Creating an instance, dataset, and table in Cloud Spanner

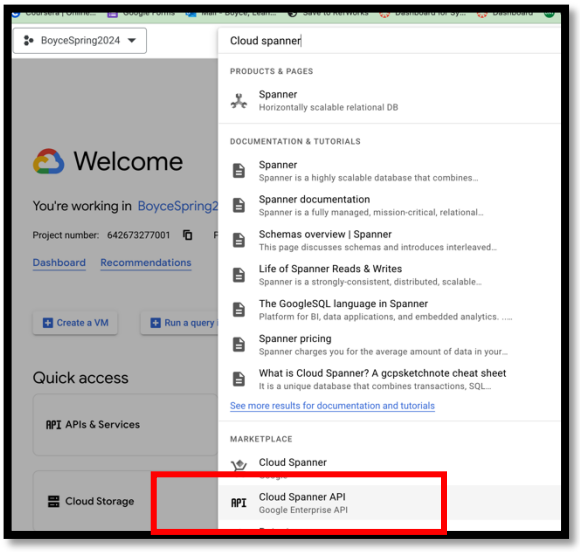
Objectives

This tutorial walks you through the following steps using the gcloud command-line tool. Remember to work with the shell you need to click on the Activate Cloud Shell in the upper right hand in the blue bar.

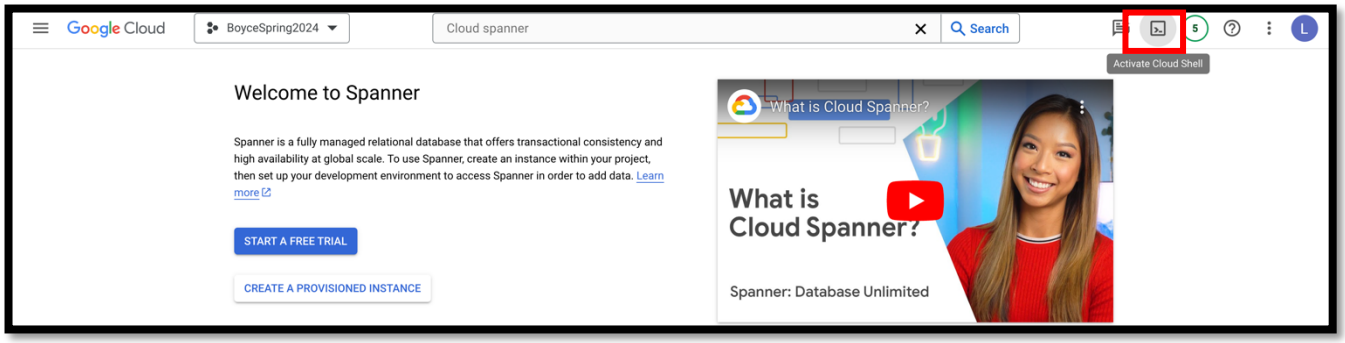
- Create a Cloud Spanner instance, database, and schema
- Write data to the database and execute SQL queries that data
- Clean up by deleting the database and instance

Before you begin

1. We will be using the project you set up for this class.
2. If Cloud Spanner API is not already enabled, you need to enable the API now. Type in Cloud Spanner API in the search bar and click on Cloud Spanner API. Then click on enable. If you forget how to enable the API, please go back to the directions for this in a previous class. Here is a hint.



3. We will work once again in the Cloud Shell for this exercise. To do so, click in the upper right-hand corner and activate Cloud shell.



Instances

When you first use Cloud Spanner, you must create an instance, which is an allocation of resources that are used by Cloud Spanner databases. When you create an instance, you choose where your data is stored and how much computing capacity the instance has.

We will create an instance using the cloud shell.

Create an instance

To create an instance named test-instance with the display name My Instance using the regional instance configuration regional-us-central1 with 1 nodes:

```
gcloud spanner instances create boyce-instance \
  --config=regional-us-central1 \
  --description="Boyce Instance" \
  --nodes=2
```

```
leann_boyce@cloudshell:~ (boycespring2024)$ gcloud spanner instances create boyce-instance \
  --config=regional-us-central1 \
  --description="Boyce Instance" \
  --nodes=2
Creating instance...done.
```

In the command above, the instance name is set to boyce-instance and --description sets the display name of the instance (“Boyce Instance”). Both of these values must be unique within a Google Cloud Platform project.

