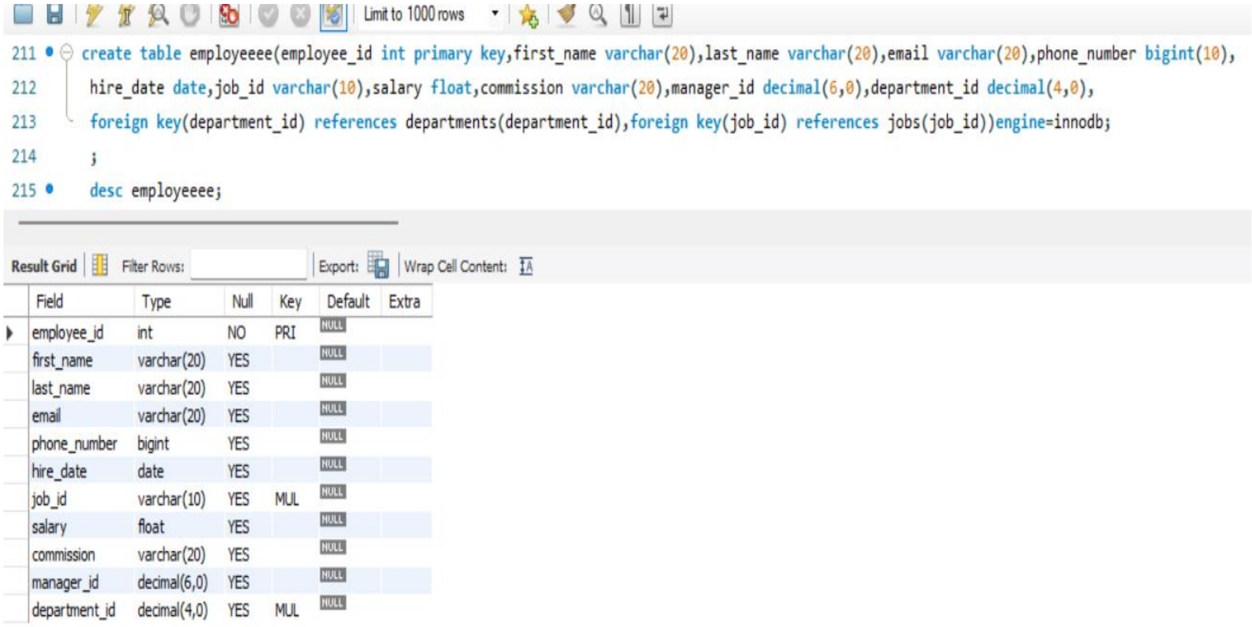
"A foreign key constraint is not required merely to join two tables. For storage engines other than InnoDB, it is possible when defining a column to use a REFERENCES tbl_name(col_name) clause, which has no actual effect, and serves only as a memo or comment to you that the column which you are currently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

