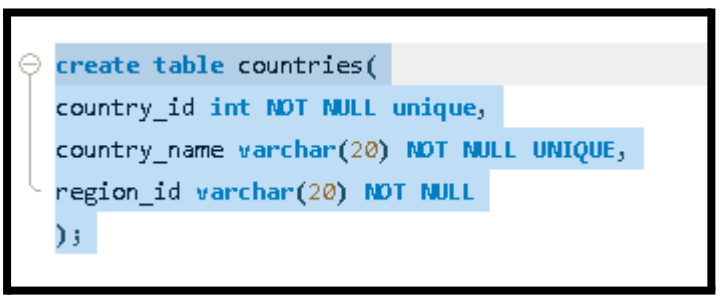
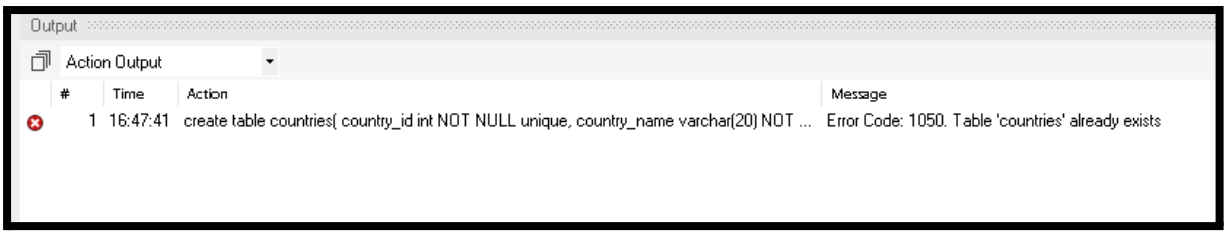
1. Write a SQL statement to create a simple table of countries including columns country_id,country_name and region_id.

```
create database Assignment1;

use Assignment1;


create table countries(

country_id int NOT NULL unique,

country_name varchar(20) NOT NULL UNIQUE,

region_id varchar(20) NOT NULL

);


insert into countries(country_id, country_name , region_id)

values

(101,"India","IN91"),

(102,"USA","US14"),

(103,"Russia","RS10"),

(104,"Canada","CN54"),

(105,"Japan","JP26");
```

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





Error as countries already exist.

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

```
create table dup_countries(
country_id int NOT NULL unique,
country_name varchar(20) NOT NULL UNIQUE,
region_id varchar(20) NOT NULL
);
```

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

To Create a copy of table:

```
CREATE TABLE new_table LIKE old_table;
INSERT INTO new_table SELECT * FROM old_table;
```

```
INSERT INTO dup_countries SELECT * FROM countries;
```

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

```
create table null_countries(
country_id int NULL,
country_name varchar(20) NULL,
region_id varchar(20) NULL
);
```

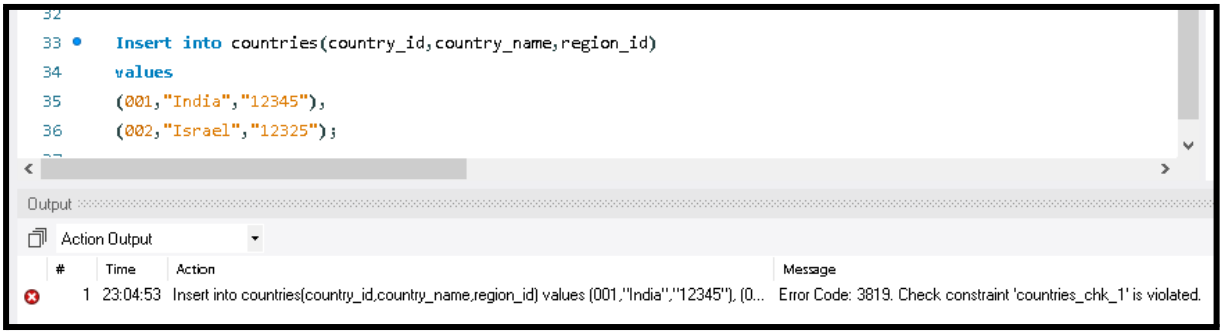
6. Write a SQL statement to create a table named jobs including columns job_id, job_title, min_salary, max_salary and check whether the max_salary amount exceeds the upper limit 25000.

```
create table jobs(
job_id int not null,
job_title varchar(20) not null,
min_salary int not null,
max_salary int not null 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 int NOT NULL ,
country_name varchar(20) NOT NULL check(country_name in
("India","China","Italy")),
region_id varchar(20) NOT NULL
```

);



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 int not null,

start_date date not null,

end_date date not null check(end_date like '--/--/----'),

job_id int not null,

department_id varchar(20) not null

);

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 int NOT NULL,

country_name varchar(20) NOT NULL UNIQUE,

region_id varchar(20) NOT NULL

);

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 int NOT NULL,
job_title varchar(20) NOT NULL default ' ',
min_salary int NOT NULL default 8000,
max_salary int default NULL
);
```

ALTER COMMAND

---

1. Write a SQL statement to rename the table countries to country_new.

```
alter table countries
rename country_new
```

2. Write a SQL statement to add a column region_id to the table locations.

```
alter table locations
add region_id varchar(20) not null;
```

3. Write a SQL statement to add a column ID as the first column of the table locations.

```
alter table locations
add ID int NOT NULL;
```

4. Write a SQL statement to add a column region_id after state_province to the table locations.

```
alter table locations
add region_id varchar(20) not null;
```

5. Write a SQL statement to change the data type of the column country_id to integer in the table locations

alter table locations

modify country_id int;

6. Write a SQL statement to drop the column city from the table locations.

Alter table locations

Drop column city;

7. Write a SQL statement to change the name of the column state_province to state, keeping the data type and size same.

Alter table locations

Change column state_province state varchar(20)

8. Write a SQL statement to add a primary key for the columns location_id in the locations table.

Alter table locations

Add primary key (location_id)

9. Write a SQL statement to add a primary key for a combination of columns location_id and country_id.

Alter table locations

Add primary key (location_id,country_id);

10. Write a SQL statement to drop the existing primary from the table locations on a combination of columns location_id and country_id.

Alter table locations

Drop Primary Key;

11. Write a SQL statement to add a foreign key on the job_id column of the job_history table referencing the primary key job_id of the jobs table.

Alter table job_history

Add Foreign Key (job_id) Reference jobs (job_id)

12. Write a SQL statement to add a foreign key constraint named fk_job_id on the job_id column of the job_history table referencing the primary key job_id of jobs table.

Alter table job_history

ADD CONSTRAINT Foreign Key (job_id) Reference jobs (job_id)

13. Write a SQL statement to drop the existing foreign key fk_job_id from the job_history table on the job_id column which is referencing the job_id of jobs table.

Alter table job_history

Drop Foreign Key fk_job_id ;

14. Write a SQL statement to add an index named indx_job_id on job_id column in the table job_history.

Alter table job_history

Add Index indx_job on job_history(indx_job_id)

15. Write a SQL statement to drop the index indx_job_id from job_history table.

Alter table job_history

Drop Index indx_job_id on job_history