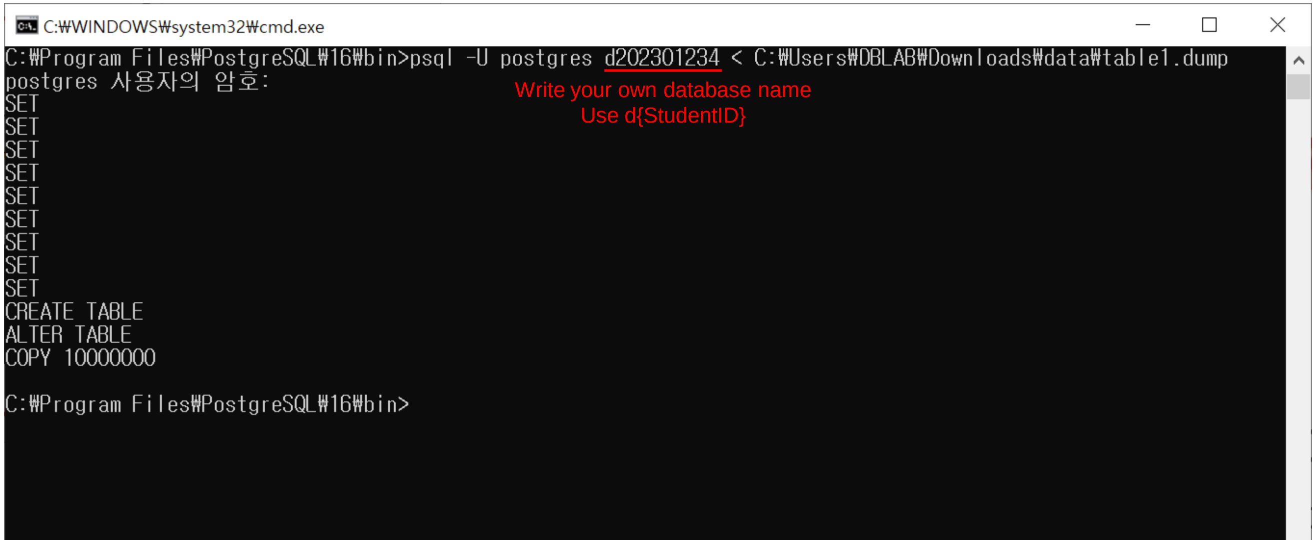
Chapter 14 - Lab

Indexing 1

Lab Setup (Windows)

- Download the “table1.dump” file from blackboard
- Open Command Prompt (cmd.exe) and type the following commands:
  1. cd C:\Program Files\PostgreSQL\16\bin
     - This is the default path. If you installed it somewhere else, go to that path.
  2. psql -U postgres d{*StudentID*} < [filepath]\table1.dump
     - For [filepath], type the path where you downloaded “table1.dump”.
  3. Type your own PostgreSQL password



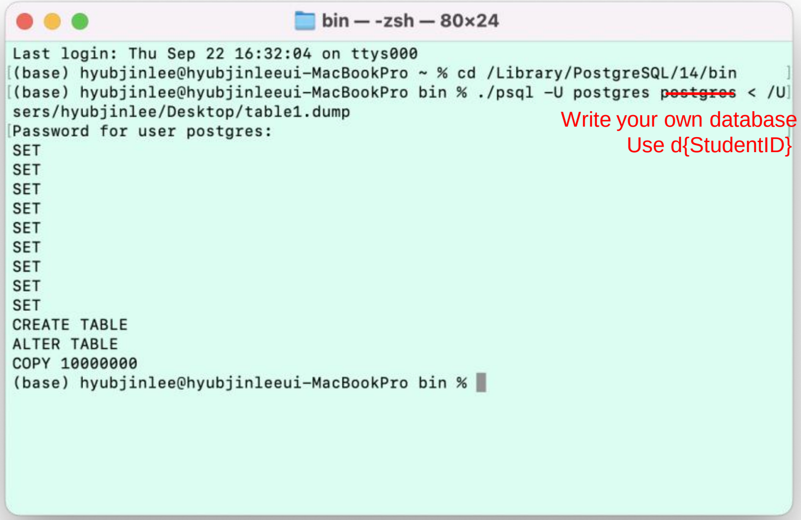
```
C:₩Program Files₩PostgreSQL₩16₩bin>psql -U postgres d202301234 < C:₩Users₩DBLAB₩Downloads₩data₩table1.dump
postgres 사용자의 암호:
SET
SET
SET
SET
SET
SET
SET
SET
SET
CREATE TABLE
ALTER TABLE
COPY 10000000

C:₩Program Files₩PostgreSQL₩16₩bin>
```

Write your own database name
Use d{StudentID}

Lab Setup (Max OS X)

- Download the “table1.dump” file from blackboard
- Open Terminal and type the following commands:
  1. cd /Library/PostgreSQL/16/bin
     - This is the default path. If you installed it somewhere else, go to that path.
  2. psql -U postgres d{*StudentID*} < [filepath]\table1.dump
     - For [filepath], type the path where you downloaded “table1.dump”.
  3. Type your own PostgreSQL password

