

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!)

Oracle Cloud Infrastructure

Oracle Cloud Free Tier

Get started with...

Always-Free access to essential services including:

- Autonomous Database
- Object storage

Plus, \$300 of credits for 30 days to use on even more services:

Account Information

Country/Territory

First Name

Last Name

Email

☐ I am human

hCaptcha
Privacy - Terms

Verify your email, then create a password you can remember.

For Customer type, choose Individual.

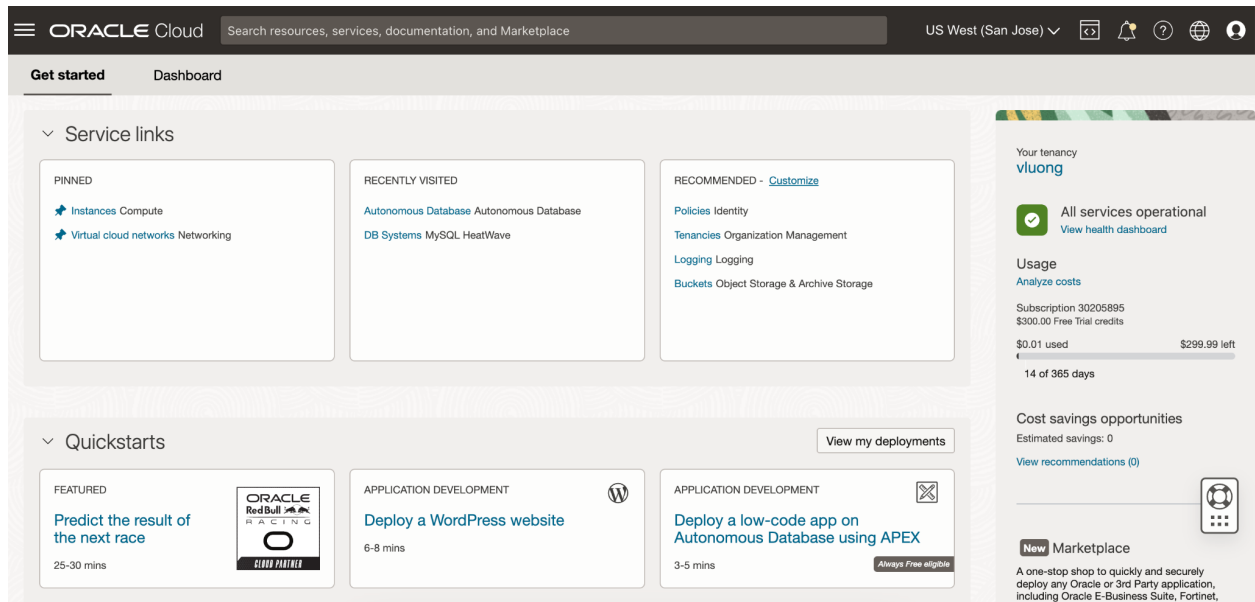
Choose a Cloud Account Name you can remember.

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.

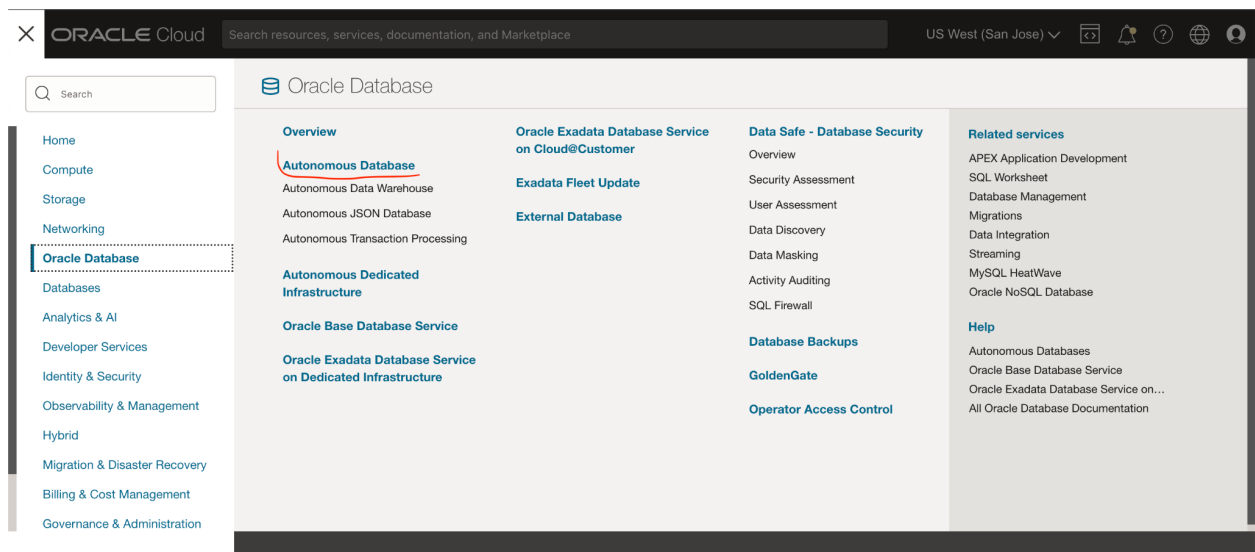
They may ask you to install Oracle Authenticator on your phone for 2-factor authentication.

