# SQL Assignment 1:

create database bikestores;

use database BIKESTORES;

create schema Production;

create schema Sales;

use schema Production;

-- Creating the tables Brands, Categories, Products, and Stocks in Production Schema And Customers, Orders, Order\_Items, Staffs, Stocks, and Stores in Sales Schema.

create table Brands

( Brand\_ID number(2) not null,

Brand\_name varchar(50),

Primary key(Brand\_ID)

);

create table Stocks

( Store\_ID number(1) not null,

Product\_ID number(3) not null,

Quantity number(2),

CONSTRAINT PK\_Store\_Product\_key PRIMARY KEY(Store\_ID,Product\_ID)

);

create table Products

( Product\_ID number(3) not null,

Product\_name varchar(100),

Brand\_ID number(2),

Category\_ID number(1),

Model\_year number(4),

List\_Price float,

Primary key(Product\_ID)

);

create table Categories

( Category\_ID number(1) not null,

Category\_name varchar(30),

Primary key(Category\_ID)

);

use schema sales;

create or replace table Order\_Items

( Order\_ID number(4) not null,

Item\_ID number(1) not null ,

Product\_ID number(3),

Quantity number(1),

List\_Price float,

Discount float,

CONSTRAINT PK\_Order\_Items PRIMARY KEY (Order\_ID,Item\_ID)

);

create table Orders

( Order\_ID number(4),

Customer\_ID number(4),

Order\_Status number(8),

Order\_date varchar(20),

Required\_date varchar(20),

Shipped\_date varchar(20),

Store\_ID number(1),

Staff\_ID number(2),

PRIMARY KEY(Order\_ID)

);

create or replace table Staffs

( Staff\_ID number(2) Primary KEY,

First\_name varchar(50),

Last\_name varchar(50),

Email varchar(120),

Phone varchar(20),

Active int default 0,

Store\_ID number(1),

Manager\_ID number(2)

);

select \* from staffs

where MANAGER\_ID is NULL;

create or replace table Stores

(Store\_ID number(1) identity(1,1),

Store\_name varchar(50),

Phone varchar(20),

Email varchar (120),

Street varchar(50),

City varchar(50),

State varchar(20),

Zip\_Code int,

primary key(Store\_ID)

);

create or replace table Customers

( Customer\_ID number(4) identity(1,1),

First\_name varchar(50),

Last\_name varchar(50),

Phone varchar(20),

Email varchar(100),

Street varchar(120),

City varchar(100),

State varchar(50),

Zip\_Code int,

PRIMARY KEY(Customer\_ID)

);

A screenshot of a computer

Description automatically generated

--2.FOREIGN KEY implementation

alter table Orders add FOREIGN KEY(Customer\_ID) REFERENCES customers(Customer\_ID);

alter table order\_items add FOREIGN KEY(Order\_ID) REFERENCES orders(Order\_ID);

alter table orders add FOREIGN KEY(Store\_ID) REFERENCES stores(Store\_ID);

alter table orders add FOREIGN KEY(Staff\_ID) REFERENCES staffs(Staff\_ID);

alter table staffs add FOREIGN KEY(Store\_ID) REFERENCES stores(Store\_ID);

alter table staffs add FOREIGN KEY(Manager\_ID) REFERENCES staffs(Staff\_ID);

use production;

alter table products add FOREIGN KEY(Category\_ID) REFERENCES BIKESTORES.PRODUCTION.CATEGORIES(Category\_ID);

alter table products add FOREIGN KEY(Brand\_ID) REFERENCES brands(Brand\_ID);

alter table sales.order\_items add FOREIGN KEY(Product\_ID) REFERENCES production.products(Product\_ID);

alter table production.stocks add FOREIGN KEY(Store\_ID) REFERENCES sales.stores(Store\_ID);

alter table stocks add FOREIGN KEY(Product\_ID ) REFERENCES products(Product\_ID);

--3.Does any of the table has missing or NULL value ? If yes which are those and what are their counts ?

use schema sales;

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

-- 4.Does the datasets has any DUPLICATE(identical rows) ? If yes â€“ can you just keep the first record

--and remove all rest if its possible without using any JOINS or WINDOW function

--Checking Duplicate in Table Brands

select \*, count(\*) as Duplicate

from BRANDS

group by Brand\_ID, Brand\_name;

--Checking Duplicate in Table BIKESTORES.PRODUCTION.CATEGORIES

select \*, count(\*) as Duplicate

from CATEGORIES

group by CATEGORY\_ID, CATEGORY\_NAME;

