SUPPLY CHAIN MANAGEMENT DATABASE Project Report

Presented By - Divyanshu Kumar, Arnav Goel, Robin John,

Harshitha Thippegundi Nagaraj, David Osei-Poku

Fall 2022

Date - 08-12-2022

### Table of Contents

|  |  |
| --- | --- |
| Topics | Page Number |
| 1. Summary | 3 |
| 1. The problem Statement | 3 |
| 1. Requirements | 4 |
| 1. Conceptual Design | 6 |
| Relational Schema | 9 |
| 1. DATA DICTIONARY | 10 |
| 1. SCHEMA and DATA | 18 |
| 8. DATA QUERIES | 32 |
| 9. DATA MANIPULATION | 42 |
| 10. SUMMARY | 59 |
| 11. Appendix | 61 |

**1. Summary**

At its most basic level, supply chain management (SCM) is the control of the movement of materials, information, and money associated with a good or service from the acquisition of raw materials to the delivery of the goods at the final location.

Even though the supply chain and logistics are sometimes confused, logistics is actually only one part of the supply chain. Modern SCM systems are digitally based and incorporate material handling and software for all parties engaged in the creation of goods and services, order fulfillment, and information tracking, including suppliers, manufacturers, wholesalers, carriers, and retailers.

Procurement, product lifecycle management, supply chain planning (including inventory planning and the upkeep of enterprise assets and production lines), logistics (including transportation and fleet management), and order management are just a few of the activities that fall under the purview of the supply chain. SCM can also be applied to tasks related to international trade, such as managing multinational supply chains and production operations.

**2. The problem statement**

* **Goals (of the application)**

The Supply chain database is a database system for one of the Business-to-Consumer (B2C) e-commerce companies. Supply chain management is the management of goods to make sure the right quality and quantity of products are procured from vendors and shipped to the customers. The database system stores data about customers, vendors, and warehouse or inventory stock. The goal of this system is to create a single database that allows the company to procure, package, and ship products to customers. The database combines vendor and customer demographic information along with warehouse inventory to help in understanding demand and supply. We have assumed a hypothetical e-commerce company for our project which sells all ranges of products from clothing, and electronics, to household items.

* **Context (why this system is important and how this system will be used)**

A supply chain database is very crucial for any e-commerce industry. Currently, the data related to vendors, customers, and inventory are in different data systems. Customer demographic information such as addresses, interests, and orders are stored as a single database. Vendor data system stores vendor information with products and services. This data system needs to be unified with warehouse inventory so that it would be easier to track the movement of the product within the company. The current data systems within the company have data redundancies and creating a unified database helps to overcome the redundancies. And the final data system can be used to analyze the data and further improve the quality and efficiency and route optimization of the delivery.

* **Scope**
  + **In-scope**

This database will store data about orders placed by individual customers, products included in the said orders, all available products in stock, a list of vendors supplying products, and employees involved in the facilitation of the supply chain. An employee in this instance may be either a manager of a warehouse or personnel that aid in the delivery of ordered items. The scope of this database will include data about individual warehouses as well as information about products.

The supply-chain database system will also track invoices presented to customers stating the individual and total costs of products ordered. This database will also include refunds made to customers in the scenario where orders are canceled, in addition to information about the orders returned.

* + **Out-scope**

This database will not track all other employees that may be involved in a supply chain. It will also not track payments made to vendors in the procurement of products. Information about the route of orders to be delivered as well as distribution centers are excluded in addition. The database will not consider information about the shipment of items as this is not a requirement in the supply chain being operated.

1. **Requirements**

* Analysis of Current System (if there is a current database)
* Data requirements

● The system stores customer information in the form of their unique ID, name, address, and phone number.

● The order placed by the customer is given a unique ID and stored alongside the order date and amount.

● The vendor information including their ID, name, address, and phone number is stored as he supplies the products

● Products have their information stored in the form of their ID, name, category as well as cost price and sale price.

● Order line items have their Id and description as well as amount stored.

● The warehouse which receives the product has its information stored in the form of its ID and address.

● The warehouse can have employees in the form of managers and delivery drivers, who have their information stored in the form of ID, name (first name and last name), employment date, working hours, and pay rate.

● Delivery details are stored in form of an Id and delivery date.

● The delivery driver has their license number and state stored.

● Managers have their years of experience stored.

● The order returning information is stored in the form of its ID and return date.

● The return line item has its Id and description as well as the amount stored.

* Business rules and Logic

● Delivery drivers can deliver to multiple customers.

● An order can consist of many products.

● Vendors can send products to multiple warehouses.

● A customer can get deliveries from multiple delivery drivers.

● Number of order returns can not be more than the orders placed.

● One order can be delivered in multiple parts depending upon the availability of items.

● Price of a product may vary over time.

● Delivery time depends upon when you place the order.

● Purchase orders must contain shipment mode and delivery date.

● Order refunds can only be issued once the order has been returned.

* **Any other assumption**
* The way you solved the problem, if relevant
* The way you obtained (created) the data
* The hardware and software (Personal Oracle, CCI Oracle, LiveSQL, Mysql, Version number) you used and any special features or considerations

### **Conceptual Design**

### The ERD (***Copy and paste your ERD to the report file***; do not say “ see the attached draw.io file for the ERD”. Make sure your ERD is legible and fonts are readable.)



* Documentation on the ERD (Provide explanation on any non-intuitive names such as abbreviation, acronyms, or shortened names used in entities, meaning of relationships, attributes, and business rules governing the cardinality.)

● A Customer may place one or multiple Orders, but each Order can be placed by only one.

● An Order can be made up of one or multiple Products, but each Product ordered belongs to one and only one Order.

● A Customer may request multiple Order Returns in case the Customer would like to cancel their Order.

● In this ERD, there is an entity for Employees, with a specialization for two employee types: Delivery Driver and Manager.

● The “Optional” constraint in the Employee subclass indicates that there could be other employees that do not belong to one of the Delivery Driver or Manager subclasses. “Or” also indicates that each employee belongs to only one subclass.

● A Customer may receive multiple Deliveries, but each Delivery is received by a single Customer

● A Delivery Driver may complete multiple Deliveries, but each Delivery is completed by a single Delivery Driver.

● A Delivery constitutes one or multiple Orders, but each Order may be part of only one Delivery.

● Each Manager manages one and only one warehouse, and each warehouse is managed by one and only one Manager.

● There is a ternary relationship between Vendor, Warehouse, and Product:

○ For a given pair of Vendor and Warehouse, there could be one or many Products supplied.

○ For a given pair of Vendors and Products, there is one and only one Warehouse supplied.

○ For a given pair of Warehouse and Products, there is one and only one Vendor that supplies.

● A Vendor may be an individual or an organization so a Vendor does not include a first and last name.

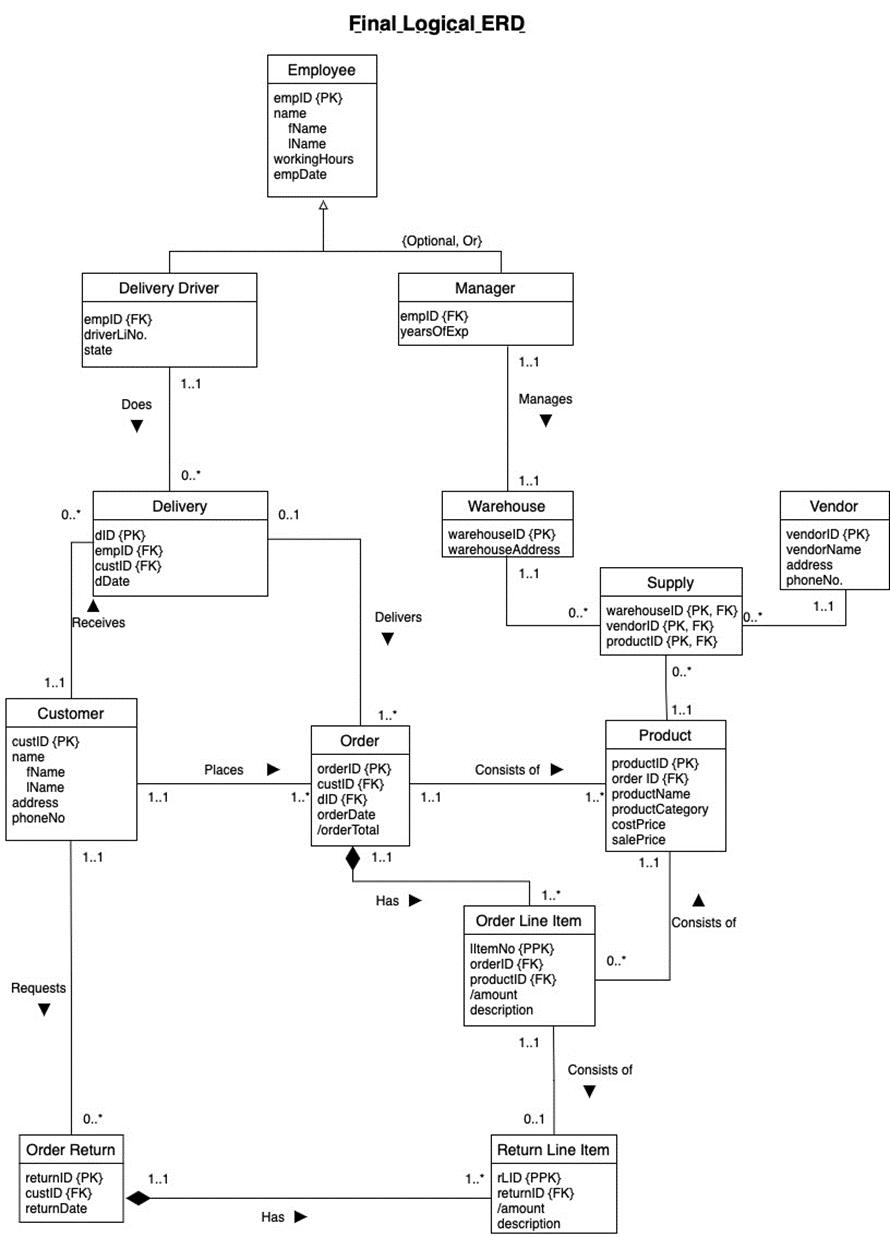
● An Order has one or multiple Order Line Items, but each Order Line Item belongs to a single Order.

● An Order Line Item consists of a single Product, whereas each Product may be part of multiple Order Line Items.

● An Order Line Item may be included as A Return Line Item, but each Return Line Item is only one Order Line Item.

**Logical ERD**

* Logical ERD



### **Relational schema**

### Employee (**empID**, fName, lName, workingHours, payRate, empDate)

Delivery Driver (**empID**, driverLiNo., state)

Foreign key empID references Employee (empID)

Manager (**empID**, yearOfExp)

Foreign key empID references Employee (empID)

Delivery(**dID**, empID, custID, dDate)

Foreign key empID references Employee (empID)

Foreign key custID references Customer (custID)

Customer (**custID**, fName, lName, address, phoneNo)

Order (**orderID**, custID, dID, orderDate)

Foreign key custID references Customer (custID)

Foreign key dID references Delivery(dID)

Warehouse (**warehouseID**, warehouseAddress)

Vendor (**vendorID**, vendorName, address, phoneNo)

Product (**productID**, orderID, productName, productCategory, costPrice, salePrice)

Foreign key orderID references Order (orderID)

Supply (**warehouseID, vendorID, productID**)

Foreign key warehouseID references Warehouse (warehouseID)

Foreign key vendorID references Vendor (vendorID)

Foreign key productID references Product (productID)

Order Return (**returnID**, custID, returnDate)

Foreign key custID references Customer (custID)

Order Line Item (**litemNo**, **orderID**, productID, description)

Foreign key orderID references Order (orderID)

Foreign key productID references Product (productID)

Return Line Item (**rLID**, **returnID**, litemNo, description)

Foreign key returnID references Order Return (returnID)

Foreign key litemNo references Order Line Item (litemNo)

### **DATA DICTIONARY**

|  |  |
| --- | --- |
| **Table Name** | **No. of Columns** |
| EMPLOYEE | 6 |
| DELIVERY DRIVER | 3 |
| MANAGER | 2 |
| CUSTOMER | 7 |
| DELIVERY | 4 |
| ORDER | 4 |
| WAREHOUSE | 2 |
| VENDOR | 4 |
| PRODUCT | 6 |
| SUPPLY | 3 |
| ORDER RETURN | 3 |
| ORDER LINE ITEM | 4 |
| RETURN LINE ITEM | 4 |

**EMPLOYEE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Field Size, Format** | **Description** | **NOT NULL, PK, FK** |
| empID | varchar2 | 11 | The unique id given to each employee | PK, NOT NULL |
| fName | varchar2 | 40 | First name | NOT NULL |
| lName | varchar2 | 40 | Last name | NOT NULL |
| workingHours | Number | 2 | Number of working hours | NOT NULL |
| payRate | Number | 2,2 | Hourly pay rate | NOT NULL |
| empDate | date | YYYY-MM-DD | The start date of employment | NOT NULL |

