School of Computer Science

COMP9120 Relational Database Systems

Tutorial Week 3: The Relational Model

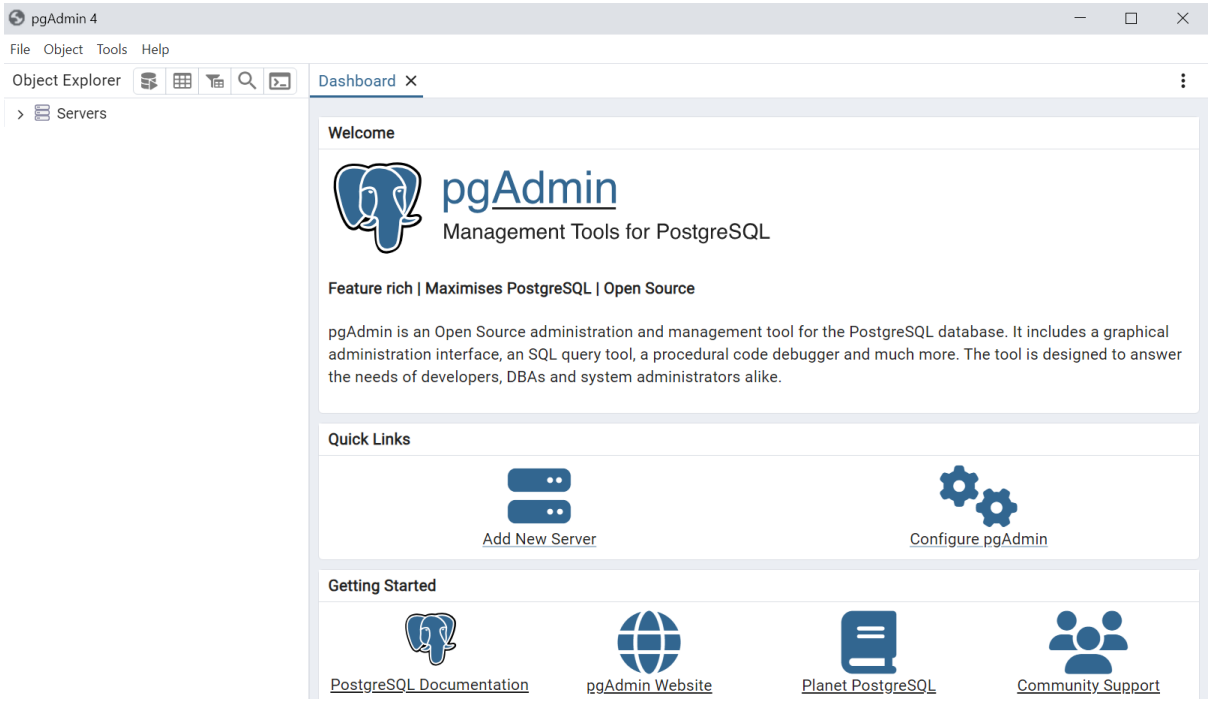
Exercise 1. Set up Remote PostgreSQL Account

You can remotely access the PostgreSQL server that is maintained by School of CS via pgAdmin. Note that, if you want to access PostgreSQL server from outside the university network, you will need to first connect securely to the university network using VPN – refer to the following link:

