**ALL SQL HOMEWORKS**

**USE ROLE sysadmin;**

**---------------------------------**

**-- CREATING WAREHOUSES & DATABASES**

**---------------------------------**

**-- Create a new WAREHOUSE for table creation and data loading.**

**-- Remember, WAREHOUSES process data. DATABASES store data.**

**CREATE OR REPLACE WAREHOUSE ghosts\_loading\_wh WITH**

**WAREHOUSE\_SIZE='SMALL'**

**AUTO\_RESUME = TRUE -- default**

**AUTO\_SUSPEND = 180 -- default (SECONDS)**

**INITIALLY\_SUSPENDED = TRUE; -- default**

**-- Create a new database called "investigation".**

**CREATE DATABASE IF NOT EXISTS investigation;**

**-- Set the CONTEXT. (i.e. Information about your environment.)**

**-- The PUBLIC schema is the default schema that all users have access to.**

**-- It is often used for development and testing.**

**USE investigation.public;**

**USE WAREHOUSE ghost\_loading\_wh;**

**-- Create the table, where trip data will be stored. (DDL is free.)**

**CREATE OR REPLACE TABLE crime\_details (**

**crime\_id integer,**

**victim\_id integer,**

**date date,**

**time time,**

**location string,**

**type string**

**);**

**CREATE OR REPLACE TABLE victim\_profiles (**

**victim\_id integer,**

**name string,**

**age integer,**

**occupation string,**

**address string,**

**district string**

**);**

**---------------------------------**

**-- CREATING STAGES**

**---------------------------------**

**-- Create a STAGE that specifies the location of the S3 bucket.**

**-- STAGES are locations where files are stored for loading or unloading.**

**-- This is public S3 bucket, so no credentials are required.**

**CREATE OR REPLACE STAGE investigation1026**

**url = 's3://investigation-1026/';**

**-- List the files in the stage. (It does NOT show delimiters.)**

**LIST @investigation1026;**

**-- Review the properties of the stage. (Delimiters are noted.)**

**DESCRIBE STAGE investigation1026;**

**-- Create a FILE FORMAT that matches the data structure.**

**-- It describes what the staged files look like.**

**-- This statement includes several default values, just as an example.**

**CREATE OR REPLACE FILE FORMAT ghosts\_csv**

**type='csv'**

**compression = 'auto' -- compression checked automatically (default)**

**field\_delimiter = ',' -- columns separated by commas (default)**

**record\_delimiter = '\n' -- rows separated by newlines (default)**

**skip\_header = 1 -- does not skip lines in the file (default)**

**trim\_space = true -- does not remove white space from fields (default)**

**null\_if = ('-') -- replace these values with SQL NULL**

**date\_format = 'auto' -- date checked automatically (default)**

**timestamp\_format = 'auto' -- time checked automatically (default)**

**field\_optionally\_enclosed\_by = '\042' -- strings closed by " " (octal)**

**comment = 'file format for investigation data'**

**;**

**---------------------------------**

**-- LOADING DATA**

**---------------------------------**

**-- Use a COPY COMMAND to load staged data into the table.**

**COPY INTO crime\_details -- the destination table**

**FROM @investigation1026/crime\_details -- the source stage**

**file\_format = ghosts\_csv -- the file format**

**; -- the filenames to match**

**-- Use a COPY COMMAND to load staged data into the table.**

**COPY INTO victim\_profiles -- the destination table**

**FROM @investigation1026/victim\_profiles -- the source stage**

**file\_format = ghosts\_csv -- the file format**

**; -- the filenames to match**

**---------------------------------**

**-- ANALYZING DATA**

**---------------------------------**

**-- Normally, analytics users have a different role than "sysadmin",**

**-- Also, queries may be done in other tools, like PowerBI.**

**-- Create a new warehouse for analysis (queries).**

**-- This warehouse separation avoids interference and allows for monitoring.**

**-- For example, "How much are we spending on data loading?"**

**CREATE OR REPLACE WAREHOUSE ghosts\_query\_wh WITH**

**WAREHOUSE\_SIZE='SMALL'**

**AUTO\_RESUME = TRUE -- default**

**AUTO\_SUSPEND = 180 -- default**

**INITIALLY\_SUSPENDED = TRUE; -- default**

**-- Determine the date range of the crimes. What is the date of the first crime?**

**SELECT TO\_CHAR(MIN(date), 'MM/DD/YY') AS first\_crime\_date,**

**FROM crime\_details;**

**-- Count the rows in the crime details table. How many crimes are in the data?**

**SELECT COUNT(\*) AS total\_crimes**

**FROM crime\_details;**

**-- Count the number of crimes by type**

**SELECT type, COUNT(\*) AS crime\_count**

**FROM crime\_details**

**GROUP BY type**

**ORDER BY crime\_count DESC;**

**-- Count the number of crimes by location.**

**SELECT location, COUNT(\*) AS crime\_count**

**FROM crime\_details**

**GROUP BY location**

**ORDER BY crime\_count DESC;**

**-- Categorize and count crimes by time of day.**

**SELECT**

**CASE**

**WHEN time >= '06:00:00' AND time < '12:00:00' THEN 'Morning'**

**WHEN time >= '12:00:00' AND time < '18:00:00' THEN 'Afternoon'**

**WHEN time >= '18:00:00' AND time < '21:00:00' THEN 'Evening'**

**ELSE 'Night'**

**END AS time\_of\_day,**

**COUNT(\*) AS crime\_count**

**FROM crime\_details**

**GROUP BY time\_of\_day**

**ORDER BY crime\_count DESC;**

**-- Calculate the average age of homicide victims.**

**SELECT AVG(age) AS average\_age\_homicide\_victims**

**FROM victim\_profiles**

**WHERE victim\_id IN (**

**SELECT victim\_id**

**FROM crime\_details**

**WHERE type = 'Homicide'**

**);**

**-- List homicide victims and their occupations.**

**SELECT v.name, v.occupation**

**FROM victim\_profiles v**

**JOIN crime\_details c ON v.victim\_id = c.victim\_id**

**WHERE c.type = 'Homicide';**

**-- Select victims with one of the following occupations: AI Researcher, City Council Member, Civil Rights Lawyer, Community Organizer, Investigative Reporter.**