**DELIVERY DRIVER**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Field Size, Format** | **Description** | **NOT NULL, PK, FK** |
| empID | varchar2 | 11 | The unique id given to each delivery driver | FK, NOT NULL |
| driverLiNo | varchar2 | 11 | License number of driver | NOT NULL |
| state | varchar2 | 2 | State where the license number is registered | NOT NULL |

**MANAGER**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Field Size, Format** | **Description** | **NOT NULL, PK, FK** |
| empID | varchar2 | 11 | The unique id given to each manager | FK, NOT NULL |
| yearOfExp | varchar2 | 2 | Number of years of experience |  |

**CUSTOMER**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Field Size, Format** | **Description** | **NOT NULL, PK, FK** |
| custID | varchar2 | 11 | The unique id given to each employee | PK, NOT NULL |
| fName | varchar2 | 20 | First name | NOT NULL |
| lName | varchar2 | 20 | Last name |  |
| address | varchar2 | 40 | Address of customer | NOT NULL |
| city | Varchar2 | 20 | City of customer | NOT NULL |
| state | Char | 2 | State of customer | NOT NULL |
| phoneNo | char | 12 | Phone number of customer | NOT NULL |

**DELIVERY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Field Size, Format** | **Description** | **NOT NULL, PK, FK** |
| dID | char | 4 | The ID assigned to a delivery | PK, NOT NULL |
| empID | varchar2 | 11 | The unique id given to each delivery driver | FK, NOT NULL |
| custId | varchar2 | 11 | The ID of customer receiving delivery | FK, NOT NULL |
| dDate | DATE | YYYY-MM-DD | Date of delivery | NOT NULL |

**ORDER**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Field Size, Format** | **Description** | **NOT NULL, PK, FK** |
| orderID | Varchar2 | 20 | The ID assigned to an order | PK, NOT NULL |
| custId | varchar2 | 11 | The ID of customer order | FK, NOT NULL |
| dID | char | 4 | The ID assigned to delivery diver | FK, NOT NULL |
| orderDate | DATE | YYYY-MM-DD | Date of order | NOT NULL |

**WAREHOUSE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Field Size, Format** | **Description** | **NOT NULL, PK, FK** |
| warehouseID | varchar2 | 11 | The ID assigned to a warehouse | PK, NOT NULL |
| warehouseAddress | varchar2 | 40 | Address of the warehouse | NOT NULL |

**VENDOR**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Field Size, Format** | **Description** | **NOT NULL, PK, FK** |
| vendorID | varchar2 | 11 | The unique id given to each vendor | PK, NOT NULL |
| vendorName | varchar2 | 40 | Name of the vendor | NOT NULL |
| address | varchar2 | 60 | Address of vendor | NOT NULL |
| phoneNo | Varchar2 | 15 | Phone number of vendor | NOT NULL |

**PRODUCT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Field Size, Format** | **Description** | **NOT NULL, PK, FK** |
| productID | varchar2 | 11 | The unique id given to each product | PK, NOT NULL |
| orderID | varchar2 | 20 | The unique id given to each order | FK,NOT NULL |
| productName | varchar2 | 20 | Name of the product | NOT NULL |
| productCategory | varchar2 | 20 | Category of the product | NOT NULL |
| costPrice | number | 10 | Cost price of the product | NOT NULL |
| salePrice | Number | 10 | Sale price of the product | NOT NULL |

**SUPPLY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Field Size, Format** | **Description** | **NOT NULL, PK, FK** |
| warehouseID | varchar2 | 11 | The unique id given to each warehouse | FK, NOT NULL |
| vendorID | varchar2 | 11 | The unique id given to each vendor | FK, NOT NULL |
| productID | varchar2 | 11 | The unique id given to each product | FK, NOT NULL |

**ORDER RETURN**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Field Size, Format** | **Description** | **NOT NULL, PK, FK** |
| returnID | varchar2 | 11 | The unique id given to each order return | PK, NOT NULL |
| custID | varchar2 | 11 | The unique id given to each customer | FK, NOT NULL |
| returnDate | Date | YYYY-MM-DD | Date of product return | NOT NULL |

**ORDER LINE ITEM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Field Size, Format** | **Description** | **NOT NULL, PK, FK** |
| itemNo | varchar2 | 11 | The unique id given to each item | PK, NOT NULL |
| orderID | varchar2 | 20 | The unique id given to each order | FK, NOT NULL |
| productID | varchar2 | 11 | The unique id given to each product | FK, NOT NULL |
| description | Varchar2 | 200 | Description of each item | NOT NULL |

**RETURN LINE ITEM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Field Size, Format** | **Description** | **NOT NULL, PK, FK** |
| rLID | varchar2 | 11 | The unique id given to each return line item | PK, NOT NULL |
| returnID | varchar2 | 11 | The unique id given to each order return | FK, NOT NULL |
| itemNo | varchar2 | 11 | The unique id given to each item | FK, NOT NULL |
| description | Varchar2 | 200 | Description of each return item | NOT NULL |

1. **SCHEMA and DATA** 
   1. CREATE TABLE commands

drop table Employee cascade constraints;

CREATE TABLE Employee

(empID varchar2(11) not null,

fName varchar2(40) not null,

lName varchar2(40) not null,

workingHours number(10) not null,

payRate number(12,2) not null,

empDate DATE not null,

constraint Employee\_pk primary key (empID));

drop table Customer cascade constraints;

CREATE TABLE Customer

(custID varchar2(11) not null,

fName varchar2(20),

lName varchar2(20),

address varchar2(40) null,

city varchar2(20) null,

state char(2) not null,

phoneNo char(12) not null,

constraint Customer\_PK PRIMARY KEY (custID));

drop table Delivery cascade constraints;

CREATE TABLE Delivery

(dID char(4) not null,

empID varchar2(11) not null,

custID varchar2(11) not null,

dDate DATE not null,

constraint Delivery\_PK PRIMARY KEY (dID),

constraint Delivery\_FK\_Employee FOREIGN KEY (empID) references Employee,

constraint Delivery\_FK\_Customer FOREIGN KEY (custID) references Customer);

drop table Orders cascade constraints;

CREATE TABLE Orders

(orderID varchar2(20) not null,

custID varchar2(11) not null,

dID char(4) not null,

orderDate DATE not null,

constraint Orders\_PK PRIMARY KEY (orderID),

constraint Orders\_FK\_Customer FOREIGN KEY (custID) references Customer,

constraint Orders\_FK\_Delivery FOREIGN KEY (dID) references Delivery);

drop table DeliveryDriver cascade constraints;

CREATE TABLE DeliveryDriver

(empID varchar2(11) not null,

driverLiNo varchar2(11) not null,

state char(2) not null,

constraint DeliveryDriver\_FK\_Employee FOREIGN KEY (empID) references Employee);

drop table Manager cascade constraints;

CREATE TABLE Manager

(empID varchar2(11) not null,

yearsOfExp varchar2(10),

constraint Manager\_FK\_Employee FOREIGN KEY (empID) references Employee);

drop table OrderReturn cascade constraints;

CREATE TABLE OrderReturn

(returnID varchar2(11) not null,

custID varchar2(11) not null,

returnDate DATE not null,

constraint OrderReturn\_PK PRIMARY KEY (returnID),

constraint OrderReturn\_FK\_Customer FOREIGN KEY (custID) references Customer);

drop table Warehouse cascade constraints;

CREATE TABLE Warehouse

(warehouseID varchar2(11) not null,

warehouseAddress varchar2(40) not null,

constraint Warehouse\_PK PRIMARY KEY (warehouseID));

drop table Vendor cascade constraints;

CREATE TABLE Vendor

(vendorID varchar2(11) not null,

vendorName varchar2(40) not null,

address varchar2(60) not null,

phoneNo varchar2(15) not null,

constraint Vendor\_PK PRIMARY KEY (vendorID));

drop table Product cascade constraints;

CREATE TABLE Product

(productID varchar2(11) not null,

orderID varchar2(20) not null,

productName varchar2(20) not null,

productCategory varchar2(20) not null,

costPrice number(10) not null,

salePrice number(10) not null,

constraint Product\_PK PRIMARY KEY (productID),

constraint Product\_FK\_Order FOREIGN KEY (orderID) references Orders);

drop table Supply cascade constraints;

CREATE TABLE Supply

(warehouseID varchar2(11) not null,

vendorID varchar2(11) not null,

productID varchar2(11) not null,

constraint Supply\_FK\_Warehouse FOREIGN KEY (warehouseID) references Warehouse,

constraint Supply\_FK\_Vendor FOREIGN KEY (vendorID) references Vendor,

constraint Supply\_FK\_Product FOREIGN KEY (productID) references Product);

drop table OrderLineItem cascade constraints;

CREATE TABLE OrderLineItem

(itemNo varchar2(11) not null,

orderID varchar2(20) not null,

productID varchar2(11) not null,

description varchar2(200) not null,

constraint OrderLineItem\_PK PRIMARY KEY (itemNo),

constraint OrderLineItem\_FK\_Orders FOREIGN KEY (orderID) references Orders,

constraint OrderLineItem\_FK\_Product FOREIGN KEY (productID) references Product);

drop table ReturnLineItem cascade constraints;

CREATE TABLE ReturnLineItem

(rLID varchar2(11) not null,

returnID varchar2(11) not null,

itemNo varchar2(11) not null,

description varchar2(100) not null,

constraint ReturnLineItem\_PK PRIMARY KEY (rLID),

constraint ReturnLineItem\_FK\_OrderReturn FOREIGN KEY (returnID) references OrderReturn,

constraint ReturnLineItem\_FK\_OrderLineItem FOREIGN KEY (itemNo) references OrderLineItem);

* 1. **INSERT INTO Commands**

INSERT INTO Employee VALUES ('75838', 'Joy', 'Roy', '40', '24.00', TO\_DATE('2006-07-23', 'YYYY-MM-DD'));

INSERT INTO Employee VALUES ('74838', 'Josh', 'Milan', '40', '43.00', TO\_DATE('2012-07-23', 'YYYY-MM-DD'));

INSERT INTO Employee VALUES ('86947', 'Amy', 'Vice', '40', '32.00', TO\_DATE('2015-04-19', 'YYYY-MM-DD'));

INSERT INTO Employee VALUES ('38956', 'Roy', 'King', '40', '28.00', TO\_DATE('2008-05-08', 'YYYY-MM-DD'));

INSERT INTO Employee VALUES ('86943', 'Sam', 'John', '40', '24.00', TO\_DATE('2006-02-04', 'YYYY-MM-DD'));

INSERT INTO Employee VALUES ('74837', 'Ram', 'Kumar', '40', '34.00', TO\_DATE('2006-09-13', 'YYYY-MM-DD'));

INSERT INTO Employee VALUES ('24478', 'Shyam', 'Reddy', '40', '45.00', TO\_DATE('2011-03-15', 'YYYY-MM-DD'));

INSERT INTO Employee VALUES ('96740', 'Avi', 'Singh', '40', '63.00', TO\_DATE('2005-07-07', 'YYYY-MM-DD'));

INSERT INTO Employee VALUES ('26473', 'Ping', 'Yong', '40', '48.00', TO\_DATE('2018-04-18', 'YYYY-MM-DD'));

INSERT INTO Employee VALUES ('58484', 'Pete', 'Bekham', '40', '44.00', TO\_DATE('2016-02-24', 'YYYY-MM-DD'));

INSERT INTO Customer VALUES ('648-92-1872', 'Morris', 'Taylor', '1711 Brandywine St', 'Philadelphia', 'PA', '201 946-8853');

INSERT INTO Customer VALUES ('213-46-8915', 'Edith', 'Smart', '3410 Blonde St.', 'Boston', 'MA', '801 826-0752');

INSERT INTO Customer VALUES ('427-17-2319', 'Chelsea', 'Rhinney', '3675 Market Street', 'Philadelphia', 'PA', '408 286-2428');

INSERT INTO Customer VALUES ('527-72-3246', 'Richeal', 'Adams', '22 Compton Av.', 'Los Angeles', 'CA', '913 843-0462');

INSERT INTO Customer VALUES ('486-29-1786', 'Claude', 'Quartey', '30 Broadway Ave.', 'San Francisco', 'CA', '415 843-2991');

INSERT INTO Delivery VALUES ('0877', '74837', '527-72-3246', TO\_DATE('2016-12-21', 'YYYY-MM-DD'));

INSERT INTO Delivery VALUES ('0736', '96740', '648-92-1872', TO\_DATE('2020-1-30', 'YYYY-MM-DD'));

INSERT INTO Delivery VALUES ('1389', '74837', '486-29-1786', TO\_DATE('2017-3-22', 'YYYY-MM-DD'));

INSERT INTO Delivery VALUES ('3588', '74837', '527-72-3246', TO\_DATE('2021-3-23', 'YYYY-MM-DD'));

INSERT INTO Delivery VALUES ('1922', '74838', '427-17-2319', TO\_DATE('2022-11-13', 'YYYY-MM-DD'));

INSERT INTO Orders VALUES ('AB-123-DEF-425-1Z3', '427-17-2319', '1922', TO\_DATE('2022-11-6', 'YYYY-MM-DD'));

INSERT INTO Orders VALUES ('BS-345-DSE-860-1F2', '527-72-3246', '0877', TO\_DATE('2016-12-12', 'YYYY-MM-DD'));

