use Commercial\_Project

-- CUSTOMER TABLE

select \* from Customers

-- Add a new column named 'full\_name' to merge the 'first\_name' and 'last\_name' columns, separated by a space

ALTER TABLE Customers

ADD full\_name VARCHAR(50)

UPDATE Customers

SET full\_name = CONCAT\_WS(' ', first\_name, last\_name)

-- Create a new column named 'birth\_year' to extract the year from the 'birthdate' column, and format as varchar

ALTER TABLE Customers

ADD birth\_year VARCHAR(50)

UPDATE Customers

SET birth\_year = DATEPART(YEAR, birthdate)

-- Create a conditional column named 'has\_children' which equals 'N' if 'total\_children' = 0, otherwise 'Y'

ALTER TABLE Customers

ADD has\_children VARCHAR(50)

UPDATE Customers

SET has\_children =

(SELECT CASE WHEN total\_children = 0 THEN 'N'

ELSE 'Y'

END AS has\_children)

A screenshot of a computer

Description automatically generated

use Commercial\_Project

-- PRODUCTS TABLE

select \* from Products

-- Use the statistics tools to return the number of distinct product brands, followed by distinct product names

select COUNT(DISTINCT product\_brand) AS number\_of\_unique\_product\_brands,

COUNT(DISTINCT product\_name) AS number\_of\_unique\_product\_names

from Products

-- Add a calculated column named 'discount\_price', equal to 90% of the original retail price

-- Format as a fixed decimal number, and then use the rounding tool to round to 2 digits

ALTER TABLE products

ADD discount\_price DECIMAL(10,2)

UPDATE Products

SET discount\_price = CAST((product\_retail\_price \* 0.90) AS DECIMAL(10,2))

A screenshot of a computer

Description automatically generated

-- Select "product\_brand" and use the Group By option to calculate the average retail price by brand

-- You should see an average retail price of $2.18 for Washington products, and $2.21 for Green Ribbon

select product\_brand, AVG(product\_retail\_price) as AVG\_product\_retail\_price from Products

where product\_brand IN ('Washington', 'Green Ribbon')

group by product\_brand

-- BY USING A COMMON TABLE EXPRESSION

WITH CTE AS (select product\_brand, AVG(product\_retail\_price) as AVG\_product\_retail\_price from Products

group by product\_brand)

select \* from CTE

where product\_brand IN ('Washington', 'Green Ribbon')

A screenshot of a computer

Description automatically generated

-- OVER() CLAUSE WINDOW FUNCTION

select product\_brand, product\_retail\_price,

AVG(product\_retail\_price) OVER() AS company\_avg\_product\_retail\_price,

AVG(product\_retail\_price) OVER(PARTITION BY product\_brand) AS avg\_product\_retail\_price\_by\_product\_brand

from Products

A screenshot of a computer

Description automatically generated

-- Replace "null" values with zeros in both the "recyclable" and "low-fat" columns

UPDATE Products

SET recyclable = 0

WHERE recyclable IS NULL

UPDATE Products

SET low\_fat = REPLACE(low\_fat, NULL, 0)

-- STORES TABLE

select \* from Stores

-- Add a calculated column named 'full\_address', by merging 'store\_city', 'store\_state', and 'store\_country', separated by a comma and space

ALTER TABLE Stores

ADD full\_address VARCHAR(50)

UPDATE stores

SET full\_address = CONCAT(store\_city, ', ', store\_state, ', ', store\_country)

-- Add a column named "area\_code", by extracting the characters before the dash ("-") in the "store\_phone" field

ALTER TABLE stores

ADD area\_code VARCHAR(50)

UPDATE stores

SET area\_code = SUBSTRING(store\_phone, 1, 3)

A screenshot of a computer

Description automatically generated

-- CALENDAR TABLE

select \* from Calendar

-- Add these columns: Start of Week (starting Sunday), Name of Day, Start of Month, Name of Month, Quarter of Year, Year

ALTER TABLE Calendar

ADD Start\_Of\_Week Date

UPDATE Calendar

SET start\_of\_week = DATETRUNC(WEEK, Date)

ALTER TABLE Calendar

ADD Name\_Of\_day VARCHAR(50)

UPDATE Calendar

SET Name\_Of\_day = DATENAME(DW, Date)

-- OR I could have used the FORMAT function too

UPDATE Calendar

SET Name\_Of\_day = FORMAT(date, 'dddd')

ALTER TABLE Calendar

ADD Start\_Of\_Month Date

UPDATE Calendar

SET Start\_of\_Month = DATETRUNC(MONTH, Date)

ALTER TABLE Calendar

ADD Name\_Of\_Month Date

ALTER TABLE Calendar

ALTER COLUMN Name\_Of\_Month VARCHAR(50)

UPDATE Calendar

SET Name\_Of\_Month = DATENAME(MONTH, Date)

ALTER TABLE Calendar

ADD Quarter\_Of\_Year INTEGER

UPDATE Calendar

SET Quarter\_Of\_Year = DATEPART(QUARTER, Date)

ALTER TABLE Calendar

ADD Year INTEGER

UPDATE Calendar

SET Year = DATEPART(YEAR, Date)

A screenshot of a computer

Description automatically generated

-- Update the date fields e.g. across the Customers table to the "M/d/yyyy" format

select FORMAT(birthdate, 'M/d/yyyy') from Customers