**SELECT v.name, v.occupation**

**FROM victim\_profiles v**

**JOIN crime\_details c ON v.victim\_id = c.victim\_id**

**WHERE c.type = 'Homicide'**

**AND v.occupation IN (**

**'AI Researcher',**

**'City Council Member',**

**'Civil Rights Lawyer',**

**'Community Organizer',**

**'Investigative Reporter'**

**);**

**---------------------------------**

**-- CREATING WAREHOUSES & DATABASES**

**---------------------------------**

**-- Create a new WAREHOUSE for table creation and data loading.**

**-- Remember, WAREHOUSES process data. DATABASES store data.**

**CREATE OR REPLACE WAREHOUSE ghosts\_loading\_wh WITH**

**WAREHOUSE\_SIZE='SMALL'**

**AUTO\_RESUME = TRUE -- default**

**AUTO\_SUSPEND = 180 -- default (SECONDS)**

**INITIALLY\_SUSPENDED = TRUE; -- default**

**-- Create a new database called "investigation".**

**CREATE DATABASE IF NOT EXISTS investigation;**

**-- Set the CONTEXT. (i.e. Information about your environment.)**

**-- The PUBLIC schema is the default schema that all users have access to.**

**-- It is often used for development and testing.**

**USE investigation.public;**

**USE WAREHOUSE ghost\_loading\_wh;**

**-- Create the table, where trip data will be stored. (DDL is free.)**

**CREATE OR REPLACE TABLE forum\_activities (**

**post\_id integer,**

**user\_id integer,**

**user\_ip\_address string,**

**post\_title string,**

**post\_category string,**

**post\_date date**

**);**

**CREATE OR REPLACE TABLE city\_officials (**

**official\_id integer,**

**name string,**

**position string,**

**department string,**

**office\_location string,**

**tenure\_start date,**

**ip\_address string**

**);**

**---------------------------------**

**-- CREATING STAGES**

**---------------------------------**

**-- Create a STAGE that specifies the location of the S3 bucket.**

**-- STAGES are locations where files are stored for loading or unloading.**

**-- This is public S3 bucket, so no credentials are required.**

**CREATE OR REPLACE STAGE investigation2134**

**url = 's3://investigation-2134/';**

**-- List the files in the stage. (It does NOT show delimiters.)**

**LIST @investigation2134;**

**-- Review the properties of the stage. (Delimiters are noted.)**

**DESCRIBE STAGE investigation2134;**

**-- Create a FILE FORMAT that matches the data structure.**

**-- It describes what the staged files look like.**

**CREATE OR REPLACE FILE FORMAT ghosts\_csv**

**type='csv'**

**compression = 'auto' -- compression checked automatically (default)**

**field\_delimiter = ',' -- columns separated by commas (default)**

**record\_delimiter = '\n' -- rows separated by newlines (default)**

**skip\_header = 1 -- does not skip lines in the file (default)**

**trim\_space = true -- does not remove white space from fields (default)**

**null\_if = ('-') -- replace these values with SQL NULL**

**date\_format = 'auto' -- date checked automatically (default)**

**timestamp\_format = 'auto' -- time checked automatically (default)**

**field\_optionally\_enclosed\_by = '\042' -- strings closed by " " (octal)**

**comment = 'file format for investigation data'**

**;**

**---------------------------------**

**-- LOADING DATA**

**---------------------------------**

**-- Use a COPY COMMAND to load staged data into the table.