INSERT INTO Orders VALUES ('GH-542-NAD-713-9F9', '486-29-1786', '1389', TO\_DATE('2022-3-15', 'YYYY-MM-DD'));

INSERT INTO Orders VALUES ('ZS-645-CAT-415-1B2', '527-72-3246', '3588', TO\_DATE('2021-3-12', 'YYYY-MM-DD'));

INSERT INTO Orders VALUES ('AX-532-FED-452-2Z7', '648-92-1872', '0736', TO\_DATE('2020-1-22', 'YYYY-MM-DD'));

INSERT INTO DeliveryDriver VALUES ('96740', 'L145D', 'PA');

INSERT INTO DeliveryDriver VALUES ('58484', 'O345I', 'MA');

INSERT INTO DeliveryDriver VALUES ('38956', 'U834R', 'NJ');

INSERT INTO DeliveryDriver VALUES ('74837', 'L145D', 'CA');

INSERT INTO DeliveryDriver VALUES ('74838', 'L145D', 'PA');

INSERT INTO Manager VALUES ('75838', '18');

INSERT INTO Manager VALUES ('86943', '5');

INSERT INTO Manager VALUES ('26473', '7');

INSERT INTO Manager VALUES ('24478', '12');

INSERT INTO Manager VALUES ('86947', '32');

INSERT INTO OrderReturn VALUES ('57384', '213-46-8915', TO\_DATE('2022-10-31', 'YYYY-MM-DD'));

INSERT INTO OrderReturn VALUES ('66254', '527-72-3246', TO\_DATE('2022-5-5', 'YYYY-MM-DD'));

INSERT INTO OrderReturn VALUES ('34952', '648-92-1872', TO\_DATE('2022-7-17', 'YYYY-MM-DD'));

INSERT INTO OrderReturn VALUES ('44978', '486-29-1786', TO\_DATE('2022-11-22', 'YYYY-MM-DD'));

INSERT INTO OrderReturn VALUES ('19564', '427-17-2319', TO\_DATE('2022-9-5', 'YYYY-MM-DD'));

INSERT INTO Warehouse VALUES ('W20221', '5720 McAuley St.');

INSERT INTO Warehouse VALUES ('W20222', '4 Upland Hts.');

INSERT INTO Warehouse VALUES ('W20223', '3410 Blonde St.');

INSERT INTO Warehouse VALUES ('W20224', '3 Silver Ct.');

INSERT INTO Warehouse VALUES ('W20225', '22 Cleveland Av. #1');

INSERT INTO Vendor VALUES ('V20221', 'Honeywell', '6223 Bateman St.#410, Philadelphia, PA', '415 658-9932');

INSERT INTO Vendor VALUES ('V20222', 'Philips', '309 63rd St. #411, Oakland, CA', '425 658-9932');

INSERT INTO Vendor VALUES ('V20223', 'Revlon', '589 Darwin Ln. #412, Berkeley, CA', '435 658-9932');

INSERT INTO Vendor VALUES ('V20224', 'Sony', '67 Seventh Av.#413, San Francisco, CA', '445 658-9932');

INSERT INTO Vendor VALUES ('V20225', 'LG', '3 Balding Pl.#414, Seattle, WA', '455 658-9932');

INSERT INTO Product VALUES ('123547', 'BS-345-DSE-860-1F2', 'Book', 'Stationery', '7', '10');

INSERT INTO Product VALUES ('354723', 'ZS-645-CAT-415-1B2', 'Earphones', 'Electronics', '450', '750');

INSERT INTO Product VALUES ('542337', 'AX-532-FED-452-2Z7', 'Pen', 'Stationery', '3', '5');

INSERT INTO Product VALUES ('723232', 'GH-542-NAD-713-9F9', 'Mobile', 'Electronics', '650', '1000');

INSERT INTO Product VALUES ('324234', 'AB-123-DEF-425-1Z3', 'iPad', 'Electronics', '750', '1050');

INSERT INTO Supply VALUES ('W20221', 'V20221', '123547');

INSERT INTO Supply VALUES ('W20222', 'V20222', '354723');

INSERT INTO Supply VALUES ('W20223', 'V20223', '542337');

INSERT INTO Supply VALUES ('W20224', 'V20224', '723232');

INSERT INTO Supply VALUES ('W20225', 'V20225', '324234');

INSERT INTO OrderLineItem VALUES ('233242', 'BS-345-DSE-860-1F2', '123547', 'Thriller Novel from a famous author');

INSERT INTO OrderLineItem VALUES ('423234', 'ZS-645-CAT-415-1B2', '354723', 'Apple Airpods Pro');

INSERT INTO OrderLineItem VALUES ('923423', 'AX-532-FED-452-2Z7', '542337', 'Parker ballpoint Pen');

INSERT INTO OrderLineItem VALUES ('559022', 'GH-542-NAD-713-9F9', '723232', 'Iphone 14 Pro');

INSERT INTO OrderLineItem VALUES ('454311', 'BS-345-DSE-860-1F2', '324234', 'Ipad Air 2');

INSERT INTO ReturnLineItem VALUES ('55978', '66254', '454311', 'product damaged');

INSERT INTO ReturnLineItem VALUES ('40013', '57384', '559022', 'not needed anymore');

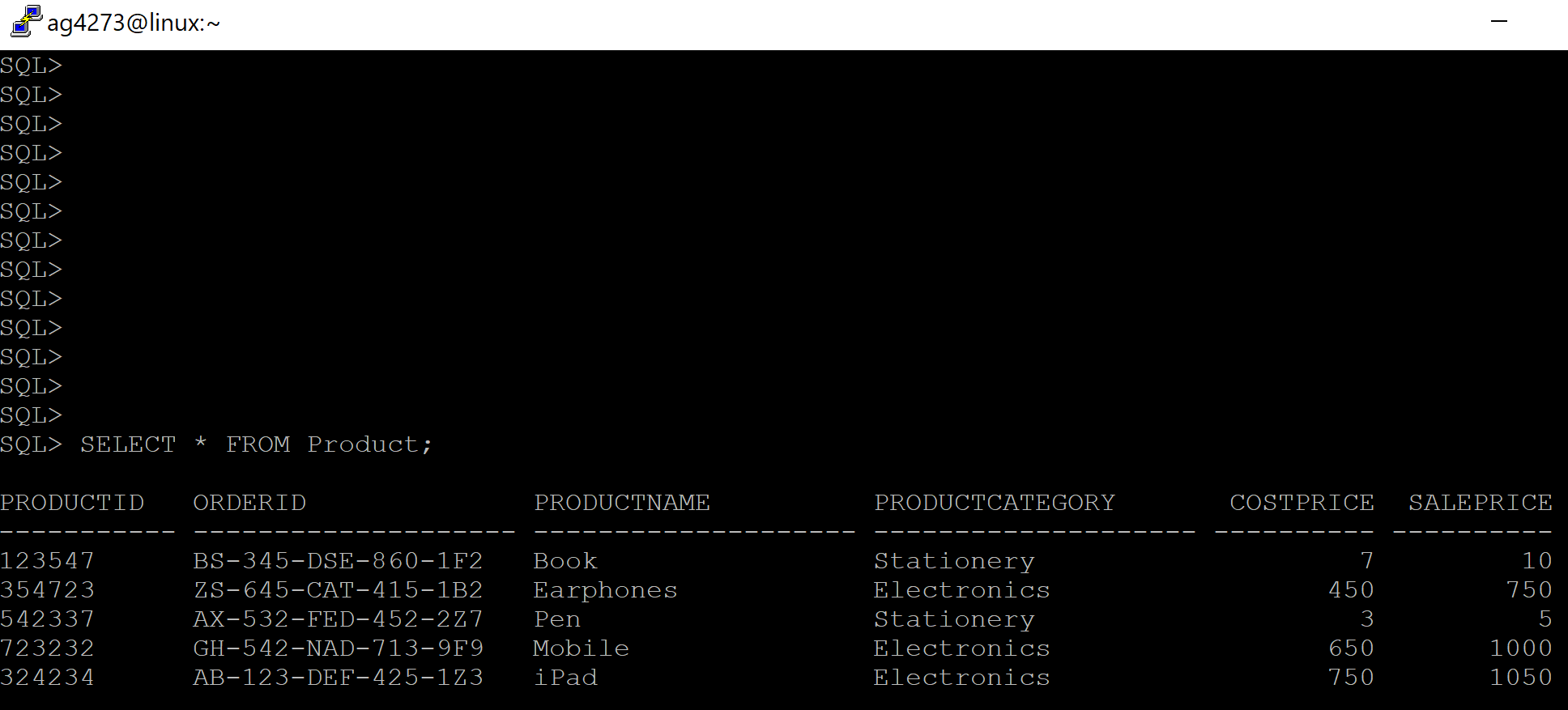
INSERT INTO ReturnLineItem VALUES ('90675', '34952', '923423', 'item not working properly');

INSERT INTO ReturnLineItem VALUES ('11902', '44978', '423234', 'better price available');

INSERT INTO ReturnLineItem VALUES ('21311', '19564', '233242', 'bought by mistake');

