SQL Queries

Part 1: Loading Data in SQL Server

This document outlines steps to create a database in SQL Server, manage data, and ensure data quality checks. Each step includes explanations and potential pitfalls to help users navigate the process smoothly.

Step 1: List All Existing Databases

Retrieve information about all existing databases to understand the current environment and avoid naming conflicts.

- **SELECT name, database_id, create_date, state_desc FROM sys.databases;**

Step 2: Create a New Database

Create a new database named ccdb. This name should be unique.

- **CREATE DATABASE ccdb;**

Step 3: Check Current Database Context

Before proceeding, verify your current database to ensure you're in the correct context.

- **SELECT DB_NAME() AS CurrentDatabase;**

Step 4: Switch to the 'ccdb' Database

Switch to ccdb so subsequent actions occur within this database.

- **USE ccdb;**

Step 5: Confirm Current Database Context

Run this query again to ensure you're now in ccdb.

- **SELECT DB_NAME() AS CurrentDatabase;**

Step 6: Close and Refresh Data

Potential Issues to Avoid

1. **Permissions**: Ensure the executing user has the appropriate permissions to create databases.
2. **Database Existence**: Confirm ccdb does not exist already.
3. **Correct Context**: Verify you're executing these commands in the appropriate tool (SSMS or another SQL client).

Data Quality Checks and Validation

- **Retrieve Records for Review -** Fetch all records from tables for initial review and analysis.
  - **Credit Card Table**
    - **select count(*) from dbo.credit_card;**
    - **SELECT top 10 * FROM dbo.credit_card;**
  - **Customer Table**
    - **select count(*) from customer;**
    - **SELECT top 10 * FROM customer;**
- **Missing Values Check**
  - **Credit Card Table -**Check for missing values in key columns.

```
SELECT
SUM(CASE WHEN Client_Num IS NULL THEN 1 ELSE 0 END) AS
Missing_Client_Num,
SUM(CASE WHEN Card_Category IS NULL THEN 1 ELSE 0 END) AS
Missing_Card_Category,
SUM(CASE WHEN Annual_Fees IS NULL THEN 1 ELSE 0 END) AS
Missing_Annual_Fees,
SUM(CASE WHEN Activation_30_Days IS NULL THEN 1 ELSE 0 END) AS
Missing_Activation_30_Days,
SUM(CASE WHEN Customer_Acq_Cost IS NULL THEN 1 ELSE 0 END) AS
Missing_Customer_Acq_Cost,
SUM(CASE WHEN Week_Start_Date IS NULL THEN 1 ELSE 0 END) AS
Missing_Week_Start_Date,
SUM(CASE WHEN Week_Num IS NULL THEN 1 ELSE 0 END) AS Missing_Week_Num,
SUM(CASE WHEN Qtr IS NULL THEN 1 ELSE 0 END) AS Missing_Qtr,
SUM(CASE WHEN current_year IS NULL THEN 1 ELSE 0 END) AS
Missing_current_year,
SUM(CASE WHEN Credit_Limit IS NULL THEN 1 ELSE 0 END) AS
Missing_Credit_Limit,
SUM(CASE WHEN Total_Revolving_Bal IS NULL THEN 1 ELSE 0 END) AS
Missing_Total_Revolving_Bal,
SUM(CASE WHEN Total_Trans_Amt IS NULL THEN 1 ELSE 0 END) AS
Missing_Total_Trans_Amt,
SUM(CASE WHEN Total_Trans_Vol IS NULL THEN 1 ELSE 0 END) AS
Missing_Total_Trans_Vol,
SUM(CASE WHEN Avg_Utilization_Ratio IS NULL THEN 1 ELSE 0 END) AS
Missing_Avg_Utilization_Ratio,
SUM(CASE WHEN Use_Chip IS NULL THEN 1 ELSE 0 END) AS Missing_Use_Chip,
SUM(CASE WHEN Exp_Type IS NULL THEN 1 ELSE 0 END) AS Missing_Exp_Type,
SUM(CASE WHEN Interest_Earned IS NULL THEN 1 ELSE 0 END) AS
Missing_Interest_Earned,
SUM(CASE WHEN Delinquent_Acc IS NULL THEN 1 ELSE 0 END) AS
Missing_Delinquent_Acc
FROM credit_card;
```

  - **Customer Table -** Data quality check for the 'customer' table

```
SELECT
```

```
SUM(CASE WHEN Client_Num IS NULL THEN 1 ELSE 0 END) AS
Missing_Client_Num,
SUM(CASE WHEN Customer_Age IS NULL THEN 1 ELSE 0 END) AS
Missing_Customer_Age,
SUM(CASE WHEN Gender IS NULL THEN 1 ELSE 0 END) AS Missing_Gender,
SUM(CASE WHEN Dependent_Count IS NULL THEN 1 ELSE 0 END) AS
Missing_Dependent_Count,
SUM(CASE WHEN Education_Level IS NULL THEN 1 ELSE 0 END) AS
Missing_Education_Level,
SUM(CASE WHEN Marital_Status IS NULL THEN 1 ELSE 0 END) AS
Missing_Marital_Status,
SUM(CASE WHEN State_cd IS NULL THEN 1 ELSE 0 END) AS Missing_State_cd,
SUM(CASE WHEN Zipcode IS NULL THEN 1 ELSE 0 END) AS Missing_Zipcode,
SUM(CASE WHEN Car_Owner IS NULL THEN 1 ELSE 0 END) AS Missing_Car_Owner,
SUM(CASE WHEN House_Owner IS NULL THEN 1 ELSE 0 END) AS
Missing_House_Owner,
SUM(CASE WHEN Personal_Loan IS NULL THEN 1 ELSE 0 END) AS
Missing_Personal_Loan,
SUM(CASE WHEN Contact IS NULL THEN 1 ELSE 0 END) AS Missing_Contact,
SUM(CASE WHEN Customer_Job IS NULL THEN 1 ELSE 0 END) AS
Missing_Customer_Job,
SUM(CASE WHEN Income IS NULL THEN 1 ELSE 0 END) AS Missing_Income,
SUM(CASE WHEN Cust_Satisfaction_Score IS NULL THEN 1 ELSE 0 END) AS
Missing_Cust_Satisfaction_Score
FROM customer;
```

- **Duplicates and Value Checks**
- **Duplicate Client Numbers -** Identify duplicate Client_Num entries in both tables.
    - **Credit Card Table**:
        - **SELECT Client_Num, COUNT(*) AS Count FROM credit_card GROUP BY Client_Num HAVING COUNT(*) > 1;**
    - **Customer Table**:
        - **SELECT Client_Num, COUNT(*) AS Count FROM customer GROUP BY Client_Num HAVING COUNT(*) > 1;**
- **Distinct Gender Values -** Verify the consistency of gender values.
    - **SELECT DISTINCT Gender FROM customer;**

****Import Credit_add and customer_add**