A

Group Project Final Report

on

**Relational Database Management System for WashCare Company Stores**

By

**Group-8 (Team Y)**

Chandu Yerragopu

Saran Prasad Balasubramaniam

Wen Xie

Under the guidance of

Dr. Benjamin Mitchell, Ph.D.

Course: ISQS 6338 Database Concepts

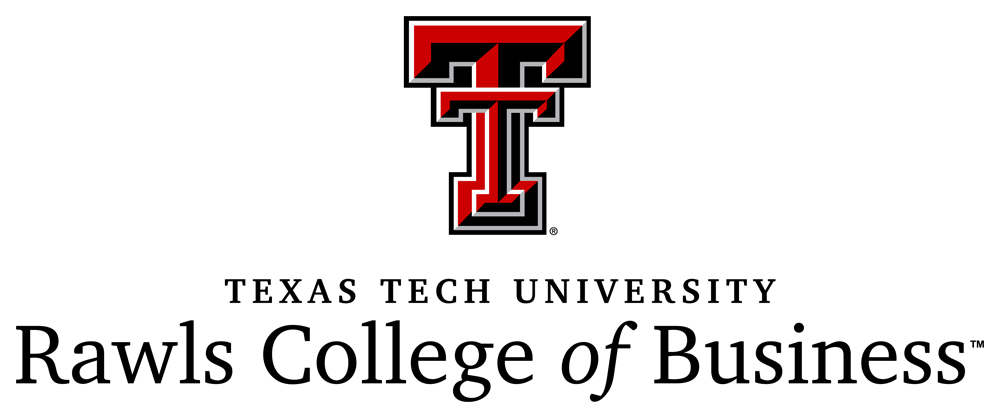


Table of contents

1 Introduction 3

1.1 Objectives for using Database in this environment 3

1.2 Problem solved and value added by using database 3

1.3 Boundaries of the Proposed model 4

1.4 Assumptions 4

2 Conceptual Model Diagrams 6

3 Logical Model Diagrams 10

4 Physical Model Diagram 16

5 SQL Statements 17

5.1 Table Creation statements 17

5.2 Inserting the data 21

5.3 Checking the data after insertion 23

6 Narrative Behind projects data 24

6.1 Questions of interest to data scientists 24

6.2 Answering those questions using SQL 26

7 Conclusions and Future Work 28

# Introduction

This project aims to build a relational database for WashCare, a company that primarily sells the laundry products, such as liquid detergents, powder, etc. This project come s under the domain of retail industry where databases have profound use because of the many advantages they bring. In the next sections of this report, we explain the reason for using a database in this environment, including what problems are solved and what advantages are realized.

## Objectives for using Database in this environment

In this case, we need to store the data of various entities such as customer, product, orders, store, employee and their salary, product sales, etc. Generally, this is done in a way such as book keeping where the data is stored in excel tables. Storing the data is excel has its own disadvantages. Inorder to overcome such disadvantages and efficiently store the data and later use it for identifying the customer purchasing behaviors, including what product are purchased, how many times customers purchase in a month or year, how much stock is available, what products are in demand, etc. a database is needed in this case.

## Problem solved and value added by using database

Using a database solves the problem of bookkeeping or storing the various transaction details in a file-based system. File-based systems are not very efficient because of the below problems’

* Data redundancy and inconsistency – The same details of the product must be stored again and again.
* Security Problems – The sales information stored in a file format has been often not safe.
* Difficulty in analyzing data – We cannot get a summary about the data.
* Update Anomalies – Suppose an employee working in a shift forgets to update the file or updates it mistakenly.
* Data Consistency and Integrity problems – Whenever there is an update to the data in one place, it should be changed at multiple locations else it will result in loss of consistency of the data and we cannot rely on that data in future.

In this way, we face several problems when use a traditional book-keeping using a file-based system. By introducing the databases in this environment, most of the above problems can be addressed.

## Boundaries of the Proposed model

In this project, we try to construct the database to serve two important purposes:

1. The first one is to maintain the data of the customer, product, day and the order details and how these entities are linked to each other. This data when properly saved in a database and then analyzed helps us in understanding various facts and obtain logical insights which will be explained in detail in the next steps of the project.

2. Also, this model maintains the data about the Employees working in this company, their personal and the salary information, the details about the store, etc. This data will be useful for the management to maintain the data about their employees.

The above explanation helps us in understanding the boundaries of the model. In the next update about the project, we will explain in detail about the exact list of tables used in this project using various diagrams and then proceed with constructing the database.

## Assumptions

Below are some of the assumptions of the proposed database model:

1. The company currently has only 2 stores in the city.

2. It sells only the laundry products such as detergents.

3. Only a limited number of data records are taken into consideration in this project.

4. There is only one product per each sale.

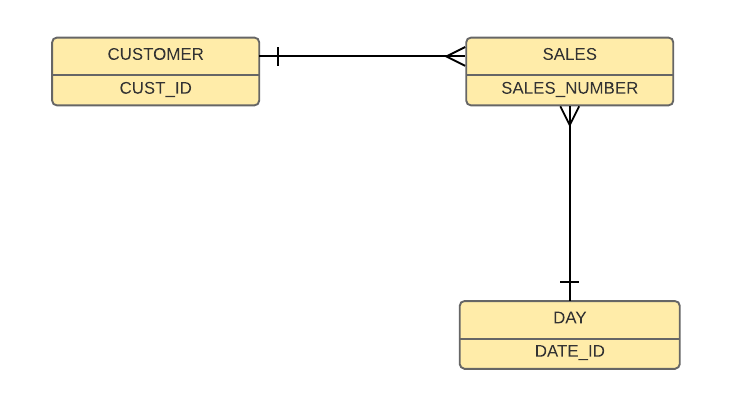
5. There is only product associated with each order.

# Conceptual Model Diagrams

**Definition:** The conceptual model is also known as the data model as data model can be used to describe the conceptual schema when a database system is implemented. It hides the internal details of physical storage and targets on describing entities, datatype, relationships and constraints.