* 1. Snipped output from SELECT \* FROM *your\_table\_name*; for all tables

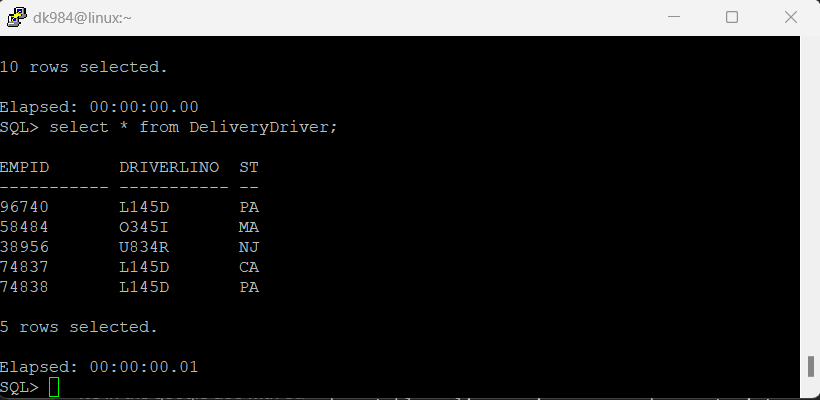
PRODUCT TABLE



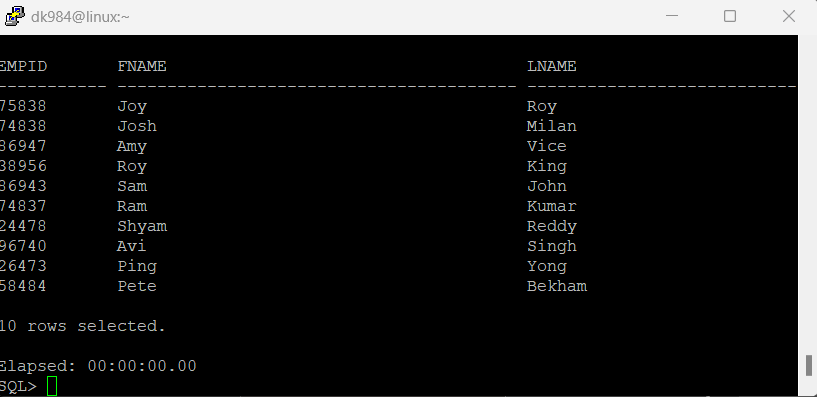
Order LIne Item Table



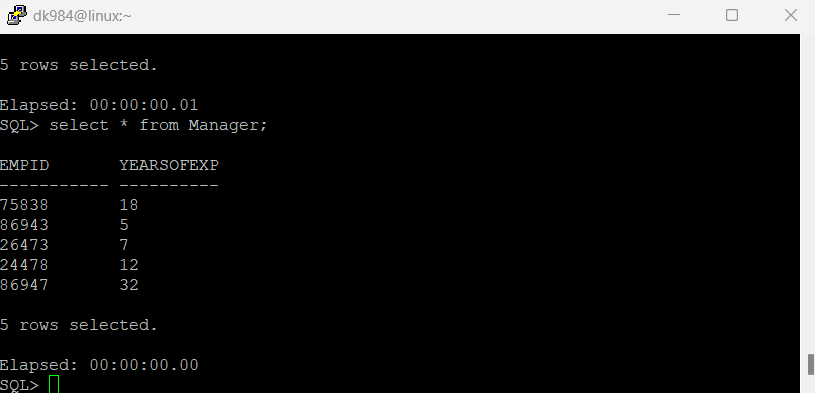
Display Delivery Driver Table



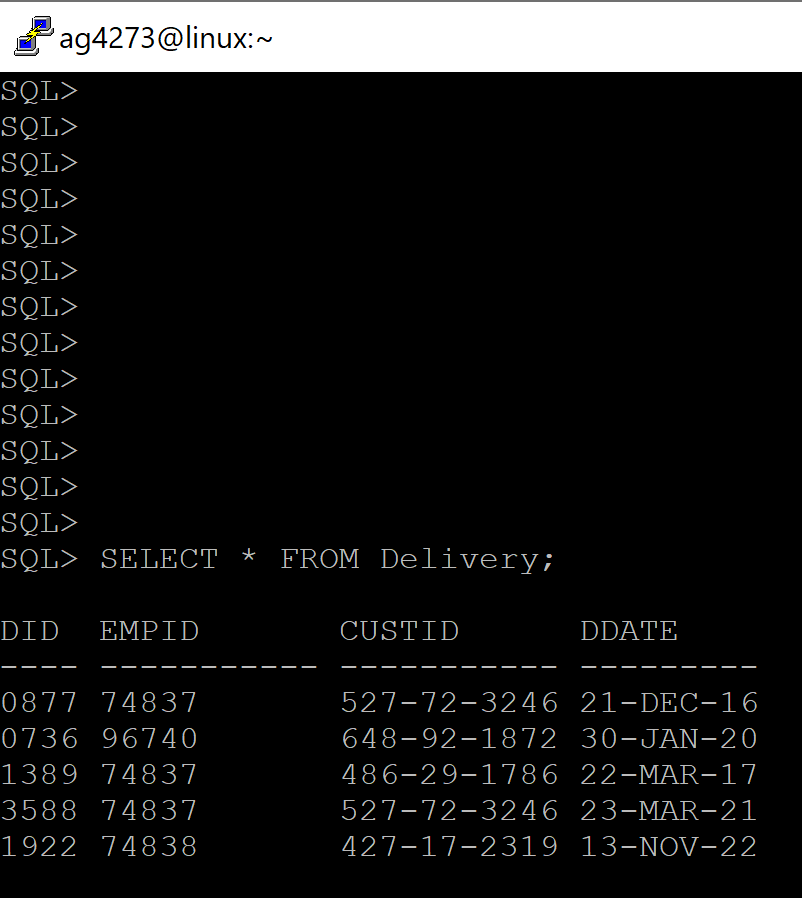
Display Employee table



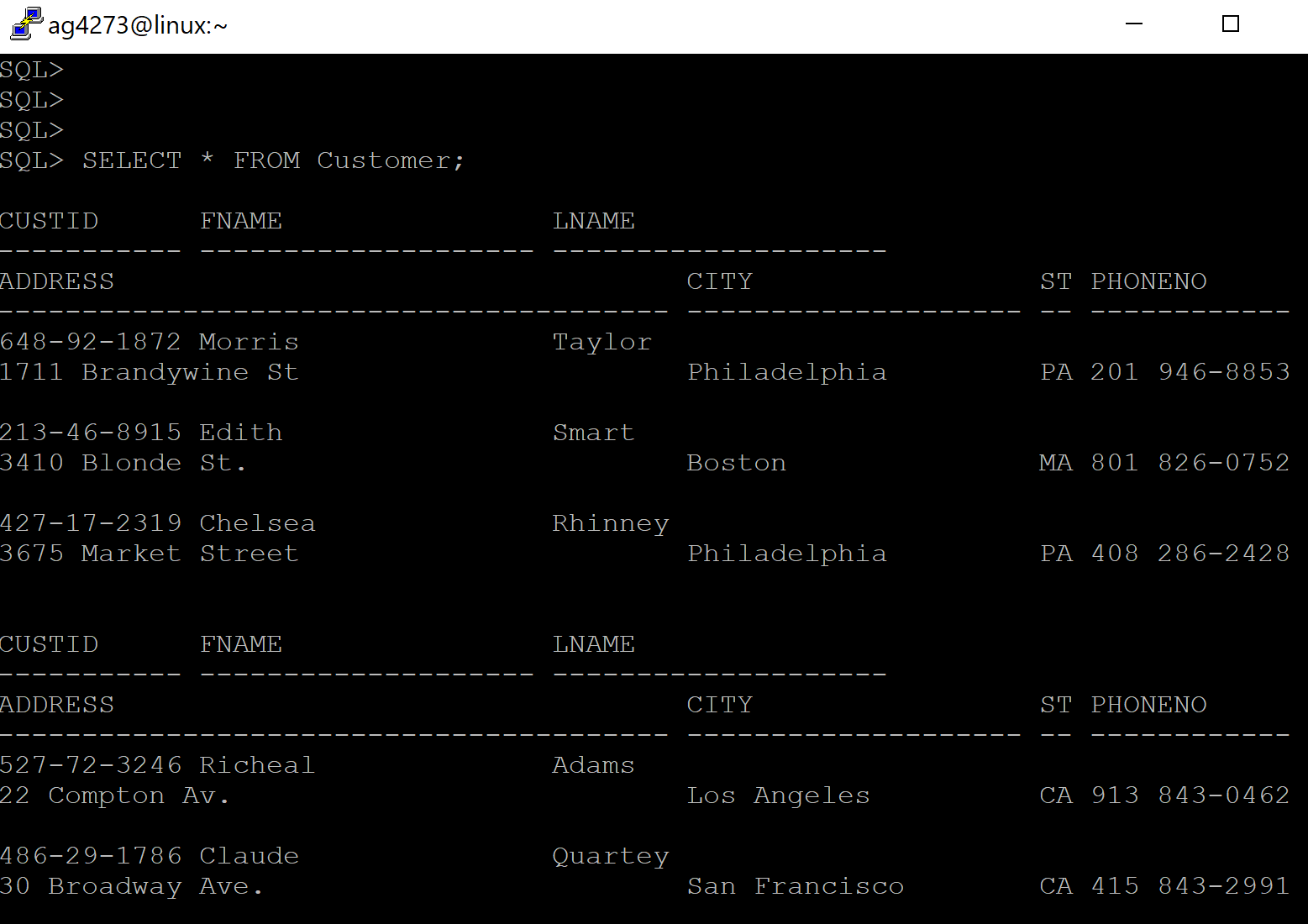
Display Manager Table



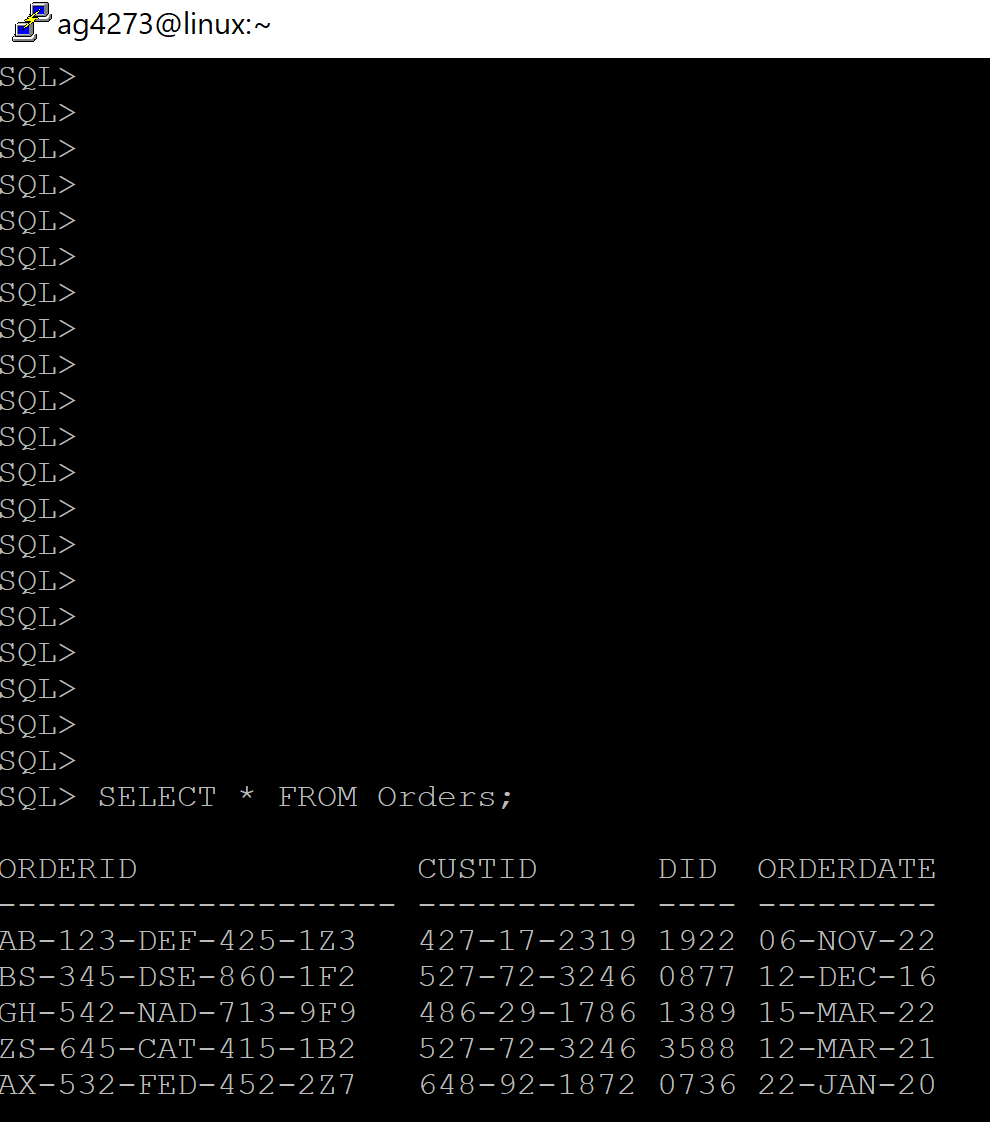
Delivery



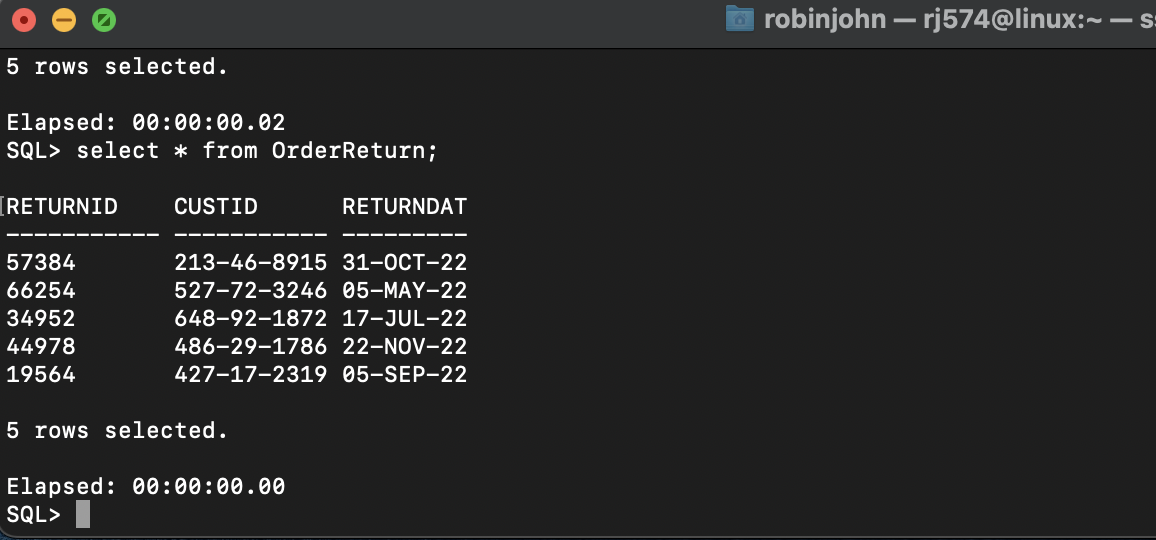
Customer



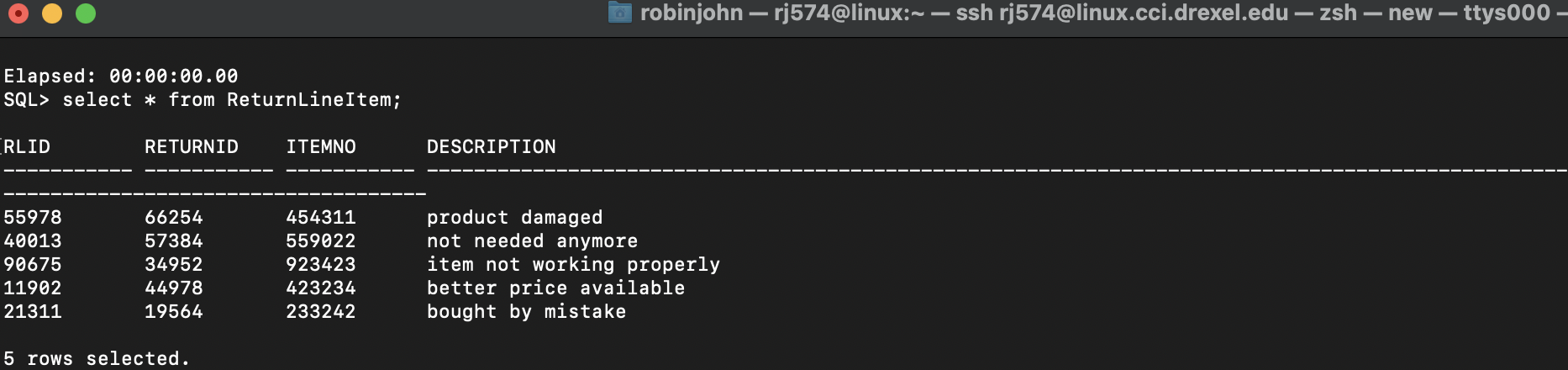
Order



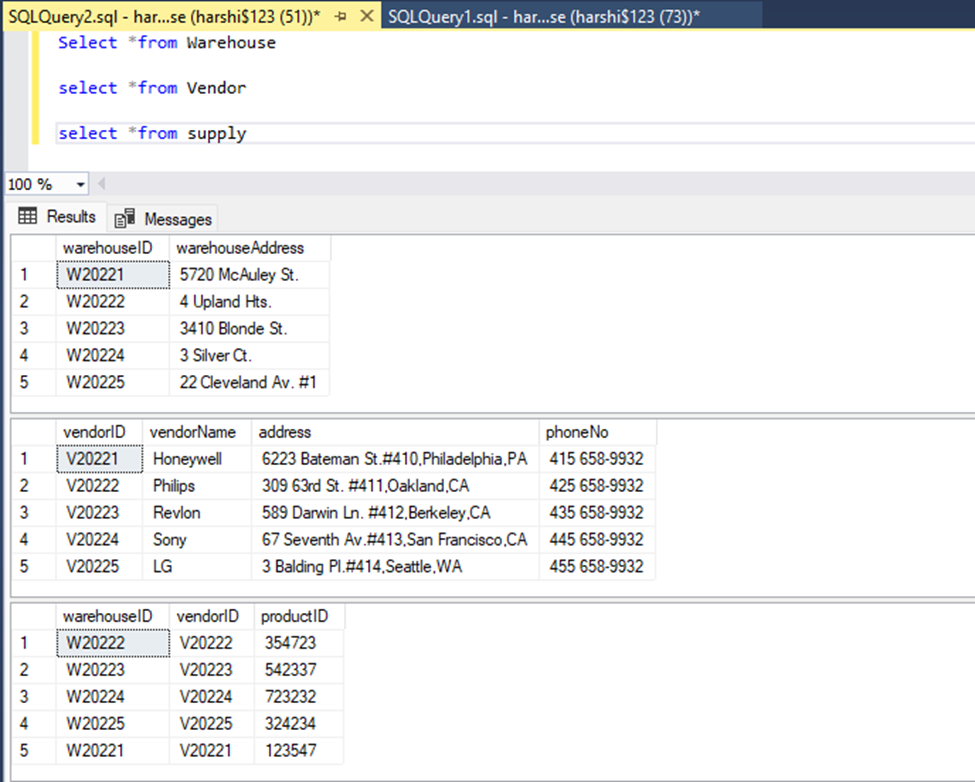
Order Return



Return Line Item



Warehouse, Vendor and Supply



1. DATA QUERIES
   1. Queries by David Osei-Poku
      1. Query 1

Display the first name and last name and delivery date of customers delivery date is before 2018:

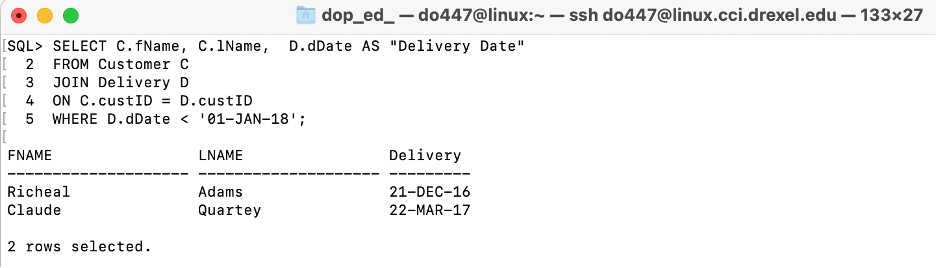
SELECT C.fName, C.lName, D.dDate AS “Delivery Date”

FROM Customer C

JOIN Delivery D

ON C.custID = D.custID

WHERE D.dDate < ‘01-JAN-18’;



* + 1. Query 2

Show the employeeID and number of deliveries of delivery drivers that have completed more than one delivery:

SELECT R.empID, COUNT( D.custID)

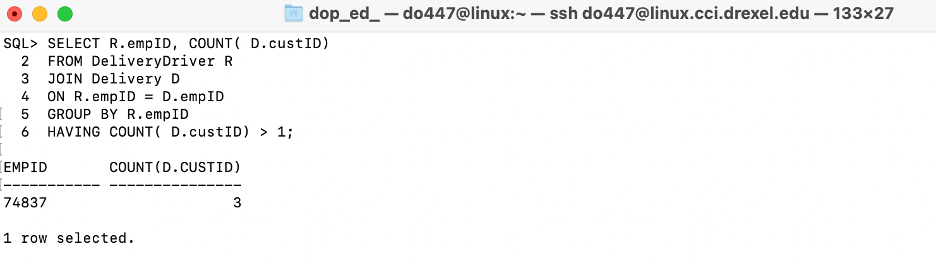
FROM DeliveryDriver R

JOIN Delivery D

ON R.empID = D.empID

GROUP BY R.empID

HAVING COUNT( D.custID) > 1;



* + 1. Query 3

Find the Distinct Orders that include ‘Electronics’, showing the product name of the product ordered:

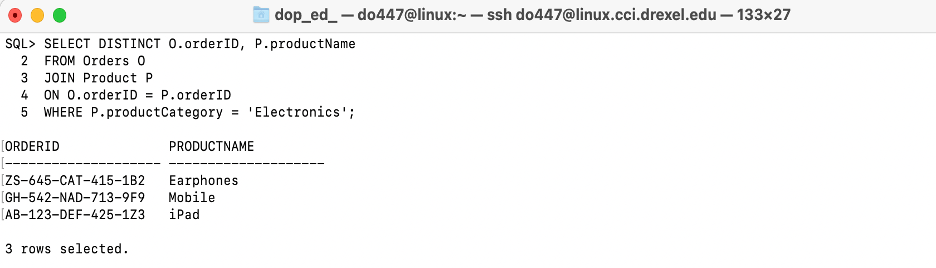
SELECT DISTINCT O.orderID, P.productName

FROM Orders O

JOIN Product P

ON O.orderID = P.orderID

WHERE P.productCategory = ‘Electronics’;



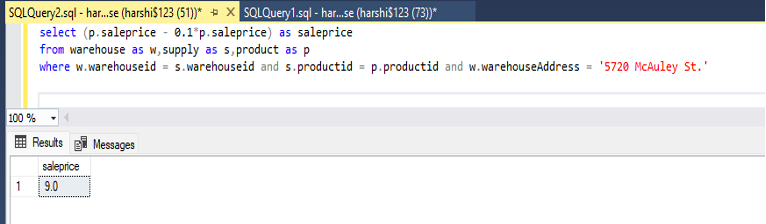
* 1. Queries by Harshitha Thippegundi Nagaraj
     1. Query 1

Calculate the sale price when the products stored at the warehouse with address ‘5720 McAuley St.’ are sold at a discount of 10%

