Cloud Spanner - Database Fundamentals

Task 1. Create an instance

1. The first step in using Cloud Spanner is to create an instance. An instance is an allocation of Google Cloud compute and storage resources. From the Console, open the navigation menu ( ) > **View All Products**. Under **Databases** section, click **Spanner**.
2. Accept any acknowledgement or information window that may appear.
3. Then click **Create a Provisioned Instance**.
4. Fill in the following fields, leave the remainder with the default values:

Item	Value
Select an edition	Enterprise
Instance Name	banking-instance
Select a configuration	____
Configure compute capacity	Unit - Nodes // Quantity - 1

5. Click **Create**. Now you can see your instance on the Instance Details page. Here you have an overview of how the instance is performing, utilization, etc.. The next step is to create a database.

Create an instance

- Select an edition
  Enterprise
- 2 Name your instance
  banking-instance
- 3 Configure your instance
- 4 Allocate compute capacity

Name your instance

An instance has both a **name** and an **ID**. The name is for display purposes only. The ID is a permanent and unique identifier.

Instance name *
banking-instance
Instance name

Instance ID *
banking-instance
Lowercase letters, numbers, hyphens allowed

BACK CONTINUE

Create an instance

- Select an edition
  Enterprise
- Name your instance
  banking-instance
- 3 Configure your instance
  us-east1 (South Carolina)
- 4 Allocate compute capacity

Choose a configuration

Determines location of nodes and data. A multi-region configuration provides higher availability and enables your application to achieve faster reads in multiple locations. Configuration choice affects cost, performance, and replication. Learn more