Note: Use the instance ID, not the display name, when referring to an instance in **gcloud spanner** commands.

Set the default instance

You can set the default instance that Cloud Spanner uses when you have not specified an

instance in your command. To set the default instance:

```
gcloud config set spanner/instance boyce-instance
```

Create a database

Create a database named example-db.

```
gcloud spanner databases create example-db
```

Create a schema

Use Cloud Spanner's Data Definition Language (DDL) to create, alter, or drop tables, and to create or drop indexes. Enter each one separately. Let's create two tables. The first is for Players1 and the second is for PlayerStats1. When creating tables and entering data, it is very important that you do not have curly quotes enabled on your computer. You will need to go into your preferences on your computer and change this or you will receive an error for unrecognized arguments. If you do not know how to do this, do a quick Google search to get some help. You can also get with me or a TA during office hours. It is best to use a text editor and make all text "Plain Text."

Players1

gcloud spanner databases ddl update example-db \
--ddl='CREATE TABLE Players1 (PlayersId INT64 NOT NULL, FirstName STRING(1024), LastName STRING(1024)) PRIMARY KEY (PlayersId)'

PlayerStats1

gcloud spanner databases ddl update example-db \
--ddl='CREATE TABLE PlayerStats1 (PlayersId INT64 NOT NULL, GameId INT64 NOT NULL, OpposingTeam STRING(MAX), PointsScore INT64, OpposingScore INT64) PRIMARY KEY (PlayersId, GameId), INTERLEAVE IN PARENT Players1 ON DELETE CASCADE'

You may also try having the OpposingTeam STRING(1024) instead of STRING(MAX). I doubt you will need “MAX” as it would allow a string in size, about 2 GB.

```
leann_boyce@cloudshell:~ (boycespring2024)$ gcloud spanner databases ddl update example-db \
--ddl='CREATE TABLE Players1 ( PlayersId INT64 NOT NULL, FirstName STRING(1024), LastName STRING(1024) ) PRIMARY KEY (PlayersId)'
Schema updating...done.
leann_boyce@cloudshell:~ (boycespring2024)$ gcloud spanner databases ddl update example-db \
--ddl='CREATE TABLE PlayerStats1 ( PlayersId INT64 NOT NULL, GameId INT64 NOT NULL, OpposingTeam STRING(MAX), PointsScore INT64, OpposingScore INT64) PRIMARY KEY (PlayersId, GameId), INTERLEAVE IN
 PARENT Players1 ON DELETE CASCADE'
Schema updating...done.
leann_boyce@cloudshell:~ (boycespring2024)$
```

Write data

Let's add some sample data to our database. We will add 3 rows of data to each table. Enter each separately.

Enter Data

gcloud spanner rows insert --instance=boyce-instance --database=example-db \

--table=Players1 \

--data=PlayersId=1,FirstName='Lebron',LastName='James'

gcloud spanner rows insert --instance=boyce-instance --database=example-db \

--table=Players1 \

--data=PlayersId=2,FirstName=’Stephen’,LastName=‘Curry’

gcloud spanner rows insert --instance=boyce-instance --database=example-db \

--table=Players1 \

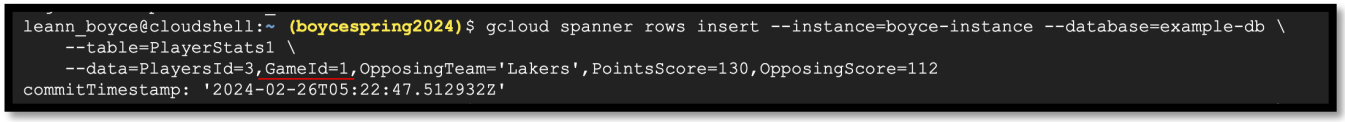
--data=PlayersId=3,FirstName=‘Kevin’,LastName=‘Durant’

```
leann_boyce@cloudshell:~ (boycespring2024)$ gcloud spanner rows insert --instance=boyce-instance --database=example-db \
    --table=Players1 \
    --data=PlayersId=1,FirstName='Lebron',LastName='James'
commitTimestamp: '2024-02-26T05:12:16.821980Z'
leann_boyce@cloudshell:~ (boycespring2024)$ gcloud spanner rows insert --instance=boyce-instance --database=example-db \
    --table=Players1 \
    --data=PlayersId=2,FirstName=’Stephen’,LastName=‘Curry’
commitTimestamp: '2024-02-26T05:14:07.868360Z'
leann_boyce@cloudshell:~ (boycespring2024)$
leann_boyce@cloudshell:~ (boycespring2024)$ gcloud spanner rows insert --instance=boyce-instance --database=example-db \
    --table=Players1 \
    --data=PlayersId=3,FirstName=’Kevin’,LastName=‘Durant’
commitTimestamp: '2024-02-26T05:14:54.068515Z'
```

