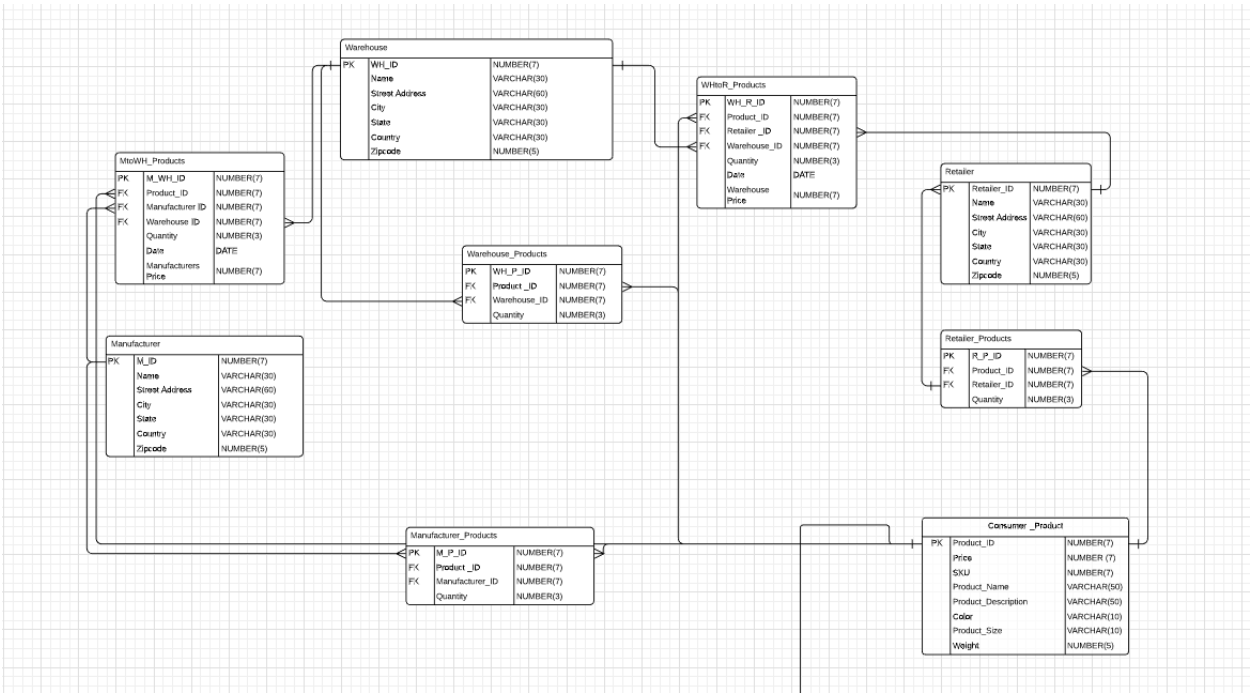
Data Management Final:

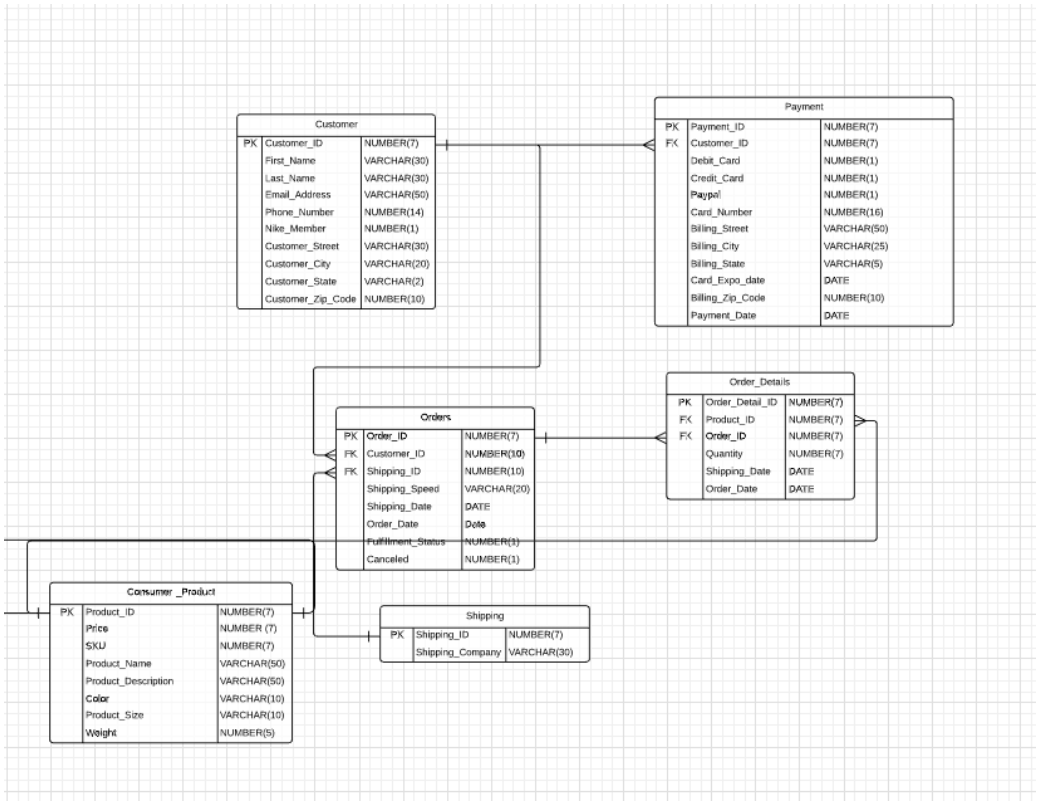
India Lindsay, Daniel Stern, Luke Bravo, Ram Kapistalam, Ali Sayyed

Data Models

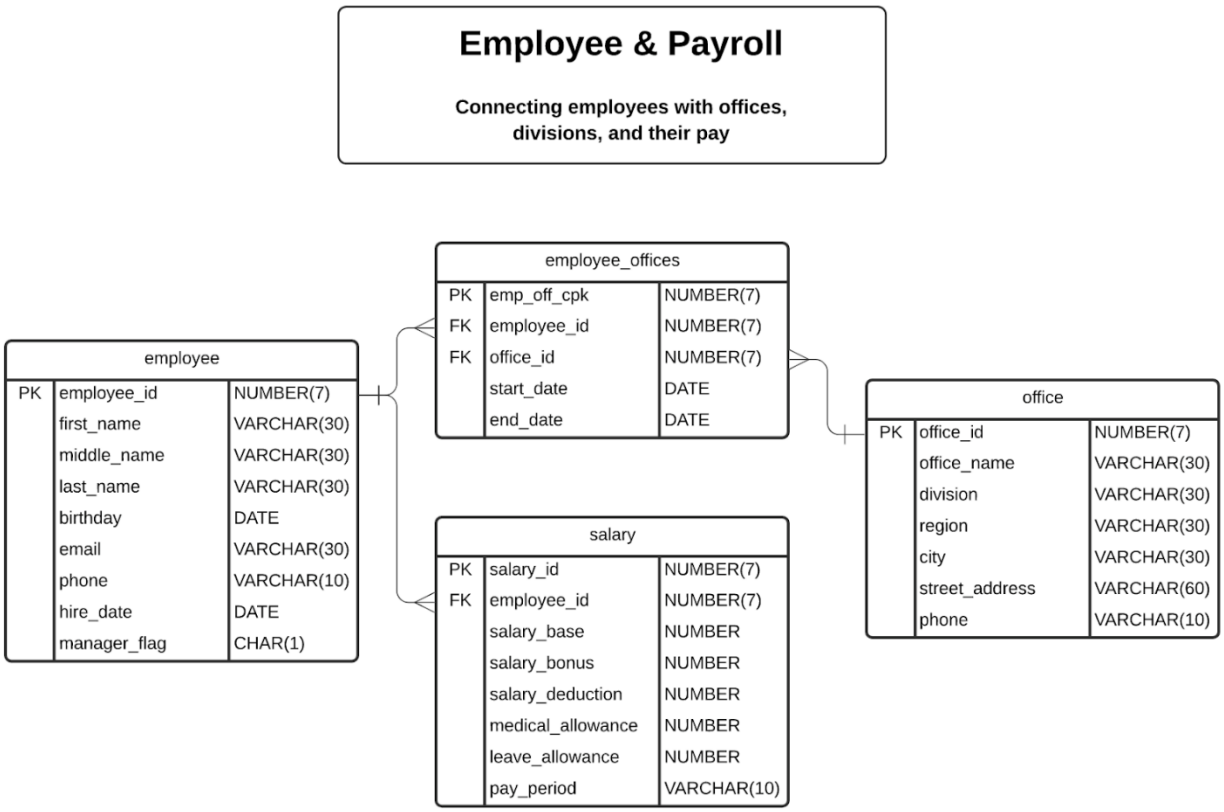
Inventory management:



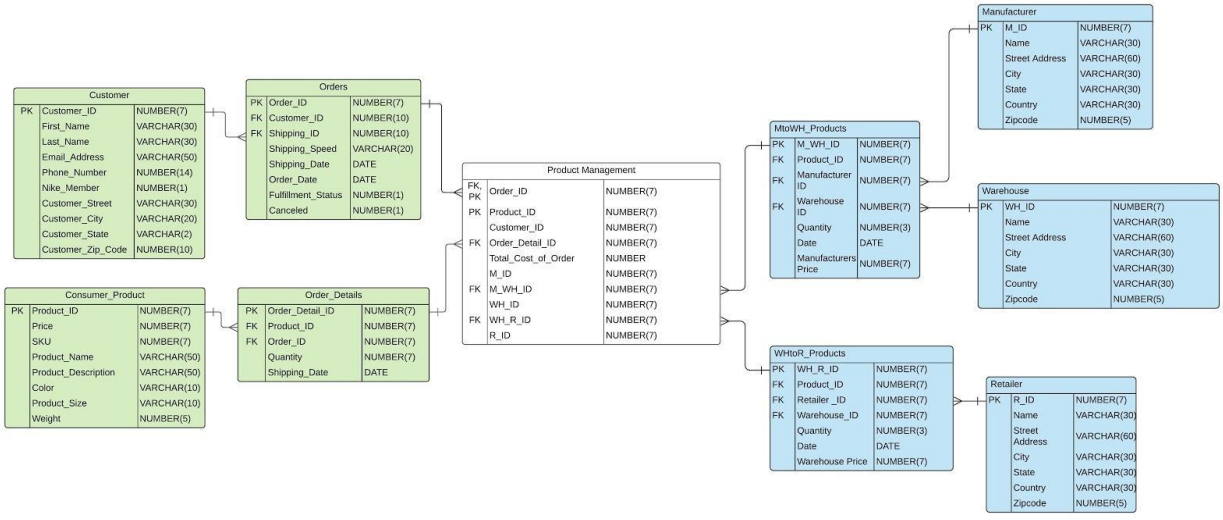
Order Processing:



Employee & Payroll



Data Warehouse:



DDL SCRIPT: CREATING AND POPULATING TABLES RELATED TO

1. EMPLOYEES, OFFICES, AND PAYROLL
2. ORDER PROCESSING
3. INVENTORY MANAGEMENT

```
--Dropping Tables

ALTER TABLE employee_offices
DROP CONSTRAINT fk_employee_id_empoff;
ALTER TABLE employee_offices
DROP CONSTRAINT fk_office_id;
DROP TABLE employee_offices;

ALTER TABLE salary
DROP CONSTRAINT fk_employee_id_sal;
DROP TABLE salary;

DROP TABLE employee;
DROP TABLE office;

DROP TABLE Order_Details;
DROP TABLE Orders;
DROP TABLE Payment;
DROP TABLE Shipping;
DROP TABLE Customer;


DROP TABLE Retailer_Products;
DROP TABLE Warehouse_Products;
DROP TABLE WHtoR_Products;
DROP TABLE Manufacturer_Products;
DROP TABLE MtoWH_Products;
DROP TABLE Warehouse;
```

```
DROP TABLE Retailer;
DROP TABLE Manufacturer;
DROP TABLE Consumer_Product;

--Dropping Sequences

DROP SEQUENCE employee_id_seq;
DROP SEQUENCE employee_offices_seq;
DROP SEQUENCE office_id_seq;
DROP SEQUENCE salary_id_seq;

DROP SEQUENCE cust_id_seq;
DROP SEQUENCE prod_id_seq;
DROP SEQUENCE shipping_id_seq;
DROP SEQUENCE payment_id_seq;
DROP SEQUENCE order_id_seq;
DROP SEQUENCE order_detail_id_seq;

DROP SEQUENCE wh_id_seq;
DROP SEQUENCE r_id_seq;
DROP SEQUENCE m_id_seq;
DROP SEQUENCE m_wh_id_seq;
DROP SEQUENCE m_p_seq;
DROP SEQUENCE wh_r_id_seq;
DROP SEQUENCE wh_p_seq;
DROP SEQUENCE r_p_seq;

-----------------------------------------

-- CREATE sequences
```

```
-----------------------------------------

--SELECT * FROM USER_CONSTRAINTS WHERE TABLE_NAME = "employee";

--CREATING SEQUENCES &
TABLES-----------------------------------------------------------------------------------------------

CREATE SEQUENCE employee_id_seq
   START WITH 1000000
   INCREMENT BY 1;

CREATE SEQUENCE cust_id_seq
START WITH 1000000 INCREMENT BY 1;

CREATE SEQUENCE prod_id_seq
START WITH 1000000 INCREMENT BY 1;

CREATE SEQUENCE shipping_id_seq
START WITH 1000000 INCREMENT BY 1;

CREATE SEQUENCE payment_id_seq
START WITH 1000000 INCREMENT BY 1;

CREATE SEQUENCE order_id_seq
START WITH 1000000 INCREMENT BY 1;

CREATE SEQUENCE order_detail_id_seq
START WITH 1000000 INCREMENT BY 1;
```

```
-- create warehouse sequence
CREATE SEQUENCE wh_id_seq
START WITH 1000000 INCREMENT BY 1;

-- create retailer sequence
CREATE SEQUENCE r_id_seq
START WITH 1000000 INCREMENT BY 1;

-- create manufacturer sequence
CREATE SEQUENCE m_id_seq
START WITH 1000000 INCREMENT BY 1;

-- create manufacturer to warehouse sequence
CREATE SEQUENCE m_wh_id_seq
START WITH 1000000 INCREMENT BY 1;

-- create manufacturer products in stock
CREATE SEQUENCE m_p_seq
START WITH 1 INCREMENT BY 1;

-- create warehouse to retailer sequence
CREATE SEQUENCE wh_r_id_seq
START WITH 1000000 INCREMENT BY 1;

-- create warehouse products in stock
CREATE SEQUENCE wh_p_seq
START WITH 1000000 INCREMENT BY 1;

-- create retailer products in stock
```

```
CREATE SEQUENCE r_p_seq
START WITH 1000000 INCREMENT BY 1;

CREATE SEQUENCE office_id_seq
   START WITH 1000000
   INCREMENT BY 1;



CREATE SEQUENCE employee_offices_seq
   START WITH 1000000
   INCREMENT BY 1;

CREATE SEQUENCE salary_id_seq
   START WITH 1000000
   INCREMENT BY 1;



-- EMPLOYEE TABLES

CREATE TABLE employee(
   employee_id     NUMBER(7)      DEFAULT employee_id_seq.NEXTVAL,
   first_name      VARCHAR(30)     NOT NULL,
   middle_name     VARCHAR(30),
   last_name       VARCHAR(30)     NOT NULL,
   birthday        DATE,
   email           VARCHAR(30)     NOT NULL,
   phone           VARCHAR(10)     NOT NULL,
   hire_date       DATE            DEFAULT SYSDATE,
   manager_flag    NUMBER          DEFAULT 0,
```

```
    CONSTRAINT pk_employee_id PRIMARY KEY (employee_id)
    );


CREATE TABLE office(
    office_id       NUMBER(7)       DEFAULT office_id_seq.NEXTVAL,
    office_name     VARCHAR(30)     NOT NULL UNIQUE,
    division        VARCHAR(30)     NOT NULL,
    region          VARCHAR(30),
    city            VARCHAR(30),
    street_address  VARCHAR(60)     NOT NULL,
    phone           VARCHAR(10)     NOT NULL,

    CONSTRAINT pk_office_id PRIMARY KEY (office_id)
    );


CREATE TABLE employee_offices(
    emp_off_cpk     NUMBER(7)       DEFAULT employee_offices_seq.NEXTVAL,
    employee_id     NUMBER(7),
    office_id       NUMBER(7),
    start_date      DATE NOT NULL,
    end_date        DATE,

    CONSTRAINT cpk_employee_offices PRIMARY KEY (employee_id, office_id),
    CONSTRAINT fk_employee_id_empoff FOREIGN KEY (employee_id) REFERENCES
employee(employee_id),
    CONSTRAINT fk_office_id FOREIGN KEY (office_id) REFERENCES office(office_id)
```

```
    );


CREATE TABLE salary(
    salary_id          NUMBER(7)      DEFAULT salary_id_seq.NEXTVAL,
    employee_id        NUMBER(7),
    salary_base        NUMBER         NOT NULL,
    salary_bonus       NUMBER,
    salary_deduction   NUMBER,
    medical_allowance  NUMBER,
    leave_allowance    NUMBER,
    pay_period         VARCHAR(10),

    CONSTRAINT pk_salary_id PRIMARY KEY (salary_id),
    CONSTRAINT fk_employee_id_sal FOREIGN KEY (employee_id) REFERENCES employee(employee_id)
    );

-- ORDER PROCESSING

CREATE TABLE Customer (
  Customer_ID NUMBER(7) DEFAULT cust_id_seq.NEXTVAL,
  First_Name VARCHAR(30)    NOT NULL,
  Last_Name VARCHAR(30)     NOT NULL,
  Email_Address VARCHAR(50) NOT NULL,
  Phone_Number NUMBER(14),
  Nike_Member NUMBER(1)    DEFAULT 0,
  Customer_Street VARCHAR(30)   NOT NULL,
  Customer_City VARCHAR(20)     NOT NULL,
  Customer_State VARCHAR(2)     NOT NULL,
```

```
  Customer_Zip_Code NUMBER(10)  NOT NULL,
  PRIMARY KEY (Customer_ID),
  CONSTRAINT email_length_check_
    CHECK (LENGTH(email_address)>=7)
);

CREATE TABLE Consumer_Product (
  Product_ID NUMBER(7)  DEFAULT prod_id_seq.NEXTVAL,
  Price NUMBER(7)   NOT NULL,
  SKU NUMBER(7)     NOT NULL,
  Product_Name VARCHAR(50)      NOT NULL,
  Product_Description VARCHAR(50),
  Color VARCHAR(10),
  Product_size VARCHAR(10),
  Weight NUMBER(5),
  PRIMARY KEY (Product_ID)
);

CREATE TABLE Shipping  (
  Shipping_ID NUMBER(7) DEFAULT shipping_id_seq.NEXTVAL,
  Shipping_Company VARCHAR(30),
  PRIMARY KEY (Shipping_ID)
);


CREATE TABLE Payment (
  Payment_ID NUMBER(7)  DEFAULT payment_id_seq.NEXTVAL,
  Customer_ID NUMBER(7),
  Debit_Card NUMBER(1)    DEFAULT 0,
```

```
  Credit_Card NUMBER(1)    DEFAULT 0,
  Paypal NUMBER(1)         DEFAULT 0,
  Card_Number NUMBER(16),
  Billing_Street VARCHAR(50)    NOT NULL,
  Billing_City VARCHAR(25)      NOT NULL,
  Billing_State VARCHAR(5)      NOT NULL,
  Card_Expo_date DATE,
  Billing_Zip_Code NUMBER(10)   NOT NULL,
  Payment_Date DATE,
  PRIMARY KEY (Payment_ID),
  CONSTRAINT payment_fk
          FOREIGN KEY (Customer_ID) REFERENCES Customer (Customer_ID),
  CONSTRAINT payment_type CHECK (debit_card+credit_card+paypal=1)
);

CREATE TABLE Orders (
  Order_ID NUMBER(7)    DEFAULT order_id_seq.NEXTVAL,
  Customer_ID NUMBER(10),
  Shipping_ID NUMBER(10),
  Shipping_Speed VARCHAR(20),
  Shipping_Date DATE,
  Order_Date DATE,
  Fulfillment_Status NUMBER(1),
  Canceled NUMBER(1)   DEFAULT 0,
  PRIMARY KEY (Order_ID),
  CONSTRAINT order_fk_customer
          FOREIGN KEY (Customer_ID) REFERENCES Customer (Customer_ID),
  CONSTRAINT order_fk_shipping
          FOREIGN KEY (Shipping_ID) REFERENCES Shipping (Shipping_ID),
