CS 174A F23 Project Setup Guide

Mostly written by Kha-Dinh (Jacob) with some additions from Momin
Last Updated: 11/12/2023 8:52 AM

Overview

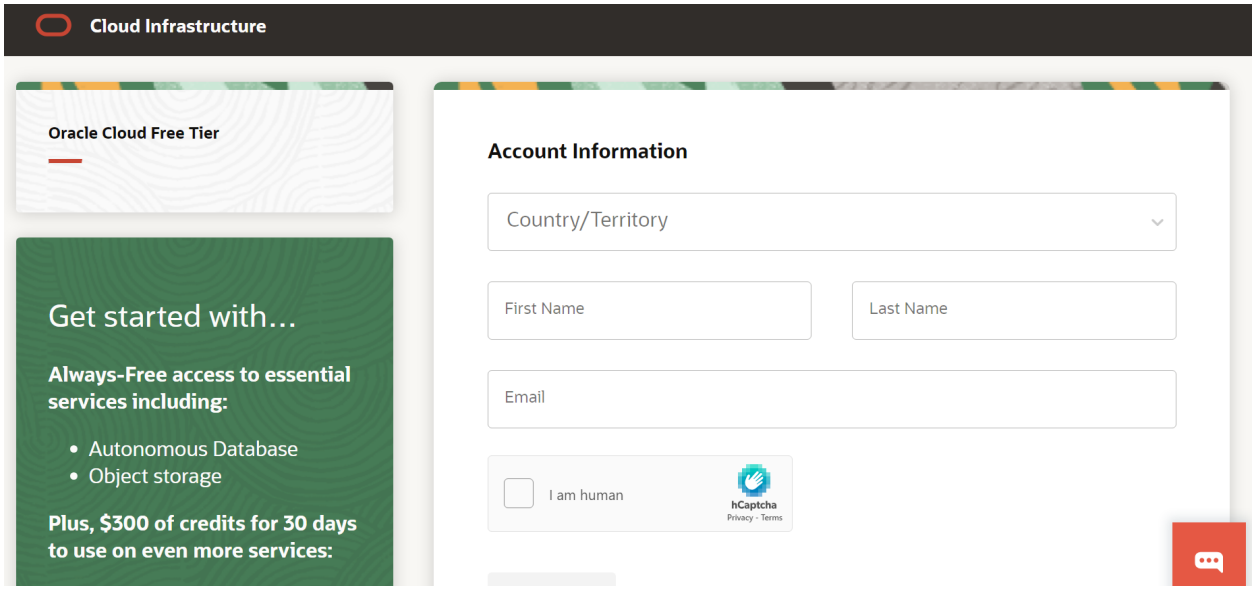
The goal of this document is to get you up and running for your CS 174A project this quarter. Here is a rough outline of what we'll accomplish:

1. Create an Oracle Cloud account to store your database **EXTERNAL** to your local device
2. Set up SQL Developer
    a. We'll install SQL Developer so you can easily *visualize* the tables in your database (you can also make queries/updates with it)
3. Set up Java Database Connectivity (JDBC)
    a. This will help you start actually constructing your project (which should be written in Java) by giving you a simple script with which you can make queries to your online database

Oracle Cloud

Update: If you've received an email from Oracle Academy on Professor Su's behalf, **FOLLOW THAT LINK** to create an Oracle account and you may not need to enter any payment info when you make your Oracle Cloud account (the two are linked if you use your UCSB email for both).

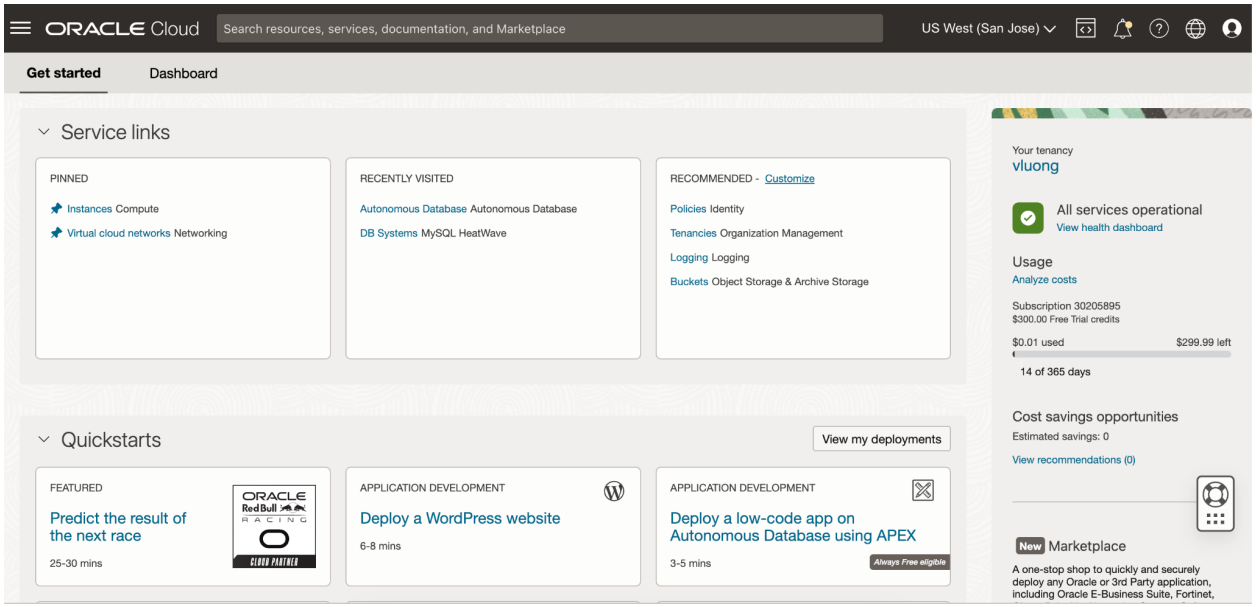
Otherwise, go to signup.cloud.oracle.com and create an Oracle Cloud account (use your UCSB email!)