**

**COPY INTO forum\_activities -- the destination table**

**FROM @investigation2134/forum\_activities -- the source stage**

**file\_format = ghosts\_csv -- the file format**

**; -- the filenames to match**

**-- Use a COPY COMMAND to load staged data into the table.**

**COPY INTO city\_officials -- the destination table**

**FROM @investigation2134/city\_officials -- the source stage**

**file\_format = ghosts\_csv -- the file format**

**; -- the filenames to match**

**--------------------------------**

**--SWITCH TO QUERY WH**

**--------------------------------**

**CREATE OR REPLACE WAREHOUSE ghosts\_query\_wh WITH**

**WAREHOUSE\_SIZE='SMALL'**

**AUTO\_RESUME = TRUE -- default**

**AUTO\_SUSPEND = 180 -- default**

**INITIALLY\_SUSPENDED = TRUE;**

**-------------------------------**

**--WRITE QUERIES**

**-------------------------------**

**--1. Count the number of forum posts --360**

**SELECT COUNT(\*) AS total\_posts**

**FROM forum\_activities;**

**--2. What is the date of the earliest post? --2023-04-10**

**SELECT MIN(post\_date) AS earliest\_post\_date**

**FROM forum\_activities;**

**SELECT**

**MIN(post\_date) AS earliest\_post\_date,**

**MAX(post\_date) AS latest\_post\_date**

**FROM forum\_activities;**

**SELECT**

**MIN(date) AS earliest\_crime\_date,**

**MAX(date) AS latest\_crime\_date**

**FROM crime\_details;**

**--3. Count the posts and cumulative posts by year and month. (Store results in Excel.) -- By the end of September, how many posts had been made? -- 243**

**WITH post\_counts AS (**

**SELECT**

**YEAR(post\_date) AS post\_year,**

**MONTH(post\_date) AS post\_month,**

**COUNT(\*) AS post\_count**

**FROM forum\_activities**

**GROUP BY YEAR(post\_date), MONTH(post\_date)**

**ORDER BY post\_year, post\_month**

**)**

**SELECT**

**post\_year,**

**post\_month,**

**post\_count,**

**SUM(post\_count) OVER (ORDER BY post\_year, post\_month) AS cumulative\_posts**

**FROM post\_counts;**

**--4.Rank the top 10 most active users based on # of posts. What is the user ID for the most active user? 902842**

**WITH user\_post\_counts AS (**

**SELECT**

**user\_id,**

**COUNT(\*) AS post\_count**

**FROM forum\_activities**

**GROUP BY user\_id**

**)**

**SELECT**

**user\_id,**

**post\_count,**

**RANK() OVER (ORDER BY post\_count DESC) AS rank**

**FROM user\_post\_counts**

**ORDER BY rank**

**;**

**--5. Count the number of city officials. How many officials are in the data? --98**

**SELECT COUNT(\*) AS total\_officials**

**FROM city\_officials;**

**SELECT**

**department,**

**COUNT(\*) AS official\_count**

**FROM city\_officials**

**GROUP BY department**

**ORDER BY official\_count DESC;**

**--6. Calculate each city official's tenure in months. What is Tom Johnson's tenure in months? --104**

**SELECT**

**name,**

**tenure\_start,**

**DATEDIFF(MONTH, tenure\_start, '2025-01-01') AS tenure\_months**

**FROM city\_officials**

**ORDER BY tenure\_months DESC;**

**--7. Rank officials in each department by tenure (longest to shortest). Who is the longest tenured official in "smart grid planning"? --Amelia Scott**