```

```
  CONSTRAINT orders_date_check CHECK (shipping_date>=order_date)

);


CREATE TABLE Order_Details (
  Order_Detail_ID NUMBER(7) DEFAULT order_detail_id_seq.NEXTVAL,
  Product_ID NUMBER(7),
  Order_ID NUMBER(7),
  Quantity NUMBER,
  Shipping_Date DATE,
  Order_Date DATE,
  PRIMARY KEY (Order_Detail_ID),
  CONSTRAINT order_details_fk_consumer_product
                FOREIGN KEY (Product_ID) REFERENCES Consumer_Product (Product_ID),
  CONSTRAINT order_details_fk_Order_ID
                FOREIGN KEY (Order_ID) REFERENCES Orders (Order_ID),
  CONSTRAINT shipping_date_check CHECK (shipping_date>=order_date)
);

--- Inventory Warehouse

CREATE TABLE Warehouse (
  WH_ID NUMBER(7) default wh_id_seq.NEXTVAL,
  WH_Name VARCHAR(30),
  Address VARCHAR(60),
  City VARCHAR(30),
  WH_State VARCHAR(30),
  Country VARCHAR(30),
```

```
  Zipcode NUMBER(5),
  PRIMARY KEY (WH_ID)
);

CREATE TABLE Manufacturer (
  M_ID NUMBER(7) default m_id_seq.NEXTVAL,
  M_Name VARCHAR(30),
  Address VARCHAR(60),
  City VARCHAR(30),
  M_State VARCHAR(30),
  Country VARCHAR(30),
  Zipcode NUMBER(5),
  PRIMARY KEY (M_ID)
);

CREATE TABLE Retailer (
  R_ID NUMBER(7) default r_id_seq.NEXTVAL,
  R_Name VARCHAR(30),
  Address VARCHAR(60),
  City VARCHAR(30),
  R_State VARCHAR(30),
  Country VARCHAR(30),
  Zipcode NUMBER(5),
  PRIMARY KEY (R_ID)
);


CREATE TABLE MtoWH_Products (
  M_WH_ID NUMBER(7) default m_id_seq.NEXTVAL,
```

```
  Product_ID NUMBER(7) REFERENCES Consumer_Product(Product_ID),
  M_ID NUMBER(7) REFERENCES Manufacturer(M_ID),
  WH_ID NUMBER(7) REFERENCES Warehouse(WH_ID),
  Quantity NUMBER(3),
  M_WH_Date DATE,
  M_WH_Price NUMBER(7),
  PRIMARY KEY (M_WH_ID)
);

CREATE TABLE Manufacturer_Products (
  M_P_ID NUMBER(7) default m_p_seq.NEXTVAL,
  Product_ID NUMBER(7) REFERENCES Consumer_Product(Product_ID),
  M_ID NUMBER(7) REFERENCES Manufacturer(M_ID),
  Quantity NUMBER(3),
  PRIMARY KEY (M_P_ID)
);

CREATE TABLE WHtoR_Products (
  WH_R_ID NUMBER(7) default wh_r_id_seq.NEXTVAL,
  Product_ID NUMBER(7) REFERENCES Consumer_Product(Product_ID),
  R_ID NUMBER(7) REFERENCES Retailer(R_ID),
  WH_ID NUMBER(7) REFERENCES Warehouse(WH_ID),
  Quantity NUMBER(3),
  WH_R_Date DATE,
  WH_R_Price NUMBER(7),
  PRIMARY KEY (WH_R_ID)
);

CREATE TABLE Warehouse_Products (
```

```
  WH_P_ID NUMBER(7) default wh_p_seq.NEXTVAL,
  Product_ID NUMBER(7) REFERENCES Consumer_Product(Product_ID),
  WH_ID NUMBER(7) REFERENCES Warehouse(WH_ID),
  Quantity NUMBER(3),
  PRIMARY KEY (WH_P_ID)
);

CREATE TABLE Retailer_Products (
  R_P_ID NUMBER(7) default r_p_seq.NEXTVAL,
  Product_ID NUMBER(7) REFERENCES Consumer_Product(Product_ID),
  R_ID NUMBER(7) REFERENCES Retailer(R_ID),
  Quantity NUMBER(3),
  PRIMARY KEY (R_P_ID)
);

-- Populating tables


INSERT INTO employee

VALUES(DEFAULT,'Maarten','Luke','Bravo','02-DEC-97','luke.bravo@utexas.edu','3372964755',DEFAULT,DE
FAULT);
INSERT INTO employee

VALUES(DEFAULT,'John',NULL,'Doe','10-MAY-95','john.doe@gmail.com','5045551234',DEFAULT,DEFAULT);
INSERT INTO employee
    VALUES(DEFAULT,'Jane',NULL,'Doe','30-JUN-95','jane.doe@gmail.com','5045554321',DEFAULT,1);


INSERT INTO office
```

```
   VALUES(DEFAULT,'Order Processing','Orders Department','Texas','Austin','915 East 41st Street','5042965604');

INSERT INTO office

   VALUES(DEFAULT,'Procurement','Orders Department','Louisiana','New Orleans','31 McAlister Drive','5042964228');

INSERT INTO office

   VALUES(DEFAULT,'Retail Logistics','Retail Department','Louisiana','New Orleans','31 McAlister Drive','5042969898');


INSERT INTO employee_offices

   VALUES(DEFAULT,'1000000','1000000','21-FEB-2020',NULL);

INSERT INTO employee_offices

   VALUES(DEFAULT,'1000001','1000001','21-FEB-2020',NULL);

INSERT INTO employee_offices

   VALUES(DEFAULT,'1000002','1000001','21-FEB-2020',NULL);

INSERT INTO employee_offices

   VALUES(DEFAULT,'1000002','1000002','21-FEB-2020',NULL);


INSERT INTO salary

   VALUES(DEFAULT,'1000000',85000,12000,2000,NULL,NULL,'Annual');

INSERT INTO salary

   VALUES(DEFAULT,'1000001',90000,15000,0,NULL,NULL,'Annual');

INSERT INTO salary

   VALUES(DEFAULT,'1000002',100000,25000,5000,NULL,NULL,'Annual');


--Customer Table Inserts

insert into Customer (Customer_ID, First_Name, Last_Name, Email_Address, Phone_Number, Nike_Member, Customer_Street, Customer_City, Customer_State, Customer_Zip_Code) values (1000000, 'Weidar', 'Vaughan', 'wvaughan0@bing.com', 6024473485,0, '0537 Crest Line Pass', 'Gilbert', 'AZ', 85297);
```

```
insert into Customer (Customer_ID, First_Name, Last_Name, Email_Address, Phone_Number,
Nike_Member, Customer_Street, Customer_City, Customer_State, Customer_Zip_Code) values
(1000001, 'Kelwin', 'Alvarado', 'kalvarado1@nationalgeographic.com',6145872283,0, '0969 Texas Court',
'Columbus', 'OH', 43215);

insert into Customer (Customer_ID, First_Name, Last_Name, Email_Address, Phone_Number,
Nike_Member, Customer_Street, Customer_City, Customer_State, Customer_Zip_Code) values
(1000002, 'Connie', 'Dumbreck', 'cdumbreck2@google.com.br',2025291651,0, '23 Little Fleur Lane',
'Washington', 'DC', 20029);

insert into Customer (Customer_ID, First_Name, Last_Name, Email_Address, Phone_Number,
Nike_Member, Customer_Street, Customer_City, Customer_State, Customer_Zip_Code) values
(1000003, 'Arley', 'Chilley', 'achilley3@sciencedaily.com',5057742734,0, '2819 Eagan Hill', 'Albuquerque',
'NM',87115);

insert into Customer (Customer_ID, First_Name, Last_Name, Email_Address, Phone_Number,
Nike_Member, Customer_Street, Customer_City, Customer_State, Customer_Zip_Code) values
(1000004, 'Dylan', 'Acock', 'dacock4@symantec.com',9175274976,0, '9650 Golf Course Street', 'New York
City', 'NY',10155);

insert into Customer (Customer_ID, First_Name, Last_Name, Email_Address, Phone_Number,
Nike_Member, Customer_Street, Customer_City, Customer_State, Customer_Zip_Code) values
(1000005, 'Harald', 'Brisbane', 'hbrisbane5@ibm.com',5139743867,0, '90 Melvin Junction', 'Cincinnati',
'OH',45243);

insert into Customer (Customer_ID, First_Name, Last_Name, Email_Address, Phone_Number,
Nike_Member, Customer_Street, Customer_City, Customer_State, Customer_Zip_Code) values
(1000006, 'Janessa', 'Whightman', 'jwhightman6@chicagotribune.com',9721771046,0, '9374 Cody Pass',
'Dallas', 'TX',75236);

insert into Customer (Customer_ID, First_Name, Last_Name, Email_Address, Phone_Number,
Nike_Member, Customer_Street, Customer_City, Customer_State, Customer_Zip_Code) values
(1000007, 'Idette', 'Thornborrow', 'ithornborrow7@blogger.com',4438729297,0, '8 Hazelcrest Road',
'Annapolis', 'MD',21405);

insert into Customer (Customer_ID, First_Name, Last_Name, Email_Address, Phone_Number,
Nike_Member, Customer_Street, Customer_City, Customer_State, Customer_Zip_Code) values
(1000008, 'Kelby', 'Brabender', 'kbrabender8@va.gov',5403878433,0, '1 Valley Edge Place', 'Roanoke',
'VA',24020);

insert into Customer (Customer_ID, First_Name, Last_Name, Email_Address, Phone_Number,
Nike_Member, Customer_Street, Customer_City, Customer_State, Customer_Zip_Code) values
(1000009, 'Jaimie', 'Gilffilland', 'jgilffilland9@mayoclinic.com',8586135660,0, '42704 Sunfield Road', 'San
Diego', 'CA',92127);


--Consumer Product Table Inserts
```