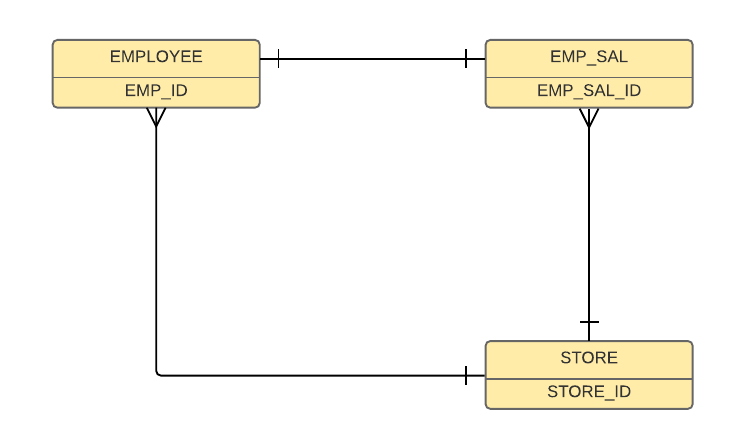
*Note: For this project, we proceed with the conceptual model diagrams of the various combinations of entities which when combined forms a final diagram that depicts the project.*

1. Customer, Sales and Day



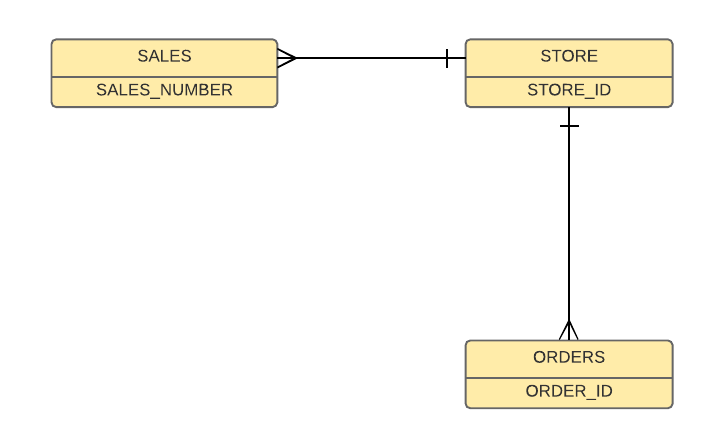
On a given day, every customer will have some sales information. One customer may have many sales records. And for each day we have many sales information.

1. Employee, Employee Salary and Store



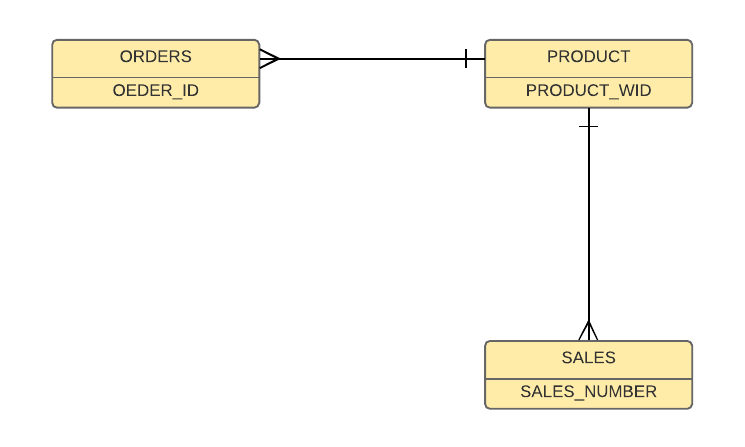
A store will have many employees and each employee will has their own salary data. One employee belongs to only one store and has one fixed salary data.

1. Store, Sales and Orders



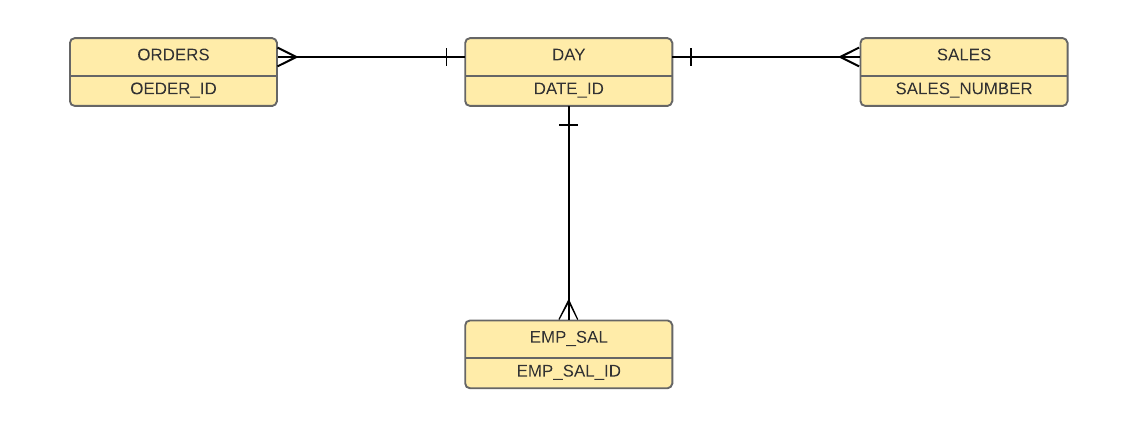
Each store will have many orders data. Also, each store has many sales related data.

1. Product, Orders and Sales



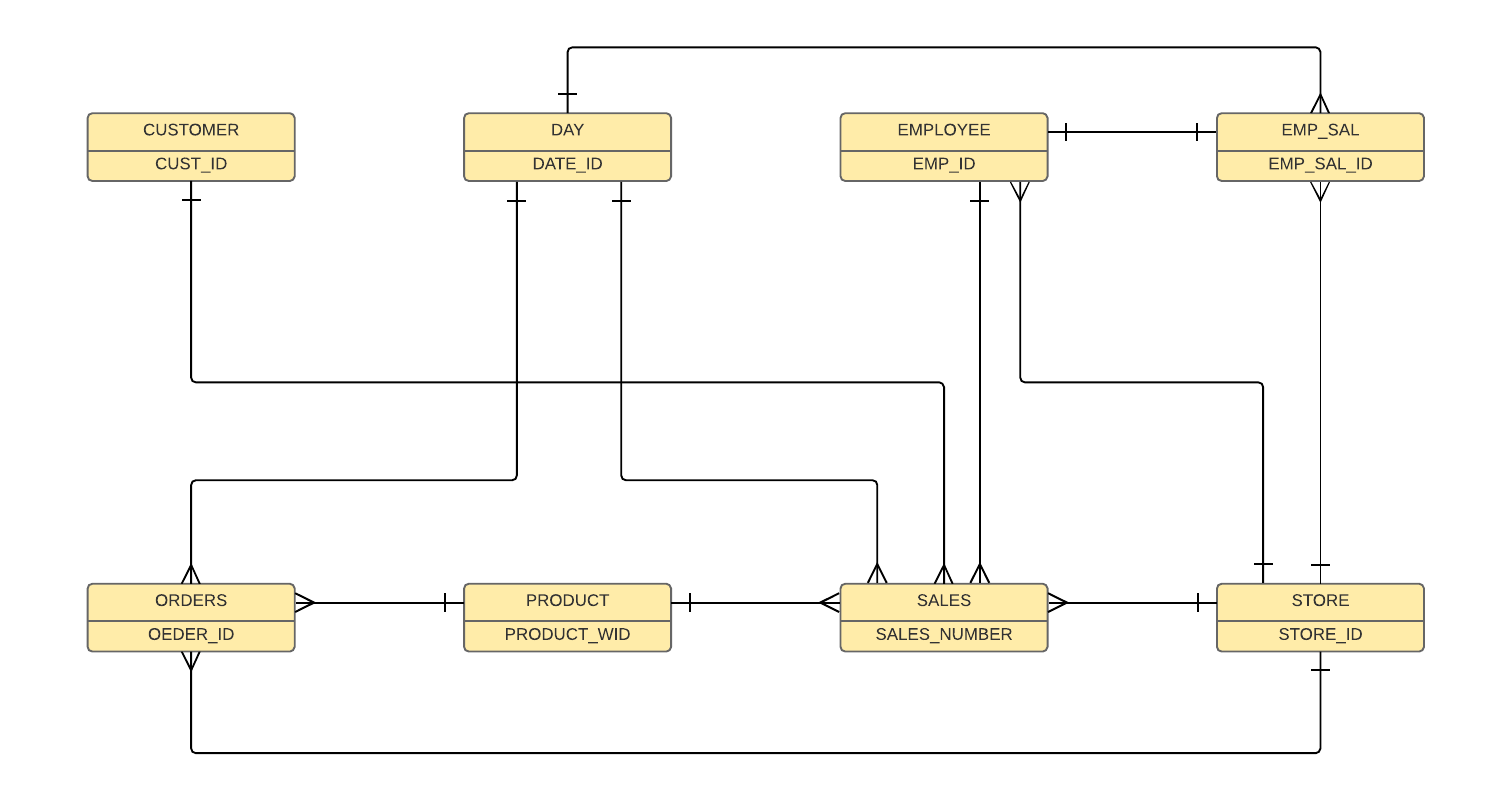
A product can be associated with multiple orders and it will have various records for sales. And each sale will have only one product.

