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

postgres=# create database assignment

CREATE DATABASE

postgres=# \c assignment

You are now connected to database "assignment" as user "postgres".

assignment=# create table country(country\_id varchar(5),country\_name text,region\_id varchar(5));

CREATE TABLE

assignment=# \d country

Table "public.country"

Column | Type | Collation | Nullable | Default

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

country\_id | character varying(5) | | |

country\_name | text | | |

region\_id | character varying(5) |

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

assignment=# create table country(country\_id varchar(5),country\_name text,region\_id varchar(5));

ERROR: relation "country" already exists

assignment=# \d country

Table "public.country"

Column | Type | Collation | Nullable | Default

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

country\_id | character varying(5) | | |

country\_name | text | | |

region\_id | character varying(5) | | |

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

assignment=# create table dup\_country As(select \* from country);

SELECT 0

assignment=# \d

List of relations

Schema | Name | Type | Owner

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

public | country | table | postgres

public | dup\_country | table | postgres

(2 rows)

4. Write a SQL statement to create a table named countries, including 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.