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 employeeee(employee_id int primary key,first_name varchar(20),last_name varchar(20),email varchar(20),phone_number bigint(10),
 hire_date date,job_id varchar(10),salary float,commission varchar(20),manager_id decimal(6,0),department_id decimal(4,0),
 foreign key(department_id) references departments(department_id),foreign key(job_id) references jobs(job_id))engine=innodb;
 ;
 desc employeeee;
```

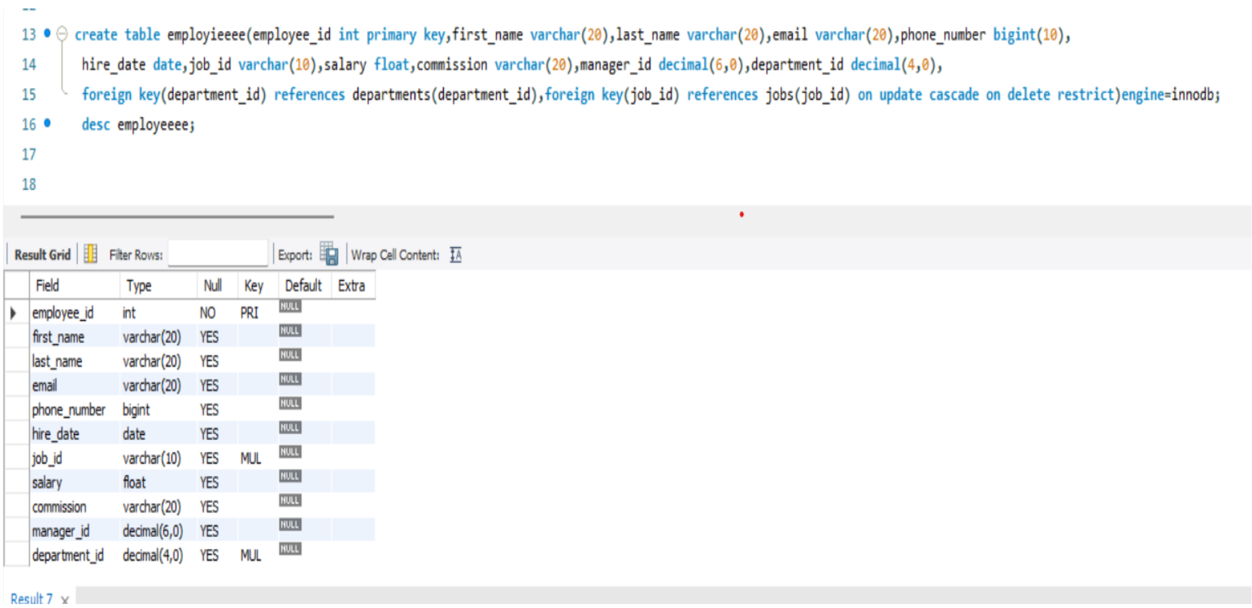
Field	Type	Null	Key	Default	Extra
employee_id	int	NO	PRI	NULL	
first_name	varchar(20)	YES		NULL	
last_name	varchar(20)	YES		NULL	
email	varchar(20)	YES		NULL	
phone_number	bigint	YES		NULL	
hire_date	date	YES		NULL	
job_id	varchar(10)	YES	MUL	NULL	
salary	float	YES		NULL	
commission	varchar(20)	YES		NULL	
manager_id	decimal(6,0)	YES		NULL	
department_id	decimal(4,0)	YES	MUL	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    |       |
+------------+--------------+------+-----+---------+-------+
```



```
--
create table employieeee(employee_id int primary key,first_name varchar(20),last_name varchar(20),email varchar(20),phone_number bigint(10),
 hire_date date,job_id varchar(10),salary float,commission varchar(20),manager_id decimal(6,0),department_id decimal(4,0),
 foreign key(department_id) references departments(department_id),foreign key(job_id) references jobs(job_id) on update cascade on delete restrict)engine=innodb;
 desc employeeee;
```

Field	Type	Null	Key	Default	Extra
employee_id	int	NO	PRI	NULL	
first_name	varchar(20)	YES		NULL	
last_name	varchar(20)	YES		NULL	
email	varchar(20)	YES		NULL	
phone_number	bigint	YES		NULL	
hire_date	date	YES		NULL	
job_id	varchar(10)	YES	MUL	NULL	
salary	float	YES		NULL	
commission	varchar(20)	YES		NULL	
manager_id	decimal(6,0)	YES		NULL	
department_id	decimal(4,0)	YES	MUL	NULL	

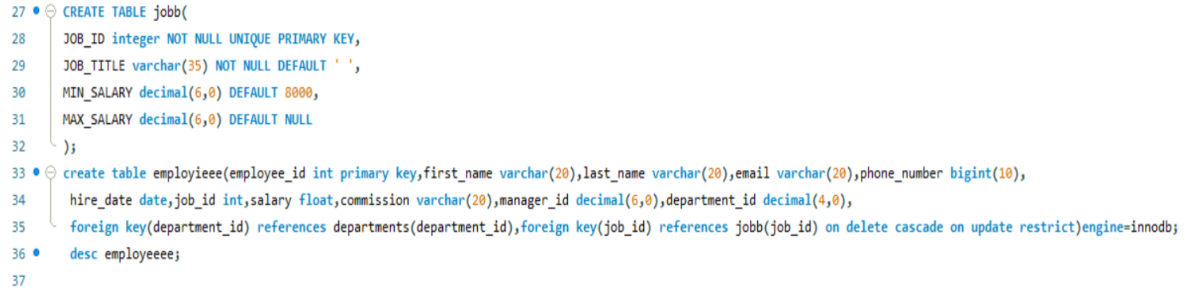
18. Write a SQL statement to create a table employees including columns employee_id, first_name, last_name, job_id, salary and make sure that, the employee_id column does not contain any duplicate value at the time of insertion, and the foreign key column job_id, referenced by the column job_id of jobs 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.