insert into Consumer_Product (Product_id, Price, SKU, Product_Name, Product_Description, Color, Product_Size, Weight) values (1000000, 559.86, 1000000, 'Nike Air Zoom Alphafly', 'Shoe', 'Maroon', 5, 16);

insert into Consumer_Product (Product_id, Price, SKU, Product_Name, Product_Description, Color, Product_Size, Weight) values (1000001, 796.17, 1000001, 'Nike Air Force 1', 'Shoe', 'Pink', 13, 18);

insert into Consumer_Product (Product_id, Price, SKU, Product_Name, Product_Description, Color, Product_Size, Weight) values (1000002, 247.47, 1000002, 'Nike Blazer Mid ''77 SE', 'Shoe', 'Cyan', 12, 20);

insert into Consumer_Product (Product_id, Price, SKU, Product_Name, Product_Description, Color, Product_Size, Weight) values (1000003, 655.88, 1000003, 'Nike Metcon 6 AMP', 'Shoe', 'Orange', 12, 11);

insert into Consumer_Product (Product_id, Price, SKU, Product_Name, Product_Description, Color, Product_Size, Weight) values (1000004, 961.38, 1000004, 'Nike Run Division', 'Jacket', 'Blue', 'Large', 16);

insert into Consumer_Product (Product_id, Price, SKU, Product_Name, Product_Description, Color, Product_Size, Weight) values (1000005, 203.62, 1000005, 'Nike NFL Dri-FIT', 'Shirt', 'Yellow', 'Medium', 16);

insert into Consumer_Product (Product_id, Price, SKU, Product_Name, Product_Description, Color, Product_Size, Weight) values (1000006, 616.53, 1000006, 'Nike Air VaporMax 2020 FK MS', 'Shoe', 'Maroon', 12, 20);

insert into Consumer_Product (Product_id, Price, SKU, Product_Name, Product_Description, Color, Product_Size, Weight) values (1000007, 395.93, 1000007, 'Nike Air Monarch IV', 'Shoe', 'White', 5, 15);

insert into Consumer_Product (Product_id, Price, SKU, Product_Name, Product_Description, Color, Product_Size, Weight) values (1000008, 624.6, 1000008, 'Nike Air Zoom Tempo Next% FlyEase', 'Shoe', 'Cyan', 12, 14);

insert into Consumer_Product (Product_id, Price, SKU, Product_Name, Product_Description, Color, Product_Size, Weight) values (1000009, 196.5, 1000009, 'Nike React Infinity Run Flyknit', 'Shoe', 'Blue', 13, 5);

--Shipping Table Inserts

insert into Shipping (Shipping_ID, Shipping_Company) values (1000000, 'Brown and Sons');

insert into Shipping (Shipping_ID, Shipping_Company) values (1000001, 'Olson-Wolff');

insert into Shipping (Shipping_ID, Shipping_Company) values (1000002, 'Harris LLC');

insert into Shipping (Shipping_ID, Shipping_Company) values (1000003, 'Willms Group');

insert into Shipping (Shipping_ID, Shipping_Company) values (1000004, 'Kub-Erdman');

insert into Shipping (Shipping_ID, Shipping_Company) values (1000005, 'Champlin-Mraz');

insert into Shipping (Shipping_ID, Shipping_Company) values (1000006, 'Ryan-Quigley');

```
insert into Shipping (Shipping_ID, Shipping_Company) values (1000007, 'Treutel-Ratke');

insert into Shipping (Shipping_ID, Shipping_Company) values (1000008, 'Pouros Group');

insert into Shipping (Shipping_ID, Shipping_Company) values (1000009, 'Kutch, Parisian and Adams');



--Payment Table Inserts

insert into Payment (Payment_ID, Customer_ID, Debit_Card, Credit_Card, Paypal, Card_Number,
Billing_Street, Billing_City, Billing_State, Card_Expo_date, Billing_Zip_Code, Payment_Date) values
(1000000, 1000009, 0, 1, 0, 3800726035908436, '9504 Transport Hill', 'El Paso', 'TX', '01-SEP-2028',
'79955', '13-JAN-2020');

insert into Payment (Payment_ID, Customer_ID, Debit_Card, Credit_Card, Paypal, Card_Number,
Billing_Street, Billing_City, Billing_State, Card_Expo_date, Billing_Zip_Code, Payment_Date) values
(1000001, 1000008, 1, 0, 0, 832865484039435, '456 Farmco Hill', 'Chattanooga', 'TN', '29-NOV-2023',
'37410', '4-JUL-2020');

insert into Payment (Payment_ID, Customer_ID, Debit_Card, Credit_Card, Paypal, Card_Number,
Billing_Street, Billing_City, Billing_State, Card_Expo_date, Billing_Zip_Code, Payment_Date) values
(1000002, 1000008, 0, 0, 1, 3485926228644878, '59727 Cottonwood Crossing', 'Dallas', 'TX',
'2-JAN-2020', '75342', '19-MAY-2020');

insert into Payment (Payment_ID, Customer_ID, Debit_Card, Credit_Card, Paypal, Card_Number,
Billing_Street, Billing_City, Billing_State, Card_Expo_date, Billing_Zip_Code, Payment_Date) values
(1000003, 1000007, 1, 0, 0, 5845985446918227, '52671 Coleman Circle', 'Tallahassee', 'FL',
'19-FEB-2022', '32399', '5-NOV-2020');

insert into Payment (Payment_ID, Customer_ID, Debit_Card, Credit_Card, Paypal, Card_Number,
Billing_Street, Billing_City, Billing_State, Card_Expo_date, Billing_Zip_Code, Payment_Date) values
(1000004, 1000000, 1, 0, 0, 4472148597127115, '68 Waxwing Way', 'Saint Petersburg', 'FL',
'23-SEP-2026', '33731', '4-NOV-2020');

insert into Payment (Payment_ID, Customer_ID, Debit_Card, Credit_Card, Paypal, Card_Number,
Billing_Street, Billing_City, Billing_State, Card_Expo_date, Billing_Zip_Code, Payment_Date) values
(1000005, 1000000, 0, 1, 0, 1882322401072074, '33 Vera Drive', 'Pittsburgh', 'PA', '9-MAY-2029', '15261',
'1-APR-2020');

insert into Payment (Payment_ID, Customer_ID, Debit_Card, Credit_Card, Paypal, Card_Number,
Billing_Street, Billing_City, Billing_State, Card_Expo_date, Billing_Zip_Code, Payment_Date) values
(1000006, 1000008, 1, 0, 0, 2831582153059070, '001 Golf Center', 'Chicago', 'IL', '31-Mar-2021', '60669',
'1-SEP-2020');

insert into Payment (Payment_ID, Customer_ID, Debit_Card, Credit_Card, Paypal, Card_Number,
Billing_Street, Billing_City, Billing_State, Card_Expo_date, Billing_Zip_Code, Payment_Date) values
```

(1000007, 1000001, 1, 0, 0, 1145732971367696, '416 Mockingbird Drive', 'Saginaw', 'MI', '19-DEC-2027', '48604', '5-JAN-2020');

insert into Payment (Payment_ID, Customer_ID, Debit_Card, Credit_Card, Paypal, Card_Number, Billing_Street, Billing_City, Billing_State, Card_Expo_date, Billing_Zip_Code, Payment_Date) values (1000008, 1000000, 0, 0, 1, 5288957928264541, '0 Moulton Pass', 'Reno', 'NV', '18-SEP-2025', '89510', '19-JUL-2020');

insert into Payment (Payment_ID, Customer_ID, Debit_Card, Credit_Card, Paypal, Card_Number, Billing_Street, Billing_City, Billing_State, Card_Expo_date, Billing_Zip_Code, Payment_Date) values (1000009, 1000004, 0, 0, 1, 3425471170323776, '88066 Sheridan Parkway', 'Montgomery', 'AL', '18-JUN-2025', '36177', '18-OCT-2020');