You can see we have 3 players: James Lebron, Stephen Curry, and Kevin Durant. Now let's enter some data into the second table.

Here is the code to enter Game 1:

gcloud spanner rows insert --instance=boyce-instance --database=example-db \
--table=PlayerStats1 \
--data=PlayersId=3,GameId=1,OpposingTeam='Lakers',PointsScore=130,OpposingScore=112

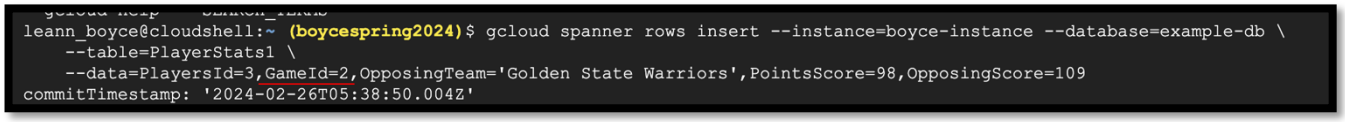


Here is the code to enter Game 2:

gcloud spanner rows insert --instance=boyce-instance --database=example-db \

--table=PlayerStats1 \

--data=PlayersId=3,GameId=2,OpposingTeam='Golden State Warriors',PointsScore=98,OpposingScore=109



Here is the code to enter Game 3:

gcloud spanner rows insert --instance=boyce-instance --database=example-db \
--table=PlayerStats1 \
--data=PlayersId=1,GameId=3,OpposingTeam='Suns',PointsScore=113,OpposingScore=123

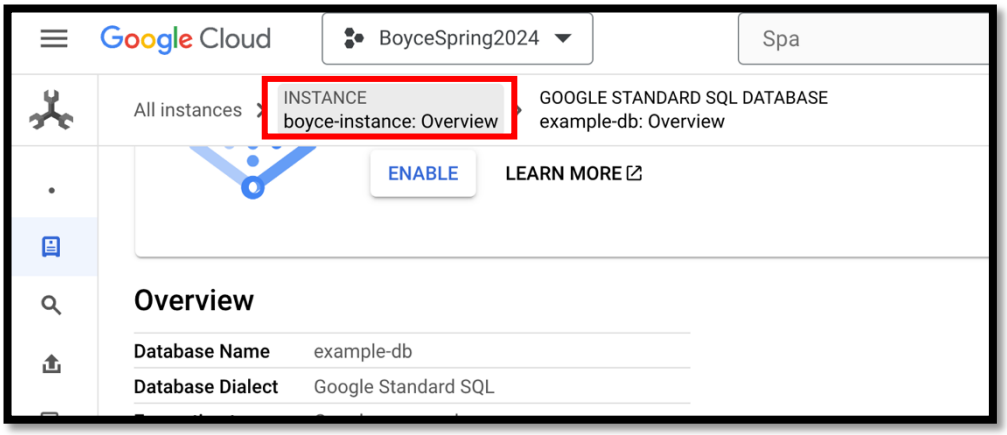
By default, comma is used to delimit items in lists. If you have commas in the a title, you will specify colon (^:^) as the delimiter.

Close the Cloud Shell and go to the Navigation Pane by clickin on “X” in the top of the terminal.

