Gemini - Create column description

Overview

This guide walks you through setting up the Gemini AI Column Description Generator in a new Google Cloud Platform (GCP) project. This tool automatically generates descriptions for database columns using Google's Gemini AI model integrated with BigQuery ML.

Step 1: Enable Required APIs

The following APIs need to be activated

1. **BigQuery API** - For data processing and ML operations
2. **Vertex AI API** - For Gemini AI model access *gcloud services enable aiplatform.googleapis.com*
3. **Cloud Resource Manager API** - For project management

Step 2: Set Up IAM Roles and Permissions

Required Roles for Script Execution

The service account or user running the script needs the following roles:

BigQuery Roles

- **BigQuery Data Editor** (`roles/bigquery.dataEditor`)
  - Create and modify datasets and tables
  - Execute queries and manage temporary tables
- **BigQuery Job User** (`roles/bigquery.jobUser`)
  - Run BigQuery jobs and queries
- **BigQuery Connection User** (`roles/bigquery.connectionUser`)
  - Use BigQuery connections for ML models

Vertex AI Roles

- **Vertex AI User** (`roles/aiplatform.user`)
  - Access Vertex AI services and models
- **ML Developer** (`roles/ml.developer`)
  - Use ML models in BigQuery ML

Assign Roles via Console

1. Go to IAM & Admin > IAM
2. Find your user account or service account
3. Click "Edit principal"
4. Add each required role listed above

Assign Roles via gcloud CLI

```
# Replace [USER_EMAIL] with the actual email
export USER_EMAIL="your-email@domain.com"
export PROJECT_ID="your-project-id"

gcloud projects add-iam-policy-binding $PROJECT_ID \
    --member="user:$USER_EMAIL" \
    --role="roles/bigquery.dataEditor"

gcloud projects add-iam-policy-binding $PROJECT_ID \
    --member="user:$USER_EMAIL" \
    --role="roles/bigquery.jobUser"

gcloud projects add-iam-policy-binding $PROJECT_ID \
    --member="user:$USER_EMAIL" \
    --role="roles/bigquery.connectionUser"

gcloud projects add-iam-policy-binding $PROJECT_ID \
    --member="user:$USER_EMAIL" \
    --role="roles/aiplatform.user"

gcloud projects add-iam-policy-binding $PROJECT_ID \
    --member="user:$USER_EMAIL" \
    --role="roles/ml.developer"
```

Step 3: Create BigQuery Datasets

Create Source Dataset

Create a dataset to hold your source tables (if not already existing):

```
CREATE SCHEMA IF NOT EXISTS `your-project-id.DM`
OPTIONS (
  description = "Data Mart - Source tables for analysis",
  location = "US"  -- Choose appropriate location
);
```

Create Output Dataset

Create a dataset for storing the AI-generated descriptions:

```
CREATE SCHEMA IF NOT EXISTS `your-project-id.TEST_GEMINI`
OPTIONS (
  description = "Gemini AI generated column descriptions",
  location = "US"  -- Must match source dataset location
);
```

Step 4: Set Up Vertex AI Connection

Create Vertex AI Connection in BigQuery (documentation)

1. **Via BigQuery Console:**
2.
    - Go to BigQuery in Google Cloud Console
    - Click **+ Add data**, and then use the search bar for data sources to search for **Vertex AI**. Click on the result for **Vertex AI** then click **Vertex AI Models: BigQuery Federation**.
    - Select **Connection type** as `Vertex AI remote models, remote functions and BigLake (Cloud Resource)` if not already selected and set **Connection ID** to `vertex-ai-connection`.

Get Connection Service Account

After creating the connection, note the service account email that's automatically created. You'll need to grant it permissions.

```
-- Get connection details
SELECT * FROM `your-project-id.us-central1.INFORMATION_SCHEMA.CONNECTIONS`
WHERE connection_name = 'vertex-ai-connection';
```

Grant Vertex AI Access to Connection Service Account

The connection service account needs access to Vertex AI:

```
# Get the service account email from the connection details above
export CONNECTION_SA="service-account-email@gcp-sa-bigquery-condel.iam.gserviceaccount.com"

gcloud projects add-iam-policy-binding $PROJECT_ID \
    --member="serviceAccount:$CONNECTION_SA" \
    --role="roles/aiplatform.user"
```

Step 5: Create Gemini AI Model in BigQuery

Create Remote Model

```
CREATE OR REPLACE MODEL `your-project-id.YOUR_DATASET.gemini_remote_model`
REMOTE WITH CONNECTION `your-project-id.europe-west1.vertex-ai-connection`
OPTIONS (
  ENDPOINT = 'gemini-2.0-flash'
);
```

Test Model Connection

```
-- Test the model with a simple query
SELECT ml_generate_text_llm_result
FROM ML.GENERATE_TEXT(
  MODEL `your-project-id.TEST_GEMINI.gemini_remote_model`,
  (SELECT "Test connection to Gemini AI" AS prompt),
  STRUCT(TRUE AS flatten_json_output)
);
```

Step 6: Configure and Deploy the Script

Update Configuration Variables

In the main script, update these variables with your project details:

```
-- Update these variables in the script
DECLARE dataset_name STRING DEFAULT 'your-project-id.DM';
DECLARE output_dataset STRING DEFAULT 'TEST_GEMINI';
DECLARE model_name STRING DEFAULT 'TEST_GEMINI.gemini_remote_model';
DECLARE sample_size INT64 DEFAULT 5;
```

Step 7: Execute the Script

1. Open BigQuery Console
2. Paste the complete script in the query editor: https://gitlab.syensqo.com/syensqo-connected-research/toolkit/toolkit/-/blob/master/SQL/gemini_column_description.sql?ref_type=heads
3. Update the configuration variables at the top
4. Run the script

Step 8: Verify Installation

Check Results

```
-- Verify the output table was created
SELECT
  table_name,
  COUNT(*) as column_count
FROM `your-project-id.TEST_GEMINI.column_descriptions_with_ai`
GROUP BY table_name;

-- View sample results
SELECT * FROM `your-project-id.TEST_GEMINI.column_descriptions_with_ai` LIMIT 5;
```

Troubleshooting

Common Issues

1. **"Permission denied" errors**
    - Verify all required APIs are enabled
    - Check IAM roles are correctly assigned
    - Ensure connection service account has Vertex AI access
2. **"Model not found" errors**
    - Verify the Vertex AI connection was created successfully
    - Check the model name matches exactly
    - Ensure the connection region matches your dataset location
3. **"No tables found" errors**
    - Verify source dataset exists and contains tables
    - Check dataset names in the configuration variables
    - Ensure the service account has access to source datasets
4. **Quota exceeded errors**
    - Check Vertex AI quotas in the Cloud Console
    - Consider reducing sample_size or processing fewer tables at once

Documentation

For issues specific to:

- **Vertex AI:** Vertex AI Documentation
- Summarize Text using SQL and LLMs in BigQuery ML