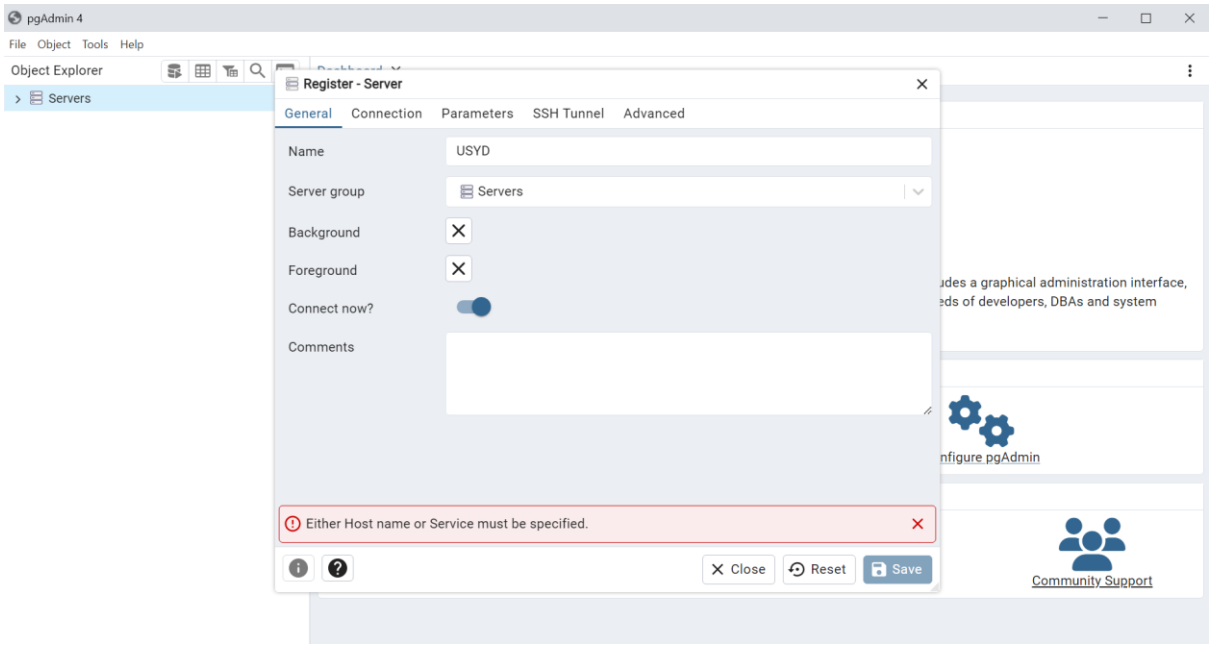
https://sydneyuni.service-now.com/sm?id=kb_article_view&sys_kb_id=c0bf9bd6db41b3485beaf9b7f49619a2&sysparm_tsqueryId=f90a62cbdb937f44c8a5773c349619f2&sysparm_rank=7

[Please check the DB connection information sent to your email]

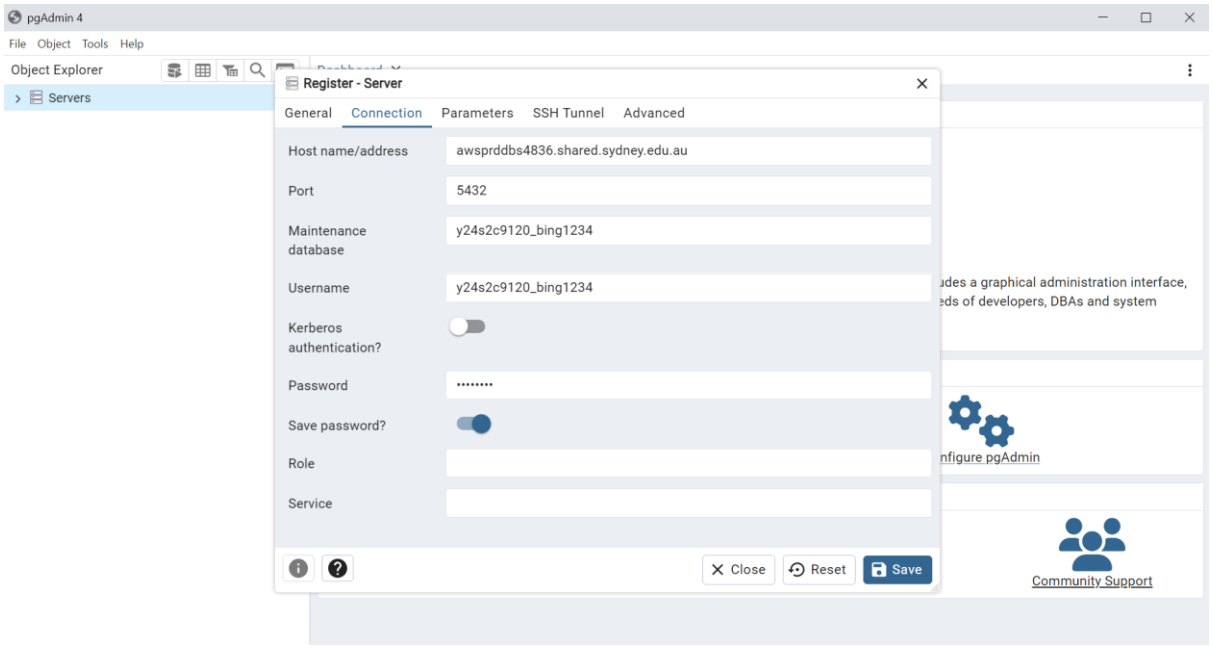
1. Start pgAdmin 4. You should see a screen like the one shown in the next picture. Type in the master password if it asks you to set the master password for pgAdmin and click OK.