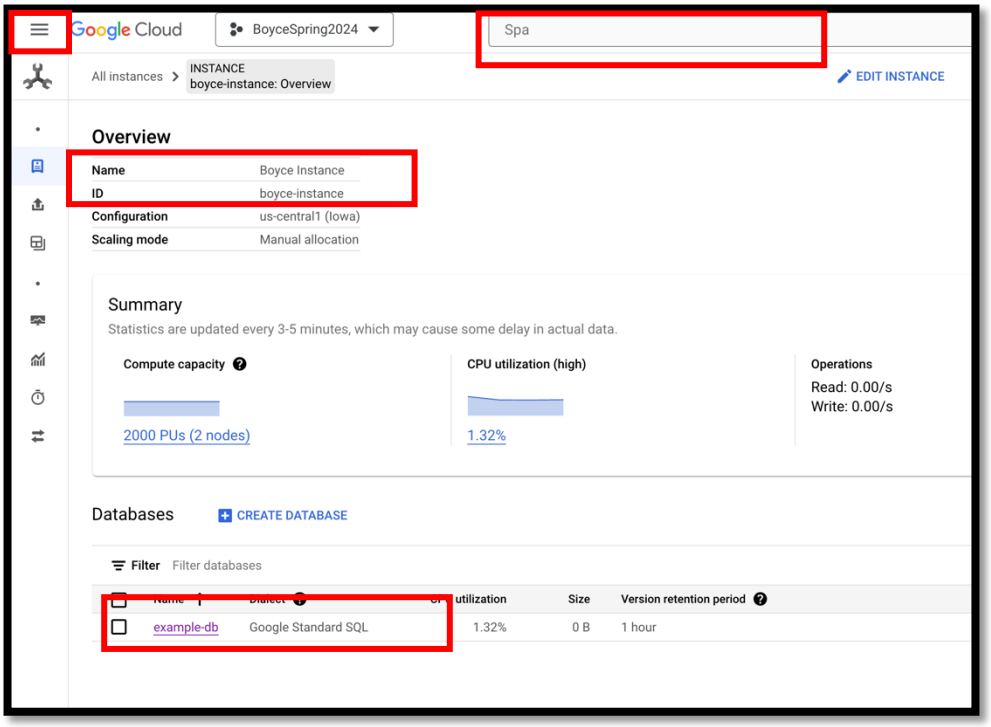
Query data using SQL

The only query we will make is to see the tables and the data created. We will make more queries in the next half of the semester. This is just a little teaser. There are many ways to make a query from Spanner. If you know a different way that is fine; just be sure you are in Spanner.

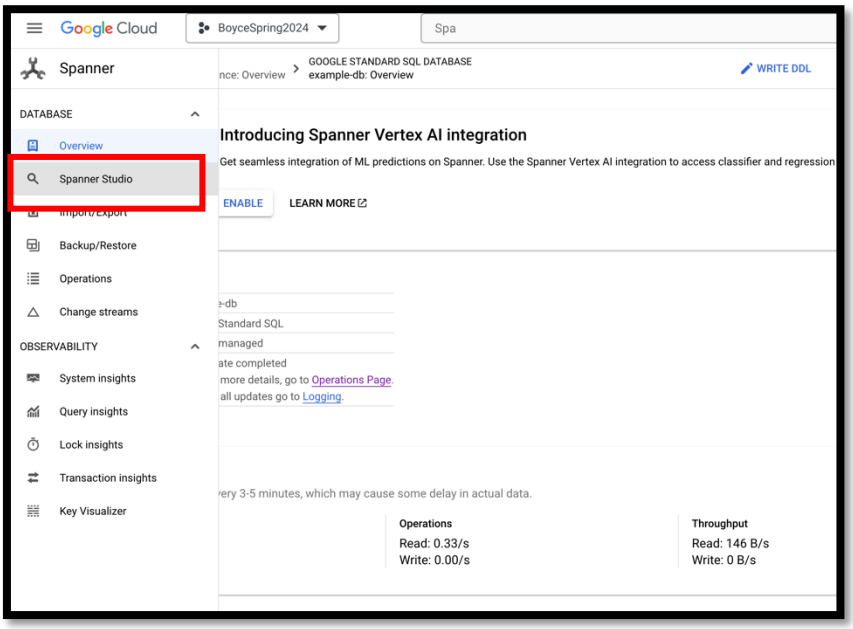
1. Click on the instance you are working in once you exit out of the terminal.