1. Day, orders, sales and Employee Salary



Each day will have various entries of orders data, sales data. Also the salary of employee table is also related to day table as salary is related to no.of days worked

1. Conceptual Model – Full diagram



The above conceptual model diagram is obtained by combining all the previous diagrams. Finally, we have eight tables namely Customer, Day, Employee, Employee salary, Orders, Product, Sales, Store.

In the next section, we have drawn the logical model diagrams for the above.

# Logical Model Diagrams

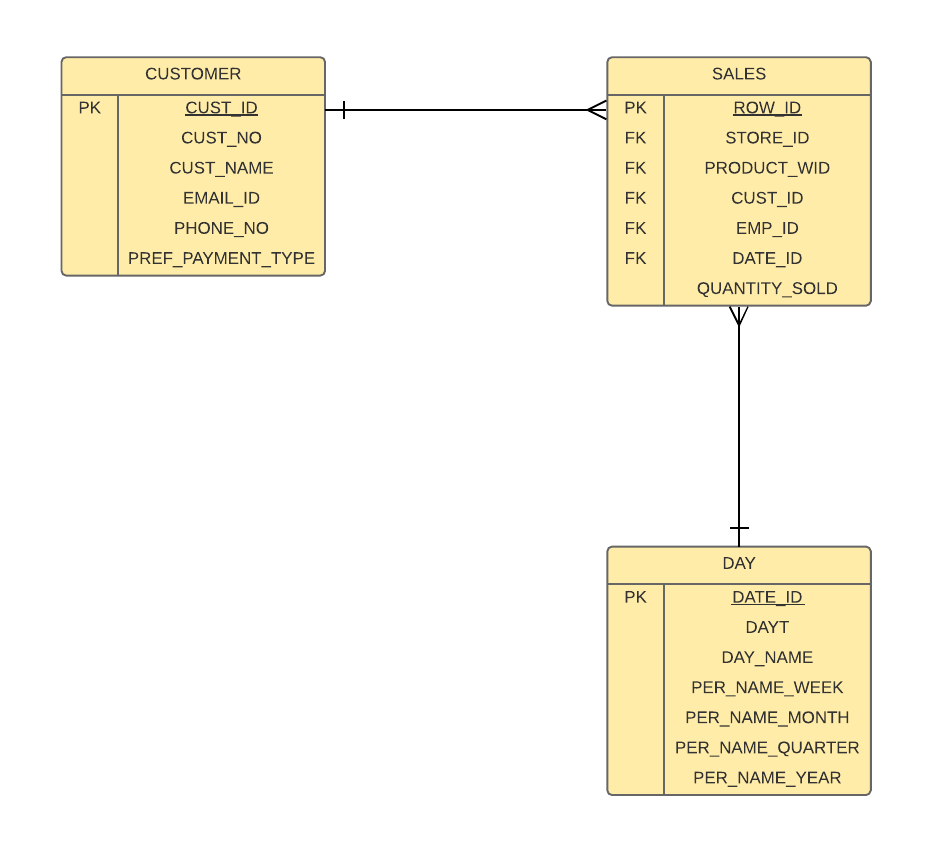
The next step after the database design process is the logical model diagrams. The definition of a logical model is as follows:

**Definition:** A logical data model or logical schema is a data model of a specific problem domain expressed independently of a database management product but in terms of data structures such as relational tables and columns.

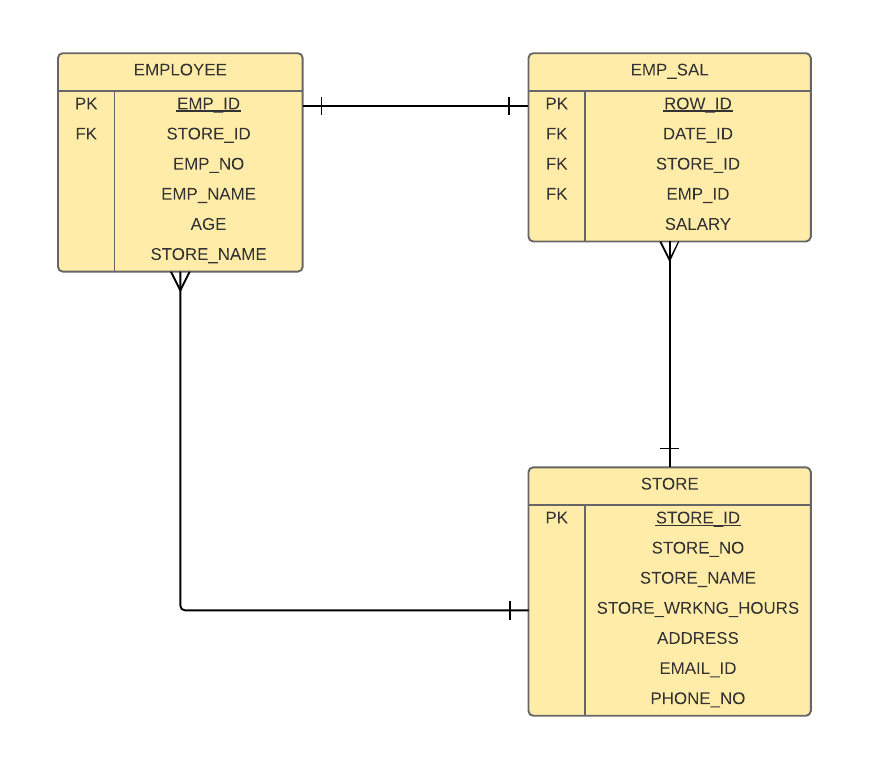
Below are the various combinations of the logical model diagrams for this database project.