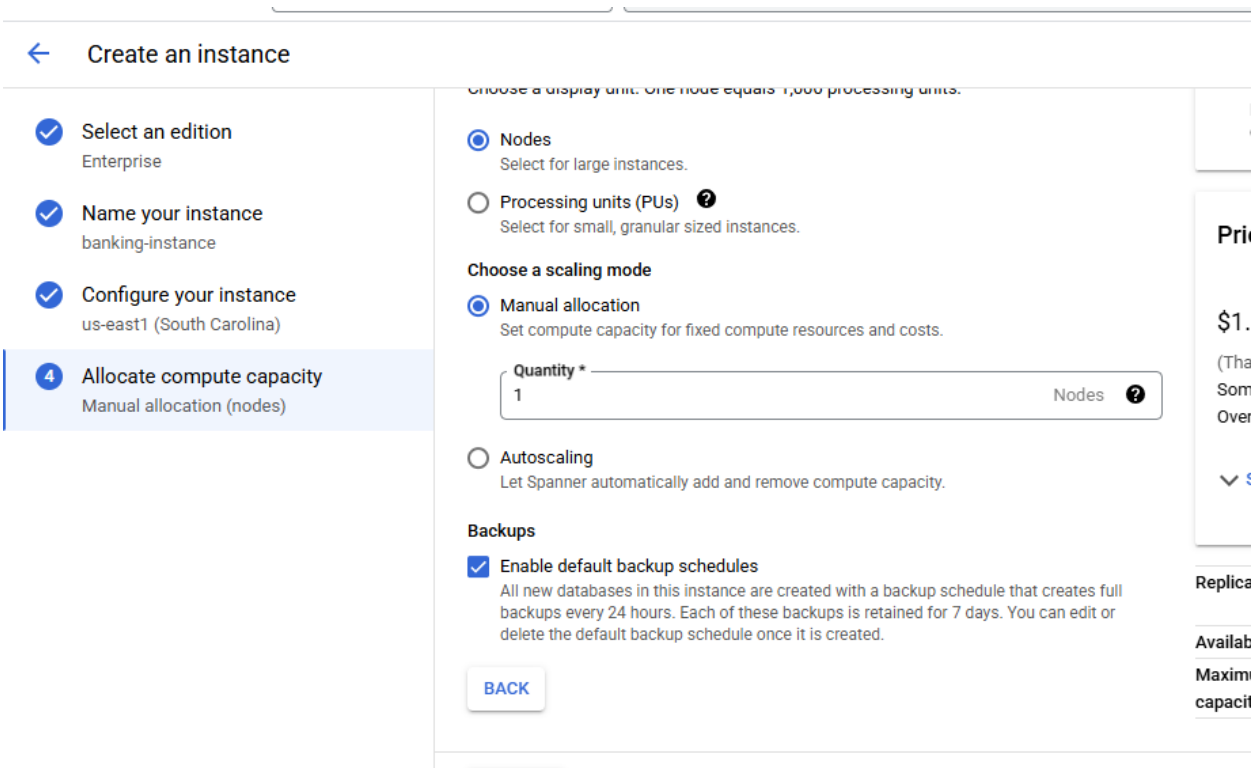
COMPARE REGION CONFIGURATIONS

- (●) Regional
  99.99% availability SLA, lower write latencies within region
- ( ) Dual-region
  99.999% availability SLA, from across two regions
- ( ) Multi-region
  99.999% availability SLA, from multi-geographic regions

Select a configuration *
us-east1 (South Carolina)

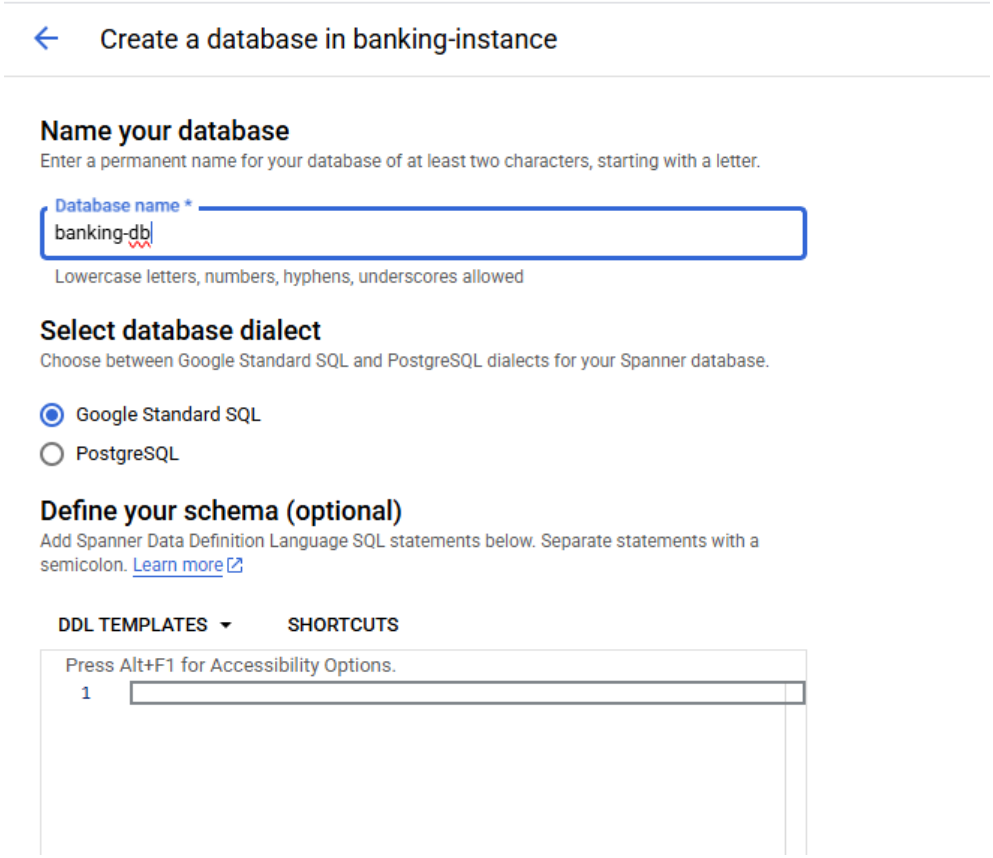
To add read-only replicas to this base configuration, create a new configuration with Cloud Shell. Learn more

