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

postgres=# CREATE DATABASE assignment1;

CREATE DATABASE

postgres=# \c assignment1

You are now connected to database "assignment1" as user "postgres".

assignment1=# CREATE TABLE countries (country\_id integer,country\_name text,region\_id varchar(10));

CREATE TABLE

assignment1=# \d countries

Table "public.countries"

Column | Type | Collation | Nullable | Default

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

country\_id | integer | | |

country\_name | text | | |

region\_id | character varying(10) | | |

assignment1=# insert into countries (country\_id,country\_name,region\_id) values (1,'india','ind'),(2,'austrila','aust'),(3,'america','ame'),(3,'africa','afr'),(4,'shrilanka','shri');

INSERT 0 5

assignment1=# select \* from countries

assignment1-# ;

country\_id | country\_name | region\_id

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

1 | india | ind

2 | austrila | aust

3 | america | ame

3 | africa | afr

4 | shrilanka | shri

(5 rows)

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

assignment1=# create table if not exists countries(country\_id integer,country\_name text,region\_id varchar(10));

NOTICE: relation "countries" already exists, skipping

CREATE TABLE

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

assignment1=# create table dup\_countries as table countries with no data;

CREATE TABLE AS

assignment1=# select \* from dup\_countries;

country\_id | country\_name | region\_id

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

(0 rows)

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

assignment1=# create table dup\_countries2 as table countries;

SELECT 5

^

assignment1=# select \* from dup\_countries2;

country\_id | country\_name | region\_id

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

1 | india | ind

2 | austrila | aust

3 | america | ame

3 | africa | afr

4 | shrilanka | shri

(5 rows)

5. 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.

assignment1=# create table countries1 (country\_id integer,country\_name text,region\_id integer,primary key(country\_id));

CREATE TABLE

assignment1=# \d countries1;

Table "public.countries1"

Column | Type | Collation | Nullable | Default

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

country\_id | integer | | not null |

country\_name | text | | |

region\_id | integer | | |

Indexes:

"countries1\_pkey" PRIMARY KEY, btree (country\_id)

assignment1=# insert into countries1 values (1,'india',001),(2,'austrila',002),(3,'africa',004);

INSERT 0 3

assignment1=# select \* from countries1

assignment1-# ;

country\_id | country\_name | region\_id

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

1 | india | 1

2 | austrila | 2

3 | africa | 4

(3 rows)

assignment1=# insert into countries1 values (1,'india',001),(2,'austrila',002),(2,'africa',004);

ERROR: duplicate key value violates unique constraint "countries1\_pkey"

DETAIL: Key (country\_id)=(1) already exists.