SELECT (p.saleprice - 0.1\*p.saleprice) as saleprice

FROM warehouse as w,supply as s,product as p

WHERE w.warehouseid = s.warehouseid and s.productid = p.productid and w.warehouseAddress = '5720 McAuley St.'



* + 1. Query 2

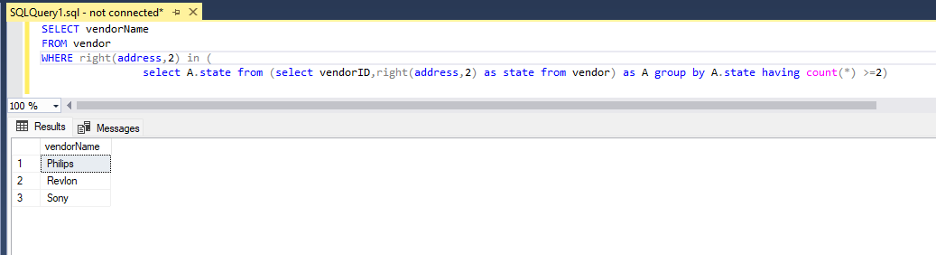
Find all the vendor names if there are more than two vendors for each state

SELECT vendorName

FROM vendor

WHERE right(address,2) in (

select A.state from (select vendorID,right(address,2) as state from vendor) as A group by A.state having count(\*) >=2)



* + 1. Query 3

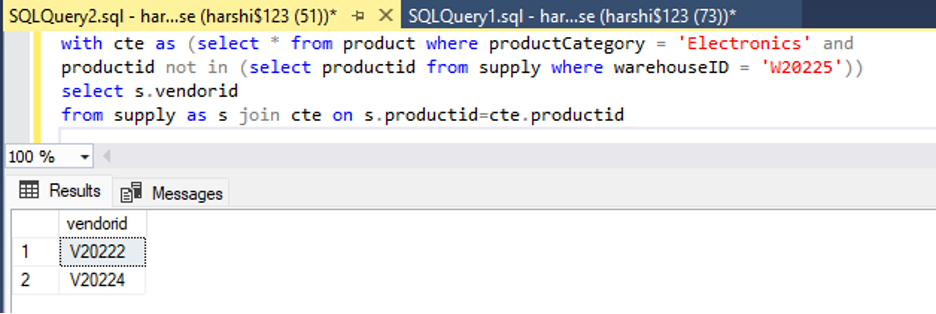
Find new vendorID for the ‘Electronics’ Product category stored at warehouse with ID W20225.

with cte as (select \* from product where productCategory = 'Electronics' and

productid not in (select productid from supply where warehouseID = 'W20225'))

SELECT s.vendorid

FROM supply as s join cte on s.productid=cte.productid



* 1. Queries by Arnav Goel
     1. Query 1

Get all the current customer names along with there order id and order date that live in Pennsylvania

SELECT o.orderID, c.fName ,c.lName, o.orderDate

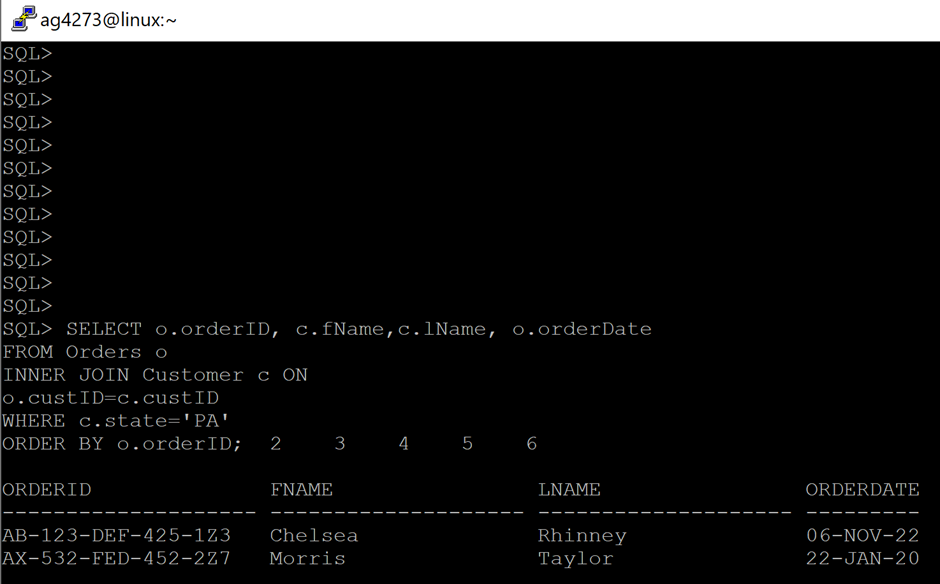
FROM Orders o

INNER JOIN Customer c ON

o.custID=c.custID

WHERE c.state='PA'

ORDER BY o.orderID;



* + 1. Query 2

Show the employee ID, first name and last name for managers working more than 20 hours and more than 10 years of experience to consider them for a raise.

SELECT m.empID, e.fname, e.lname

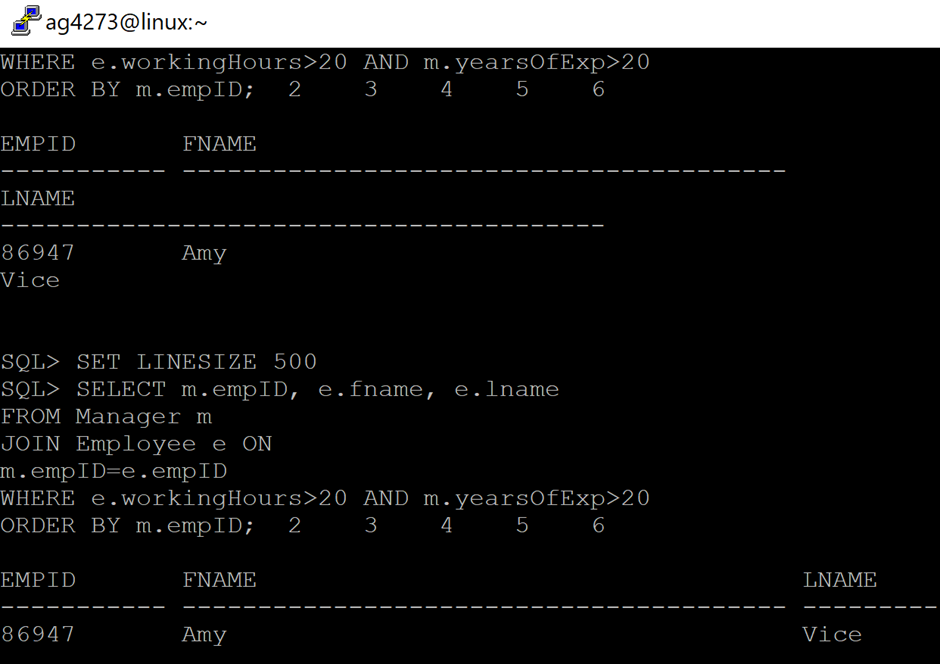
FROM Manager m

JOIN Employee e ON

m.empID=e.empID

WHERE e.workingHours>20 AND m.yearsOfExp>20

ORDER BY m.empID;



* + 1. Query 3

Get warehouse addresses that have products supplied in them with a cost price more than 300 to add more security there.