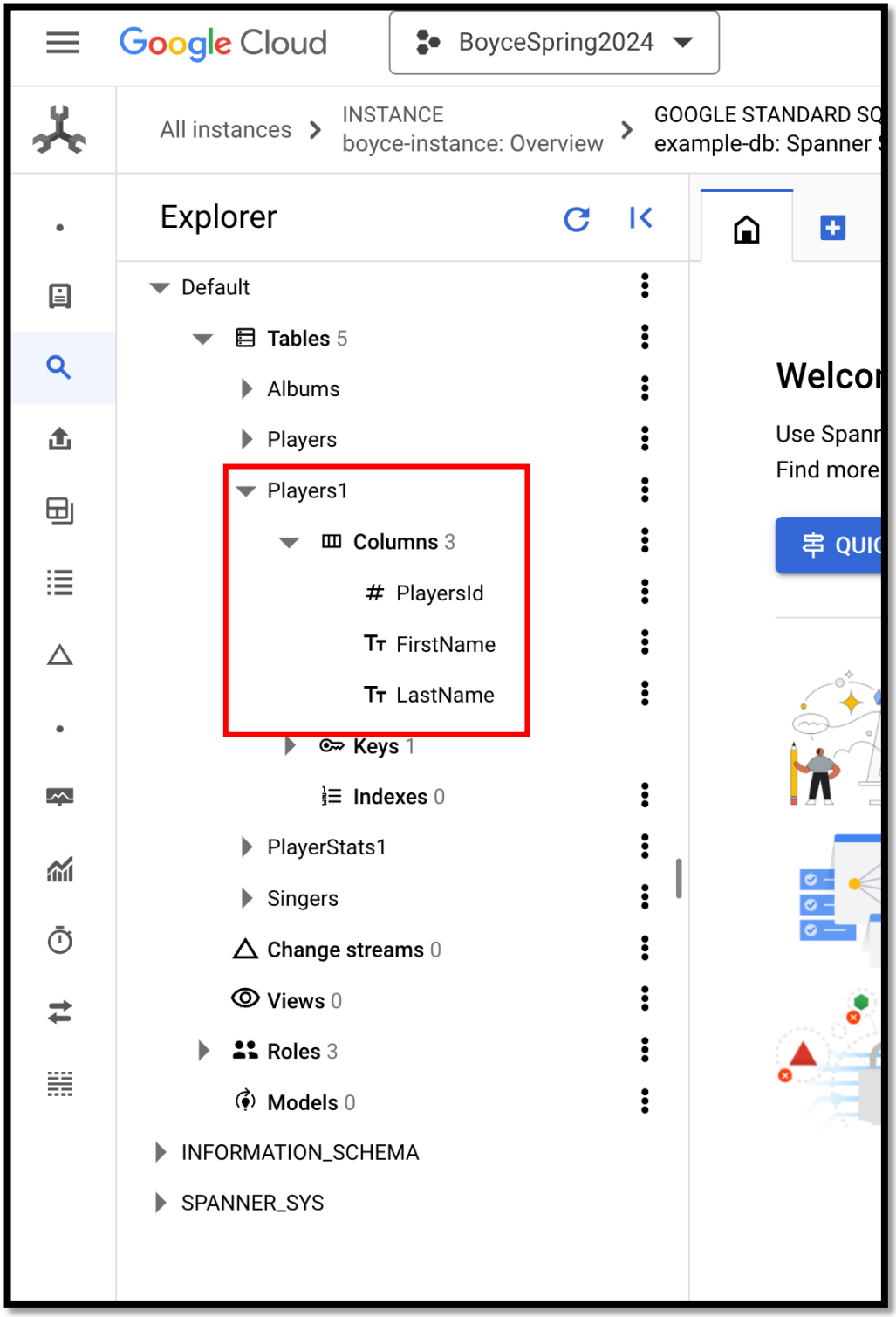
2. You can also type Spanner in the Search bar or navigate to it through the navigation pane (3 horizontal marks at the top left).
3. Once here, you will click on the name of your instance. Mine is Boyce Instance. You then want to click on your database (mine is example-db).



4. Once you click on your database, click on the magnifying glass in the navigation pane. When you scroll over it, it should read “Spanner Studio.” Click on Spanner Studio.