* In each of the diagrams, the respective primary key (PK), Foreign key (FK) are indicated.
* Most of the column names are named appropriately to increase the readability.
* Also, we have used the same column name when naming a foreign key inorder to avoid confusions to the readers.

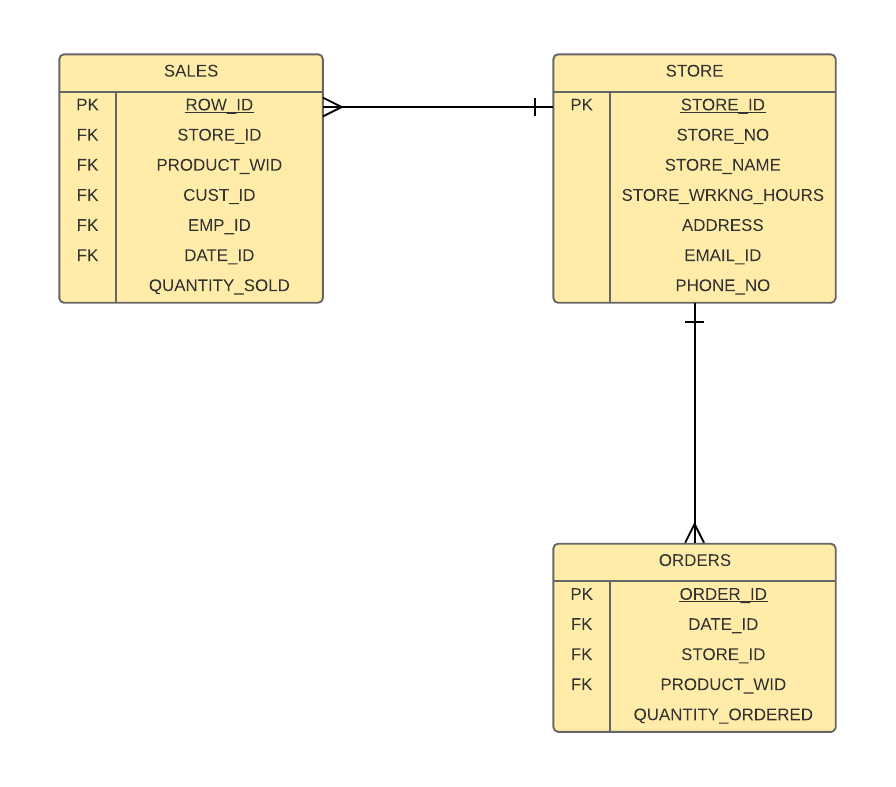
1. Customer, Sales and Day



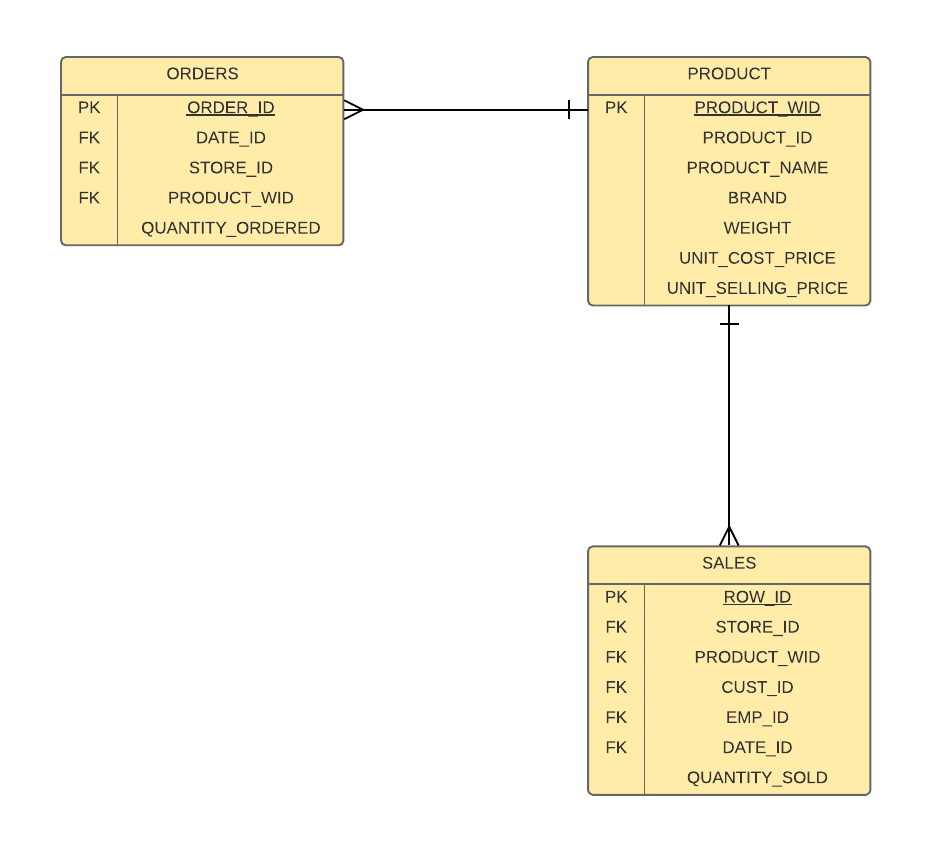
1. Employee, Employee Salary and Store



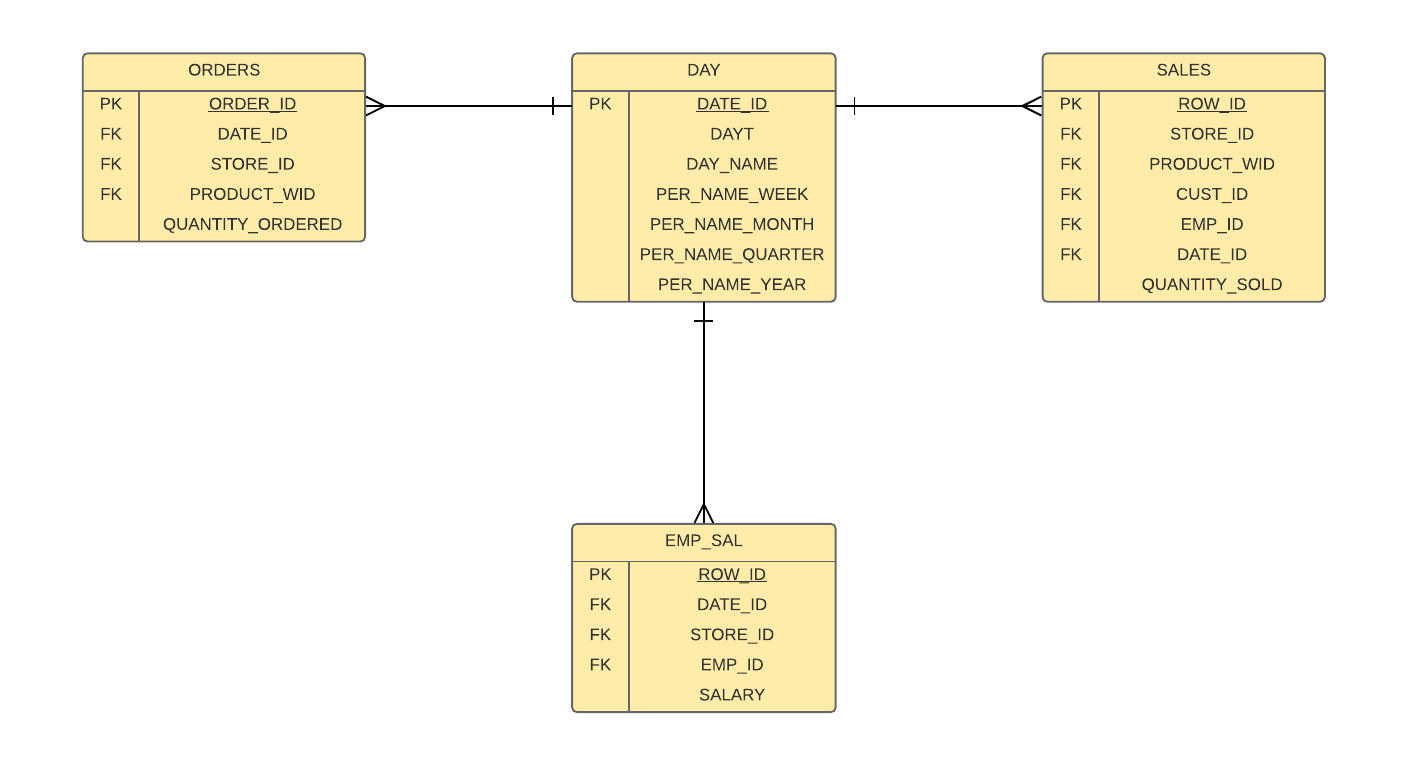
1. Store, Sales and Orders



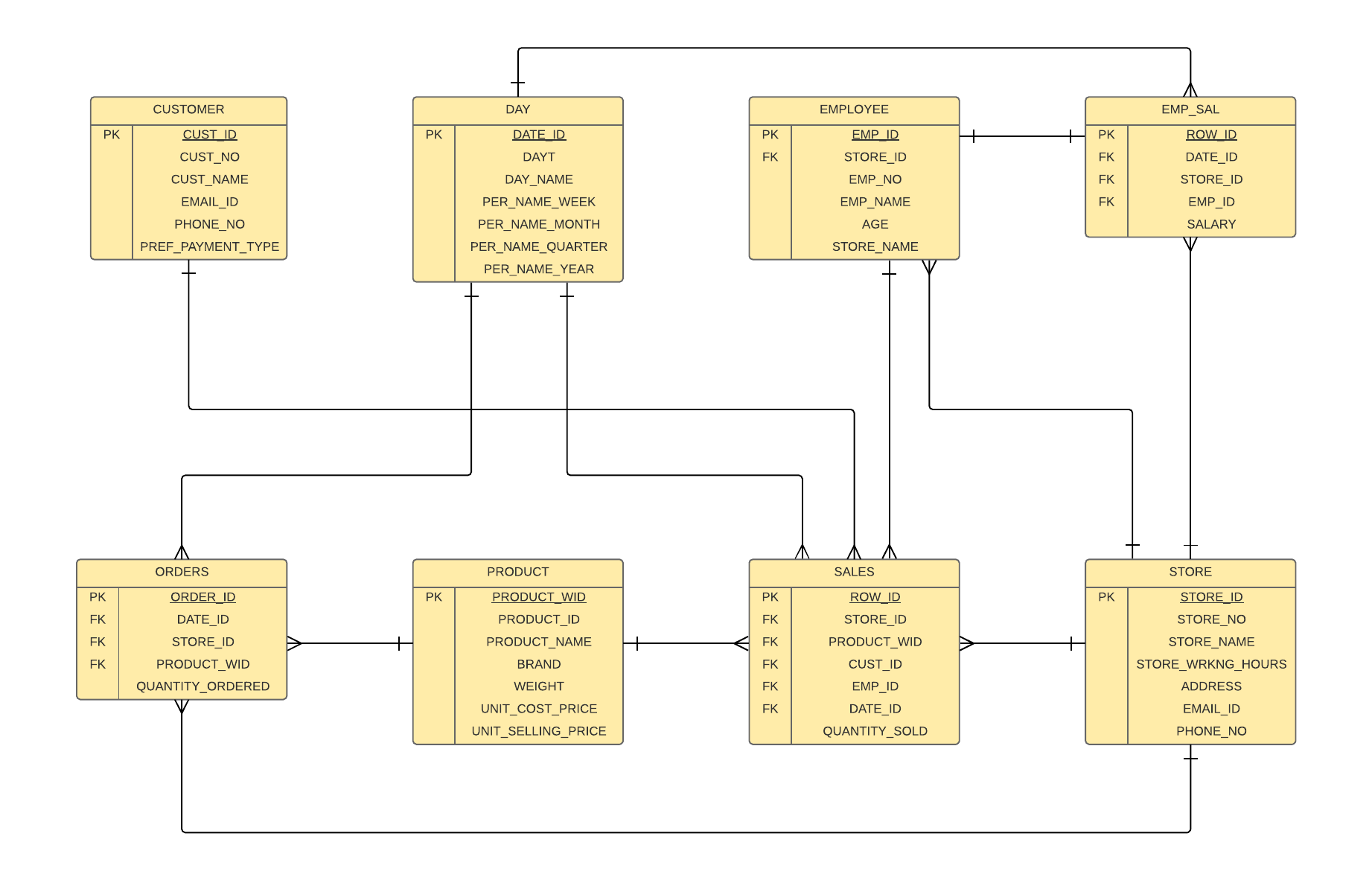
1. Product, Orders and Sales



1. Day, orders, sales and Employee Salary



1. Logical Model – Full diagram



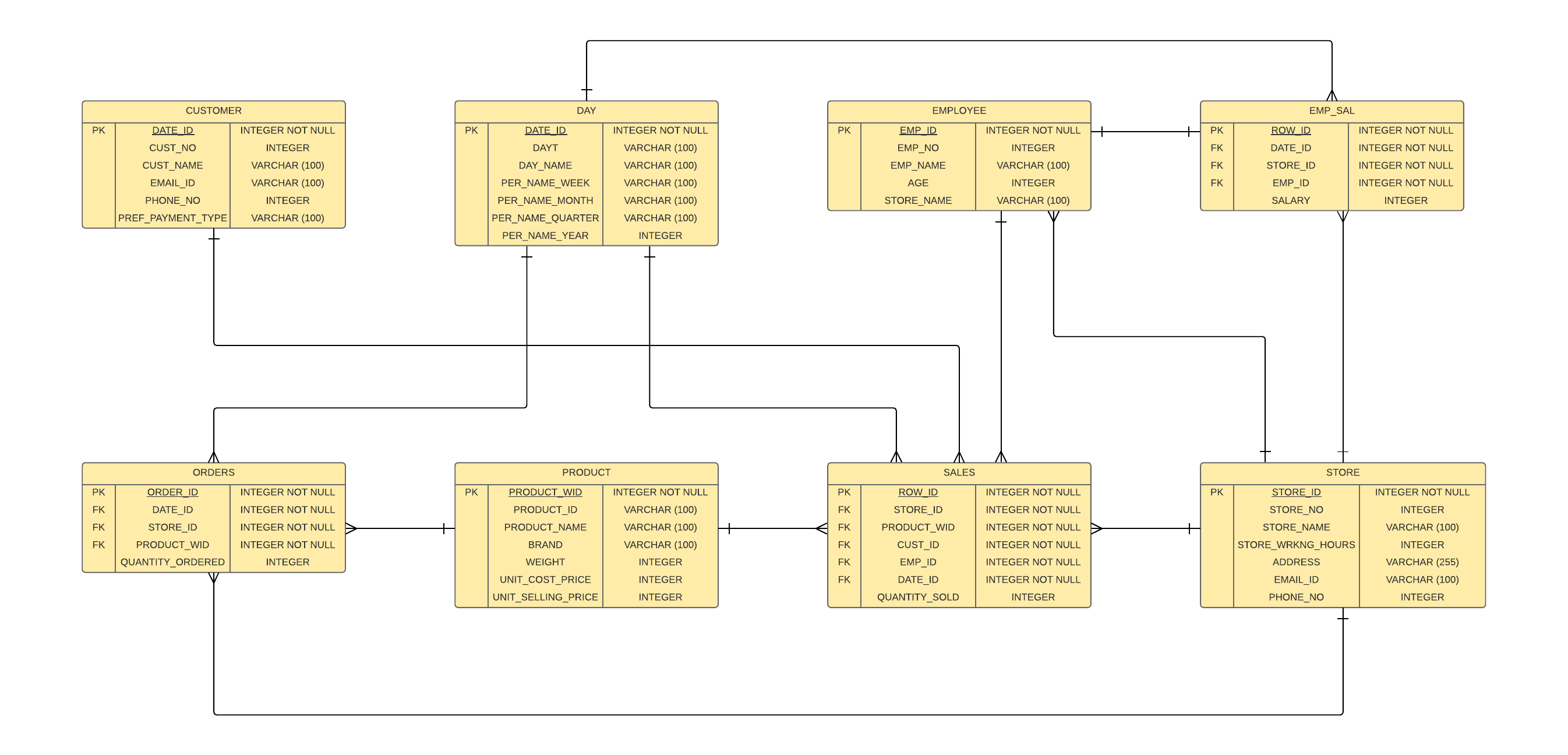
Similar to the conceptual model diagram in previous section, this logical model diagram is obtained by combining all the logical model diagrams above.

Next, we will proceed with the physical model diagrams and then converting the same into SQL and then populate data in our update-3