SELECT w.warehouseAddress

FROM Warehouse w

INNER JOIN Supply s

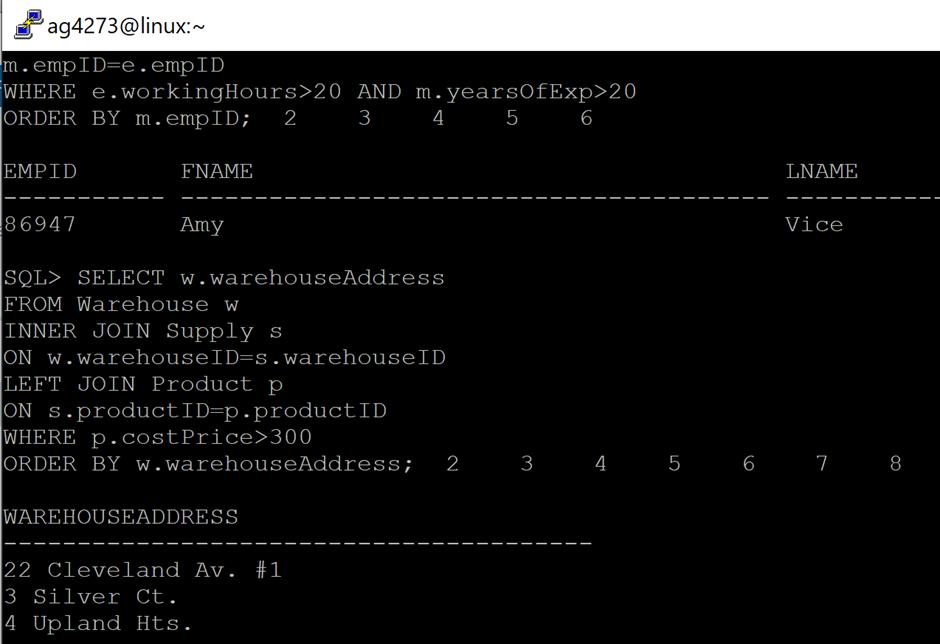
ON w.warehouseID=s.warehouseID

LEFT JOIN Product p

ON s.productID=p.productID

WHERE p.costPrice>300

ORDER BY w.warehouseAddress;



* 1. Queries by Divyanshu Kumar
     1. Query 1

Display employee ID, first name and last name for managers working more than 15 hours

SELECT ma.empID, em.fname, em.lname

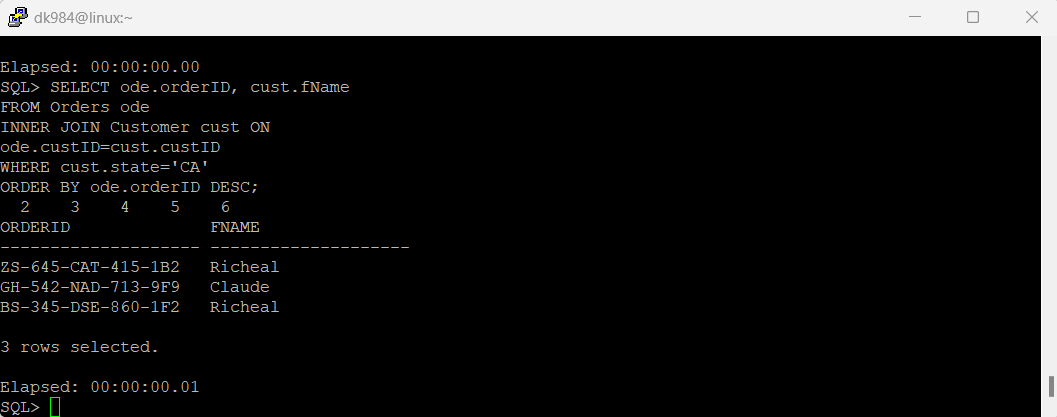
FROM Manager ma

JOIN Employee em ON

ma.empID=em.empID

WHERE em.workingHours>15

ORDER BY ma.empID;



* + 1. Query 2

Show the employeeID of delivery drivers that have state in common

SELECT E.empID, COUNT( D.state)

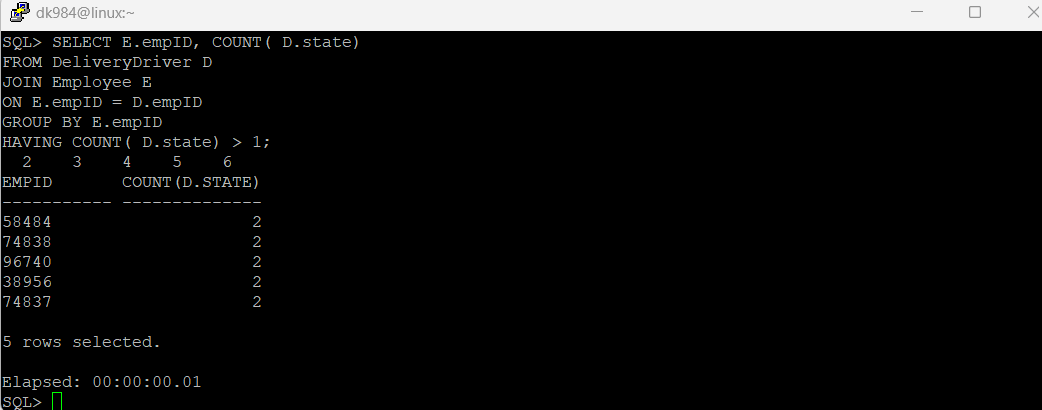
FROM DeliveryDriver D

JOIN Employee E

ON E.empID = D.empID

GROUP BY E.empID

HAVING COUNT( D.state) > 1;



* + 1. Query 3

Display details such as first name and order id of all customers from California

SELECT ode.orderID, cust.fName

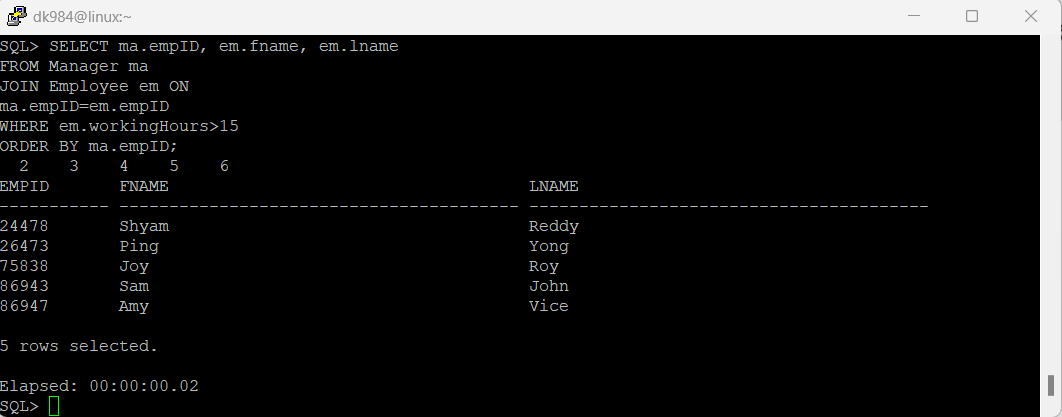
FROM Orders ode

INNER JOIN Customer cust ON

ode.custID=cust.custID

WHERE cust.state='CA'

ORDER BY ode.orderID DESC;



* 1. Queries by Robin John

i. Query 1

Get the customer's first name, last name, phone number who returned orders so that feedback can be taken by phone call.

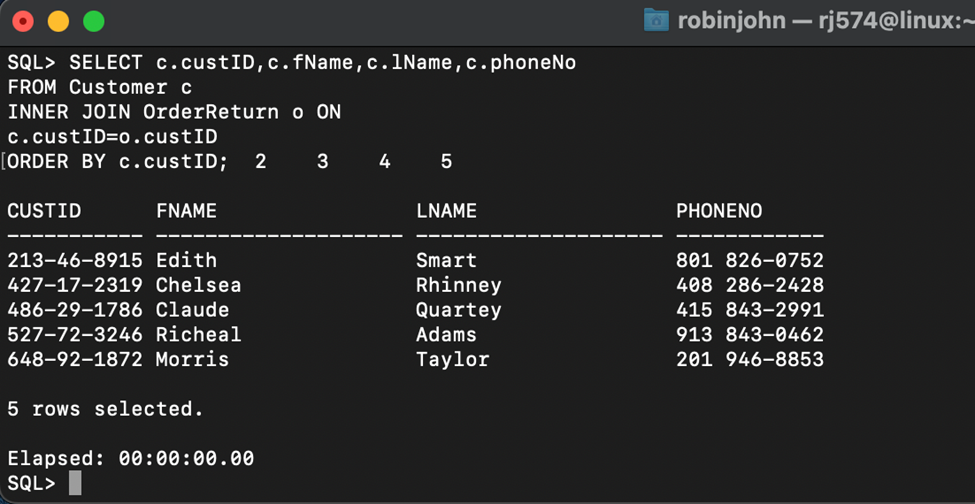
SELECT c.custID,c.fName,c.lName,c.phoneNo

FROM Customer c

INNER JOIN OrderReturn o ON

c.custID=o.custID

ORDER BY c.custID;



ii. Query 2

Getting address and orderID of customers living in CA.

SELECT cus.custID, o.orderID, cus.address, cus.phoneNo

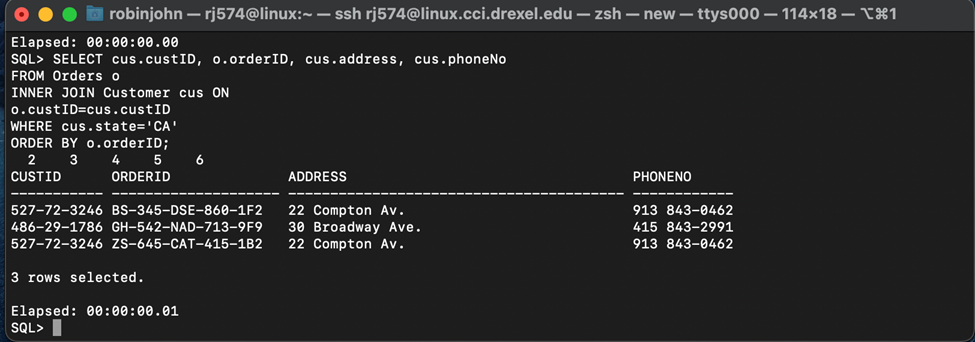
FROM Orders o

INNER JOIN Customer cus ON

o.custID=cus.custID

WHERE cus.state='CA'

ORDER BY o.orderID;



iii. Query 3

Show the first name employee ID, for managers working more than 20 hours.