--Orders Table Inserts

insert into Orders (Order_ID, Customer_ID, Shipping_ID, Shipping_Speed, Shipping_Date, Order_Date, Fulfillment_Status, Canceled) values (1000000, 1000008, 1000006, '3-5 Days Shipping', '14-APR-2019','12-APR-2019', 0, 0);

insert into Orders (Order_ID, Customer_ID, Shipping_ID, Shipping_Speed, Shipping_Date, Order_Date, Fulfillment_Status, Canceled) values (1000001, 1000000, 1000007, '3-5 Days Shipping', '9-DEC-2018','6-DEC-2018', 1, 0);

insert into Orders (Order_ID, Customer_ID, Shipping_ID, Shipping_Speed, Shipping_Date, Order_Date, Fulfillment_Status, Canceled) values (1000002, 1000004, 1000009, 'Next Day', '2-MAY-2019','2-MAY-2019', 1, 0);

insert into Orders (Order_ID, Customer_ID, Shipping_ID, Shipping_Speed, Shipping_Date, Order_Date, Fulfillment_Status, Canceled) values (1000003, 1000004, 1000003, 'Next Day', '12-JAN-2019','9-JAN-2019', 0, 1);

insert into Orders (Order_ID, Customer_ID, Shipping_ID, Shipping_Speed, Shipping_Date, Order_Date, Fulfillment_Status, Canceled) values (1000004, 1000008, 1000005, '7-10 Days Shipping', '12-JUL-2019','10-JUL-2019', 1, 0);

insert into Orders (Order_ID, Customer_ID, Shipping_ID, Shipping_Speed, Shipping_Date, Order_Date, Fulfillment_Status, Canceled) values (1000005, 1000005, 1000009, 'Next Day', '9-FEB-2019','8-FEB-2019', 1, 0);

insert into Orders (Order_ID, Customer_ID, Shipping_ID, Shipping_Speed, Shipping_Date, Order_Date, Fulfillment_Status, Canceled) values (1000006, 1000006, 1000004, '7-10 Days Shipping', '15-MAY-2019','11-MAY-2019', 1, 0);

insert into Orders (Order_ID, Customer_ID, Shipping_ID, Shipping_Speed, Shipping_Date, Order_Date, Fulfillment_Status, Canceled) values (1000007, 1000003, 1000009, 'Next Day', '14-NOV-2018','14-NOV-2018', 1, 0);

insert into Orders (Order_ID, Customer_ID, Shipping_ID, Shipping_Speed, Shipping_Date, Order_Date, Fulfillment_Status, Canceled) values (1000008, 1000004, 1000004, '3-5 Days Shipping', '28-FEB-2019','24-FEB-2019', 1, 0);

insert into Orders (Order_ID, Customer_ID, Shipping_ID, Shipping_Speed, Shipping_Date, Order_Date, Fulfillment_Status, Canceled) values (1000009, 1000006, 1000006, '3-5 Days Shipping', '16-MAY-2019','14-MAY-2019', 1, 0);

--Order Details Table Inserts

insert into Order_Details (Order_Detail_ID, Product_ID, Order_ID, Quantity, Shipping_Date, Order_Date) values (1000000, 1000006, 1000002, 50, '27-NOV-2018', '15-NOV-2018');

insert into Order_Details (Order_Detail_ID, Product_ID, Order_ID, Quantity, Shipping_Date, Order_Date) values (1000001, 1000001, 1000009, 57, '15-OCT-2019', '14-OCT-2019');

insert into Order_Details (Order_Detail_ID, Product_ID, Order_ID, Quantity, Shipping_Date, Order_Date) values (1000002, 1000000, 1000009, 10, '18-JUL-2018', '8-JUL-2018');

insert into Order_Details (Order_Detail_ID, Product_ID, Order_ID, Quantity, Shipping_Date, Order_Date) values (1000003, 1000009, 1000003, 8, '8-SEP-2019', '5-SEP-2019');

insert into Order_Details (Order_Detail_ID, Product_ID, Order_ID, Quantity, Shipping_Date, Order_Date) values (1000004, 1000003, 1000002, 56, '31-JAN-2019', '30-JAN-2019');

insert into Order_Details (Order_Detail_ID, Product_ID, Order_ID, Quantity, Shipping_Date, Order_Date) values (1000005, 1000008, 1000004, 86, '11-JAN-2019', '29-DEC-2018');

insert into Order_Details (Order_Detail_ID, Product_ID, Order_ID, Quantity, Shipping_Date, Order_Date) values (1000006, 1000002, 1000000, 72, '29-MAY-2019', '21-MAY-2019');

insert into Order_Details (Order_Detail_ID, Product_ID, Order_ID, Quantity, Shipping_Date, Order_Date) values (1000007, 1000006, 1000001, 45, '18-JUN-2019', '13-JUN-2019');

insert into Order_Details (Order_Detail_ID, Product_ID, Order_ID, Quantity, Shipping_Date, Order_Date) values (1000008, 1000000, 1000002, 90, '20-OCT-2019', '10-OCT-2019');

insert into Order_Details (Order_Detail_ID, Product_ID, Order_ID, Quantity, Shipping_Date, Order_Date) values (1000009, 1000000, 1000006, 27, '3-JUL-2019', '1-JUL-2019');

--Warehouse Table Inserts

insert into Warehouse (WH_Name, Address, City, WH_State, Country, Zipcode) values ('Leifer', '1 Esker Crossing', 'Corpus Christi', 'Texas', 'United States', '78410');

insert into Warehouse (WH_Name, Address, City, WH_State, Country, Zipcode) values ('Dance', '08 Johnson Way', 'New York City', 'New York', 'United States', '10110');

insert into Warehouse (WH_Name, Address, City, WH_State, Country, Zipcode) values ('Casserly', '8 Dottie Circle', 'Seattle', 'Washington', 'United States', '98195');

insert into Warehouse (WH_Name, Address, City, WH_State, Country, Zipcode) values ('Kohrding', '19 Morning Plaza', 'Fort Lauderdale', 'Florida', 'United States', '33355');

insert into Warehouse (WH_Name, Address, City, WH_State, Country, Zipcode) values ('Knewstubb', '963 Farragut Place', 'Huntington', 'West Virginia', 'United States', '25705');

insert into Warehouse (WH_Name, Address, City, WH_State, Country, Zipcode) values ('Serman', '308 Hallows Road', 'Dallas', 'Texas', 'United States', '75353');

insert into Warehouse (WH_Name, Address, City, WH_State, Country, Zipcode) values ('Copner', '3 Delladonna Junction', 'Miami', 'Florida', 'United States', '33129');

insert into Warehouse (WH_Name, Address, City, WH_State, Country, Zipcode) values ('Lomath', '5678 Spenser Court', 'Erie', 'Pennsylvania', 'United States', '16565');

insert into Warehouse (WH_Name, Address, City, WH_State, Country, Zipcode) values ('Braidon', '6277 Mallory Lane', 'Honolulu', 'Hawaii', 'United States', '96845');

insert into Warehouse (WH_Name, Address, City, WH_State, Country, Zipcode) values ('Scotson', '7 Oak Valley Alley', 'New York City', 'New York', 'United States', '10019');

--Retailer Table Inserts

insert into Retailer (R_Name, Address, City, R_State, Country, Zipcode) values ('McPeake', '06 1st Plaza', 'Norcross', 'Georgia', 'United States', '30092');

insert into Retailer (R_Name, Address, City, R_State, Country, Zipcode) values ('Bissex', '633 Arizona Street', 'Dearborn', 'Michigan', 'United States', '48126');

insert into Retailer (R_Name, Address, City, R_State, Country, Zipcode) values ('oldey', '48394 Anthes Street', 'Milwaukee', 'Wisconsin', 'United States', '53215');

insert into Retailer (R_Name, Address, City, R_State, Country, Zipcode) values ('Poulson', '3 5th Point', 'Atlanta', 'Georgia', 'United States', '30380');

insert into Retailer (R_Name, Address, City, R_State, Country, Zipcode) values ('Woollacott', '99 Fieldstone Lane', 'Metairie', 'Louisiana', 'United States', '70005');

insert into Retailer (R_Name, Address, City, R_State, Country, Zipcode) values ('Teggin', '61282 Browning Avenue', 'Dayton', 'Ohio', 'United States', '45432');

insert into Retailer (R_Name, Address, City, R_State, Country, Zipcode) values ('Fogarty', '81 Declaration Plaza', 'Fullerton', 'California', 'United States', '92640');

insert into Retailer (R_Name, Address, City, R_State, Country, Zipcode) values ('Giorgeschi', '743 Troy Trail', 'Denver', 'Colorado', 'United States', '80291');

insert into Retailer (R_Name, Address, City, R_State, Country, Zipcode) values ('Pyrah', '27 Hayes Court', 'Cincinnati', 'Ohio', 'United States', '45233');

insert into Retailer (R_Name, Address, City, R_State, Country, Zipcode) values ('Frankel', '6 Fulton Avenue', 'North Las Vegas', 'Nevada', 'United States', '89087');

