Indian Institute of Information Technology Surat

Lab Report on

Advanced Database Management (CS 604) Practical

Submitted by

[RAHUL KUMAR SINGH] (UI21CS44)

Course Faculty

Mr. Rishi Sharma

Department of Computer Science and Engineering

Indian Institute of Information Technology Surat

Gujarat-394190, India

Jan-2024

Lab No: 7

Aim: To Implement basic relational methods and statements using PostgreSQL.

Description:

1. Write a PSQL statement to create a simple table countries including columns country_id,country_name and region_id.
2. Write a PSQL statement to create a simple table countries including columns country_id,country_name and region_id which already exist.
3. Write a PSQL statement to create the structure of a table dup_countries similar to countries.
4. Write a PSQL statement to create a duplicate copy of countries table including structure and data by name dup_countries.
5. Write a PSQL statement to create a table countries set a constraint NULL.
6. Write a PSQL statement to create a table named jobs including columns job_id, job_title, min_salary, max_salary and check whether the max_salary amount exceeding the upper limit 25000.
7. Write a PSQL 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.
8. Write a PSQL 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.
9. Write a PSQL 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.
10. Write a PSQL 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.
11. Write a PSQL 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.

Click me to see the solution

12. Write a PSQL 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.

Source Code & Output:

Task 1:

```
postgres=# CREATE TABLE countries (
postgres(#     country_id SERIAL PRIMARY KEY,
postgres(#     country_name VARCHAR(100) NOT NULL,
postgres(#     region_id INT
postgres(# );
CREATE TABLE
```

Task 2:

```
postgres=# CREATE TABLE new_countries AS
postgres-# SELECT country_id, country_name, region_id
postgres-# FROM countries;
SELECT 0
postgres=# select * from new_countries;
 country_id | country_name | region_id
------------+--------------+-----------
```

Task 3:

```
postgres=# CREATE TABLE dup_countries (LIKE countries INCLUDING ALL);
CREATE TABLE
postgres=# select * from dup_countries;
 country_id | country_name | region_id
------------+--------------+-----------
```

Task 4:

```
postgres=# CREATE TABLE dup_countries2 AS
postgres-# SELECT *
postgres-# FROM countries
postgres-# WITH NO DATA;
CREATE TABLE AS
postgres=# INSERT INTO dup_countries
postgres-# SELECT *
postgres-# FROM countries;
INSERT 0 0
postgres=# select * from dup_countries2;
 country_id | country_name | region_id
------------+--------------+-----------
```

Task 5:

```
postgres=# CREATE TABLE countries (
postgres(#     country_id SERIAL PRIMARY KEY,
postgres(#     country_name VARCHAR(100) NULL,
postgres(#     region_id INT
postgres(# );
CREATE TABLE
postgres=# select * from countries;
 country_id | country_name | region_id
------------+--------------+-----------
```

Task 6:

```
postgres=# CREATE TABLE jobs (
postgres(#     job_id SERIAL PRIMARY KEY,
postgres(#     job_title VARCHAR(100) NOT NULL,
postgres(#     min_salary NUMERIC(10, 2) DEFAULT 0.00,
postgres(#     max_salary NUMERIC(10, 2) CHECK (max_salary <= 25000)
postgres(# );
CREATE TABLE
```

```
postgres=# INSERT INTO jobs VALUES (3, 'A', 0, 122000);
ERROR:  new row for relation "jobs" violates check constraint "jobs_max_salary_check"
DETAIL:  Failing row contains (3, A, 0.00, 122000.00).
```

Task 7:

```
postgres=# CREATE TYPE allowed_countries AS ENUM ('Italy', 'India', 'China');
CREATE TYPE
postgres=# CREATE TABLE countries (
postgres(#     country_id SERIAL PRIMARY KEY,
postgres(#     country_name allowed_countries NOT NULL,
postgres(#     region_id INT
postgres(# );
CREATE TABLE
postgres=# INSERT INTO countries VALUES (1, 'USA', 1);
ERROR:  invalid input value for enum allowed_countries: "USA"
LINE 1: INSERT INTO countries VALUES (1, 'USA', 1);
                                         ^
postgres=# INSERT INTO countries VALUES (2, 'India', 1);
INSERT 0 1
```