**SELECT**

**department,**

**name,**

**tenure\_start,**

**DATEDIFF(MONTH, tenure\_start, '2025-01-01') AS tenure\_months,**

**RANK() OVER (PARTITION BY department ORDER BY tenure\_start ASC) AS tenure\_rank**

**FROM city\_officials**

**ORDER BY department, tenure\_rank;**

**--8. List the officials whose IP appears on the forum. How many officials were identified? --5**

**SELECT DISTINCT c.name, c.department, c.ip\_address**

**FROM city\_officials c**

**JOIN forum\_activities f**

**ON c.ip\_address = f.user\_ip\_address;**

**--9. Update the previous query to count posts by those officials. Which of the following officials had the most posts? -- Christine Kirkwood**

**SELECT**

**c.name,**

**c.department,**

**c.ip\_address,**

**COUNT(f.post\_id) AS post\_count**

**FROM city\_officials c**

**JOIN forum\_activities f**

**ON c.ip\_address = f.user\_ip\_address**

**GROUP BY c.name, c.department, c.ip\_address**

**ORDER BY post\_count DESC;**

**--10. List details for each post made by these officials. Which official wrote this? "Our AI systems are reliable and trustworthy. Critics are just technophobes." -- Omar Khan**

**SELECT**

**c.name AS official\_name,**

**c.department,**

**f.post\_category,**

**f.post\_title,**

**f.post\_date**

**FROM city\_officials c**

**JOIN forum\_activities f**

**ON c.ip\_address = f.user\_ip\_address**

**WHERE c.name IN ('Christen Kirkwood', 'Omar Khan', 'Amir Khan', 'Gregory Hayes', 'Linda Johnson')**

**ORDER BY f.post\_date DESC;**

**-------------------------------**

**--DROP TABLES & RESTORE**

**-------------------------------**

**UPDATE city\_officials SET position = 'AI Specialist';**

**SELECT \* FROM city\_officials LIMIT 10;**

**-- ROLLBACK TABLES**

**-- Option #1: Find the query ID in the console.**

**-- Option #2: Find the query ID using SQL.**

**SELECT \* FROM table(information\_schema.query\_history\_by\_session (result\_limit=>10));**

**-- Restore the table using the query ID.**

**CREATE OR REPLACE TABLE city\_officials AS**

**(SELECT \* FROM city\_officials before (statement => '01bad678-0002-82eb-0007-2e3e0009828e'));**

**-- Review (It should be restored.)**

**SELECT \* FROM city\_officials LIMIT 10;**

**--------------------**

**--CREATE A NEW ROLE**

**--------------------**

**CREATE ROLE junior\_detective;**

**GRANT ROLE junior\_detective TO USER MARCESHDZ;**

**GRANT USAGE ON DATABASE investigation TO ROLE junior\_detective;**

**GRANT USAGE ON SCHEMA investigation.public TO ROLE junior\_detective;**

**GRANT SELECT ON ALL TABLES IN SCHEMA investigation.public to ROLE junior\_detective;**

**GRANT OPERATE ON WAREHOUSE ghosts\_query\_wh TO ROLE junior\_detective;**

**SHOW ROLES LIKE 'junior\_detective';**

**---------------------------------**

**-- CREATING WAREHOUSES & DATABASES**

**---------------------------------**

**-- Create a new WAREHOUSE for table creation and data loading.**

**-- Remember, WAREHOUSES process data. DATABASES store data.**

**CREATE OR REPLACE WAREHOUSE ghosts\_loading\_wh WITH**

**WAREHOUSE\_SIZE='SMALL'**

**AUTO\_RESUME = TRUE -- default**

**AUTO\_SUSPEND = 180 -- default (SECONDS)**

**INITIALLY\_SUSPENDED = TRUE; -- default**

**-- Create a new database called "investigation".**

**CREATE DATABASE IF NOT EXISTS investigation;**

**-- Set the CONTEXT. (i.e. Information about your environment.)**

**-- The PUBLIC schema is the default schema that all users have access to.**

**-- It is often used for development and testing.**

**USE investigation.public;**

**USE WAREHOUSE ghost\_loading\_wh;**

**-- Create the table, where trip data will be stored. (DDL is free.)