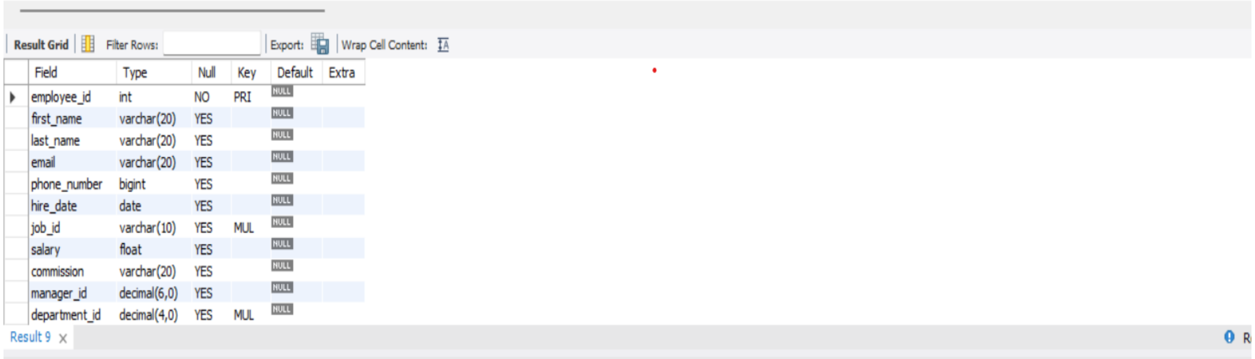
CREATE TABLE IF NOT EXISTS jobs (

```
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    |       |
+------------+--------------+------+-----+---------+-------+
```



```
CREATE TABLE jobb(
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
);
create table employieee(employee_id int primary key,first_name varchar(20),last_name varchar(20),email varchar(20),phone_number bigint(10),
 hire_date date,job_id int,salary float,commission varchar(20),manager_id decimal(6,0),department_id decimal(4,0),
 foreign key(department_id) references departments(department_id),foreign key(job_id) references jobb(job_id) on delete cascade on update restrict)engine=innodb;
 desc employeeee;
```



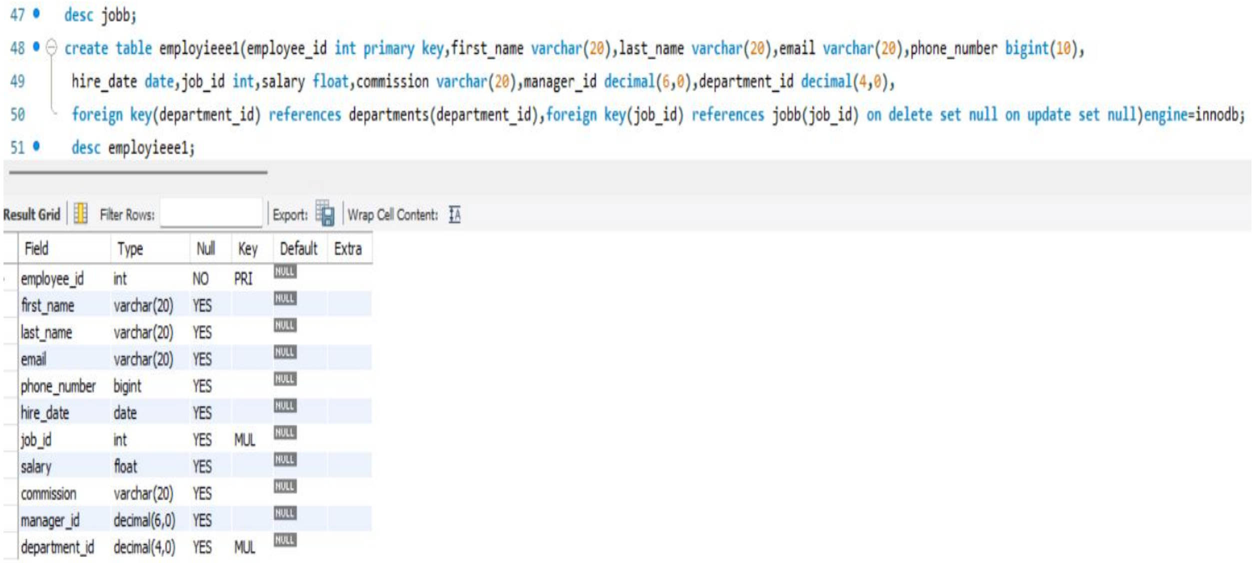
Field	Type	Null	Key	Default	Extra
employee_id	int	NO	PRI	NULL	
first_name	varchar(20)	YES		NULL	
last_name	varchar(20)	YES		NULL	
email	varchar(20)	YES		NULL	
phone_number	bigint	YES		NULL	
hire_date	date	YES		NULL	
job_id	varchar(10)	YES	MUL	NULL	
salary	float	YES		NULL	
commission	varchar(20)	YES		NULL	
manager_id	decimal(6,0)	YES		NULL	
department_id	decimal(4,0)	YES	MUL	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    |       |
+------------+--------------+------+-----+---------+-------+
```



