Q.1 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.

```
postgres=# create database assgn2
postgres-# ;
CREATE DATABASE
postgres=# \c
You are now connected to database "postgres" as user "postgres".
postgres=# \c assgn2
You are now connected to database "assgn2" as user "postgres".
assgn2=# create table jobs (job_id serial primary key, job_title text,
min_salary integer, max_salary integer);
CREATE TABLE
assgn2=# \d jobs
                                Table "public.jobs"
   Column   |  Type   | Collation | Nullable |                Default
------------+---------+-----------+----------+-----------------------
--------------
 job_id     | integer |           | not null |
nextval('jobs_job_id_seq'::regclass)
 job_title  | text    |           |          |
 min_salary | integer |           |          |
 max_salary | integer |           |          |
Indexes:
    "jobs_pkey" PRIMARY KEY, btree (job_id)


assgn2=# alter table jobs alter column min_salary set default 8000
assgn2-# ;
ALTER TABLE
assgn2=# \d jobs
                                Table "public.jobs"
   Column   |  Type   | Collation | Nullable |                Default
------------+---------+-----------+----------+-----------------------
--------------
 job_id     | integer |           | not null |
nextval('jobs_job_id_seq'::regclass)
 job_title  | text    |           |          |
 min_salary | integer |           |          | 8000
 max_salary | integer |           |          |
Indexes:
    "jobs_pkey" PRIMARY KEY, btree (job_id)
```

Q.2 Write a SQL statement to create and insert a record into the table countries to ensure that the country_id and the region_id combination will be entered once in the table.

```
assgn2=# create table countries ("country_id" integer , "country_name"
text, "region_id" integer);
CREATE TABLE
assgn2=# alter table countries add primary key(country_id, region_id);
ALTER TABLE
assgn2=# \d countries
                Table "public.countries"
    Column    |  Type   | Collation | Nullable | Default
--------------+---------+-----------+----------+---------
 country_id   | integer |           | not null |
 country_name | text    |           |          |
 region_id    | integer |           | not null |
Indexes:
    "countries_pkey" PRIMARY KEY, btree (country_id, region_id)
```

Q.3 Write a SQL statement to create and insert records into the table countries to ensure that the country_id column will not contain any duplicate data and this will be automatically incremented and the column country_name will be filled up by 'N/A' if no value is assigned to that column.

```
assgn2=# create table countries (country_id serial not null
primary key, country_name text not null default 'NA', region_id
integer not null);
CREATE TABLE
assgn2=# \d copuntries
Did not find any relation named "copuntries".
assgn2=# \d countries
                                      Table "public.countries"
    Column    |  Type   | Collation | Nullable |
Default
--------------+---------+-----------+----------+---------------
------------------------------
 country_id   | integer |           | not null |
nextval('countries_country_id_seq'::regclass)
 country_name | text    |           | not null | 'NA'::text
```

```
 region_id    | integer |           | not null |
Indexes:
    "countries_pkey" PRIMARY KEY, btree (country_id)


assgn2=# insert into countries values(101,'India',55);
INSERT 0 1
assgn2=# select * from countries
assgn2-# ;
 country_id | country_name | region_id
------------+--------------+-----------
        101 | India        |        55
(1 row)


assgn2=# insert into countries (region_id) values(65);
INSERT 0 1
assgn2=# insert into countries (region_id, country_name)
values(75,'Germany');
INSERT 0 1
assgn2=# select * from countries
assgn2-# ;
 country_id | country_name | region_id
------------+--------------+-----------
        101 | India        |        55
          1 | NA           |        65
          2 | Germany      |        75
(3 rows)
```