**

**CREATE OR REPLACE TABLE phone\_directory (**

**directory\_id integer,**

**phone\_number string,**

**name string,**

**address string,**

**district string**

**);**

**CREATE OR REPLACE TABLE call\_log (**

**call\_id integer,**

**caller\_id integer,**

**receiver\_id integer,**

**call\_duration integer,**

**call\_start\_date date,**

**call\_status string**

**);**

**---------------------------------**

**-- CREATING STAGES**

**---------------------------------**

**-- Create a STAGE that specifies the location of the S3 bucket.**

**-- STAGES are locations where files are stored for loading or unloading.**

**-- This is public S3 bucket, so no credentials are required.**

**CREATE OR REPLACE STAGE investigation3651**

**url = 's3://investigation-3651/';**

**-- List the files in the stage. (It does NOT show delimiters.)**

**LIST @investigation3651;**

**-- Review the properties of the stage. (Delimiters are noted.)**

**DESCRIBE STAGE investigation3651;**

**-- Create a FILE FORMAT that matches the data structure.**

**-- It describes what the staged files look like.**

**CREATE OR REPLACE FILE FORMAT ghosts\_csv**

**type='csv'**

**compression = 'auto' -- compression checked automatically (default)**

**field\_delimiter = ',' -- columns separated by commas (default)**

**record\_delimiter = '\n' -- rows separated by newlines (default)**

**skip\_header = 1 -- does not skip lines in the file (default)**

**trim\_space = true -- does not remove white space from fields (default)**

**null\_if = ('-') -- replace these values with SQL NULL**

**date\_format = 'auto' -- date checked automatically (default)**

**timestamp\_format = 'auto' -- time checked automatically (default)**

**field\_optionally\_enclosed\_by = '\042' -- strings closed by " " (octal)**

**;**

**---------------------------------**

**-- LOADING DATA**

**---------------------------------**

**-- Use a COPY COMMAND to load staged data into the table.**

**COPY INTO phone\_directory -- the destination table**

**FROM @investigation3651/phone\_directory -- the source stage**

**file\_format = ghosts\_csv -- the file format**

**; -- the filenames to match**

**-- Use a COPY COMMAND to load staged data into the table.**

**COPY INTO call\_log**

**FROM @investigation3651/call\_log**

**FILE\_FORMAT = (TYPE = 'CSV', NULL\_IF = ('N/A'), SKIP\_HEADER = 1);**

**-------------------------------**

**--SWITCH TO QUERY WH**

**--------------------------------**

**CREATE OR REPLACE WAREHOUSE ghosts\_query\_wh WITH**

**WAREHOUSE\_SIZE='SMALL'**

**AUTO\_RESUME = TRUE -- default**

**AUTO\_SUSPEND = 180 -- default**

**INITIALLY\_SUSPENDED = TRUE;**

**-------------------------------**

**--WRITE QUERIES**

**-------------------------------**

**--How many calls are in the data? --398390**

**SELECT COUNT(\*) AS total\_calls**

**FROM call\_log**

**;**

**--What is the date of the last call? (Use this format: MM/DD/YY.) --12/30/23**

**SELECT**

**TO\_CHAR(MIN(call\_start\_date), 'MM/DD/YY') AS first\_call\_date,**

**TO\_CHAR(MAX(call\_start\_date), 'MM/DD/YY') AS last\_call\_date**

**FROM call\_log**

**;**

**--You could use a CTE, subquery, and/or window function.**

**--What % of calls were answered? (Enter a % and round to the nearest whole number.)--78%**

**SELECT call\_status, COUNT(\*) AS count**

**FROM call\_log**

**GROUP BY call\_status**

**ORDER BY count DESC**

**;**

**WITH call\_counts AS (**

**SELECT**

**COUNT(\*) AS total\_calls,**

**COUNT(CASE WHEN call\_status = 'answered' THEN 1 END) AS calls\_answered**

**FROM call\_log**

**)**

**SELECT**

**(calls\_answered \* 100.0) / NULLIF(total\_calls, 0) AS answered\_call\_percentage**

**FROM call\_counts**

**;**

**--Calculate average call duration in minutes. -- 10.8**

**SELECT**

**AVG(call\_duration) / 60.0 AS average\_duration\_minutes**

**FROM call\_log**

**WHERE call\_status = 'answered'**

**;**

**--By the end of June 2023, how many calls had been made? --198010**

**SELECT**

**YEAR(call\_start\_date) AS year,**

**MONTH(call\_start\_date) AS month,**

**COUNT(\*) AS