# Physical Model Diagram

**Physical model definition:** A physical database model shows all table structures, including column name, column data type, column constraints, primary key, foreign key, and relationships between tables. Features of a physical data model include: Specification all tables and columns. Foreign keys are used to identify relationships between tables.

The physical model diagram for the project:



# SQL Statements

The SQL statements are grouped into two parts in this case. One set is for creation of the tables and the second set is for insertion of the data.

## Table Creation statements

We have a total of 8 tables and below is the creation script for those tables:

Customer Table:

CREATE TABLE CUSTOMER

(

CUST\_ID INTEGER NOT NULL ,

CUST\_NO INTEGER ,

CUST\_NAME VARCHAR (100) ,

EMAIL\_ID VARCHAR (100) ,

PHONE\_NO INTEGER ,

PREF\_PAYMENT\_TYPE VARCHAR (100)

) ;

ALTER TABLE CUSTOMER ADD CONSTRAINT CUSTOMER\_PK PRIMARY KEY ( CUST\_ID ) ;

Day Table:

CREATE TABLE DAY

(

DATE\_ID INTEGER NOT NULL ,

DAYT VARCHAR (100) ,

DAY\_NAME VARCHAR (100),

PER\_NAME\_WEEK VARCHAR (100) ,

PER\_NAME\_MONTH VARCHAR (100),

PER\_NAME\_QUARTER VARCHAR (100),

PER\_NAME\_YEAR INTEGER

) ;

ALTER TABLE DAY ADD CONSTRAINT DAY\_PK PRIMARY KEY ( DATE\_ID ) ;

Employee Table:

CREATE TABLE EMPLOYEE

(

EMP\_ID INTEGER NOT NULL ,

EMP\_NO INTEGER ,

EMP\_NAME VARCHAR (100) ,

AGE INTEGER ,

STORE\_NAME VARCHAR (100)

) ;

ALTER TABLE EMPLOYEE ADD CONSTRAINT EMPLOYEE\_PK PRIMARY KEY ( EMP\_ID ) ;

Employee Salary table:

CREATE TABLE EMP\_SAL

(

ROW\_ID INTEGER NOT NULL ,

DATE\_ID INTEGER NOT NULL ,

STORE\_ID INTEGER NOT NULL ,

EMP\_ID INTEGER NOT NULL ,

SALARY INTEGER

) ;

ALTER TABLE EMP\_SAL ADD CONSTRAINT EMP\_SAL\_PK PRIMARY KEY ( ROW\_ID ) ;

Orders Table:

CREATE TABLE ORDERS

(

ROW\_ID INTEGER NOT NULL ,

DATE\_ID INTEGER NOT NULL ,

STORE\_ID INTEGER NOT NULL ,

PRODUCT\_ID INTEGER NOT NULL ,

QUANTITY\_ORDERED INTEGER

) ;

ALTER TABLE ORDERS ADD CONSTRAINT ORDERS\_PK PRIMARY KEY ( ROW\_ID ) ;

Product Table:

CREATE TABLE PRODUCT

(

PRODUCT\_WID INTEGER NOT NULL ,

PRODUCT\_ID VARCHAR (100) ,

PRODUCT\_NAME VARCHAR (100) ,

BRAND VARCHAR (100) ,

WEIGHT INTEGER ,

UNIT\_COST\_PRICE INTEGER ,

UNIT\_SELLING\_PRICE INTEGER

) ;

ALTER TABLE PRODUCT ADD CONSTRAINT PRODUCT\_PK PRIMARY KEY ( PRODUCT\_WID ) ;

Sales Table:

CREATE TABLE SALES

(

ROW\_ID INTEGER NOT NULL ,

STORE\_ID INTEGER NOT NULL ,

PRODUCT\_WID INTEGER NOT NULL ,

CUST\_ID INTEGER NOT NULL ,

EMP\_ID INTEGER NOT NULL ,

DATE\_ID INTEGER NOT NULL ,

QUANTITY\_SOLD INTEGER

) ;

ALTER TABLE SALES ADD CONSTRAINT SALES\_PK PRIMARY KEY ( ROW\_ID ) ;

Store Table:

CREATE TABLE STORE

(

STORE\_ID INTEGER NOT NULL ,

STORE\_NO INTEGER ,

STORE\_NAME VARCHAR (100) ,

STORE\_WRKNG\_HOURS INTEGER ,

ADDRESS VARCHAR (255) ,

EMAIL\_ID VARCHAR (100) ,

PHONE\_NO INTEGER

) ;

ALTER TABLE STORE ADD CONSTRAINT STORE\_PK PRIMARY KEY ( STORE\_ID ) ;

Creating the Foreign Keys:

ALTER TABLE EMP\_SAL ADD CONSTRAINT EMPLOYEE\_FK1 FOREIGN KEY ( DATE\_ID ) REFERENCES DAY ( DATE\_ID ) ;

ALTER TABLE EMP\_SAL ADD CONSTRAINT EMPLOYEE\_FK2 FOREIGN KEY ( EMP\_ID ) REFERENCES EMPLOYEE ( EMP\_ID ) ;

ALTER TABLE EMP\_SAL ADD CONSTRAINT EMPLOYEE\_FK3 FOREIGN KEY ( STORE\_ID ) REFERENCES STORE ( STORE\_ID ) ;

ALTER TABLE ORDERS ADD CONSTRAINT ORDER\_FK1 FOREIGN KEY ( DATE\_ID ) REFERENCES DAY ( DATE\_ID ) ;

ALTER TABLE ORDERS ADD CONSTRAINT ORDER\_FK2 FOREIGN KEY ( PRODUCT\_WID ) REFERENCES PRODUCT ( PRODUCT\_WID ) ;

ALTER TABLE ORDERS ADD CONSTRAINT ORDER\_FK3 FOREIGN KEY ( STORE\_ID ) REFERENCES STORE ( STORE\_ID ) ;

ALTER TABLE SALES ADD CONSTRAINT SALES\_FK1 FOREIGN KEY ( CUST\_ID ) REFERENCES CUSTOMER ( CUST\_ID ) ;

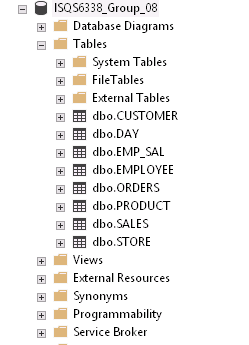
ALTER TABLE SALES ADD CONSTRAINT SALES\_FK2 FOREIGN KEY ( DATE\_ID ) REFERENCES DAY ( DATE\_ID ) ;

ALTER TABLE SALES ADD CONSTRAINT SALES\_FK3 FOREIGN KEY ( PRODUCT\_WID ) REFERENCES PRODUCT ( PRODUCT\_WID ) ;

ALTER TABLE SALES ADD CONSTRAINT SALES\_FK4 FOREIGN KEY ( STORE\_ID ) REFERENCES STORE ( STORE\_ID ) ;

ALTER TABLE SALES ADD CONSTRAINT SALES\_FK5 FOREIGN KEY ( EMP\_ID ) REFERENCES EMPLOYEE ( EMP\_ID ) ;

The above created tables are available in the ISQS6338\_Group\_08 schema as follows:



## Inserting the data

Once the tables are created, the next step is to insert the data into these tables. Below are some of the sample insert statements for each of the tables:

Insert into Customer Table:

INSERT INTO CUSTOMER VALUES (23262,1,'Candice Levy','candicelevy@yahoo.com',1434451018,'Credit Card');

INSERT INTO CUSTOMER VALUES (23263,2,'Xerxes Smith','xerxessmith@yahoo.com',1464351356,'Cash');

INSERT INTO CUSTOMER VALUES (23264,3,'Levi Douglas','levidouglas@yahoo.com',1402058219,'Cash');

INSERT INTO CUSTOMER VALUES (23265,4,'Uriel Benton','urielbenton@yahoo.com',1494804759,'Cash');

Insert into Employee Table:

INSERT INTO EMPLOYEE VALUES (1963,1963,'Arsenio Knowles',21,'Cleanzone');

INSERT INTO EMPLOYEE VALUES (1964,1964,'Noble Warner',41,'Freshbreeze');

INSERT INTO EMPLOYEE VALUES (1965,1965,'Melinda Cobb',45,'Freshbreeze');

INSERT INTO EMPLOYEE VALUES (1966,1966,'Silas Battle',28,'Freshbreeze');

Insert into Product Table:

INSERT INTO PRODUCT VALUES (2000800,'DETA800','Detafast Stain Remover - 800ml','DSR',800,6,9);

INSERT INTO PRODUCT VALUES (3000101,'SUPA101','Super Soft - Product Sample','SST',50,0.3,0);

Insert into Store Table:

INSERT INTO STORE VALUES (101,101,'Cleanzone',8,'PO 24, Lawrence, Kansas','cleanzoneinc@yahoo.com',1425551212);

INSERT INTO STORE VALUES (102,101,'Freshbreeze',7,'PO 85, Lebanon, Kansas','freshbrezeinc@yahoo.com',1425551233);

Insert into Day Table:

INSERT INTO DAY VALUES (20110101,'01-Jan-11','Saturday','2011 Week01','2011 / 01','2011 Q 1',2011);