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.

ORACLE Cloud Search resources, services, documentation, and Marketplace US West (San Jose) [Icons]

Overview > Autonomous Database > Autonomous Databases

Autonomous Database

Autonomous Database

Dedicated infrastructure

Autonomous Container Database

Autonomous Exadata VM Cluster

Exadata Infrastructure

Autonomous Databases in vluong (root) Compartment

Autonomous Database delivers fast performance and requires no database administration. It performs all routine database maintenance tasks without human intervention while the system is running. [Learn more.](#)

Create Autonomous Database

Display name	State	Compute	Storage	Workload type	Disaster recovery	Created
khadinhtestdb <small>Always Free</small>	Available	Included ⓘ	Included ⓘ	Transaction Processing	—	Sun, Oct 22, 2023, 04:45:33 UTC

Displaying 1 Autonomous Database < 1 of 1 >

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.

ORACLE Cloud Search resources, services, documentation, and Marketplace

Create Autonomous Database

Provide basic information for the Autonomous Database

Compartment

vluong (root)

Display name

J1X9X00UWCS81LNO

A user-friendly name to help you easily identify the resource.

Database name

J1X9X00UWCS81LNO

The name must contain only letters and numbers, starting with a letter. Maximum of 30 characters.

Choose a workload type

Data Warehouse

Built for decision support and data warehouse workloads. Fast queries over large volumes of data.

Transaction Processing

Built for transactional workloads. High concurrency for short-running queries and transactions. ✓

JSON

Built for JSON-centric application development. Developer-friendly document APIs and native JSON storage.

APEX

Built for Oracle APEX application development. Creation and deployment of low-code applications, with database included.

Choose Transaction Processing.

Choose a deployment type

Serverless

Run Autonomous Database on serverless architecture. ✓

Dedicated infrastructure

Run Autonomous Database on Dedicated Exadata Infrastructure.


Choose Serverless deployment type.

IMPORTANT: Turn on the "Always Free" option:

Configure the database

Always Free ⓘ

☒ Show only Always Free configuration options



If your Always Free Autonomous Database has no activity for 7 consecutive days, the database will be automatically stopped. Your data will be preserved, and you can restart the database to continue using it. If the database remains stopped for 3 months, it will be reclaimed. [Learn more.](#)

Choose database version

19c

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:

Create Autonomous Database						
Display name	State	Compute	Storage	Workload type	Disaster recovery	Created
khadinhtestdb Always Free	Available	Included ⓘ	Included ⓘ	Transaction Processing	—	Sun, Oct 22, 2023, 04:45:33 UTC
Displaying 1 Autonomous Database < 1 of 1 >						

Click on the database name to further inspect:

The screenshot shows the Oracle Cloud console interface. At the top, there's a navigation bar with the Oracle Cloud logo, a search bar, and the region 'US West (San Jose)'. Below this, the breadcrumb trail reads 'Overview > Autonomous Database > Autonomous Database details'. The main content area is titled 'khadinhtestdb' with a status 'Always Free'. On the left, there's a green 'ATP' logo and a status 'AVAILABLE'. The right side contains several tabs: 'Database actions', 'Database connection', 'Performance hub', 'Manage resource allocation', and 'More actions'. Below these, there are three sub-tabs: 'Autonomous Database information', 'Tool configuration', and 'Tags'. The 'Autonomous Database information' tab is active, showing 'General information' and 'Disaster recovery'. The 'General information' section includes details like Database name, Workload type, Compartment, OCID, Created date, License type, Database version, Lifecycle state, Instance type, Character set, National character set, and Mode. The 'Disaster recovery' section shows Role, Local, and Cross-region settings. Below this, there's a 'Backup' section with Automatic backup retention period, Total backup storage, Last automatic backup, Next long-term backup, and Long-term backup schedule. Finally, there's a 'Network' section with Access type.