SELECT e.fname, m.empID

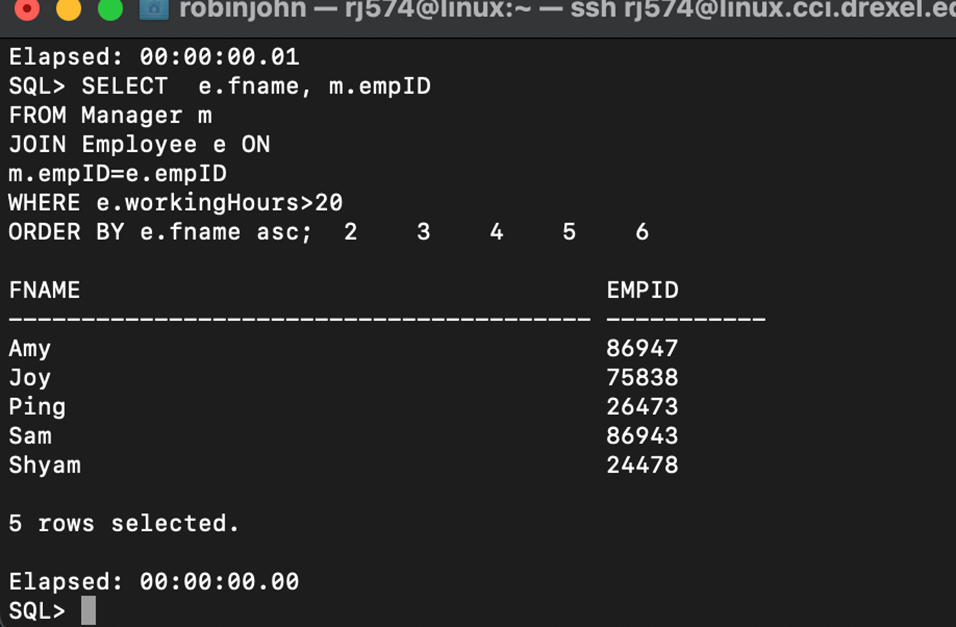
FROM Manager m

JOIN Employee e ON

m.empID=e.empID

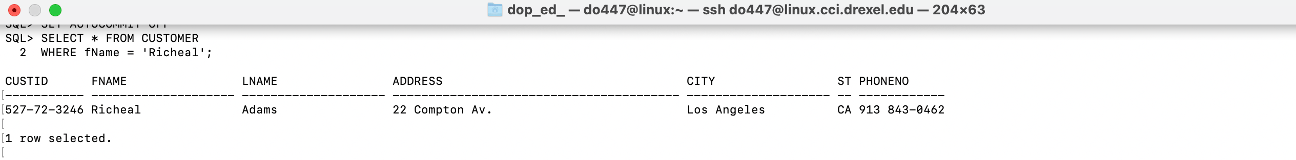
WHERE e.workingHours>20

ORDER BY e.fname asc;



1. DATA MANIPULATION
   1. DML by David Osei-Poku
      1. Data before the UPDATE command

SELECT \* FROM CUSTOMER WHERE fName = ‘Richeal’;



* + 1. UPDATE command

UPDATE CUSTOMER

SET fName = ‘Enyonam’

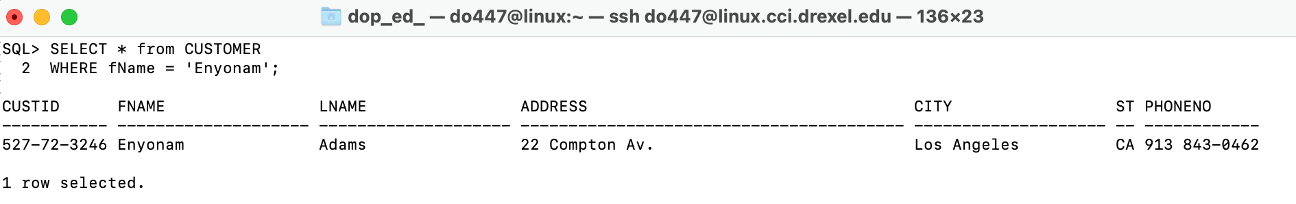
WHERE fName = ‘Richeal’;



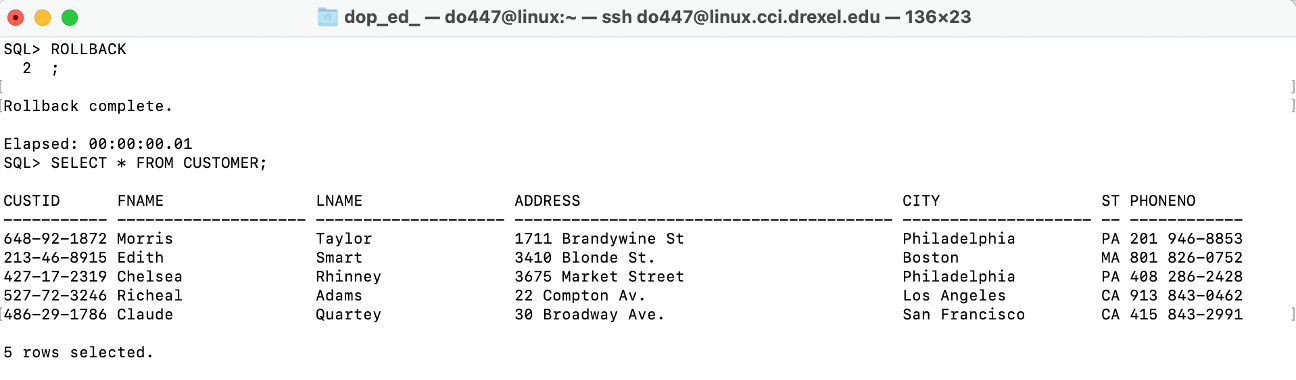
* + 1. Data after the UPDATE command

SELECT \* FROM CUSTOMER

WHERE fName = ‘Enyonam’;



* + 1. ROLLBACK

****

* + 1. Data before the DELETE command

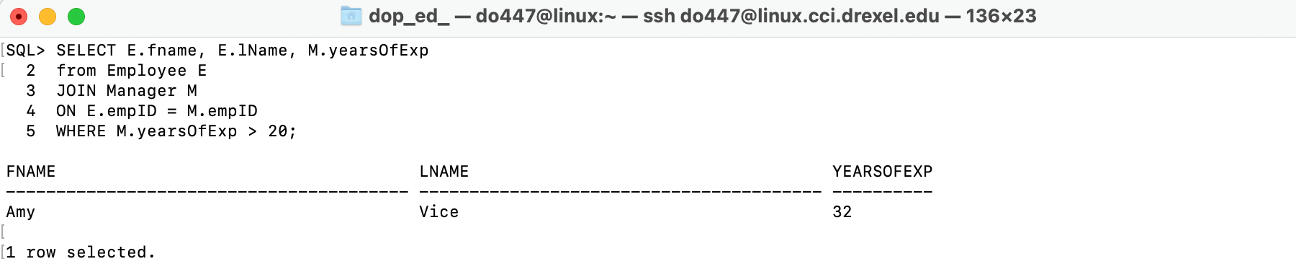
SELECT E.fName, E.lName, M.yearsofExp

FROM Employee E

JOIN Manager M

ON E.empID = M.empID

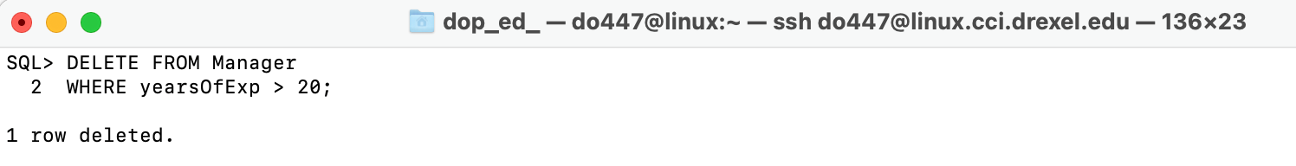
WHERE M.yearsOfExp > 20;



* + 1. DELETE command

DELETE FROM Manager

WHERE yearsOfExp > 20;



* + 1. Data after the DELETE command

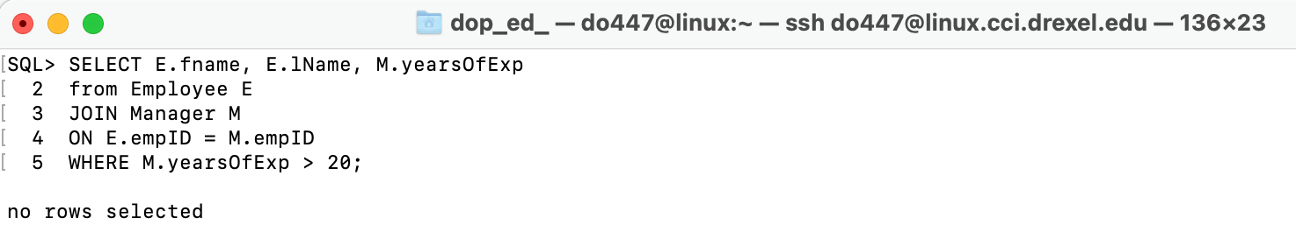
SELECT E.fName, E.lName, M.yearsofExp

FROM Employee E

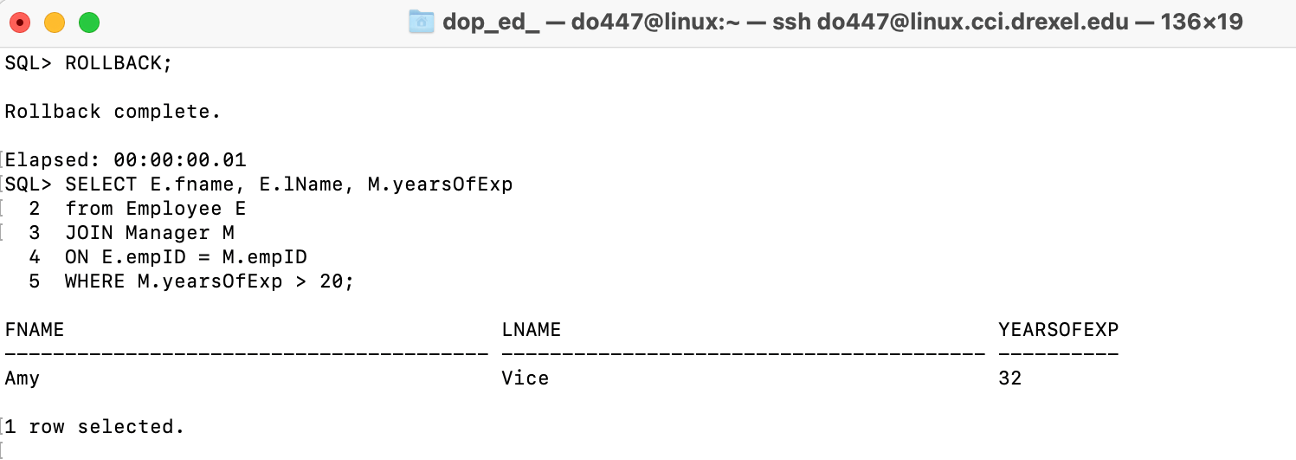
JOIN Manager M

ON E.empID = M.empID

WHERE M.yearsOfExp > 20;

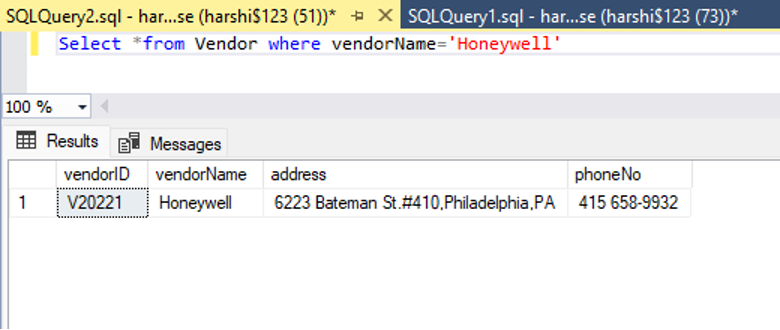


* + 1. ROLLBACK



* 1. DML Harshitha Thippegundi Nagaraj
     1. Data before the UPDATE command

Select \*from Vendor where vendorName='Honeywell'

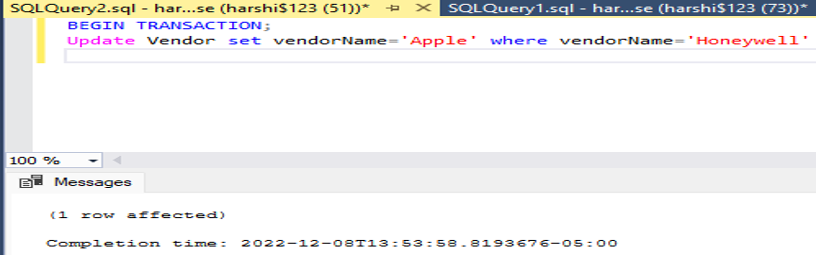


* + 1. UPDATE command

Update Honeywell Vendor name to Apple in Vendor Table.