INSERT INTO DAY VALUES (20110102,'02-Jan-11','Sunday','2011 Week02','2011 / 01','2011 Q 1',2011);

INSERT INTO DAY VALUES (20110103,'03-Jan-11','Monday','2011 Week02','2011 / 01','2011 Q 1',2011);

INSERT INTO DAY VALUES (20110104,'04-Jan-11','Tuesday','2011 Week02','2011 / 01','2011 Q 1',2011);

Insert into Sales table:

INSERT INTO SALES VALUES (1,101,2000200,23263,1951,20120706,73);

INSERT INTO SALES VALUES (2,101,2000100,23268,1952,20120712,82);

INSERT INTO SALES VALUES (3,101,1000100,23269,1953,20120603,116);

INSERT INTO SALES VALUES (4,101,2000200,23272,1954,20120731,71);

Insert into Employee Salary table:

INSERT INTO EMP\_SAL VALUES (1,20110131,101,1951,2525.33);

INSERT INTO EMP\_SAL VALUES (2,20110131,101,1952,2829.23);

INSERT INTO EMP\_SAL VALUES (3,20110131,101,1953,3987.78);

INSERT INTO EMP\_SAL VALUES (4,20110131,101,1954,2277.73);

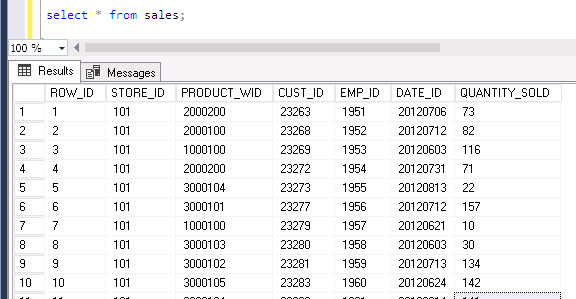
Insert into Orders table:

INSERT INTO ORDERS VALUES (1,20110102,101,3000101,150);

INSERT INTO ORDERS VALUES (2,20110102,102,3000105,230);

## Checking the data after insertion

We can check the successful insertion of data just by querying from the tables. For example,



# Narrative Behind projects data

“Data is the new oil,” Shivon Zilis, a partner with the venture capital firm Bloomberg Beta, said about data’s increasing value. Wherever there is an organized data, it can be used to do wonders. In the context of this project, we are storing the data of orders and sales of products, the employee and their salary details by store, etc. When we have data stored for a period of 2-3 years, the insights obtained from that data can be used to better plan our business, make plans to expand the business, etc. All these plans will turn out to be almost accurate as it is based on real time data. Imagine placing an order for products or thinking of opening a new store when you don’t have any previous knowledge and data to support your decision making. There will not be a defined path that you can choose. And we never know if it works well or Not. Thus, availability of previous data and understanding of it and gaining right insights is very much essential.

In the next sections, we see what kind of questions can be asked in this project that are of interest to data scientist and how the answers to those questions help in gaining actionable insights and effective decision making.

## Questions of interest to data scientists

Here are some of the sample list of questions that are of interest to data scientists:

* What is the No. of Employees by Store?
* What is the No. of Employees by month working in a store (With store working hours)
* Compare Employee’s Salary by Year
* Who are the Top 10 Employees with highest Salary in each year (Include Employee and Store details also)?
* Depict Store Level Details for each of the stores.
* Display Ordered Quantity and Total Cost for Selected Products by Year (Any 6 Products of user choice).
* Compare Total Cost and Ordered Quantity for a store at a month level.
* Display the all Product Details such as Name, Brand, Cost Price, Selling Price etc. Display Grand Total for all the metrics.
* Calculate and Display Profit % for Products.
* Which are the Top 10 days with Maximum Quantity Ordered (For Last 6 months only)? Use store Level Information.
* Who are the Top 10 Customers by Revenue?
* What is the No. of Customers and Total Revenue at brand level filtered for respective Month?
* What is Average Revenue, Average Quantity sold, Total Revenue by Product? Show this information Year Wise.
* Display the Products by Quantity Sold.
* Display Total Revenue, Quantity Sold at Store Level for different Years.

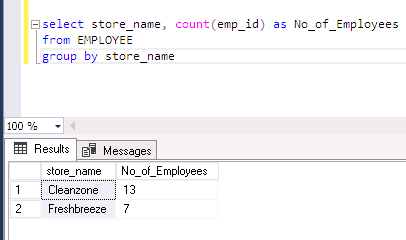
The answers to these questions can be obtained by using SQL queries on the database. When those answers are understood and analyzed, we can take right decisions.

**Also, a data scientist is not only interested in finding the insights from data, s/he is also interested in predicting the future sales, future revenue, which customers will purchase more in future etc. There come the concepts of predictive analytics which is an important area of data science.**

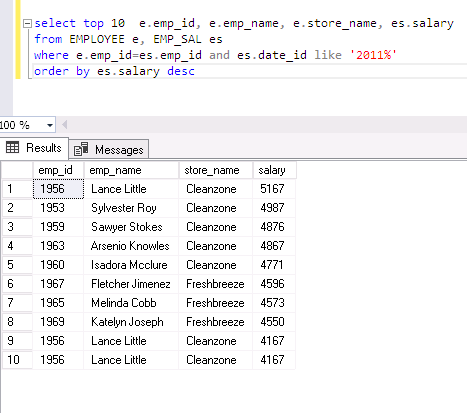
## Answering those questions using SQL

In this section of the document, we have solved some of the questions asked above which are of importance to data scientists. The answers to the questions are obtained by writing SQL statements on the projects data on MS SQL Server.

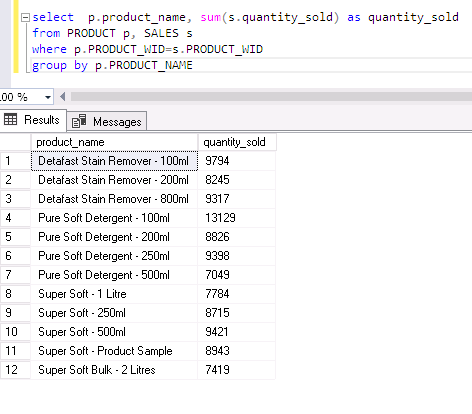
* What is the No. of Employees by Store?



* What is the No. of Employees by month working in a store in each year (With store working hours)



* Display the Products by Quantity Sold.



# Conclusions and Future Work

In future, this project can be further enhanced by adding more number of entities and by adding more attributes and capture more details. We can also introduce security to this project and the concept of views, etc.

And inorder to best utilize this data, we can plan to use software like Microsoft SSRS (Sequel Server Reporting Services) for generating the trend reports and graphs which will help us more to identify the trends in the data and thus obtain actionable insights from the same.