Verify your email, then create a password you can remember.
For Customer type, choose Individual.
Choose a Cloud Account Name you can remember.
For Home Region, choose US West (San Jose).
Provide your address and avoid providing any payment info if possible. If you have to, that's okay, but just keep track of your free credits limit.

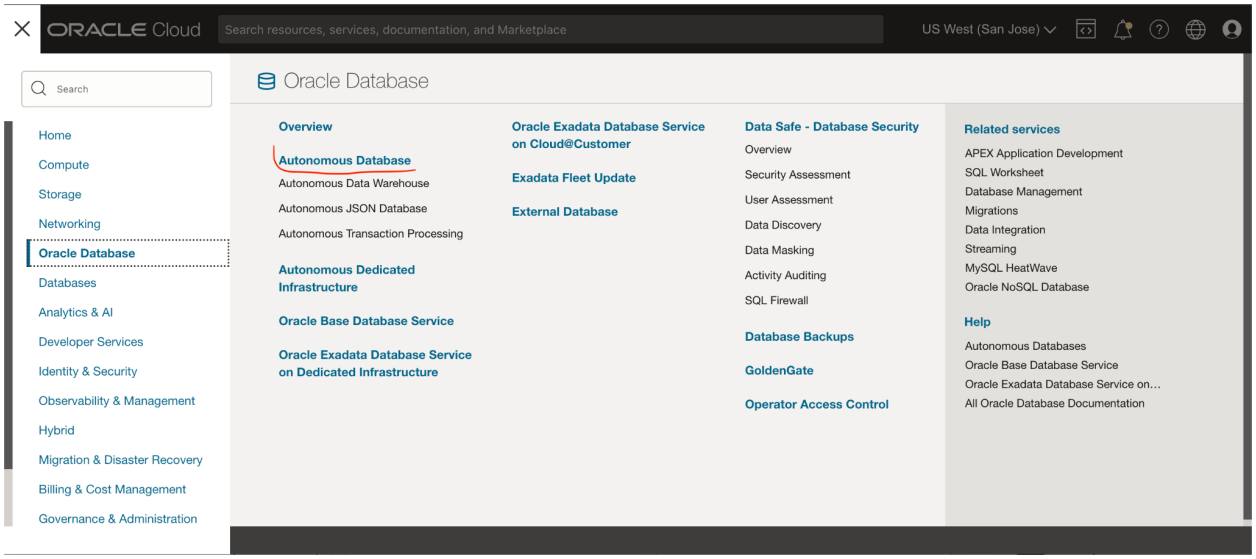
They may ask you to install Oracle Authenticator on your phone for 2-factor authentication.

Login to Oracle Cloud (cloud.oracle.com). Your home page should look like this:

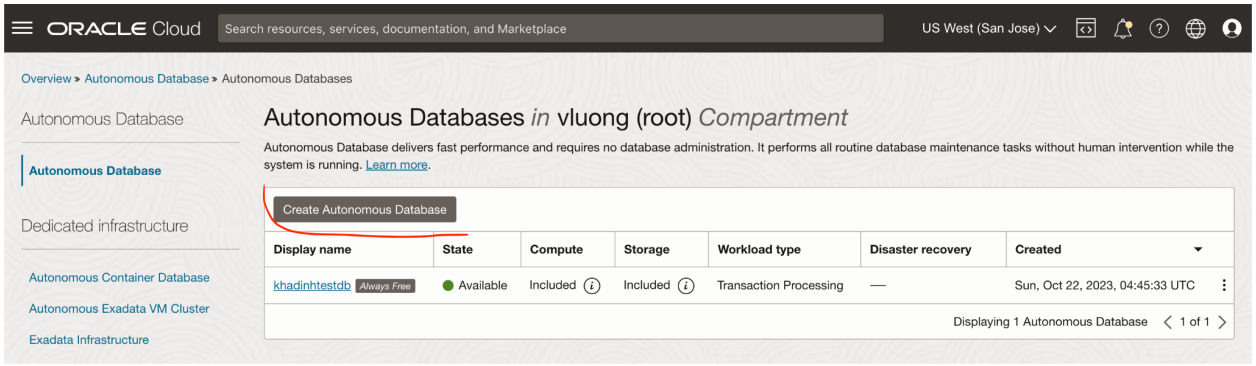


You should have a free $300 credit added to your account. All the services that we will use should either be free or only cost a few cents so we shouldn't use up most of these credits. To be safe, keep track of your free trial credits!

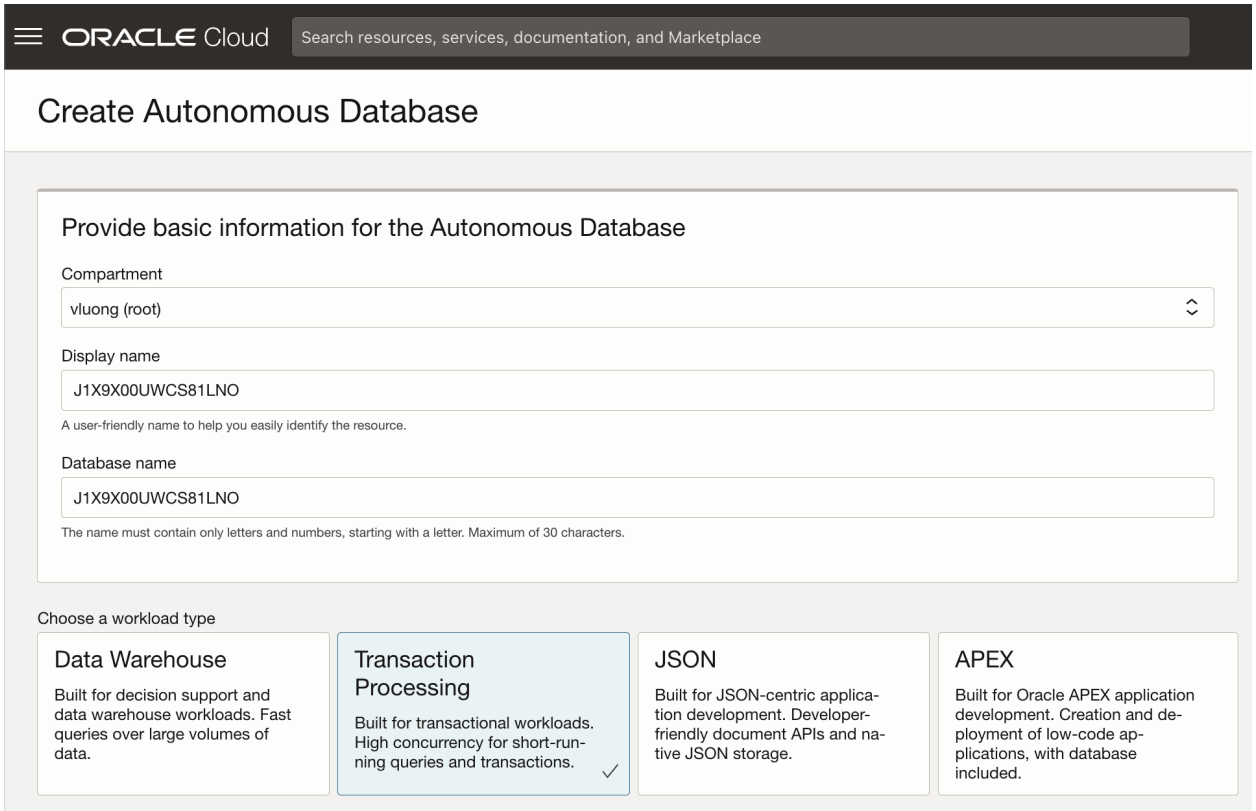
Click the hamburger menu in the top left corner.
We will create an Autonomous Database.
Go to Oracle Database -> Autonomous Database



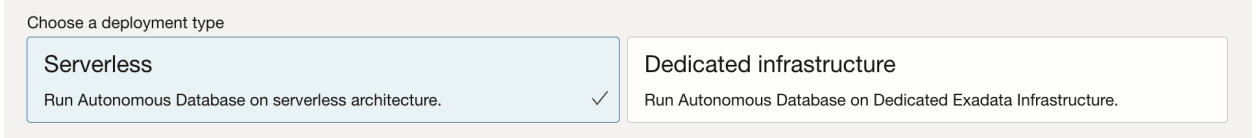
Click on Autonomous Database, you should land on this page.