Task 8:

```
postgres=# CREATE TABLE countries (
postgres(#     country_id SERIAL PRIMARY KEY,
postgres(#     country_name VARCHAR(100) NOT NULL,
postgres(#     region_id INT,
postgres(#     CONSTRAINT unique_country_id UNIQUE (country_id)
postgres(# );
CREATE TABLE
postgres=# INSERT INTO countries VALUES (1, 'India', 1);
INSERT 0 1
postgres=# INSERT INTO countries VALUES (1, 'China', 2);
ERROR:  duplicate key value violates unique constraint "unique_country_id"
DETAIL:  Key (country_id)=(1) already exists.
```

Task 9:

```
postgres=# CREATE TABLE jobs (
postgres(#     job_id SERIAL PRIMARY KEY,
postgres(#     job_title VARCHAR(100) DEFAULT '',
postgres(#     min_salary NUMERIC(10, 2) DEFAULT 8000.00,
postgres(#     max_salary NUMERIC(10, 2) DEFAULT NULL
postgres(# );
CREATE TABLE
postgres=# INSERT INTO jobs DEFAULT VALUES;
INSERT 0 1
postgres=# SELECT * FROM jobs;
 job_id | job_title | min_salary | max_salary
--------+-----------+------------+------------
      1 |           |    8000.00 |
(1 row)
```

Task 10:

```
postgres=# CREATE TABLE countries (
postgres(#     country_id SERIAL PRIMARY KEY,
postgres(#     country_name VARCHAR(100) NOT NULL,
postgres(#     region_id INT,
postgres(#     CONSTRAINT unique_country_id UNIQUE (country_id)
postgres(# );
CREATE TABLE
postgres=# INSERT INTO countries VALUES (1, 'India', 1);
INSERT 0 1
postgres=# INSERT INTO countries VALUES (1, 'China', 2);
ERROR:  duplicate key value violates unique constraint "unique_country_id"
DETAIL:  Key (country_id)=(1) already exists.
```

Task 11:

```
postgres=# CREATE TABLE jobs (
postgres(#     job_id SERIAL PRIMARY KEY,
postgres(#     job_title VARCHAR(100),
postgres(#     min_salary NUMERIC(10, 2),
postgres(#     max_salary NUMERIC(10, 2)
postgres(# );
CREATE TABLE
postgres=# INSERT INTO jobs
postgres-# (job_title, min_salary, max_salary)
postgres-# VALUES ('AI', 0, 10000);
INSERT 0 1
postgres=# SELECT * FROM jobs;
 job_id | job_title | min_salary | max_salary
--------+-----------+------------+------------
      1 | AI        |       0.00 |   10000.00
(1 row)
```

Task 12:

```
postgres=# CREATE TABLE countries (
postgres(#     country_id INT,
postgres(#     country_name VARCHAR(100) NOT NULL,
postgres(#     region_id INT NOT NULL,
postgres(#     CONSTRAINT unique_country_region UNIQUE (country_id, region_id)
postgres(# );
CREATE TABLE
postgres=# INSERT INTO countries VALUES (1, 'India', 1);
INSERT 0 1
postgres=# INSERT INTO countries VALUES (1, 'China', 2);
INSERT 0 1
postgres=# INSERT INTO countries VALUES (1, 'Italy', 2);
ERROR:  duplicate key value violates unique constraint "unique_country_region"
DETAIL:  Key (country_id, region_id)=(1, 2) already exists.
```

Conclusion:

- PSQL is PostgreSQL command-line interface for database management.
- Similarity to MySQL language in terms of both statements and structures.
- SERIAL in place of AUTO_INCREMENT
- Doesn't have an explicit statement for displaying a list of table (but could be done by using command **\dt**)