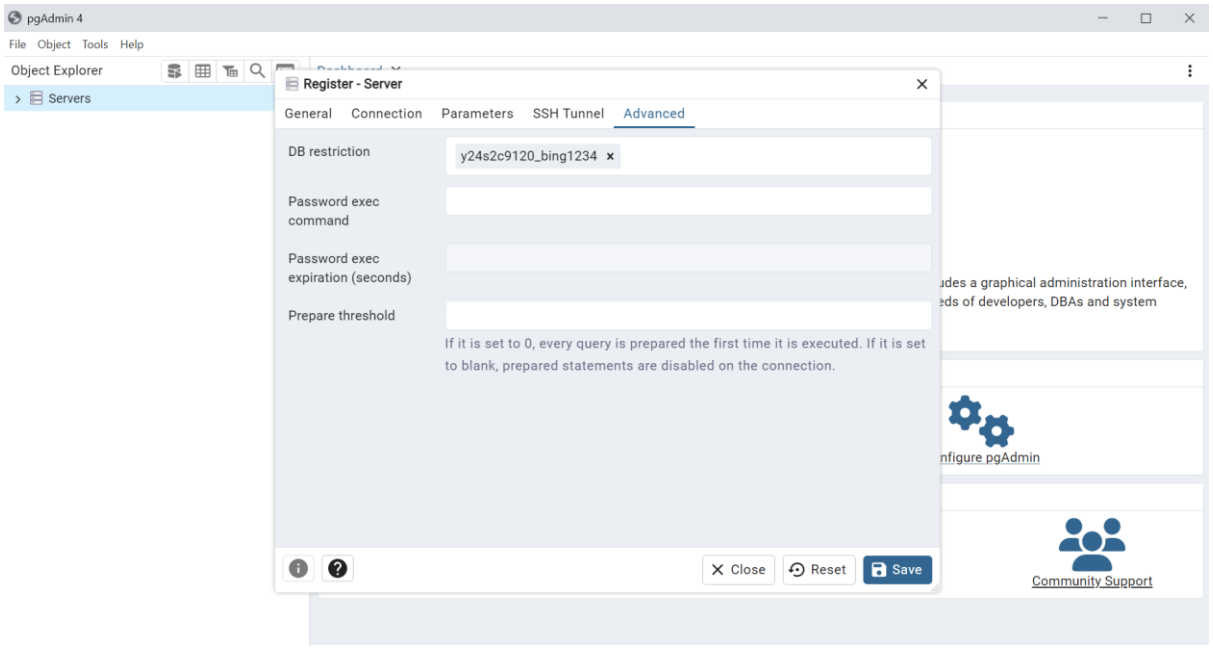
2. Right-click “Servers->Register->Server”. You should see the following screen. Type in a Name (this can be anything you like, e.g. USYD).