Click "Create Autonomous Database":
You may set your display name to be something easy to recognize.
Set your database name to something you can remember: you'll need it for JDBC.

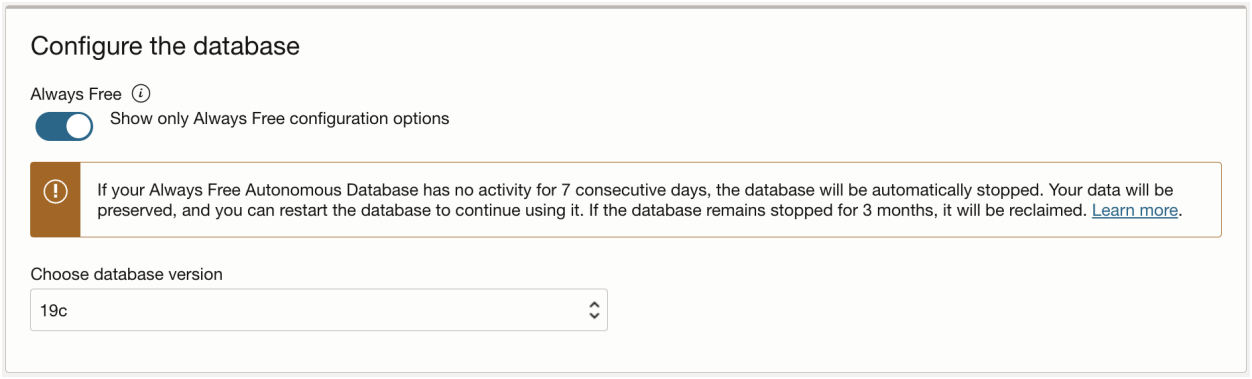


Choose Transaction Processing.



Choose Serverless deployment type.

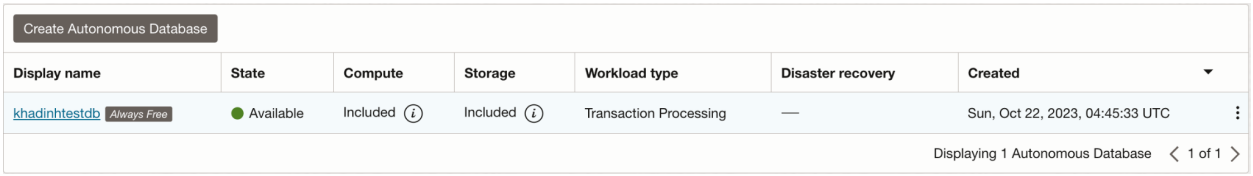
IMPORTANT: Turn on the "Always Free" option:



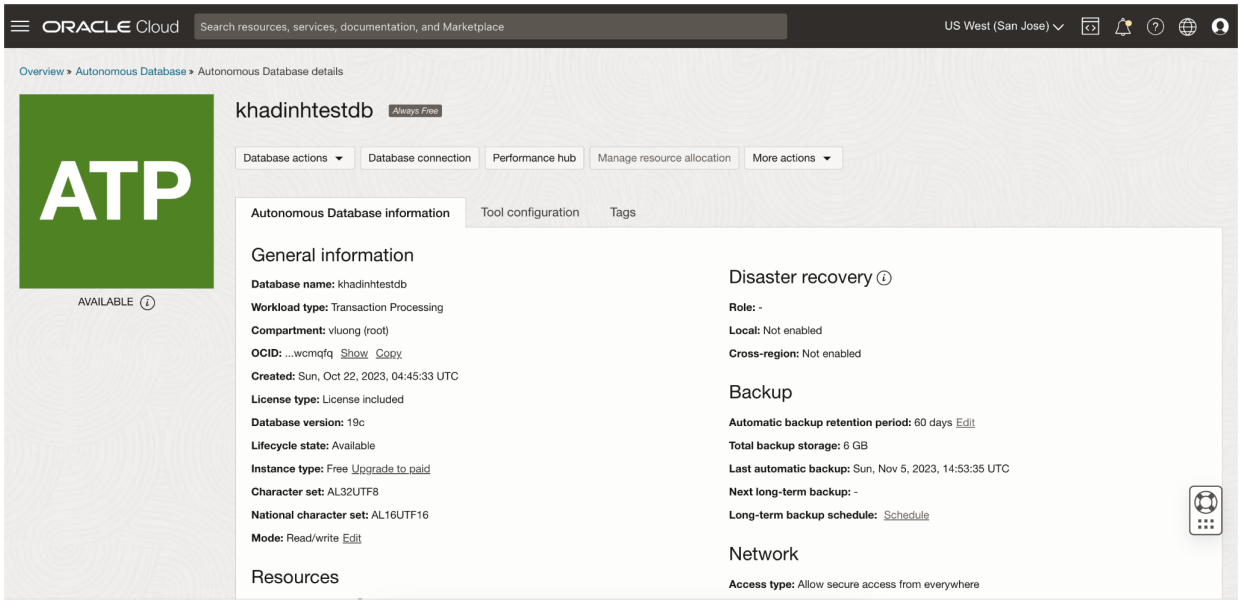
You may leave your username for this database as “ADMIN”.
Choose a password for this database and REMEMBER IT, because you’ll need it to connect with SQL Developer and JDBC.