--Manufacturer Table Inserts

insert into Manufacturer (M_Name, Address, City, M_State, Country, Zipcode) values ('Wetherby', '0439 Manitowish Point', 'Philadelphia', 'Pennsylvania', 'United States', '19146');

insert into Manufacturer (M_Name, Address, City, M_State, Country, Zipcode) values ('Closs', '5114 Morrow Place', 'Philadelphia', 'Pennsylvania', 'United States', '19120');

insert into Manufacturer (M_Name, Address, City, M_State, Country, Zipcode) values ('Clews', '315 Lighthouse Bay Plaza', 'Lawrenceville', 'Georgia', 'United States', '30245');

insert into Manufacturer (M_Name, Address, City, M_State, Country, Zipcode) values ('Hasel', '10305 Shelley Point', 'Phoenix', 'Arizona', 'United States', '85020');

insert into Manufacturer (M_Name, Address, City, M_State, Country, Zipcode) values ('Jimes', '40350 Kensington Avenue', 'Columbia', 'South Carolina', 'United States', '29215');

insert into Manufacturer (M_Name, Address, City, M_State, Country, Zipcode) values ('Breyt', '16369 Blackbird Alley', 'West Palm Beach', 'Florida', 'United States', '33411');

insert into Manufacturer (M_Name, Address, City, M_State, Country, Zipcode) values ('Whitehead', '17022 Maple Avenue', 'Washington', 'District of Columbia', 'United States', '20062');

insert into Manufacturer (M_Name, Address, City, M_State, Country, Zipcode) values ('Cartmael', '473 Judy Trail', 'Orlando', 'Florida', 'United States', '32868');

insert into Manufacturer (M_Name, Address, City, M_State, Country, Zipcode) values ('Eayres', '00190 Hovde Street', 'Las Vegas', 'Nevada', 'United States', '89155');

insert into Manufacturer (M_Name, Address, City, M_State, Country, Zipcode) values ('Tidman', '88430 Hermina Lane', 'Mesquite', 'Texas', 'United States', '75185');

insert into MtoWH_Products (Product_ID, M_ID, WH_ID, Quantity, M_WH_Date, M_WH_Price) values (1000000, 1000000, 1000000, 8, '25-Jul-2020', 6);

insert into MtoWH_Products (Product_ID, M_ID, WH_ID, Quantity, M_WH_Date, M_WH_Price) values (1000001, 1000001, 1000001, 66, '22-Apr-2020', 233);

insert into MtoWH_Products (Product_ID, M_ID, WH_ID, Quantity, M_WH_Date, M_WH_Price) values (1000002, 1000002, 1000002, 20, '22-Jun-2020', 205);

insert into MtoWH_Products (Product_ID, M_ID, WH_ID, Quantity, M_WH_Date, M_WH_Price) values (1000003, 1000003, 1000003, 94, '16-Aug-2020', 166);

insert into MtoWH_Products (Product_ID, M_ID, WH_ID, Quantity, M_WH_Date, M_WH_Price) values (1000004, 1000004, 1000004, 68, '05-Apr-2020', 266);

insert into MtoWH_Products (Product_ID, M_ID, WH_ID, Quantity, M_WH_Date, M_WH_Price) values (1000005, 1000005, 1000005, 3, '21-May-2020', 172);

insert into MtoWH_Products (Product_ID, M_ID, WH_ID, Quantity, M_WH_Date, M_WH_Price) values (1000006, 1000006, 1000006, 98, '13-May-2020', 234);

insert into MtoWH_Products (Product_ID, M_ID, WH_ID, Quantity, M_WH_Date, M_WH_Price) values (1000007, 1000007, 1000007, 97, '06-Jul-2020', 280);

insert into MtoWH_Products (Product_ID, M_ID, WH_ID, Quantity, M_WH_Date, M_WH_Price) values (1000008, 1000008, 1000008, 28, '19-Apr-2020', 104);

insert into MtoWH_Products (Product_ID, M_ID, WH_ID, Quantity, M_WH_Date, M_WH_Price) values (1000009, 1000009, 1000009, 89, '10-Dec-2019', 71);

insert into WHtoR_Products (Product_ID, R_ID, WH_ID, Quantity, WH_R_Date, WH_R_Price) values (1000000, 1000000, 1000000, 43, '31-Aug-2020', 271);

insert into WHtoR_Products (Product_ID, R_ID, WH_ID, Quantity, WH_R_Date, WH_R_Price) values (1000001, 1000001, 1000001, 26, '31-Dec-2019', 230);

insert into WHtoR_Products (Product_ID, R_ID, WH_ID, Quantity, WH_R_Date, WH_R_Price) values (1000002, 1000002, 1000002, 82, '21-Apr-2020', 46);

insert into WHtoR_Products (Product_ID, R_ID, WH_ID, Quantity, WH_R_Date, WH_R_Price) values (1000003, 1000003, 1000003, 71, '20-Jul-2020', 168);

insert into WHtoR_Products (Product_ID, R_ID, WH_ID, Quantity, WH_R_Date, WH_R_Price) values (1000004, 1000004, 1000004, 19, '07-Mar-2020', 85);

insert into WHtoR_Products (Product_ID, R_ID, WH_ID, Quantity, WH_R_Date, WH_R_Price) values (1000005, 1000005, 1000005, 90, '02-May-2020', 202);

insert into WHtoR_Products (Product_ID, R_ID, WH_ID, Quantity, WH_R_Date, WH_R_Price) values (1000006, 1000006, 1000006, 96, '27-Aug-2020', 225);

insert into WHtoR_Products (Product_ID, R_ID, WH_ID, Quantity, WH_R_Date, WH_R_Price) values (1000007, 1000007, 1000007, 47, '03-Jan-2020', 48);

insert into WHtoR_Products (Product_ID, R_ID, WH_ID, Quantity, WH_R_Date, WH_R_Price) values (1000008, 1000008, 1000008, 57, '21-Jul-2020', 77);

