

Database creation and Data populating in PostgreSQL.

DTS207TC - DATABASE DEVELOPMENT AND DESIGN

COURSE LEADER: DR.ZHANG D

CO-TEACHER: DR. AFFAN YASIN

SCHOOL OF AI AND ADVANCED COMPUTING (AIAC), TAICANG, CHINA



Xi'an Jiaotong-Liverpool University

西交利物浦大學

Creating Database – Step 1

The image shows the pgAdmin 4 web interface. On the left, the 'Object Explorer' pane displays a tree structure: 'Servers (1)' > 'PostgreSQL 17' > 'Databases (2)'. The 'Databases (2)' folder is expanded, showing 'database' and 'postgres'. A right-click context menu is open over the 'database' folder, with options: 'Create', 'Refresh', and 'Disconnect from all databases'. The 'Create' option is highlighted, and a sub-menu is visible with the option 'Database...'. The main pane shows the 'Dashboard' tab with sub-tabs: 'Activity', 'State', 'Configuration', 'Logs', and 'System'. The 'Activity' sub-tab is active, displaying a table titled 'Tuples in' with columns for 'Inserts', 'Updates', 'Deletes', and 'Tuples'. The table has a y-axis from 0 to 100 and an x-axis with values 41, 21, and 1.

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

Servers (1)

PostgreSQL 17

Databases (2)

database

postgres

Login/Group Roles

Tablespaces

Create

Refresh

Disconnect from all databases

Database...