Other parameters can be left as the default values. Click Create Autonomous Database.

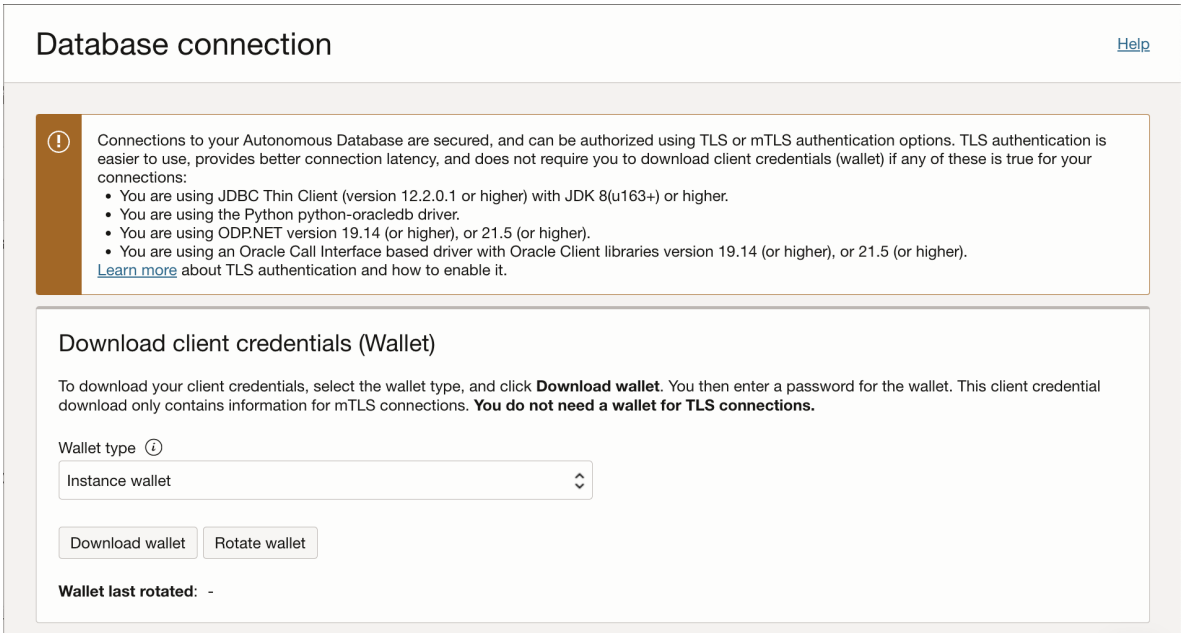
When you go back to the Autonomous Database main console, the database should be available in a few minutes and can be viewed from the list:



Click on the database name to further inspect:



Click on "Database connection":



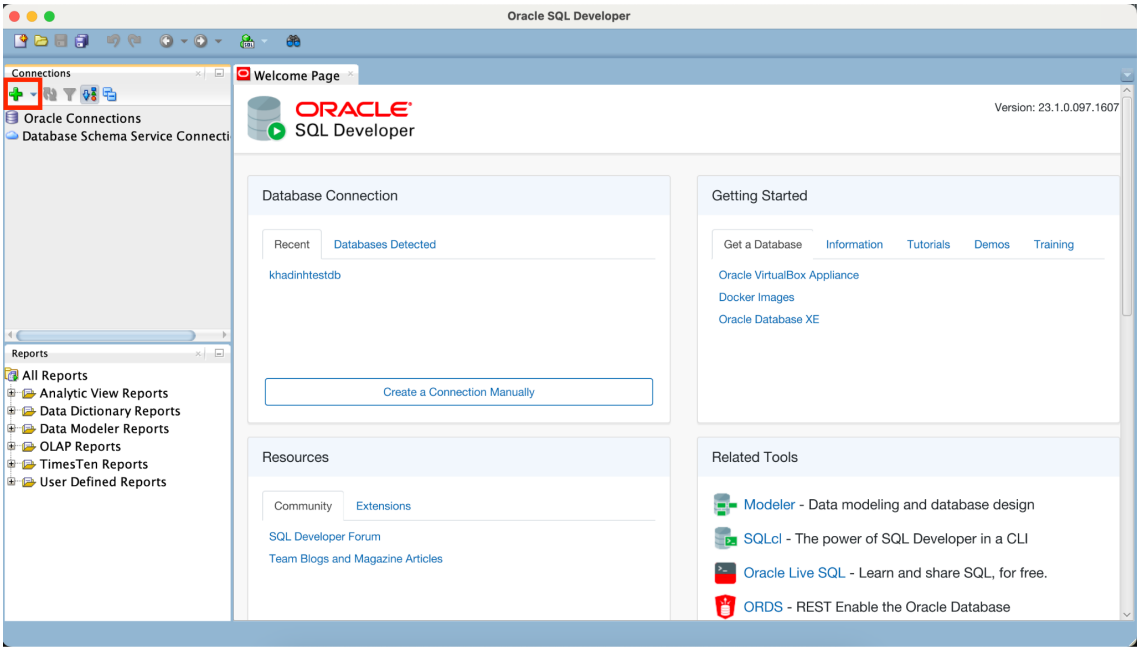
Click "Download wallet". The wallet should be downloaded as a zip file. Keep the ZIP file and also extract the ZIP to make another folder. We will use the zipped wallet to connect using SQL Developer and the unzipped wallet to connect with JDBC.