```
desc jobb;
create table employieee1(employee_id int primary key,first_name varchar(20),last_name varchar(20),email varchar(20),phone_number bigint(10),
 hire_date date,job_id int,salary float,commission varchar(20),manager_id decimal(6,0),department_id decimal(4,0),
 foreign key(department_id) references departments(department_id),foreign key(job_id) references jobb(job_id) on delete set null on update set null)engine=innodb;
 desc employieee1;
```

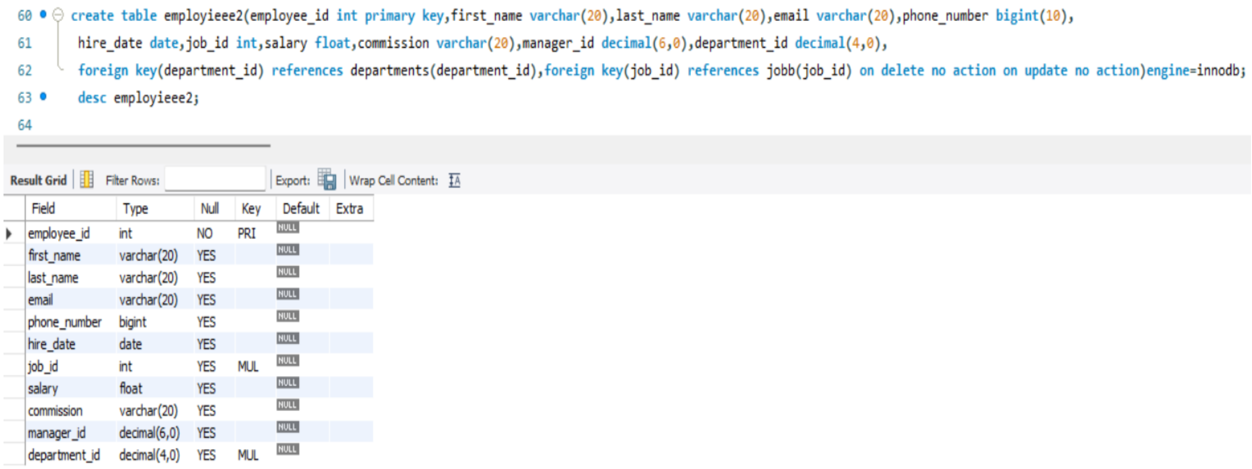
Field	Type	Null	Key	Default	Extra
employee_id	int	NO	PRI	NULL	
first_name	varchar(20)	YES		NULL	
last_name	varchar(20)	YES		NULL	
email	varchar(20)	YES		NULL	
phone_number	bigint	YES		NULL	
hire_date	date	YES		NULL	
job_id	int	YES	MUL	NULL	
salary	float	YES		NULL	
commission	varchar(20)	YES		NULL	
manager_id	decimal(6,0)	YES		NULL	
department_id	decimal(4,0)	YES	MUL	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 employieee2(employee_id int primary key,first_name varchar(20),last_name varchar(20),email varchar(20),phone_number bigint(10),
 hire_date date,job_id int,salary float,commission varchar(20),manager_id decimal(6,0),department_id decimal(4,0),
 foreign key(department_id) references departments(department_id),foreign key(job_id) references jobb(job_id) on delete no action on update no action)engine=innodb;
 desc employieee2;
```

Field	Type	Null	Key	Default	Extra
employee_id	int	NO	PRI	NULL	
first_name	varchar(20)	YES		NULL	
last_name	varchar(20)	YES		NULL	
email	varchar(20)	YES		NULL	
phone_number	bigint	YES		NULL	
hire_date	date	YES		NULL	
job_id	int	YES	MUL	NULL	
salary	float	YES		NULL	
commission	varchar(20)	YES		NULL	
manager_id	decimal(6,0)	YES		NULL	
department_id	decimal(4,0)	YES	MUL	NULL	