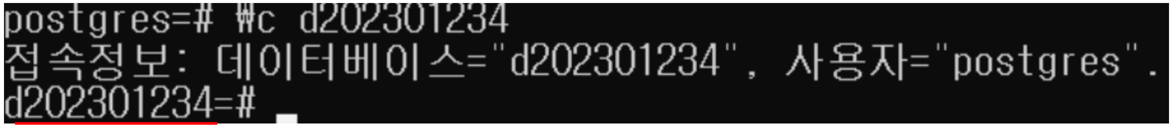


```
Last login: Thu Sep 22 16:32:04 on ttys000
(base) hyubjinlee@hyubjinleeui-MacBookPro ~ % cd /Library/PostgreSQL/14/bin
(base) hyubjinlee@hyubjinleeui-MacBookPro bin % ./psql -U postgres ~~postgres~~ < /U
sers/hyubjinlee/Desktop/table1.dump
Password for user postgres:
SET
SET
SET
SET
SET
SET
SET
SET
SET
CREATE TABLE
ALTER TABLE
COPY 10000000
(base) hyubjinlee@hyubjinleeui-MacBookPro bin % █
```

Write your own database name
Use d{StudentID}

Lab Setup

- Execute PostgreSQL SQL Shell (psql) and login your database
  - Server [localhost]: Press the enter key
  - Database [postgres]: Press the enter key
  - Port [5432]: Press the enter key
  - Username [postgres]: Press the enter key
  - Password for user postgres: Type your own password
  - \c d{StudentID}



```
postgres=# ₩c d202301234
접속정보: 데이터베이스="d202301234", 사용자="postgres".
d202301234=# 
```

Your answers must be displayed along with your student ID.

- Type on psql command line
  - SET enable_bitmapscan=false;
  - SET max_parallel_workers_per_gather=0;

Table Information

- “table1” has 10,000,000 rows
- “table1”’s schema is as follows:

Attribute	Data Type	Cardinality	Features
sorted	integer	2,000,000	Sorted
unsorted	integer	1,986,519	Unsorted
rndm	integer	100,000	Dummy field
dummy	character(40)	1	Dummy field

Exercise 1

- Create two indexes on “table1”
  - Indexed attributes are “sorted” and “unsorted”
  - Type “\h CREATE INDEX” for detailed index creation syntax

Exercise 2

- PostgreSQL supports following index-based query execution plans
  - Seq scan: All rows in a table are read sequentially
  - Index scan: Some (or all) rows in a table are read after traversing an index
  - Index only scan: Query is processed in an index, not accessing table data
- ‘EXPLAIN ANALYZE’ statement shows the query plan and execution time of the query
  - E.g., > EXPLAIN ANALYZE SELECT * FROM table1;

Exercise 2

a. Make (and execute) three queries each of which uses *seq scan*, *index scan*, and *index only scan* respectively

b. Make two queries that are expected to use indices on attributes “sorted” and “unsorted” respectively, then compare their execution times

c. Execute your queries to Exercise 2.b. after executing each of the following queries respectively, then compare their execution times
   - CLUSTER table1 USING idx_unsorted;
   - CLUSTER table1 USING idx_sorted;

d. Execute and compare the following two queries:
   - SELECT sorted, rndm FROM table1 WHERE sorted>1999231 AND rndm=1005;
   - SELECT sorted, rndm FROM table1 WHERE sorted<1999231 AND rndm=1005;
   - Explain why their query plans are different

Exercise 3

- Setup: Create a synthetic data set that has 5,000,000 rows
  - CREATE TABLE pool (val integer);
  - INSERT INTO pool(val) SELECT * FROM (SELECT generate_series(1,5000000)) as T;
  - Type on psql command line
    - SET enable_bitmapscan=false; \timing

- Consider two cases below. Which case will take a longer time?
  1. Inserting tuples in a table, and then creating index
  2. Creating index, and then inserting tuples in a table
- Compare the execution time t_1 and t_2
  - $t_1 = t_{1.insert} + t_{1.create\_index}$
    - Tuple insertion → Index creation
  - $t_2 = t_{2.create\_index} + t_{2.insert}$
    - Index creation → Tuple insertion

Exercise 3 - Hints

- Create an empty table named “table10” and “table20”
  - CREATE TABLE table10 (val integer);
  - CREATE TABLE table20 (val integer);
- Use “table10” to measure t_1 and “table20” to measure t_2
  - Inserting tuples into tables
    - Utilize the “pool” table for a synthetic dataset
    - Insert all the tuples of “pool” into “table10” and “table20”
      - E.g., > INSERT INTO table10 (SELECT * FROM pool);

Homework

- Complete today's practice exercises
- Write your queries and take screenshots of execution results
- Submit your report on blackboard
  - 10:29, October 1st, 2024
  - **Only PDF files** are accepted
  - **No late submission**

End of Lab