monthly\_total,**

**SUM(COUNT(\*)) OVER (ORDER BY YEAR(call\_start\_date), MONTH(call\_start\_date)) AS running\_total,**

**FROM call\_log**

**GROUP BY YEAR(call\_start\_date), MONTH(call\_start\_date)**

**ORDER BY year, month**

**;**

**--What was the total outbound call volume for the "Tech District”? --35606**

**SELECT**

**pd.district,**

**COUNT(cl.caller\_id) AS outbound\_call\_volume**

**FROM call\_log cl**

**JOIN phone\_directory pd**

**ON cl.caller\_id = pd.directory\_id**

**GROUP BY pd.district**

**ORDER BY outbound\_call\_volume DESC**

**;**

**--Create a view**

**DROP VIEW IF EXISTS victim\_detail\_view;**

**CREATE VIEW victim\_detail\_view AS**

**SELECT**

**vp.victim\_id,**

**vp.name,**

**vp.address,**

**cd.date AS homicide\_date,**

**pd.directory\_id**

**FROM victim\_profiles vp**

**JOIN crime\_details cd ON vp.victim\_id = cd.victim\_id**

**LEFT JOIN phone\_directory pd ON vp.address = pd.address**

**WHERE cd.type = 'Homicide'**

**ORDER BY vp.name;**

**SELECT \* FROM victim\_detail\_view;**

**USE investigation.public;**

**USE WAREHOUSE ghosts\_loading\_wh;**

**---Clear the existing call\_log table, but do NOT delete it.**

**TRUNCATE TABLE call\_log;**

**---------------------------------**

**-- CREATING STAGES**

**---------------------------------**

**-- Create a STAGE that specifies the location of the S3 bucket.**

**-- STAGES are locations where files are stored for loading or unloading.**

**-- This is public S3 bucket, so no credentials are required.**

**CREATE OR REPLACE STAGE investigation4275**

**url = 's3://investigation-4275/';**

**-- List the files in the stage. (It does NOT show delimiters.)**

**LIST @investigation4275;**

**-- Review the properties of the stage. (Delimiters are noted.)**

**DESCRIBE STAGE investigation4275;**

**-- Create a FILE FORMAT that matches the data structure.**

**-- It describes what the staged files look like.**

**CREATE OR REPLACE FILE FORMAT ghosts\_pipe**

**type='csv'**

**field\_delimiter = '|' -- columns separated by commas (default)**

**skip\_header = 1 -- does not skip lines in the file (default)**

**trim\_space = true -- does not remove white space from fields (default)**

**null\_if = ('N/A') -- replace these values with SQL NULL**

**;**

**-- Preview the listings file(s) on staging.**

**SELECT $1 FROM @investigation4275/call\_log LIMIT 100;**

**---Copy the updated call log from staging into the table.**

**COPY INTO call\_log**

**FROM @investigation4275/call\_log**

**FILE\_FORMAT = ghosts\_pipe;**

**SELECT \* FROM call\_log LIMIT 100;**

**---Unload the call\_log table to its internal table stage.**

**COPY INTO @%call\_log**

**FROM call\_log;**

**LIST @%call\_log;**

**SELECT \* FROM @%call\_log/data\_0\_0\_0.csv LIMIT 10;**

**---Switch to Query Data**

**USE WAREHOUSE query\_wh;**

**---Explore victim details**

**SELECT \* FROM victim\_detail\_view**

**LIMIT 10;**

**---Queries**

**---1. Count inbound calls received by homicide victims, including victim names ---Anita Gomez: 62 calls**

**SELECT**

**vdv.name AS victim\_name,**

**COUNT(cl.call\_id) AS inbound\_call\_count**

**FROM call\_log cl**

**INNER JOIN victim\_detail\_view vdv**

**ON cl.receiver\_id = vdv.directory\_id**

**GROUP BY vdv.name**

**ORDER BY vdv.name ASC;**

**---2. List the inbound calls received by these victims. ---Alex Martinez most recent call: 2023-12-11**

**SELECT**

**vdv.name AS victim\_name,**

**cl.caller\_id,**

**cl.call\_id,**

**cl.call\_start\_date,**

**cl.call\_status**

**FROM call\_log cl**

**INNER JOIN victim\_detail\_view vdv**

**ON cl.receiver\_id = vdv.directory\_id**

**ORDER BY**

**vdv.name ASC,**

**cl.call\_start\_date DESC;**

**---3. Copy and adjust the previous query, which listed inbound calls. Only list calls received ≤ 14 days before their homicide. Emily Thorton caller\_id: 54528.**