SQL Developer

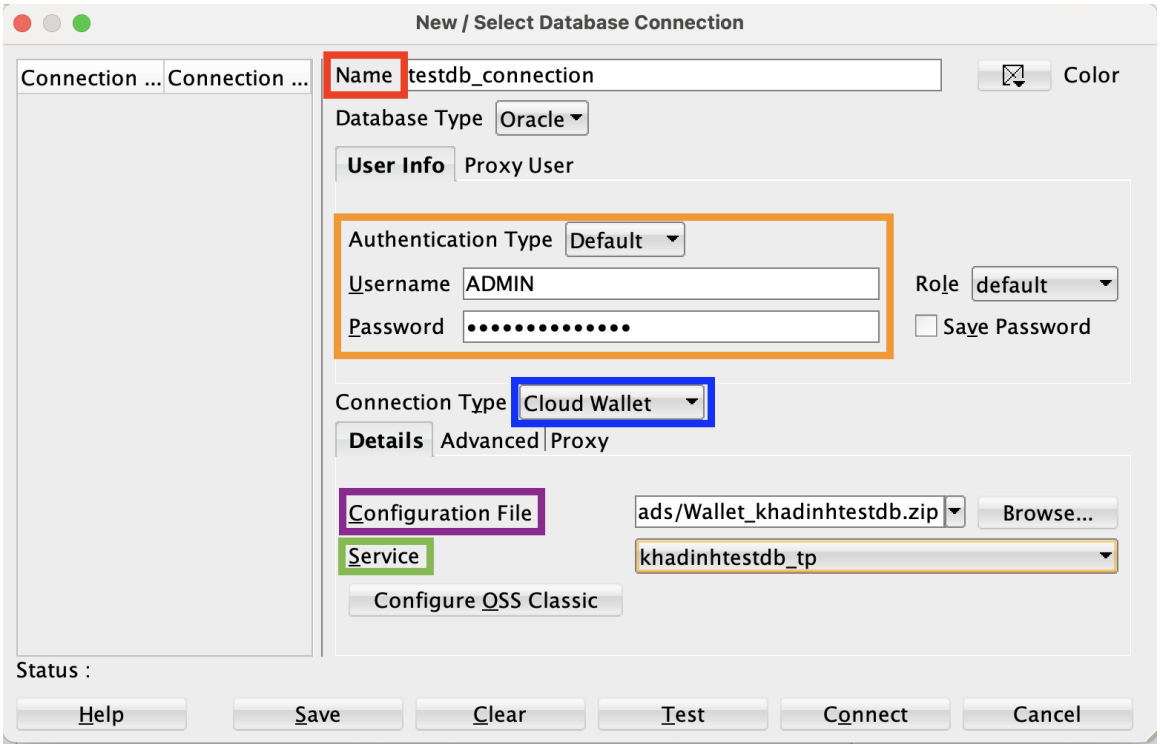
Download and install SQL Developer from https://www.oracle.com/database/sqldeveloper/technologies/download/

You may need to create an Oracle Account as well (different from Oracle Cloud Account). For Job Title, you can just put Student and for Company Name, you can put UCSB.

Open SQL Developer and click on the green plus icon to add a new connection



The console for adding a new connection should look like this:



Red: name your connection. It can be anything.

Orange: authentication for connecting to the autonomous database created. You can use 'ADMIN' for username. Password should be the password of your autonomous database.