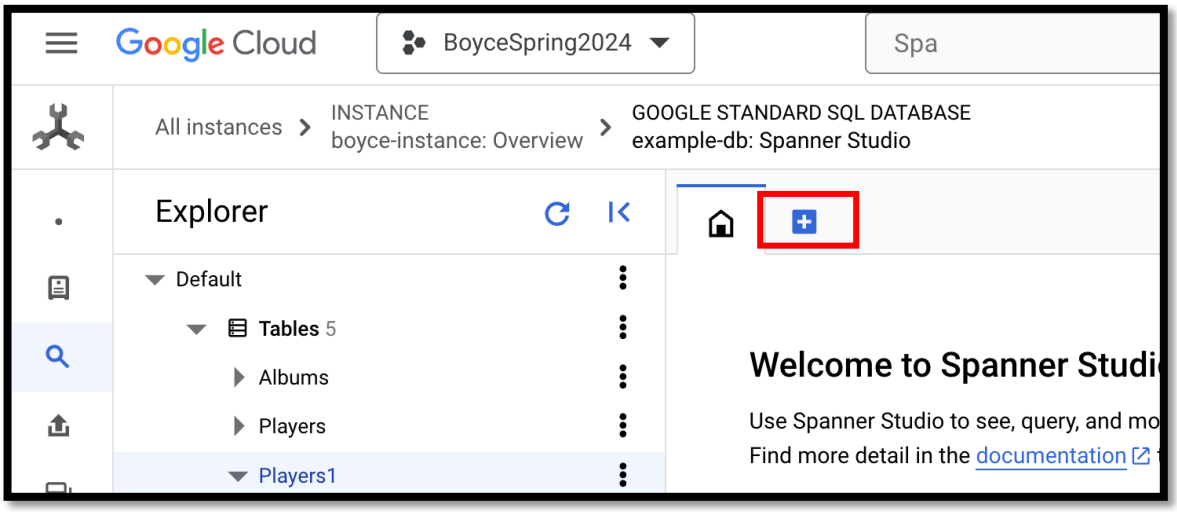


5. Click on the arrow next to Players1 and you will see the columns you created in the Players1 table: PlayersId, FirstName, and LastName.