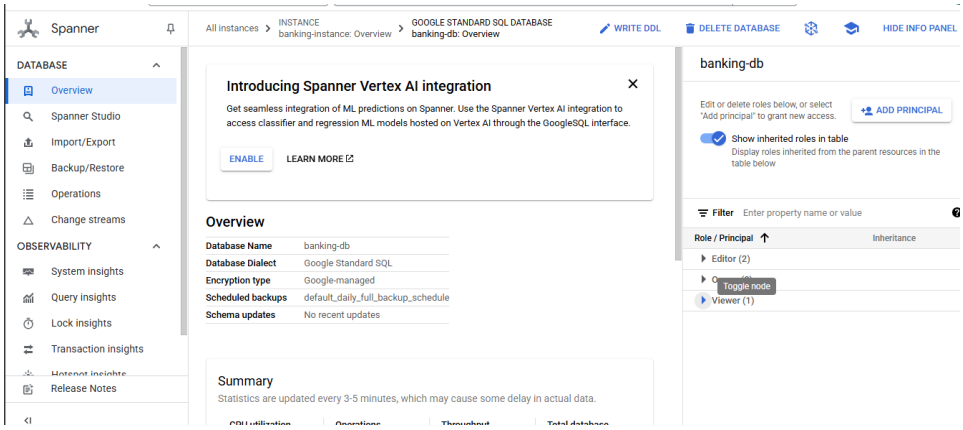
BACK CONTINUE



Task 2. Create a database

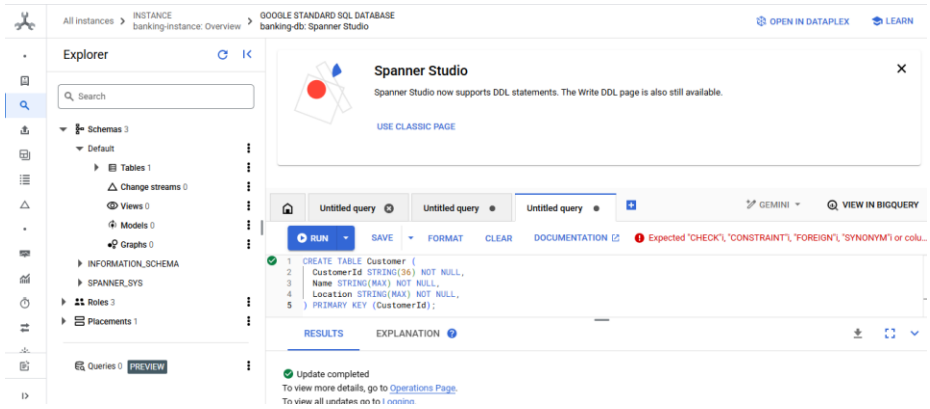
1. From the instance details page, click **Create database**.
2. For the database name, enter **banking-db**.
3. Skip the **Define your schema (optional)** step for now. You'll define your schema in the next section.
4. Click **Create**.
5. You're now on the **Overview** page for the new database you created. You can see that the page is similar to the Instance one, but the statistics refer to the specific database. Also note the new options on the left menu.
6. Click **Check my progress** to verify the objective.





Task 3. Create a table in your database

1. On the Database Details page for your **banking-db** database, scroll down the page and click **Create table**.
2. Click the blue **+** icon to open the **Query** page, enter:



3. Click **Run**.

4. When the operation is complete, click **Overview** under **Database** in the left menu. Then scroll down to **Tables** and click **Customer** to see the schema details