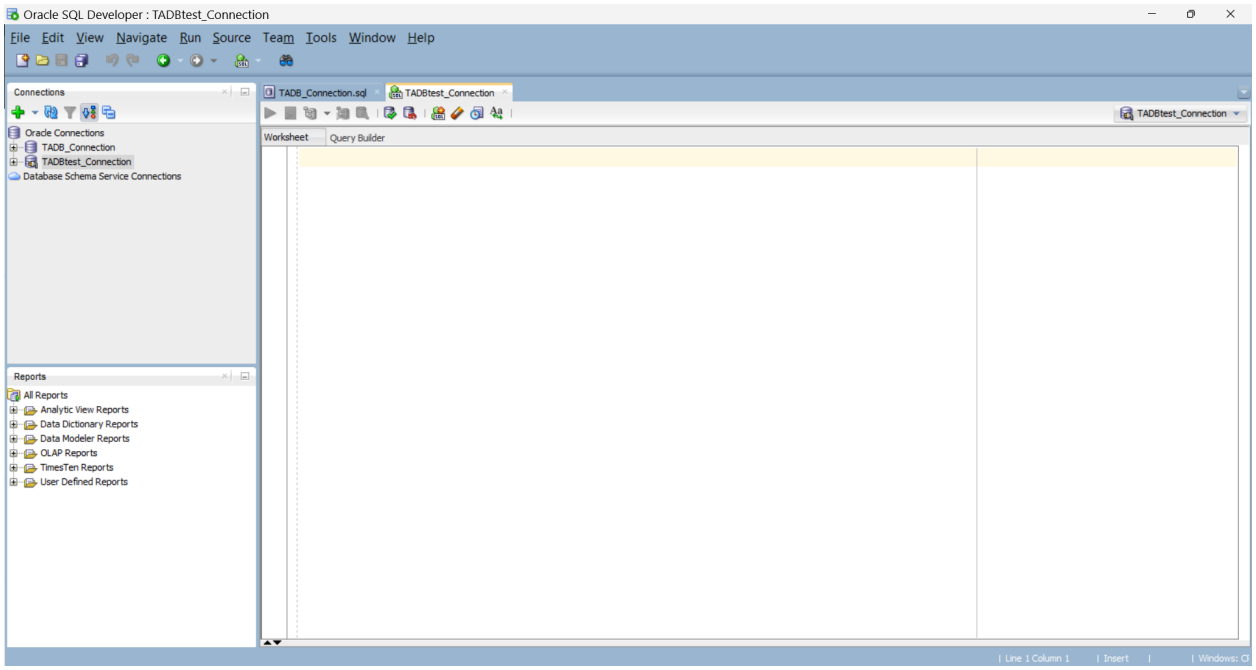
Blue: Choose 'Cloud Wallet'

Purple: Browse the .zip wallet file that we downloaded.

Green: Choose '_tp' or '_low'.

You can use the Test button to see if everything works: you should see a “Status: Success” in the bottom left corner of the window.

Hit Connect. The new connection now appears in the left box. If you expand it, you can select and view various tables in the database. Also, a new window should have opened for you to write SQL queries in.