Click on "Database connection":

The screenshot shows the 'Database connection' page. At the top, there's a title 'Database connection' and a 'Help' link. Below this, there's a warning icon and a message: 'Connections to your Autonomous Database are secured, and can be authorized using TLS or mTLS authentication options. TLS authentication is easier to use, provides better connection latency, and does not require you to download client credentials (wallet) if any of these is true for your connections:'. This is followed by a list of conditions: 'You are using JDBC Thin Client (version 12.2.0.1 or higher) with JDK 8(u163+) or higher.', 'You are using the Python python-oracledb driver.', 'You are using ODP.NET version 19.14 (or higher), or 21.5 (or higher).', and 'You are using an Oracle Call Interface based driver with Oracle Client libraries version 19.14 (or higher), or 21.5 (or higher).'. A 'Learn more' link is provided. Below this, there's a section titled 'Download client credentials (Wallet)'. It contains a paragraph: 'To download your client credentials, select the wallet type, and click **Download wallet**. You then enter a password for the wallet. This client credential download only contains information for mTLS connections. **You do not need a wallet for TLS connections.**'. Below this, there's a 'Wallet type' dropdown menu with 'Instance wallet' selected. There are two buttons: 'Download wallet' and 'Rotate wallet'. At the bottom, there's a field 'Wallet last rotated:' with a '-' sign.

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.

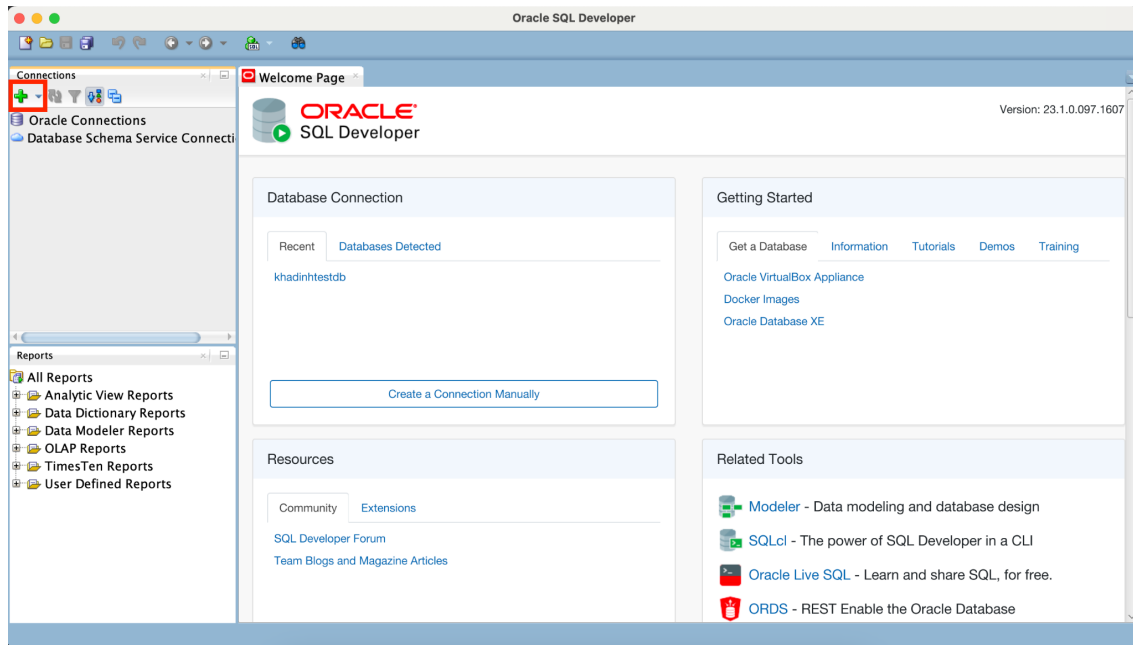
SQL Developer

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.