Dashboard

Properties

SQL

Statistics

Dependencies

Dependents

Processes

Activity

State

Configuration

Logs

System

Tuples in

Inserts

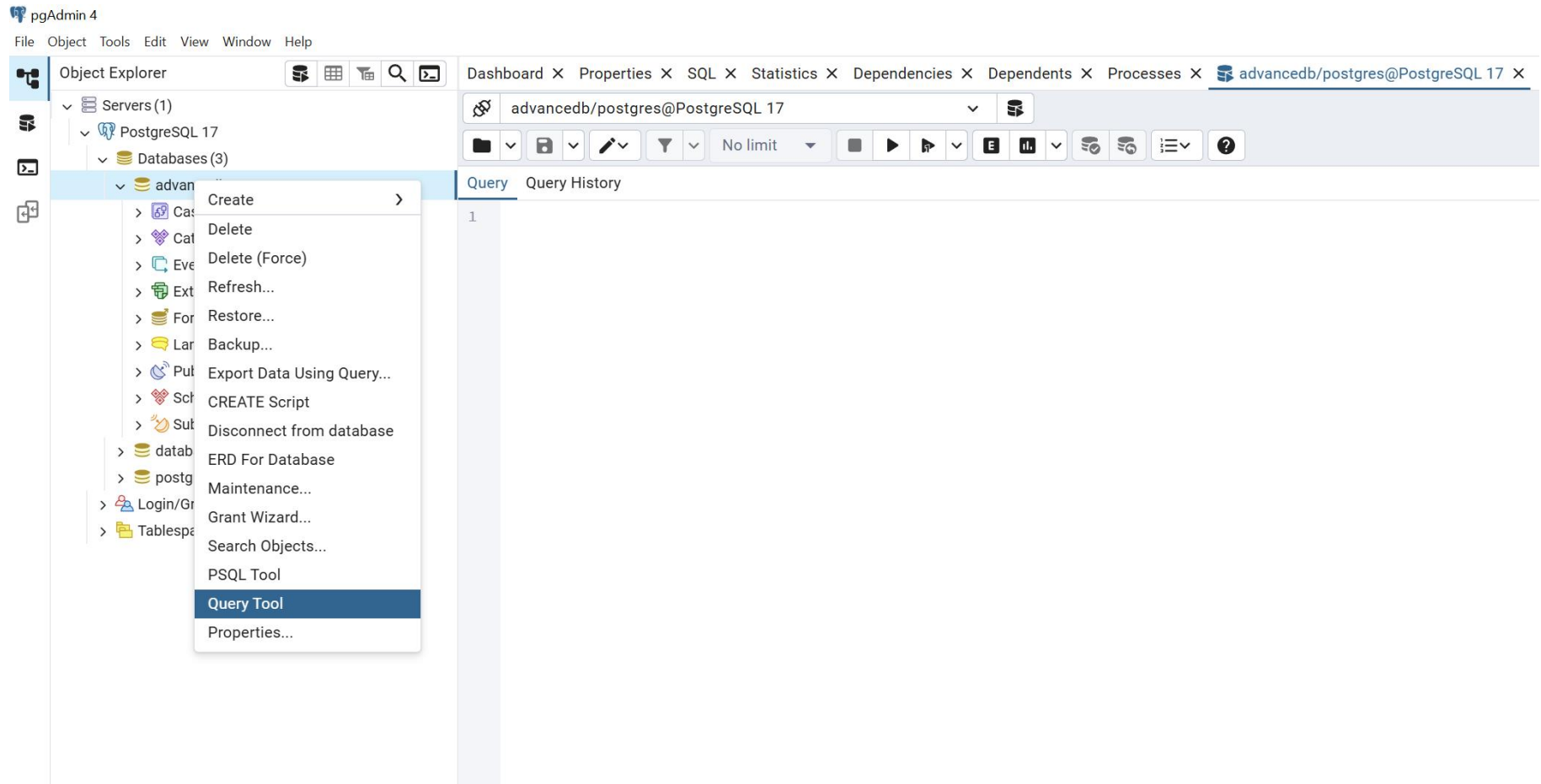
Updates

Deletes

Tuples

Inserts	Updates	Deletes	Tuples
100			
75			41
50			21
25			
0			1

Opening Query Tool – Step 2



Running the Given Queries – Step 3 to populate data

The screenshot displays the pgAdmin 4 web interface. On the left, the 'Object Explorer' pane shows the database structure: Servers (1) > PostgreSQL 17 > Databases (3) > advancedb > Schemas (1) > public > Tables (1) > students. The 'students' table is selected. The main pane shows the SQL editor with the following queries:

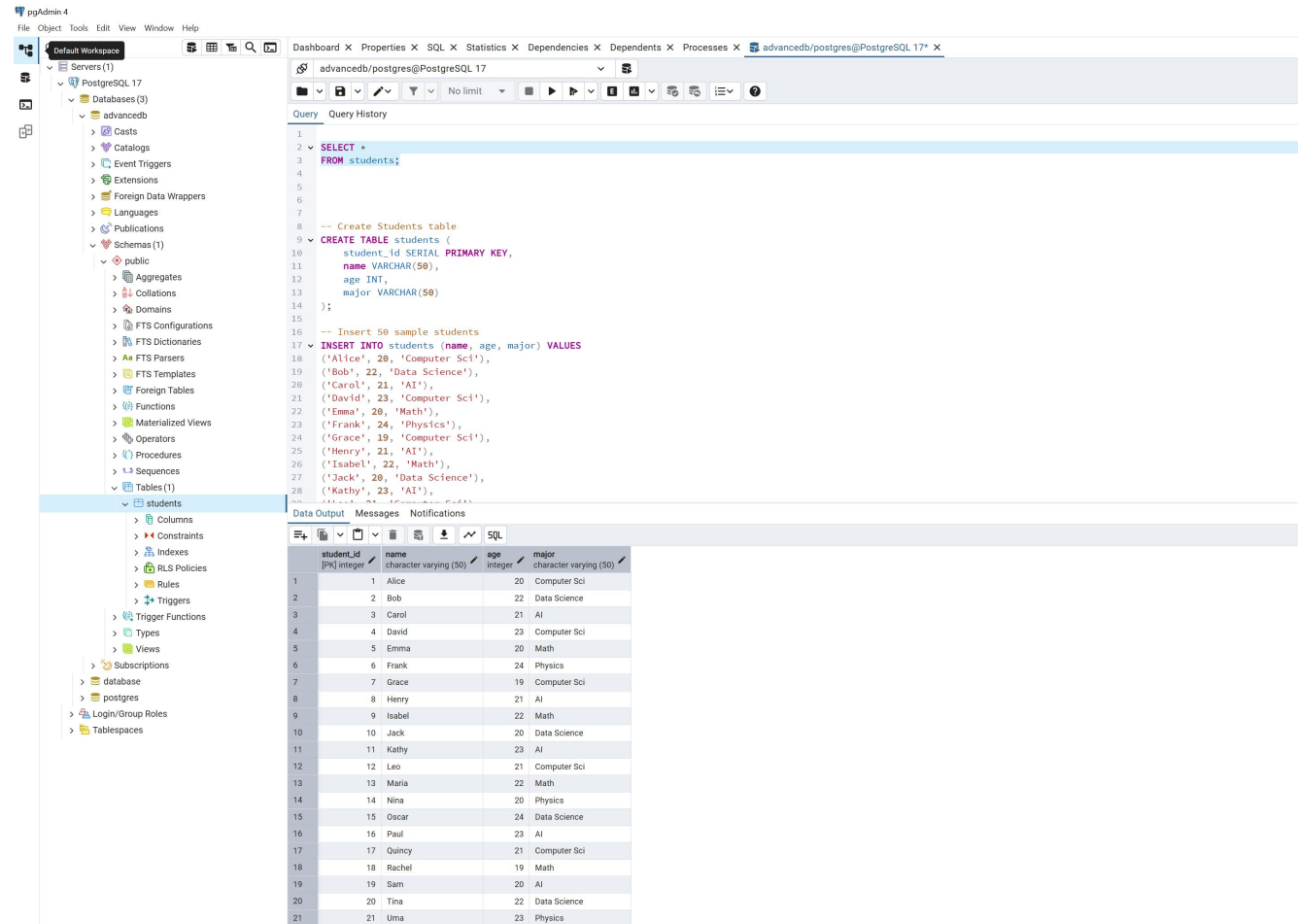
```
-- Create Students table
CREATE TABLE students (
  student_id SERIAL PRIMARY KEY,
  name VARCHAR(50),
  age INT,
  major VARCHAR(50)
);

-- Insert 50 sample students
INSERT INTO students (name, age, major) VALUES
('Alice', 20, 'Computer Sci'),
('Bob', 22, 'Data Science'),
('Carol', 21, 'AI'),
('David', 23, 'Computer Sci'),
('Emma', 20, 'Math'),
('Frank', 24, 'Physics'),
('Grace', 19, 'Computer Sci'),
('Henry', 21, 'AI'),
('Isabel', 22, 'Math'),
('Jack', 20, 'Data Science'),
('Kathy', 23, 'AI'),
('Leo', 21, 'Computer Sci'),
('Maria', 22, 'Math'),
('Nina', 20, 'Physics'),
('Oscar', 24, 'Data Science'),
('Paul', 23, 'AI'),
('Quincy', 21, 'Computer Sci'),
('Rachel', 19, 'Math');
```

