**SQL Script Documentation**

Overview

This SQL script creates and manages customer data tables, focusing on feature engineering by introducing a new derived feature (all\_charges). It also includes schema modifications and data transformations.

Steps and Explanation

1. Creating the clean\_data Table

* This table stores cleaned customer data with key attributes such as demographic details, service subscriptions, and billing information.

CREATE TABLE clean\_data (

gender VARCHAR(10),

seniorcitizen VARCHAR(10),

partner VARCHAR(10),

dependents VARCHAR(10),

phoneservice VARCHAR(10),

multiplelines VARCHAR(10),

internetservice VARCHAR(10),

onlinesecurity VARCHAR(10),

onlinebackup VARCHAR(10),

deviceprotection VARCHAR(10),

techsupport VARCHAR(10),

streamingtv VARCHAR(10),

streamingmovies VARCHAR(10),

tenure NUMERIC(10,2),

contract VARCHAR(10),

paperlessbilling VARCHAR(10),

paymentmethod VARCHAR(10),

monthlycharges NUMERIC(10,2),

totalcharges NUMERIC(10,2),

churn NUMERIC(10,2)

);

* Expected Output: A new table clean\_data is successfully created with no data initially.

1. Checking the Record Count

* A query is used to verify that the table contains data after population.

SELECT COUNT(\*) FROM clean\_data;

* Expected Output: A numerical value indicating the number of records in clean\_data.

1. Creating clean\_data\_new\_feature Table

* This table is an extended version of clean\_data, incorporating a new feature (all\_charges), which is calculated as monthlycharges \* tenure.

CREATE TABLE clean\_data\_new\_feature (

gender VARCHAR(10),

seniorcitizen VARCHAR(10),

partner VARCHAR(10),

dependents VARCHAR(10),

phoneservice VARCHAR(10),

multiplelines VARCHAR(10),

internetservice VARCHAR(10),

onlinesecurity VARCHAR(10),

onlinebackup VARCHAR(10),

deviceprotection VARCHAR(10),

techsupport VARCHAR(10),

streamingtv VARCHAR(10),

streamingmovies VARCHAR(10),

tenure NUMERIC(10,2),

contract VARCHAR(10),

paperlessbilling VARCHAR(10),

paymentmethod VARCHAR(10),

monthlycharges NUMERIC(10,2),

totalcharges NUMERIC(10,2),

churn NUMERIC(10,2),

all\_charges NUMERIC(10,2)

);

* Expected Output: A new table clean\_data\_new\_feature is successfully created with the additional column all\_charges.

1. Populating clean\_data\_new\_feature Table

* This step copies data from clean\_data and computes all\_charges as monthlycharges \* tenure.

INSERT INTO clean\_data\_new\_feature (

gender, seniorcitizen, partner, dependents, phoneservice, multiplelines,

internetservice, onlinesecurity, onlinebackup, deviceprotection, techsupport,

streamingtv, streamingmovies, tenure, contract, paperlessbilling, paymentmethod,

monthlycharges, totalcharges, churn, all\_charges

)

SELECT

gender, seniorcitizen, partner, dependents, phoneservice, multiplelines,

internetservice, onlinesecurity, onlinebackup, deviceprotection, techsupport,

streamingtv, streamingmovies, tenure, contract, paperlessbilling, paymentmethod,

monthlycharges, totalcharges, churn,

(monthlycharges \* tenure) AS all\_charges

FROM clean\_data;

* Expected Output: Records from clean\_data are successfully copied into clean\_data\_new\_feature, with all\_charges populated for each row.

1. Adding a modified\_date Column

* This column is intended to track when a record was last modified.

ALTER TABLE clean\_data\_new\_feature ADD COLUMN modified\_date DATE;

* Expected Output: A new column modified\_date is successfully added to clean\_data\_new\_feature.

Summary

* Tables Created: clean\_data, clean\_data\_new\_feature
* New Feature Added: all\_charges = monthlycharges \* tenure
* Schema Changes: Added modified\_date
* Data Transformation: Migration from clean\_data to clean\_data\_new\_feature with additional calculated field.