**SELECT**

**vdv.name AS victim\_name,**

**cl.caller\_id,**

**cl.call\_id,**

**cl.call\_start\_date,**

**cl.call\_status**

**FROM call\_log cl**

**INNER JOIN victim\_detail\_view vdv**

**ON cl.receiver\_id = vdv.directory\_id**

**WHERE**

**DATEDIFF(day, cl.call\_start\_date, vdv.homicide\_date) BETWEEN 0 AND 14**

**ORDER BY**

**vdv.name ASC,**

**cl.call\_start\_date DESC;**

**---Copy and adjust the previous query again.**

**-- List inbound calls received ≤ 14 days before victim's homicide, including caller names --Doroteya Hasson**

**SELECT**

**vdv.name AS victim\_name,**

**cl.caller\_id,**

**caller.name AS caller\_name,**

**cl.call\_id,**

**cl.call\_start\_date,**

**cl.call\_status**

**FROM call\_log cl**

**INNER JOIN victim\_detail\_view vdv**

**ON cl.receiver\_id = vdv.directory\_id**

**LEFT JOIN phone\_directory caller**

**ON cl.caller\_id = caller.directory\_id**

**WHERE**

**DATEDIFF(day, cl.call\_start\_date, vdv.homicide\_date) BETWEEN 0 AND 14**

**ORDER BY**

**vdv.name ASC,**

**cl.call\_start\_date DESC;**

**SELECT**

**vdv.name AS victim\_name,**

**cl.caller\_id,**

**caller.name AS caller\_name,**

**cl.call\_id,**

**cl.call\_start\_date,**

**cl.call\_status**

**FROM call\_log cl**

**INNER JOIN victim\_detail\_view vdv**

**ON cl.receiver\_id = vdv.directory\_id**

**LEFT JOIN phone\_directory caller**

**ON cl.caller\_id = caller.directory\_id**

**WHERE**

**cl.call\_start\_date BETWEEN DATEADD(day, -14, vdv.homicide\_date) AND vdv.homicide\_date**

**ORDER BY**

**vdv.name ASC,**

**cl.call\_start\_date DESC;**

**------------------------------**

**--Analyze Semi-Structured Data**

**------------------------------**

**--**

**CREATE OR REPLACE TABLE video\_activity\_json (**

**camera\_data VARIANT**

**);**

**-- Load JSON data into video\_activity\_json, stripping outer array brackets**

**COPY INTO video\_activity\_json**

**FROM @investigation4275/video\_activity**

**file\_format = (type = 'json'**

**strip\_outer\_array = TRUE)**

**;**

**SELECT camera\_data**

**FROM video\_activity\_json**

**LIMIT 10;**

**---Create a new view that will provide structure to the data.**

**---Create structured view for JSON data**

**CREATE OR REPLACE VIEW video\_activity\_view AS**

**SELECT**

**camera\_data:footage\_id::INTEGER AS footage\_id,**

**camera\_data:camera\_location::STRING AS camera\_location,**

**camera\_data:camera\_status::STRING AS camera\_status,**

**camera\_data:observed\_activity::STRING AS observed\_activity,**

**TO\_TIMESTAMP(camera\_data:timestamp::STRING, 'MM/DD/YYYY HH24:MI') AS timestamp**

**FROM video\_activity\_json;**

**---Determine the time span of video activity data ---2023-10-16 08:55:00.000**

**SELECT**

**MIN(timestamp) AS earliest\_activity,**

**MAX(timestamp) AS latest\_activity**

**FROM video\_activity\_view;**

**-- List video activities on homicide days. --How many video activities were collected on 12/28/23? 3**

**SELECT**

**v.\***

**FROM video\_activity\_view v**

**WHERE TO\_DATE(v.timestamp) IN (**

**SELECT DISTINCT date**

**FROM crime\_details**

**WHERE type = 'Homicide'**

**)**

**ORDER BY v.timestamp DESC;**

**USE investigation.public;**

**USE WAREHOUSE ghosts\_loading\_wh;**

**CREATE OR REPLACE STAGE investigation5921**

**url = 's3://investigation-5921/';**

**LIST @investigation5921;**

**DESCRIBE STAGE investigation5921;**

**---Query the city officials file in staging to explore it.