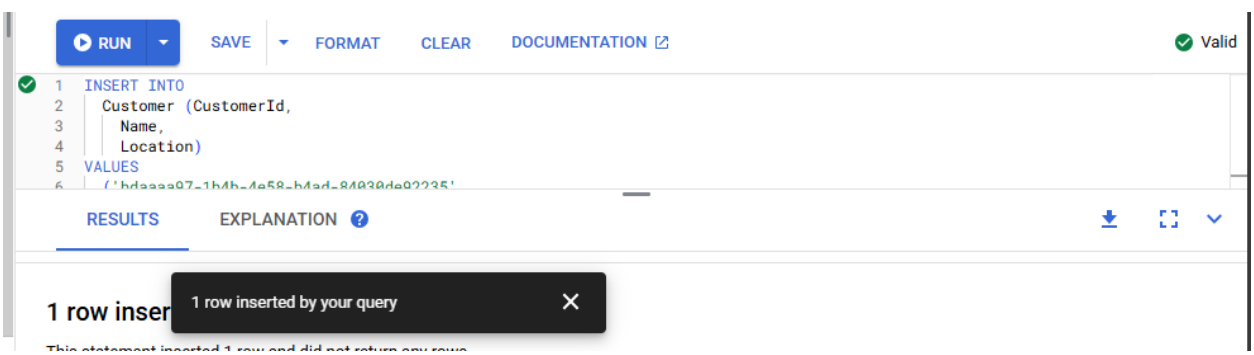
Task 4. Insert and modify data

The Cloud Console provides an interface for inserting, editing, and deleting data.

Insert data

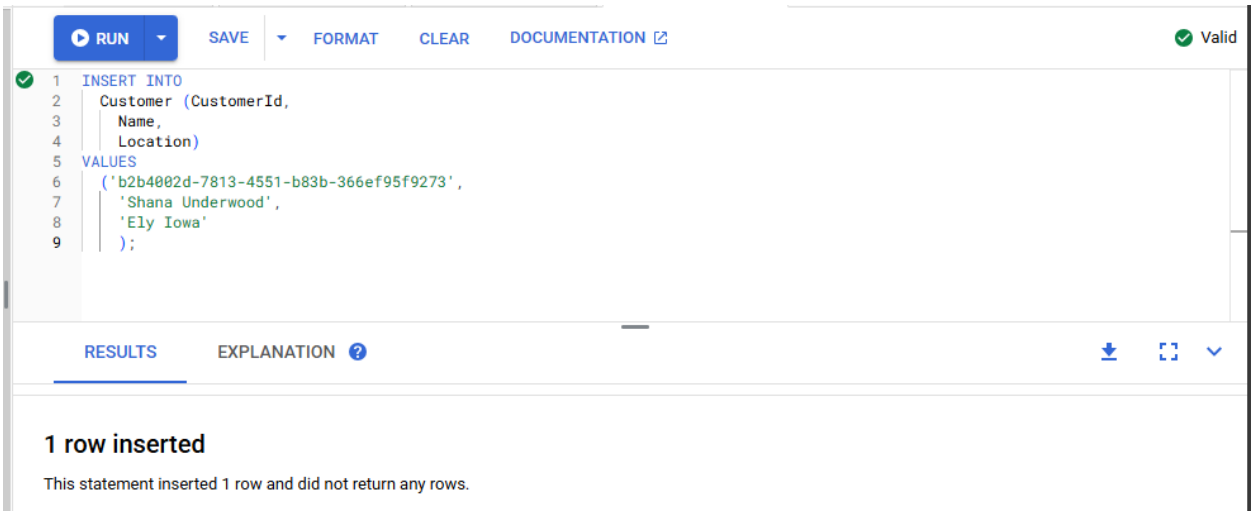
1. While on the **Schema** page, click **Data** in the left menu. Then click **Insert**.

2. This takes you to the **Query** tab of the **Spanner Studio** automatically. Click **Clear Query**, paste the query below, and click **Run**:



3. The lower page of the screen shows the result. The **Customer** table now has one row.

4. Add a second row. Replace the previous statement with the following, and click **Run**:

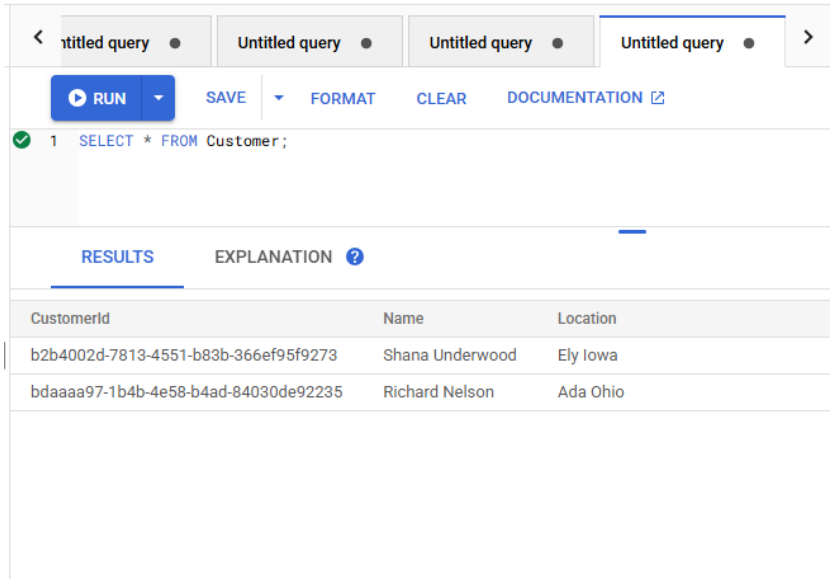


Run a query

1. You can execute a SQL statement on the query page of your database.
2. In the left pane of the Cloud Platform Console, click **Spanner Studio** to navigate to the Query UI window.
3. Click the blue **+** icon to open the **Query** page. Click **Clear Query**, paste the query below, and click Run:

SELECT * FROM Customer;

4. Click **Run**.
5. The Cloud Console displays the result of your query.



ask 5. Use the Google Cloud CLI with Cloud Spanner

The Cloud Console is very useful, but in some use cases you want to manage Spanner instances using other methods. Google Cloud services can also be managed through the command line tool named **gcloud**. The easiest way to use the **gcloud** CLI is via the Cloud Shell but it can also be installed on a wide variety of operating systems.

Create an instance with CLI

1. Creating a Spanner instance via **gcloud** is very simple. The core command is as follows:

```
gcloud spanner instances create [INSTANCE-ID] \
--config=[INSTANCE-CONFIG] \
--description="[INSTANCE-NAME]" \
--nodes=[NODE-COUNT]
```

2. In the Cloud Shell, create a new Cloud Spanner using the command below.

```
student_04_537aff948af7@cloudshell:~ (qwiklabs-gcp-00-f30df53fe800)$ gcloud spanner instances create banking-instance-2 \
--config=regional-us-east1  \
--description="Banking Instance 2" \
--nodes=2
Creating instance...done.
student_04_537aff948af7@cloudshell:~ (qwiklabs-gcp-00-f30df53fe800)$
```

Listing instances

1. You can run the following command to list the Spanner instances available in your project.

gcloud spanner instances list

```
student_04_537aff948af7@cloudshell:~ (qwiklabs-gcp-00-f30df53fe800)$ gcloud spanner instances list
NAME: banking-instance
DISPLAY_NAME: banking-instance
CONFIG: regional-us-east1
NODE_COUNT: 1
PROCESSING_UNITS: 1000
STATE: READY
INSTANCE_TYPE: PROVISIONED

NAME: banking-instance-2
DISPLAY_NAME: Banking Instance 2
CONFIG: regional-us-east1
NODE_COUNT: 2
PROCESSING_UNITS: 2000
STATE: READY
INSTANCE_TYPE: PROVISIONED
```

Creating a database

1. You can also create databases in a Spanner instance using **gcloud**.
2. In the Cloud Shell, create a new database using the command below.

gcloud spanner databases create banking-db-2 --instance=banking-instance-2

```
tudent_04_537aff948af7@cloudshell:~ (qwiklabs-gcp-00-f30df53fe800)$ gcloud spanner databases create banking-db-2 --instance=banking-instance-2
reating database...done.
tudent_04_537aff948af7@cloudshell:~ (qwiklabs-gcp-00-f30df53fe800)$
```

Modifying number of nodes

Remember that it is important to provision enough nodes to keep CPU utilization and storage utilization below the recommended maximum values. However, sometimes it is necessary to reduce the number of nodes.