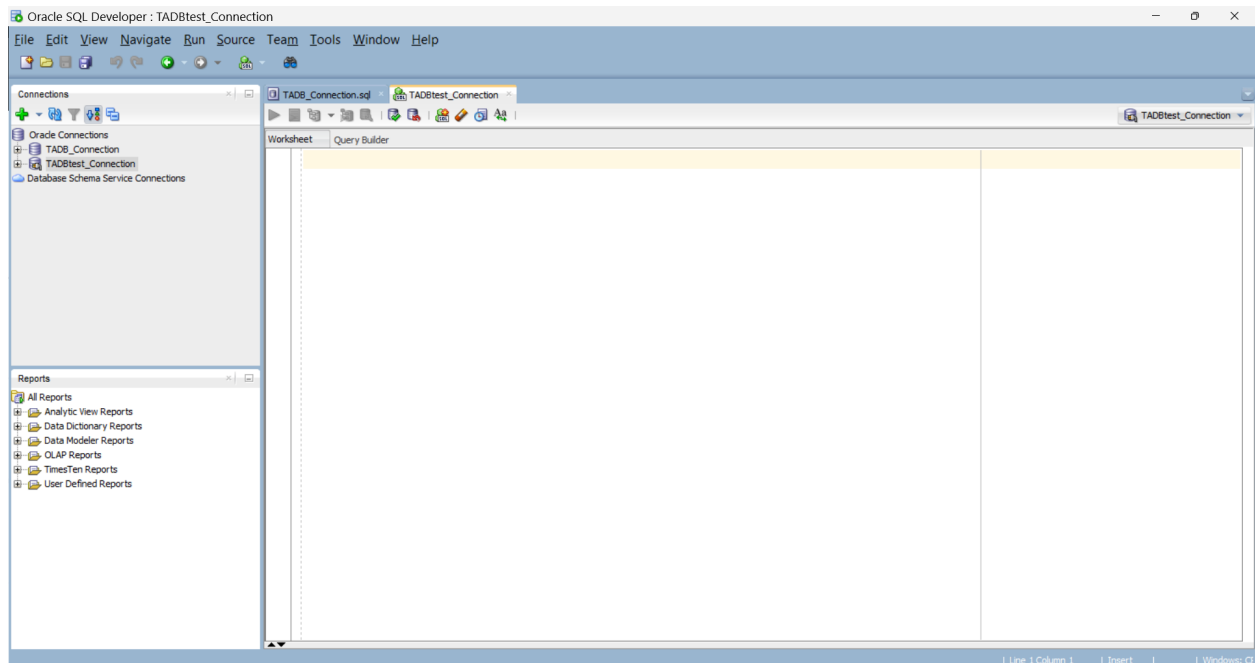
Blue: Choose 'Cloud Wallet'

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.

