

## 02 BigTable Demo -KirkYagami

### Table 1: User Profile Data Table

This table will store user profile information, including both personal details and activity data. We'll use two column families: `cf_personal` and `cf_activity`.

#### Example Schema for User Profile Data:

```
Row Key: user_id
Column Families:
  - cf_personal:
    - name
    - email
    - birthdate
  - cf_activity:
    - last_login
    - total_logins
    - account_status
```

- ◆ **Row Key:** `user_id`, which uniquely identifies each user.
- ◆ **Column Family 1 (`cf_personal`):** Stores personal information.
  - ◆ `name`: User's full name.
  - ◆ `email`: User's email address.
  - ◆ `birthdate`: User's date of birth.
- ◆ **Column Family 2 (`cf_activity`):** Stores user activity information.
  - ◆ `last_login`: Timestamp of the user's last login.
  - ◆ `total_logins`: Total number of times the user has logged in.
  - ◆ `account_status`: Current status of the user's account (e.g., active, suspended).

### Table 2: Product Catalog Data Table

This table will store data related to products in an e-commerce platform, divided into descriptive and pricing information. We'll use two column families: `cf_description` and `cf_pricing`.

#### Example Schema for Product Catalog Data:

```
Row Key: product_id
Column Families:
  - cf_description:
    - name
```

- category
- brand
- description
- cf\_pricing:
  - price
  - discount
  - availability
  - last\_updated

- ◆ **Row Key:** `product_id`, which uniquely identifies each product.
- ◆ **Column Family 1 (`cf_description`):** Stores product description details.
  - ◆ `name`: Name of the product.
  - ◆ `category`: Category under which the product is listed.
  - ◆ `brand`: Brand or manufacturer of the product.
  - ◆ `description`: Detailed description of the product.
- ◆ **Column Family 2 (`cf_pricing`):** Stores pricing and availability details.
  - ◆ `price`: Current price of the product.
  - ◆ `discount`: Any discount applied to the product.
  - ◆ `availability`: Stock availability (e.g., in stock, out of stock).
  - ◆ `last_updated`: Timestamp of the last update to the pricing or availability.

## Step-03: Create Tables in Bigtable Instance

---

### Table 1: User Profile Data Table

---

1. Navigate to Bigtable -> `mybigtableins1` -> Instance -> Tables -> CREATE TABLE.
2. Table ID: `user_profiles`
3. Add Column Families:
  - ◆ Column Family 1 Name: `cf_personal`
  - ◆ Column Family 2 Name: `cf_activity`
4. Click on CREATE.

### Table 2: Product Catalog Data Table

---

1. Navigate to Bigtable -> `mybigtableins1` -> Instance -> Tables -> CREATE TABLE.
2. Table ID: `product_catalog`
3. Add Column Families:
  - ◆ Column Family 1 Name: `cf_description`
  - ◆ Column Family 2 Name: `cf_pricing`
4. Click on CREATE.

### Using `cbt` CLI (Optional):

---

For more advanced users, the `cbt` CLI can be used to create these tables with multiple column families:

### 1. Create the User Profile Data Table:

```
cbt createtable user_profiles
cbt createfamily user_profiles cf_personal
cbt createfamily user_profiles cf_activity
```

### 2. Create the Product Catalog Data Table:

```
cbt createtable product_catalog
cbt createfamily product_catalog cf_description
cbt createfamily product_catalog cf_pricing
```

## Notes:

---

- ◆ **Column Families** allow you to logically group related data, which can improve the performance of read and write operations in Bigtable.
- ◆ The **Row Key** design is essential for ensuring even distribution of data across Bigtable nodes, avoiding performance bottlenecks.
- ◆ Using multiple column families can help in organizing data and optimizing queries, especially when different types of data are accessed with different frequencies.

By following these steps, you'll be able to create two tables that showcase the use of multiple column families in Google Cloud Bigtable, suitable for handling user profiles and product catalog data in an efficient and scalable manner.

## Cloud BigTable - Basics

---

### Step-01: Introduction

---

- ◆ Web Console
  - ◆ Create Big Table Instance
  - ◆ Create Table
- ◆ `cbt cli`

### Step-02: Create BigTable Instance

---

- ◆ Go to BigTable -> CREATE INSTANCE
- ◆ Name your Instance
  - ◆ Instance Name: mybigtableins1
  - ◆ Instance ID: mybigtableins1

- ◆ Select your Storage type: SSD
- ◆ Configure your first cluster
  - ◆ Select a cluster ID: mybigtableins1-c1
  - ◆ Select Location: europe-west10(Berlin)
  - ◆ Zone: Any
- ◆ Choose node scaling mode
  - ◆ Autoscaling Minimum: 1
  - ◆ Autoscaling Maximum: 3
  - ◆ CPU utilization target: 60%
  - ◆ Storage utilization target: 2.5%
- ◆ REST ALL LEAVE TO DEFAULTS

## Step-03: Create Table in BigTable Instance

---

- ◆ Go to BigTable -> mybigtableins1 -> Instance -> Tables -> CREATE TABLE
- ◆ Table ID: mytable1
- ◆ Add Column Family
  - ◆ Column Family Name: cf1
- ◆ Click on **CREATE**

## Step-04: cbt cli: Create BigTable Instance, Table using cbt cli

---

- ◆ [cbt reference](#) 

```
# Set Project
gcloud config set project PROJECT_ID
gcloud config set project gcplearn9

# Install cbt cli
gcloud components install cbt
[or]
sudo apt-get install google-cloud-sdk-cbt

# List Instances
cbt listinstances

# Setup Project and BigTable Instance for cbt cli
echo project = PROJECT_ID >> ~/.cbtrc && echo instance = INSTANCE_NAME >>
~/.cbtrc
echo project = gcplearn9 >> ~/.cbtrc && echo instance = mybigtableins1 >>
~/.cbtrc
cat $HOME/.cbtrc

# Create Table (Optional)
cbt createtable mytable1
```

```
# List Tables
cbt ls

# Add one column family (Optional)
cbt createfamily mytable1 cf1

# List column families
cbt ls mytable1

# Write the values test-value1 and test-value2 to the row r1, using the
column family cf1 and the column qualifier c1:
cbt set mytable1 r1 cf1:c1=test-value1
cbt set mytable1 r1 cf1:c1=test-value2
cbt set mytable1 r1 cf1:c1=test-value3

# Read the data you added to the table
cbt read mytable1
Important Note: In this demo you set only three cells, but Bigtable lets you
set up to 10,000 cells in a single write request.

# Delete Table (DONT DELETE)
cbt deletetable mytable1

# Delete BigTable Instance (DONT DELETE)
cbt deleteinstance mybigtableins1
```

## Additional References

---

- ◆ <https://cloud.google.com/bigtable/docs/cbt-reference> 
- ◆ <https://cloud.google.com/bigtable/docs/create-instance-write-data-cbt-cli> 