**

**SELECT $1, $2, $3, $4, $5, $6, $7, $8**

**FROM @investigation5921/city\_officials\_augmented.csv LIMIT 100;**

**---What is the "official\_id" for the first official in the staged file? 101147**

**---Next, clear the "city\_officials" table**

**DROP TABLE city\_officials;**

**CREATE OR REPLACE FILE FORMAT ghosts\_csv**

**type='csv'**

**compression = 'auto' -- compression checked automatically (default)**

**field\_delimiter = ',' -- columns separated by commas (default)**

**record\_delimiter = '\n' -- rows separated by newlines (default)**

**skip\_header = 1 -- does not skip lines in the file (default)**

**trim\_space = true -- does not remove white space from fields (default)**

**null\_if = ('-') -- replace these values with SQL NULL**

**date\_format = 'auto' -- date checked automatically (default)**

**timestamp\_format = 'auto' -- time checked automatically (default)**

**field\_optionally\_enclosed\_by = '\042' -- strings closed by " " (octal)**

**;**

**---Add a new column to the table -- "phone\_directory\_id".**

**CREATE OR REPLACE TABLE city\_officials (**

**official\_id integer,**

**name string,**

**position string,**

**department string,**

**office\_location string,**

**phone\_directory\_id string,**

**tenure\_start date,**

**ip\_address string**

**);**

**---Reload the table using the file in staging.**

**COPY INTO city\_officials**

**FROM (**

**SELECT $1, $2, $3, $4, $5, $6, $7, $8**

**FROM @investigation5921/city\_officials\_augmented.csv (file\_format => ghosts\_csv)**

**);**

**---What is Charles Lee’s office location? (Provide the name, not a building number.)**

**UPDATE city\_officials**

**SET office\_location = CASE office\_location**

**WHEN 'Sector 1' THEN 'Business District'**

**WHEN 'Sector 2' THEN 'Downtown'**

**WHEN 'Sector 3' THEN 'East Market'**

**WHEN 'Sector 4' THEN 'Highview'**

**WHEN 'Sector 5' THEN 'Midtown'**

**WHEN 'Sector 6' THEN 'Riverside'**

**WHEN 'Sector 7' THEN 'Tech District'**

**WHEN 'Sector 8' THEN 'West End'**

**ELSE office\_location**

**END;**

**SELECT \* FROM city\_officials;**

**---Create a new view -- "calls\_to\_victims".**

**CREATE OR REPLACE VIEW calls\_to\_victims AS**

**SELECT**

**vdv.name AS victim\_name,**

**cl.caller\_id,**

**caller.name AS caller\_name,**

**cl.call\_id,**

**cl.call\_start\_date,**

**cl.call\_status**

**FROM call\_log cl**

**INNER JOIN victim\_detail\_view vdv**

**ON cl.receiver\_id = vdv.directory\_id**

**LEFT JOIN phone\_directory caller**

**ON cl.caller\_id = caller.directory\_id**

**WHERE**

**cl.call\_start\_date BETWEEN DATEADD(day, -14, vdv.homicide\_date) AND vdv.homicide\_date**

**ORDER BY**

**vdv.name ASC,**

**cl.call\_start\_date DESC;**

**-- Query the view to confirm it loaded properly**

**SELECT \* FROM calls\_to\_victims LIMIT 100;**

**-- Query the calls\_to\_victims view to show only calls from city officials**

**---Which city official called Jane Patterson? Enter their phone ID.**

**SELECT**

**ctv.victim\_name,**

**ctv.caller\_id,**

**ctv.caller\_name,**

**co.position AS official\_position,**

**co.department AS official\_department,**

**co.office\_location,**

**co.phone\_directory\_id,**

**ctv.call\_id,**

**ctv.call\_start\_date,**

**ctv.call\_status**

**FROM calls\_to\_victims ctv**

**INNER JOIN city\_officials co**

**ON ctv.caller\_id = co.phone\_directory\_id**

**ORDER BY**

**ctv.victim\_name ASC,**

**ctv.call\_start\_date DESC;**