Below the SQL editor, the 'Data Output' tab shows the execution results:

```
INSERT 0 49
Query returned successfully in 36 msec.
```

Reconfirm data is successfully imported – Step 4



The screenshot displays the pgAdmin 4 interface. On the left, the 'Default Workspace' pane shows the database structure: 'PostgreSQL 17' > 'Databases (3)' > 'advancedb' > 'Schemas (1)' > 'public' > 'Tables (1)' > 'students'. The 'students' table is selected, showing its columns: 'student_id' (integer, PK), 'name' (character varying (50)), 'age' (integer), and 'major' (character varying (50)).

The central pane shows the SQL editor with the following query:

```
1 SELECT *
2 FROM students;
3
4
5
6
7
8 -- Create Students Table
9 CREATE TABLE students (
10     student_id SERIAL PRIMARY KEY,
11     name VARCHAR(50),
12     age INT,
13     major VARCHAR(50)
14 );
15
16 -- Insert 50 sample students
17 INSERT INTO students (name, age, major) VALUES
18 ('Alice', 20, 'Computer Sci'),
19 ('Bob', 22, 'Data Science'),
20 ('Carol', 21, 'AI'),
21 ('David', 23, 'Computer Sci'),
22 ('Emma', 20, 'Math'),
23 ('Frank', 24, 'Physics'),
24 ('Grace', 19, 'Computer Sci'),
25 ('Henry', 21, 'AI'),
26 ('Isabel', 22, 'Math'),
27 ('Jack', 20, 'Data Science'),
28 ('Kathy', 23, 'AI'),
29 ('Leo', 21, 'Computer Sci'),
30 ('Maria', 22, 'Math'),
31 ('Nina', 20, 'Physics'),
32 ('Oscar', 24, 'Data Science'),
33 ('Paul', 23, 'AI'),
34 ('Quincy', 21, 'Computer Sci'),
35 ('Rachel', 19, 'Math'),
36 ('Sam', 20, 'AI'),
37 ('Tina', 22, 'Data Science'),
38 ('Uma', 23, 'Physics');
```

The bottom pane shows the 'Data Output' tab with the following data:

student_id	name	age	major
1	Alice	20	Computer Sci
2	Bob	22	Data Science
3	Carol	21	AI
4	David	23	Computer Sci
5	Emma	20	Math
6	Frank	24	Physics
7	Grace	19	Computer Sci
8	Henry	21	AI
9	Isabel	22	Math
10	Jack	20	Data Science
11	Kathy	23	AI
12	Leo	21	Computer Sci
13	Maria	22	Math
14	Nina	20	Physics
15	Oscar	24	Data Science
16	Paul	23	AI
17	Quincy	21	Computer Sci
18	Rachel	19	Math
19	Sam	20	AI
20	Tina	22	Data Science
21	Uma	23	Physics

Queries to run

All the SQL queries required to populate data for this lab are provided in a separate **.txt** file. Please refer to the **Learning Mall course folder** to access them.