insert into WHtoR_Products (Product_ID, R_ID, WH_ID, Quantity, WH_R_Date, WH_R_Price) values (1000009, 1000009, 1000009, 100, '15-Jun-2020', 112);

```
insert into Manufacturer_Products (Product_ID, M_ID, Quantity) values (1000000, 1000000, 75);
insert into Manufacturer_Products (Product_ID, M_ID, Quantity) values (1000001, 1000001, 54);
insert into Manufacturer_Products (Product_ID, M_ID, Quantity) values (1000002, 1000002, 21);
insert into Manufacturer_Products (Product_ID, M_ID, Quantity) values (1000003, 1000003, 92);
insert into Manufacturer_Products (Product_ID, M_ID, Quantity) values (1000004, 1000004, 13);
insert into Manufacturer_Products (Product_ID, M_ID, Quantity) values (1000005, 1000005, 31);
insert into Manufacturer_Products (Product_ID, M_ID, Quantity) values (1000006, 1000006, 34);
insert into Manufacturer_Products (Product_ID, M_ID, Quantity) values (1000007, 1000007, 42);
insert into Manufacturer_Products (Product_ID, M_ID, Quantity) values (1000008, 1000008, 93);
insert into Manufacturer_Products (Product_ID, M_ID, Quantity) values (1000009, 1000009, 61);

insert into Warehouse_Products (Product_ID, WH_ID, Quantity) values (1000000, 1000000, 87);
insert into Warehouse_Products (Product_ID, WH_ID, Quantity) values (1000001, 1000001, 68);
insert into Warehouse_Products (Product_ID, WH_ID, Quantity) values (1000002, 1000002, 71);
insert into Warehouse_Products (Product_ID, WH_ID, Quantity) values (1000003, 1000003, 94);
insert into Warehouse_Products (Product_ID, WH_ID, Quantity) values (1000004, 1000004, 59);
insert into Warehouse_Products (Product_ID, WH_ID, Quantity) values (1000005, 1000005, 69);
insert into Warehouse_Products (Product_ID, WH_ID, Quantity) values (1000006, 1000006, 83);
insert into Warehouse_Products (Product_ID, WH_ID, Quantity) values (1000007, 1000007, 68);
insert into Warehouse_Products (Product_ID, WH_ID, Quantity) values (1000008, 1000008, 9);
insert into Warehouse_Products (Product_ID, WH_ID, Quantity) values (1000009, 1000009, 68);

insert into Retailer_Products (Product_ID, R_ID, Quantity) values (1000000, 1000000, 21);
insert into Retailer_Products (Product_ID, R_ID, Quantity) values (1000001, 1000001, 7);
insert into Retailer_Products (Product_ID, R_ID, Quantity) values (1000002, 1000002, 53);
insert into Retailer_Products (Product_ID, R_ID, Quantity) values (1000003, 1000003, 83);
insert into Retailer_Products (Product_ID, R_ID, Quantity) values (1000004, 1000004, 41);
insert into Retailer_Products (Product_ID, R_ID, Quantity) values (1000005, 1000005, 26);
```

```
insert into Retailer_Products (Product_ID, R_ID, Quantity) values (1000006, 1000006, 95);
insert into Retailer_Products (Product_ID, R_ID, Quantity) values (1000007, 1000007, 62);
insert into Retailer_Products (Product_ID, R_ID, Quantity) values (1000008, 1000008, 19);
insert into Retailer_Products (Product_ID, R_ID, Quantity) values (1000009, 1000009, 76);
```

DDL FOR DATA WAREHOUSE:

```
DROP TABLE Product_Management;
DROP VIEW orders_view;
DROP VIEW order_details_view;
DROP VIEW MTOWH_PRODUCTS_VIEW;
DROP VIEW WHTOR_PRODUCTS_VIEW;

CREATE TABLE Product_Management(
  Order_ID NUMBER(7),
  Product_ID NUMBER(7),
  Customer_ID NUMBER(7),
  Order_Detail_ID NUMBER(7),
  Total_Cost_of_Order NUMBER,
  M_ID NUMBER(7),
  M_WH_ID NUMBER(7),
  WH_ID NUMBER(7),
  WH_R_ID NUMBER(7),
  R_ID NUMBER(7),
  CONSTRAINT pk_product_m PRIMARY KEY (Order_ID,Product_ID)
);

CREATE OR REPLACE VIEW orders_view AS
SELECT
```

```
Order_ID,
Customer_ID
FROM orders;

CREATE OR REPLACE VIEW order_details_view AS
SELECT
Order_ID,
o.Product_ID as Product_ID,
Order_Detail_ID,
Quantity * cp.price as Total_Cost_of_Order
FROM Order_Details o JOIN Consumer_Product cp ON cp.Product_ID = o.Product_ID;

CREATE OR REPLACE VIEW mtowh_products_view AS
SELECT
M_WH_ID,
M_ID,
WH_ID,
Product_ID
FROM MtoWH_Products;

CREATE OR REPLACE VIEW whtor_products_view AS
SELECT
WH_R_ID,
R_ID,
Product_ID
FROM WHtoR_Products;
```

ETL:

```
CREATE OR REPLACE PROCEDURE pm_etl2 (

pm_id NUMBER DEFAULT 0)

IS

BEGIN

-- Insert

INSERT INTO Product_Management

   SELECT *

   FROM (SELECT
od.Order_ID,mh.Product_ID,o.Customer_ID,od.Order_Detail_ID,od.Total_Cost_of_Order,mh.M_ID,mh.M
_WH_ID,mh.WH_ID,wh.WH_R_ID,wh.R_ID

   FROM mtowh_products_view mh JOIN order_details_view od ON mh.product_id=od.product_id

   JOIN whtor_products_view wh ON mh.product_id=wh.product_id JOIN orders_view o ON
od.order_id=o.order_id)

   WHERE (Order_ID,Product_ID) NOT IN (SELECT Order_ID,Product_ID FROM Product_Management);

COMMIT;

--Update

MERGE INTO Product_Management pm

USING

(

SELECT
od.Order_ID,mh.Product_ID,o.Customer_ID,od.Order_Detail_ID,od.Total_Cost_of_Order,mh.M_ID,mh.M
_WH_ID,mh.WH_ID,wh.WH_R_ID,wh.R_ID

FROM mtowh_products_view mh JOIN order_details_view od ON mh.product_id=od.product_id JOIN
whtor_products_view wh ON

      mh.product_id=wh.product_id JOIN orders_view o ON od.order_id=o.order_id

)t3

ON(pm.product_id=t3.product_id AND pm.order_id=t3.product_id)

WHEN MATCHED THEN UPDATE SET

pm.Customer_ID=t3.customer_id,

pm.Order_Detail_ID=t3.order_detail_id,
```

```
pm.Total_Cost_of_Order=t3.total_cost_of_order,
pm.M_ID=t3.m_id,
pm.M_WH_ID=t3.m_wh_id,
pm.WH_ID=t3.wh_id,
pm.WH_R_ID=t3.wh_r_id,
pm.R_ID=t3.r_id;
END;

begin
   pm_etl2;
end;
```

DATA LAKE MODEL

```
CREATE OR REPLACE VIEW PM_view AS
SELECT
pm.order_id,
pm.product_id,
pm.customer_id,
pm.order_detail_id,
pm.total_cost_of_order,
pm.m_id,
pm.m_wh_id,
pm.wh_id,
pm.wh_r_id,
pm.r_id,
cp.price,
cp.sku,
cp.product_name,
```

```
cp.product_description,
cp.color,
cp.product_size,
cp.weight,
mw.M_WH_Price as MW_price,
mw.quantity as MW_quantity,
wr.WH_R_Price as WR_price,
wr.Quantity as WR_Quantity,
mw.M_WH_Price*mw.quantity as MW_total_product_cost,
wr.WH_R_Price*wr.Quantity as WR_total_product_cost
FROM Product_Management pm
  FULL JOIN Consumer_Product cp ON cp.Product_ID = pm.Product_ID
  FULL JOIN MtoWH_Products mw ON pm.product_id = mw.product_id
  FULL JOIN WHtoR_Products wr ON pm.product_id = wr.product_id
```

CODE FOR ANALYSIS PATTERNS

Using Spark:

1. Looking into the cost distribution of products between warehouse, manufacturer, retail, and ultimate selling price:

```
%sql

Select product_id, Price, MW_price, WR_price
FROM df
WHERE product_id IS NOT NULL
GROUP BY product_id, Price, MW_price, WR_price
```

2. Comparing the costs and revenue for each warehouse

```
%sql
```

```
Select wh_id, AVG(MW_total_product_cost) AS AVG_WH_Cost , AVG(WR_total_product_cost)
AS AVG_WH_Revenue
FROM df
WHERE wh_id IS NOT NULL and MW_total_product_cost IS NOT NULL and
MW_total_product_cost IS NOT NULL
GROUP BY wh_id
```

3. Proportion of each product sold

```
%sql

Select  product_name
FROM df
WHERE product_name IS NOT NULL
GROUP BY product_name
```