1. You are now going to reduce the number of nodes of the instance **banking-instance-2** from two to one.
2. Use the following gcloud command to adjust the instance:

gcloud spanner instances update banking-instance-2 --nodes=1

```
student_04_537aff948af7@cloudshell:~ (qwiklabs-gcp-00-f30df53fe800)$ gcloud spanner instances update banking-instance-2 --nodes=1
Updating instance...done.
student_04_537aff948af7@cloudshell:~ (qwiklabs-gcp-00-f30df53fe800)$ gcloud spanner instances list
NAME: banking-instance
DISPLAY_NAME: banking-instance
CONFIG: regional-us-east1
NODE_COUNT: 1
PROCESSING_UNITS: 1000
STATE: READY
INSTANCE_TYPE: PROVISIONED

NAME: banking-instance-2
DISPLAY_NAME: Banking Instance 2
CONFIG: regional-us-east1
NODE_COUNT: 1
PROCESSING_UNITS: 1000
STATE: READY
INSTANCE_TYPE: PROVISIONED
student_04_537aff948af7@cloudshell:~ (qwiklabs-gcp-00-f30df53fe800)$
```

3. After completion, check that the number of nodes has been reduced:

gcloud spanner instances list

Task 6. Use Automation Tools with Cloud Spanner

Verify Terraform installation

1. Terraform comes pre-installed in the Cloud Shell. Using the previous Cloud Shell (or open it again if you closed it)

   terraform –version

```
tudent_04_537aff948af7@cloudshell:~ (qwiklabs-gcp-00-f30df53fe800)$ terraform -version
erraform v1.5.7
n linux_amd64

our version of Terraform is out of date! The latest version
s 1.12.2. You can update by downloading from https://www.terraform.io/downloads.html
tudent_04_537aff948af7@cloudshell:~ (qwiklabs-gcp-00-f30df53fe800)$
```

Create Terraform Configuration

1. In the cloud shell enter the following command to invoke the **Nano** text editor and create a new empty configuration file named **spanner.tf**.

nano spanner.tf

2. In the **Nano** editor, paste the code block listed below.

```
CLOUD SHELL
Terminal (qwiklabs-gcp-00-f30df53fe800)
  GNU nano 7.2                                                    spanner.tf
resource "google_spanner_instance" "banking-instance-3" {
  name         = "banking-instance-3"
  config       = "regional-us-east1"
  display_name = "Banking Instance 3"
  num_nodes    = 2
  labels = {
  }
}
```

3. Press **Ctrl+X** to exit Nano, **Y** to confirm the update, and press **Enter** to save your changes.

Deploy

1. The next step is to make sure all the Terraform service providers are available (in this case, the Spanner service provider). For that, run the following command in the Cloud Shell:

terraform init

```
CLOUD SHELL
Terminal (qwiklabs-gcp-00-f30df53fe800)
Initializing provider plugins...
- Finding latest version of hashicorp/google...
- Installing hashicorp/google v6.41.0...
- Installed hashicorp/google v6.41.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

2. Next instruct Terraform to create an execution plan that is based on the configuration file that you created a few steps ago. Run the following command:

terraform plan

```
      }
    + force_destroy                = false
    + id                           = (known after apply)
    + instance_type                = (known after apply)
    + name                         = "banking-instance-3"
    + num_nodes                    = 2
    + processing_units             = (known after apply)
    + project                      = "qwiklabs-gcp-00-f30df53fe800"
    + state                        = (known after apply)
    + terraform_labels             = {
      + "goog-terraform-provisioned" = "true"
      }
  }

Plan: 1 to add, 0 to change, 0 to destroy.


Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
student_04_537aff948af7@cloudshell:~ (qwiklabs-gcp-00-f30df53fe800)$
```

3. The output show details about the new instance that will be created. Run the following command to apply the plan to your project:

terraform apply

```
    + force_destroy                = false
    + id                           = (known after apply)
    + instance_type                = (known after apply)
    + name                         = "banking-instance-3"
    + num_nodes                    = 2
    + processing_units             = (known after apply)
    + project                      = "qwiklabs-gcp-00-f30df53fe800"
    + state                        = (known after apply)
    + terraform_labels             = {
      + "goog-terraform-provisioned" = "true"
      }
  }

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value:
```

4. The plan will be displayed again and Terraform will pause for approval to continue. Type **yes** and Terraform will create the new instance.

Task 7. Deleting instances

1. A very quick way to delete an instance is using the CLI. Run the following command:

gcloud spanner instances delete banking-instance-2

2. To confirm that **banking-instance-2** was deleted run the following command:

gcloud spanner instances list