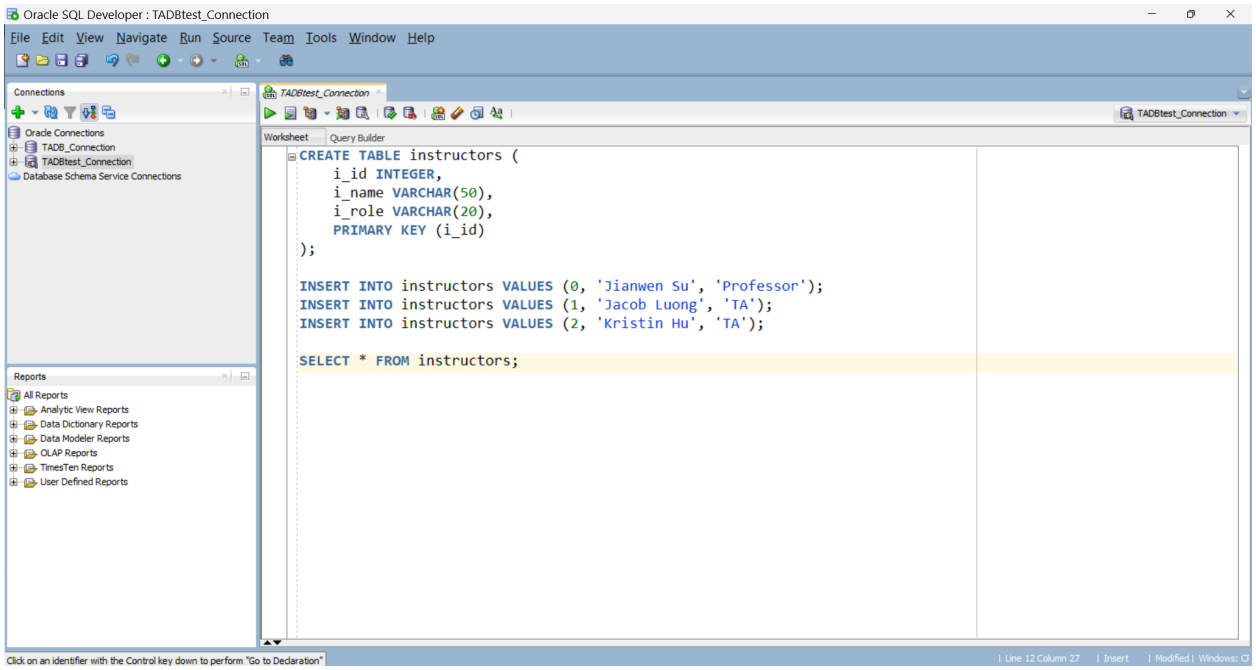


Add the following queries into your worksheet:

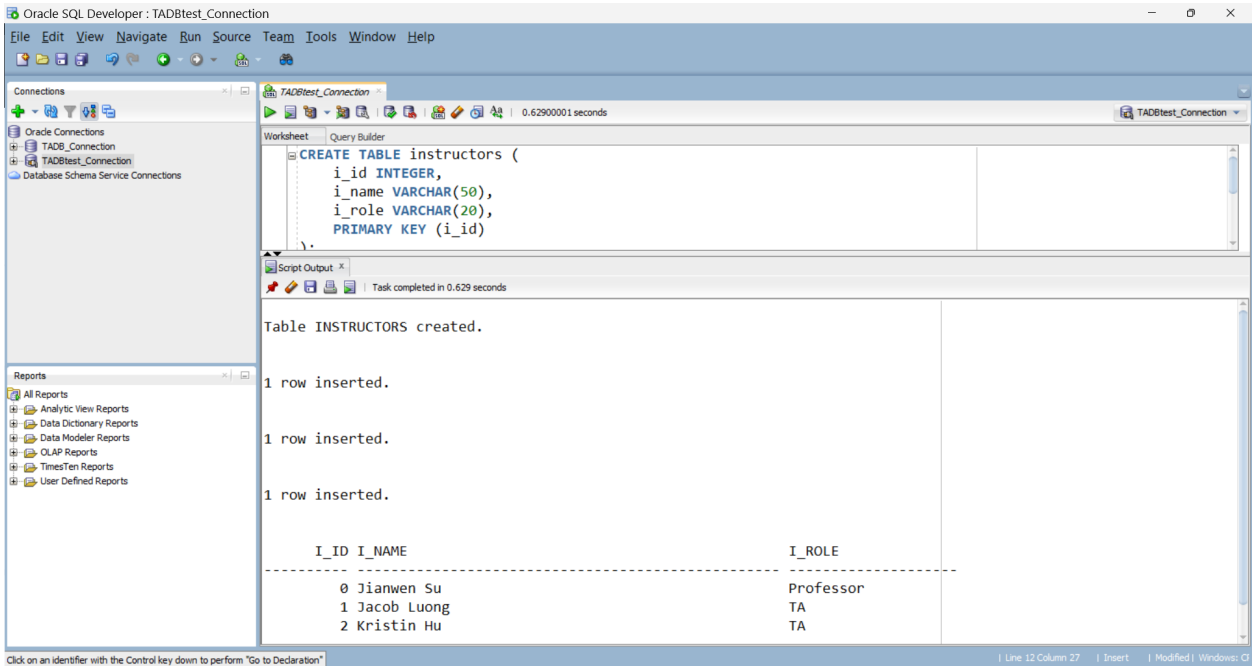
```
CREATE TABLE instructors (
    i_id INTEGER,
    i_name VARCHAR(50),
    i_role VARCHAR(20),
    PRIMARY KEY (i_id)
);

INSERT INTO instructors VALUES (0, 'Jianwen Su', 'Professor');
INSERT INTO instructors VALUES (1, 'Jacob Luong', 'TA');
INSERT INTO instructors VALUES (2, 'Kristin Hu', 'TA');

SELECT * FROM instructors;
```



Hitting Run (either the second button in the middle toolbar or F5) will execute the SQL commands in the worksheet, but they *won't* persist to your cloud database.



IMPORTANT: in order for the SQL operations to persist to your cloud database, hit the Commit button (either the sixth button in the middle toolbar or F11).

But wait… one of your instructors is missing! We'll use JDBC to add him into the database :)

Java Database Connectivity (JDBC)

To set up JDBC, start by setting up a Java project. If you're using VS Code, your project setup should look something like this repo:

https://github.com/hmomin/cs174a-jdbc

Alternatively, the main source file you need can be accessed on Canvas under:

Files → Project → TestConnection.java

Continue on from here using the instructions at the top of that file…