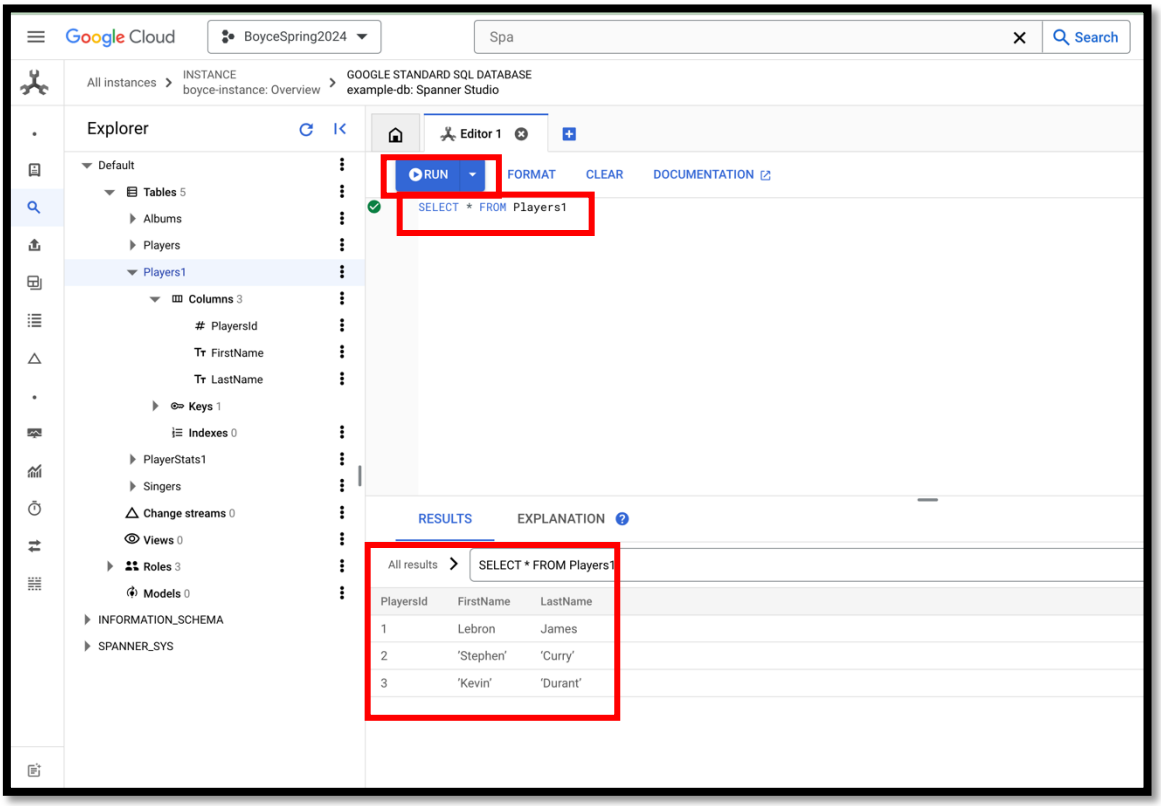


6. I encourage you to explore and see what you find. Maybe you want to go back and create more tables and more data.

7. Click on the “+” sign next to the home icon.



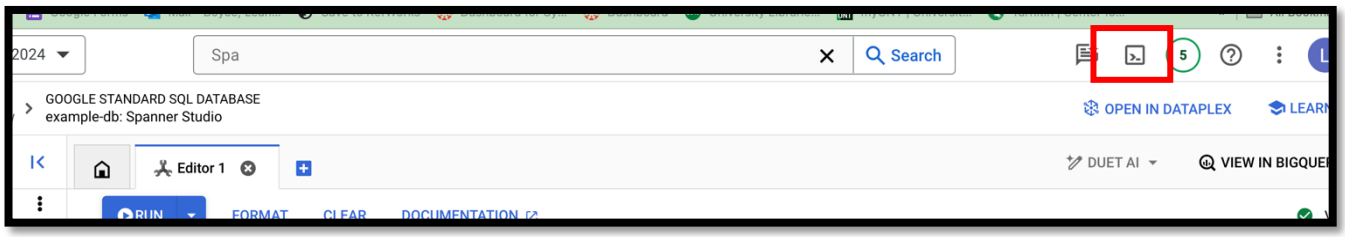
8. Type in the command: SELECT * FROM Players1
9. Hit Run
10. You will see the results of what is entered in the Players1 table.



Cleanup

To avoid incurring additional charges to your Google Cloud account for the resources used in this tutorial, drop the database and delete the instance that you created. There are many ways to accomplish this. I will stick with the Cloud Shell and then after the midterm, you will work more with the console.

1. Start by getting back into the Cloud Shell
2. Click on the "Activate Cloud Shell."



3. You will have to set the property again. Type in:

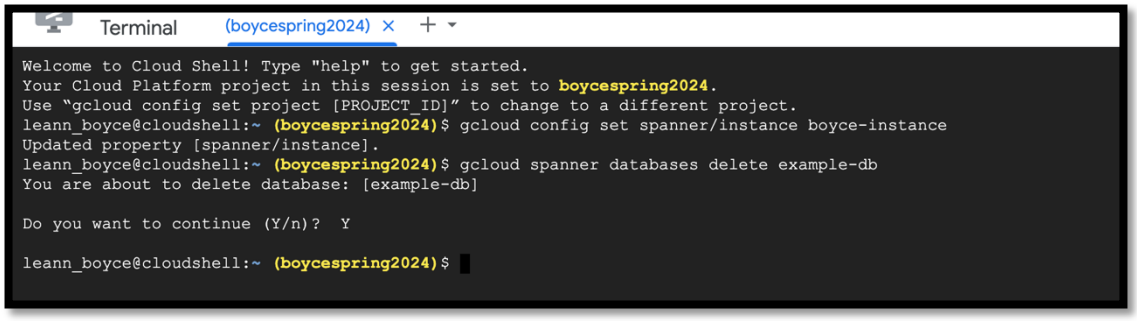
   gcloud config set spanner/instance boyce-instance

Drop a database

1. To delete an existing database, enter:

   gcloud spanner databases delete example-db

2. You will be asked if you want to delete the database. Enter Y. (This may take a little while.)
3. You might be asked if you authorize. If so, click on Authorize.



Delete an instance

To delete an existing instance, enter:

gcloud spanner instances delete boyce-instance

Note that deleting an instance also drops all of the databases in that instance. Deleting an instance is not reversible. Also note, you can always go back to make sure your instance was deleted (both Spanner & SQL).

Information obtained from (with some modification):
https://cloud.google.com/spanner/docs/getting-started/gcloud