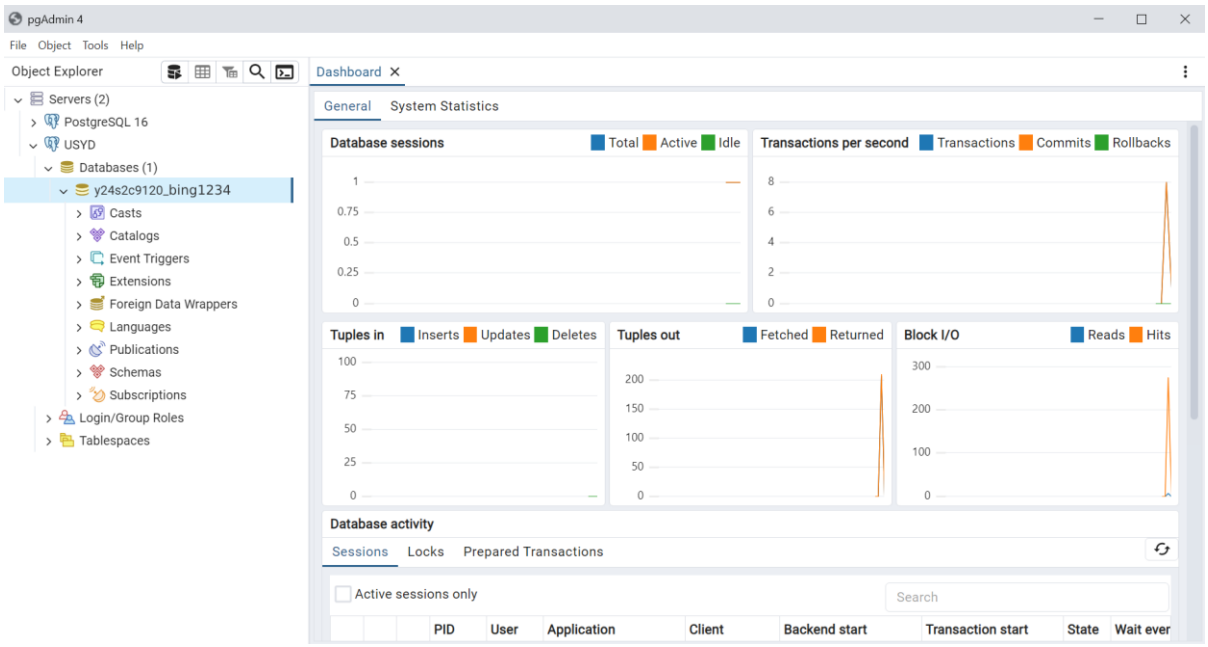
3. Click the tab “Connection”. Type in Host name/address, Maintenance database, Username, Password, as per your DB connection details.



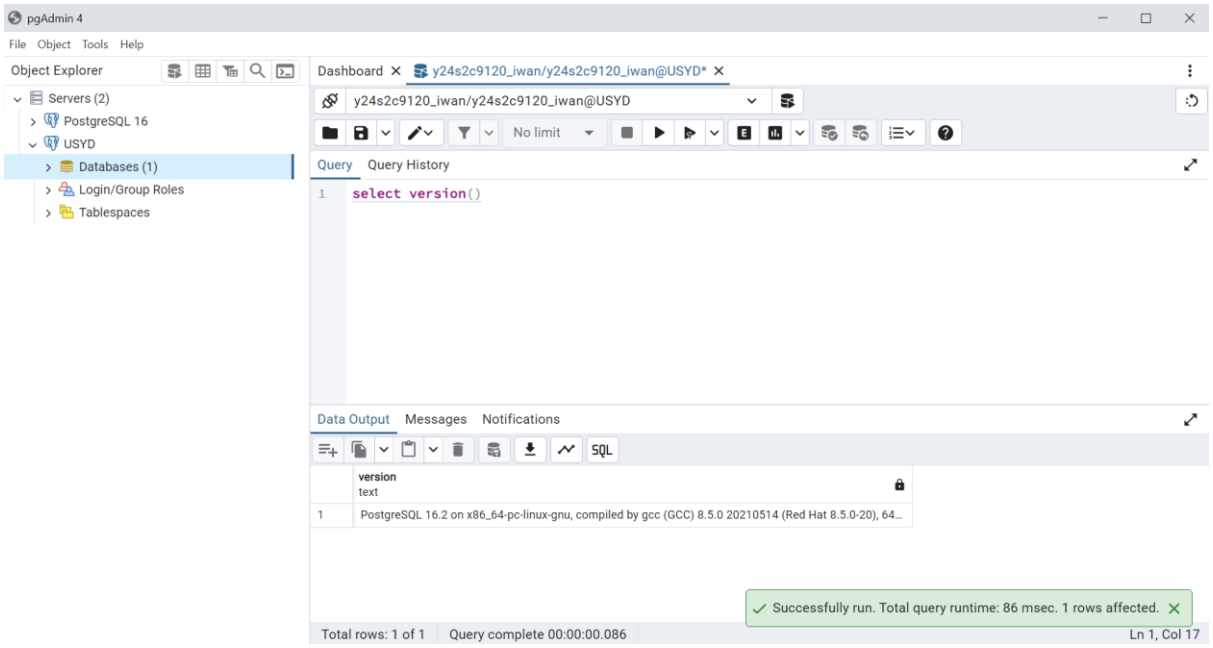
4. Click the tab “Advanced”, and copy the name of your DB onto the field “DB restriction”.