--Checking Duplicate in Table BIKESTORES.PRODUCTION.PRODUCTS

select \*, count(\*) as Duplicate

from BIKESTORES.PRODUCTION.PRODUCTS

group by PRODUCT\_ID, PRODUCT\_NAME, BRAND\_ID, CATEGORY\_ID, MODEL\_YEAR, LIST\_PRICE;

--Checking Duplicate in Table BIKESTORES.PRODUCTION.STOCKS

select \*, count(\*) as Duplicate

from BIKESTORES.PRODUCTION.STOCKS

group by STORE\_ID, PRODUCT\_ID, QUANTITY

--Checking Duplicate in Table BIKESTORES.SALES.CUSTOMERS

select \*, count(\*) as Duplicate

from BIKESTORES.SALES.CUSTOMERS

group by CUSTOMER\_ID, FIRST\_NAME, LAST\_NAME, PHONE, EMAIL, STREET, CITY, STATE, ZIP\_CODE;

--Checking Duplicate in Table BIKESTORES.SALES.ORDERS

select \*, count(\*) as Duplicate

from BIKESTORES.SALES.ORDERS

group by ORDER\_ID, CUSTOMER\_ID, ORDER\_STATUS, ORDER\_DATE, REQUIRED\_DATE, SHIPPED\_DATE, STORE\_ID, STAFF\_ID;

--Checking Duplicate in Table BIKESTORES.SALES.ORDER\_ITEMS

select \*, count(\*) as Duplicate

from BIKESTORES.SALES.ORDER\_ITEMS

group by ORDER\_ID, ITEM\_ID, PRODUCT\_ID, QUANTITY, LIST\_PRICE, DISCOUNT;

--Checking Duplicate in Table BIKESTORES.SALES.STAFFS

select \*, count(\*) as Duplicate

from BIKESTORES.SALES.STAFFS

group by STAFF\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE, ACTIVE, STORE\_ID, MANAGER\_ID;

--Checking Duplicate in Table BIKESTORES.SALES.STORES

select \*, count(\*) as Duplicate

from BIKESTORES.SALES.STORES

group by STORE\_ID, STORE\_NAME, PHONE, EMAIL, STREET, CITY, STATE, ZIP\_CODE;

--4.How many total serving customer BikeStore has ?

select count(customer\_id) as Total\_serving\_Customers from customers;

A close up of a website

Description automatically generated

-- 5.How many total orders are there ?

select count(order\_id) as Total\_Orders

from orders;

A close-up of a screen

Description automatically generated

--6.Which store has the highest number of sales ?

select distinct(STORE\_ID), Quantity

from BIKESTORES.PRODUCTION.STOCKS

order by Quantity desc;

select Store\_ID, Store\_name

from BIKESTORES.SALES.STORES

where store\_id=1 or store\_id=2 or store\_id=3;

A screenshot of a phone

Description automatically generated

--7.Which month the sales was highest and for which store ?

Ans

NOT known

--8.How many orders each customer has placed (give me top 10 customers)

select Top 10 CUSTOMER\_ID, count(ORDER\_ID) as Total\_Orders

from ORDERS

group by customer\_id

order by Total\_orders desc;

A white sheet with black text

Description automatically generated

--9. Which are the TOP 3 selling product ?

select top 3 Product\_ID, Item\_ID, Quantity, count(order\_id) as Total\_orders

from order\_items

group by Product\_ID,Item\_ID,Quantity

order by Total\_orders desc;

A white line on a white surface

Description automatically generated

select Product\_ID, Product\_name

from BIKESTORES.PRODUCTION.PRODUCTS

where PRODUCT\_ID in (16,6,8);

A screenshot of a product list

Description automatically generated

--13.Add a column TOTAL\_PRICE with appropriate data type into the sales.order\_items

alter table BIKESTORES.SALES.ORDER\_ITEMS

add column TOTAL\_PRICE float;

A white background with black text

Description automatically generated

--14.Calculate TOTAL\_PRICE = quantity \* list price and Update the value for all rows in the

--sales.order\_items table.

update BIKESTORES.SALES.ORDER\_ITEMS

set Total\_Price = Quantity\*List\_Price;

A white sheet with black text

Description automatically generated

select \* from BIKESTORES.SALES.ORDER\_ITEMS;

--14.What is the value of the TOTAL\_PRICE paid for all the sales.order\_items ?

select sum(total\_price) as total\_price\_paid

from BIKESTORES.SALES.ORDER\_ITEMS;

A screenshot of a phone

Description automatically generated