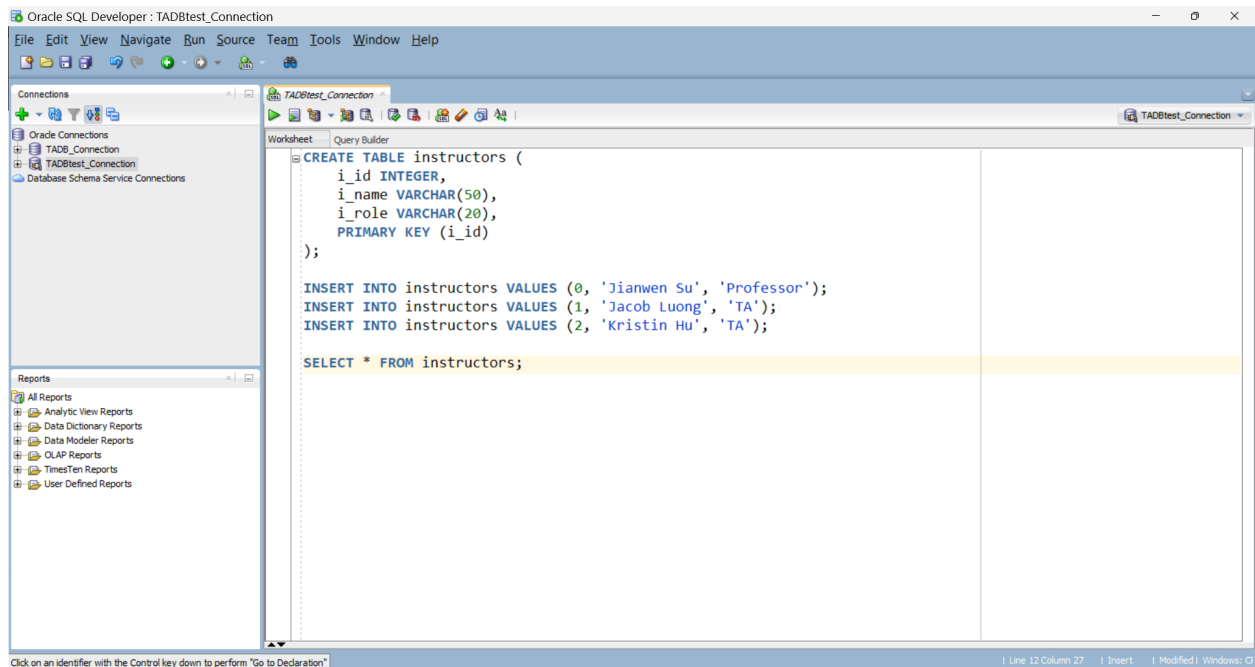


Add the following queries into your worksheet:

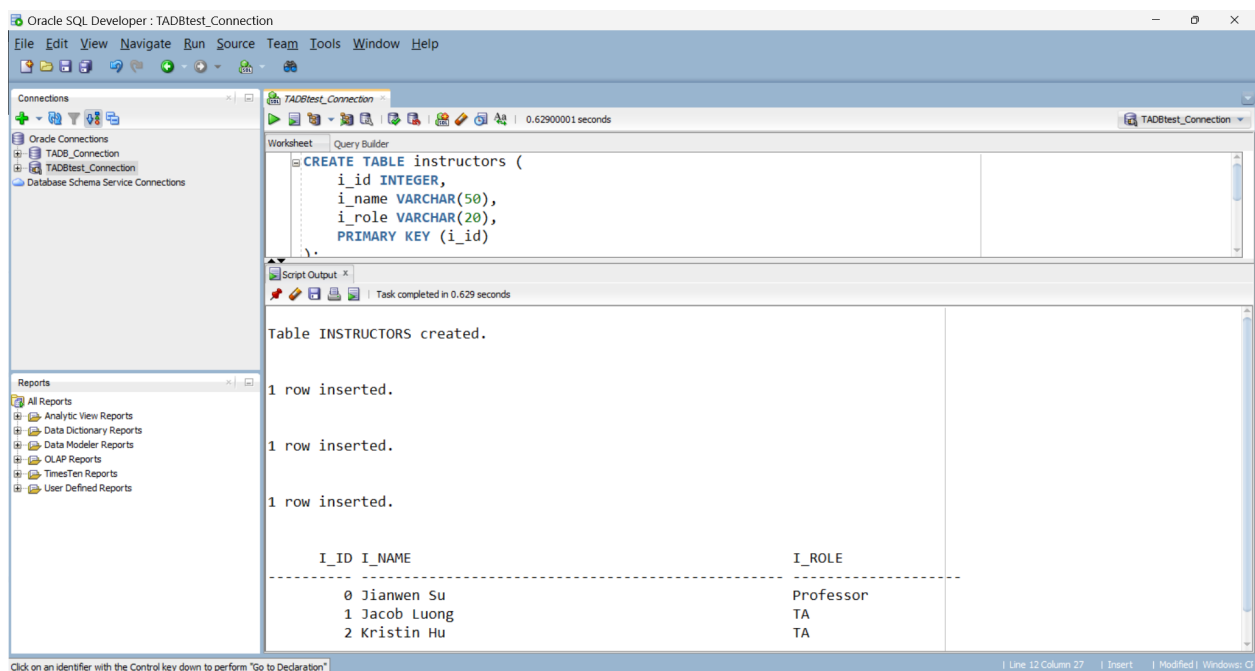
```
CREATE TABLE instructors (  
    i_id INTEGER,  
    i_name VARCHAR(50),  
    i_role VARCHAR(20),  
    PRIMARY KEY (i_id)  
);
```

```
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...