5. Click “Save”. You should see the following screen:



6. Select your Database, and right-click your Database -> “Query Tool”. You should see the following screen that allows you to write and execute SQL commands



7. Change your password by executing the following SQL command:

ALTER ROLE y24s2c9120_unikey WITH PASSWORD ‘a new secure password’;

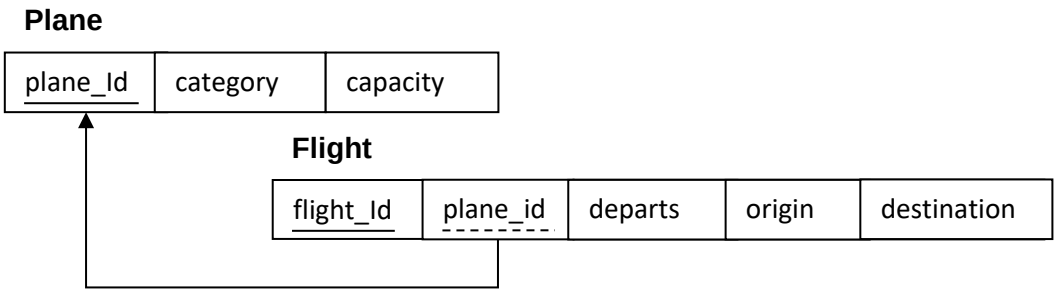
Replace ‘a new secure password’ with a new password that you wish to set for yourself. **Please remember your new password as you will need this in this course.**

Click  “Execute script” or F5.

You should then see “Query returned successfully in ** msec.” displayed on the Messages tab in the Output panel, meaning that your password was changed successfully!

Exercise 2: Flight Booking Schema

Here is a relational model diagram for part of a flight booking database schema:



Write DDL statements for these two relations. Execute the statements within your PostgreSQL account to test them out. You are using PostgreSQL syntax so you may wish to refer to the PostgreSQL CREATE TABLE documentation.

You should choose appropriate data types for your attributes based upon the following details:

(i) Each plane has a unique alphanumeric ID of up to 8 characters, a category (either 'jet' or 'turboprop'), and capacity (maximum number of passengers);

(ii) A flight has a unique numerical ID, departure date, the plane making the flight, the origin and destination;

Check the documentation on PostgreSQL data types.

Also make sure you capture all the appropriate key constraints (there is one foreign key and two primary keys).

Once the tables are created, add another integrity constraint to capture the extra rule that a plane cannot make more than one flight a day.

If you'd like to remove one of the tables that you've created, you may do so with DROP TABLE, after which you can create the table afresh.

Exercise 3: Populating the DB

Add data to your relations using INSERT statements. Then try inserting, updating and deleting data to violate the integrity constraints of the database. See if you can trigger an error for each of the classes of constraints described in Exercise 2.

You can inspect the contents of each relation with, e.g.:

```
SELECT * FROM Plane;
SELECT * FROM Flight;
```