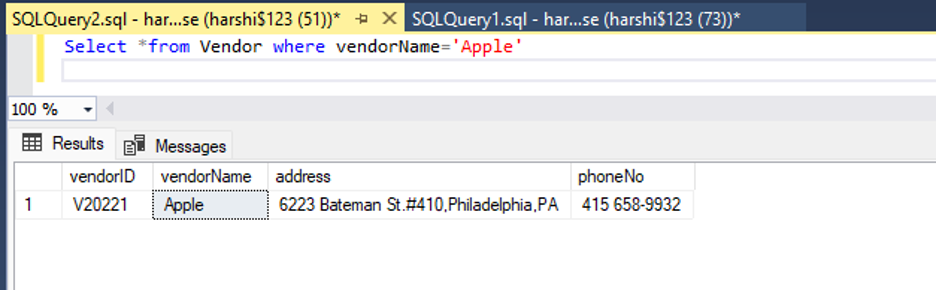
BEGIN TRANSACTION

Update Vendor set vendorName='Apple' where vendorName='Honeywell'



* + 1. Data after the UPDATE command

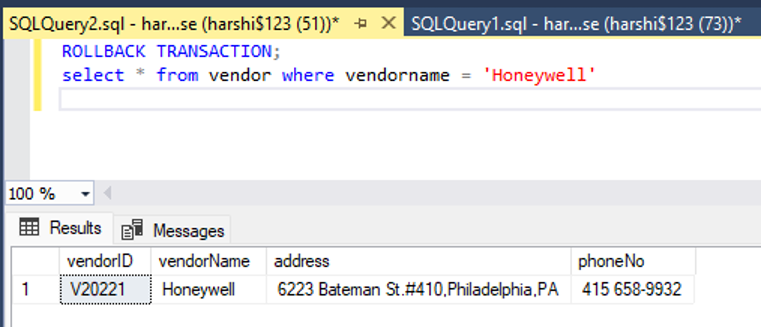
Select \*from Vendor where vendorName='Apple'



* + 1. ROLLBACK

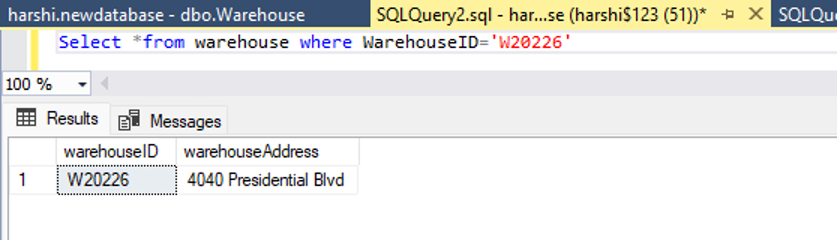
ROLLBACK TRANSACTION;

select \* from vendor where vendorname = 'Honeywell'



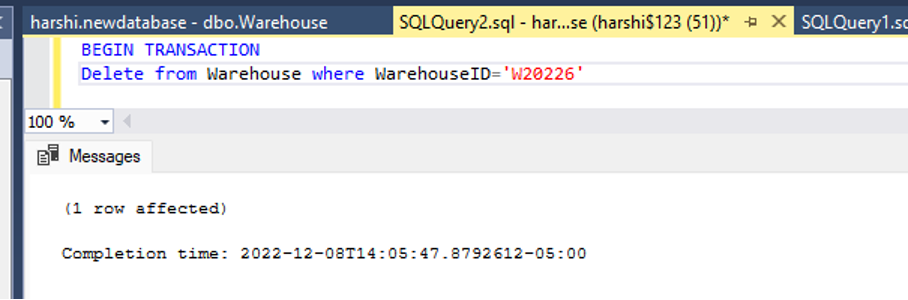
* + 1. Data before the DELETE command

Select \*from warehouse where WarehouseID='W20226'

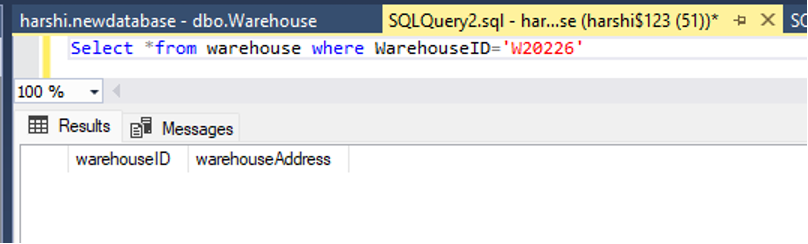


* + 1. DELETE command

Delete from Warehouse where WarehouseID='W20226'



* + 1. Data after DELETE command



* + 1. ROLLBACK

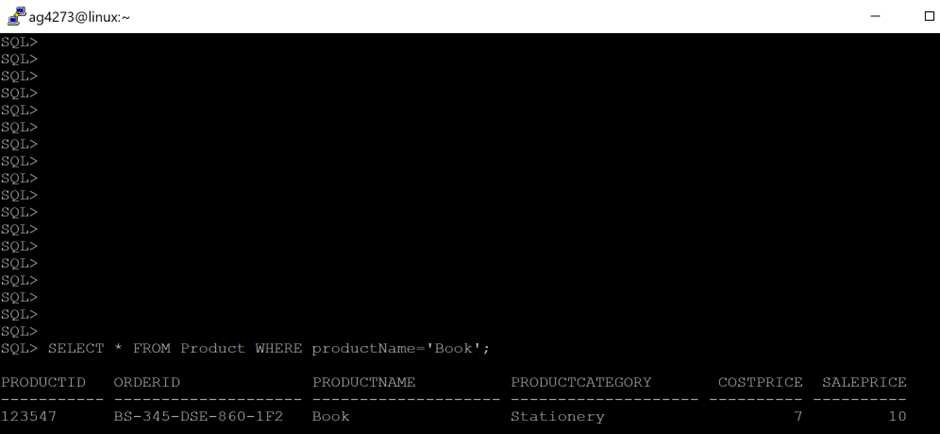
ROLLBACK TRANSACTION

Select \*from warehouse where WarehouseID='W20226'



* 1. DML by Arnav Goel
     1. Data before the UPDATE command

SELECT \* FROM Product WHERE productName='Book';



* + 1. UPDATE command

UPDATE Product

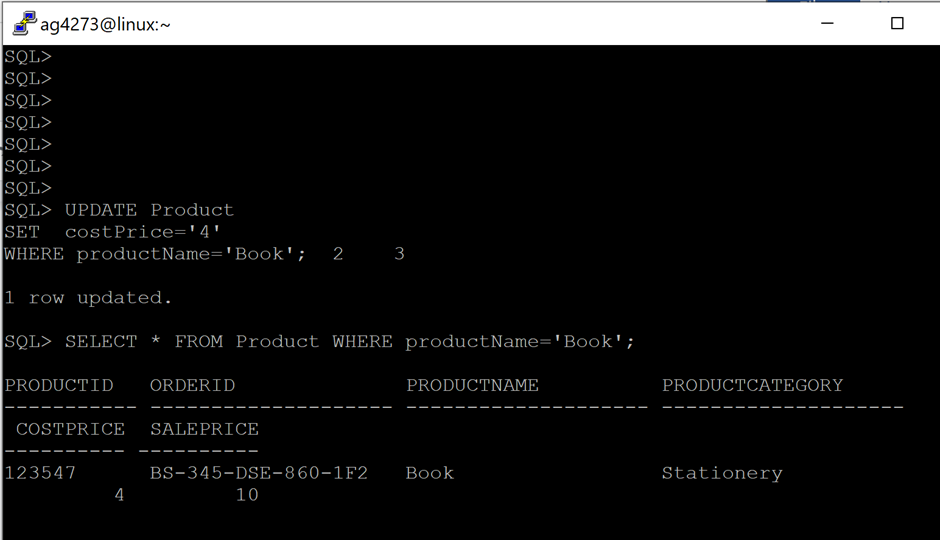
SET costPrice='4'

WHERE productName='Book';



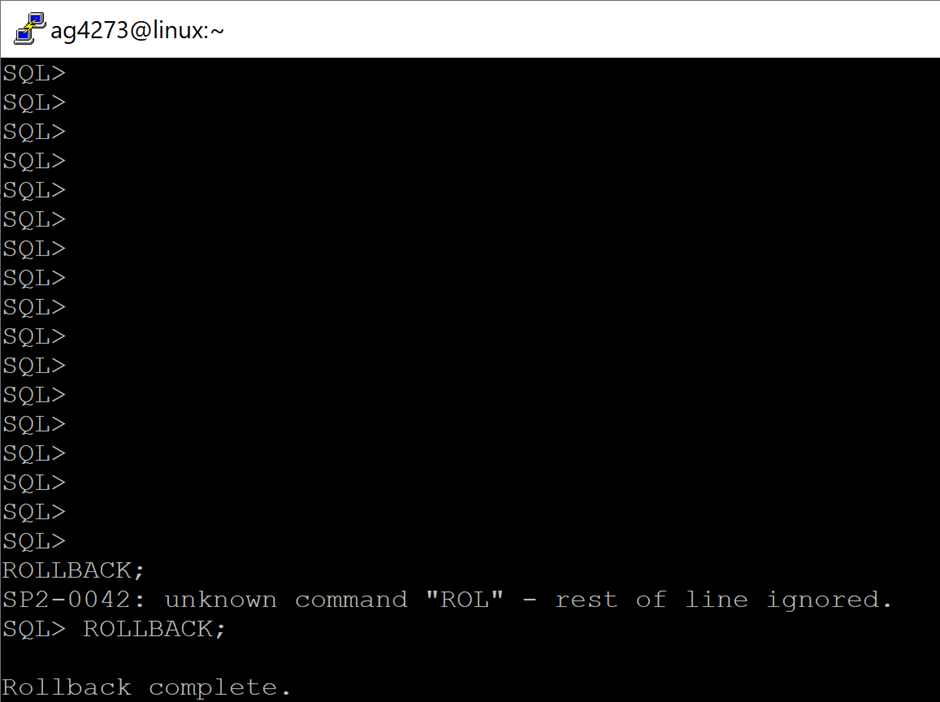
* + 1. Data after the UPDATE command

SELECT \* FROM Product WHERE productName='Book';



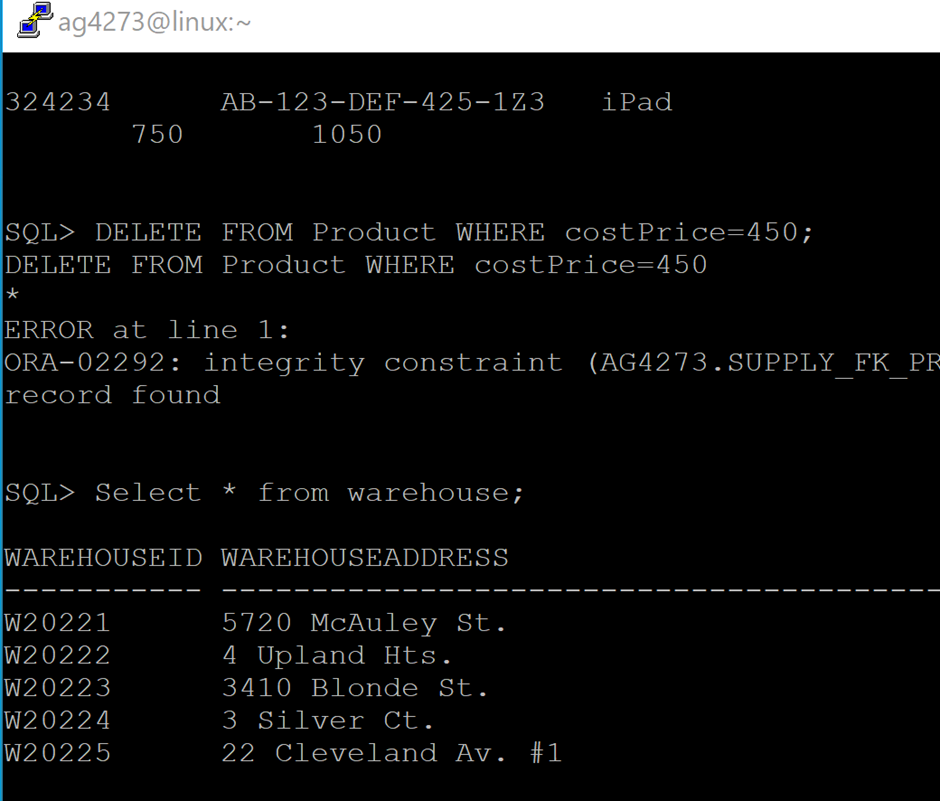
* + 1. ROLLBACK

ROLLBACK;



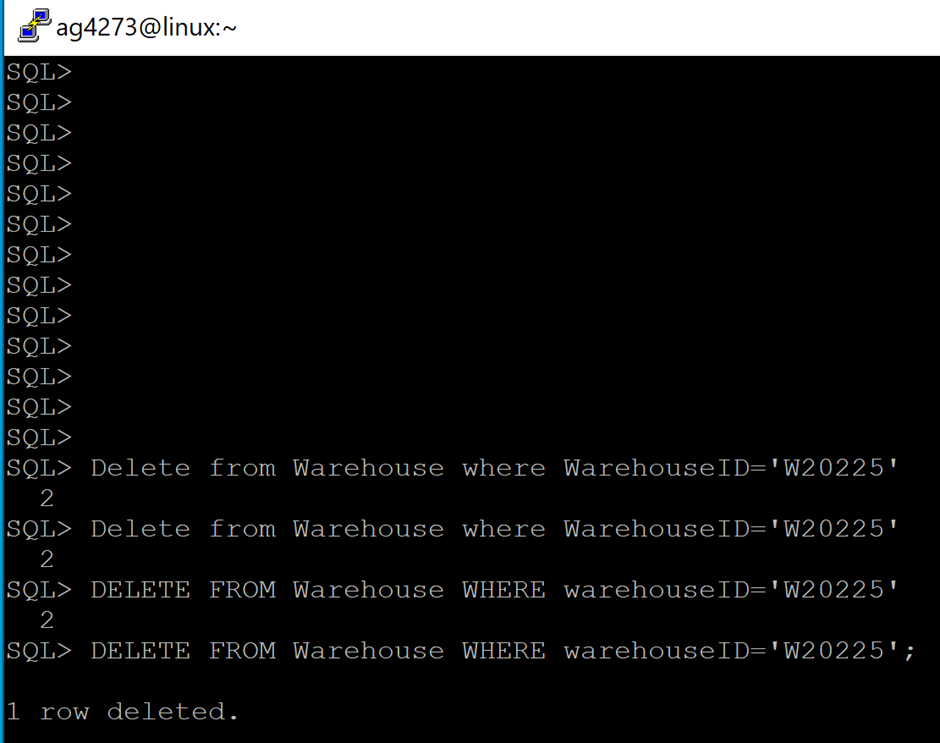
* + 1. Data before the DELETE command

Select \* from warehouse;



* + 1. DELETE command

DELETE FROM Warehouse WHERE warehouseID='W20225';



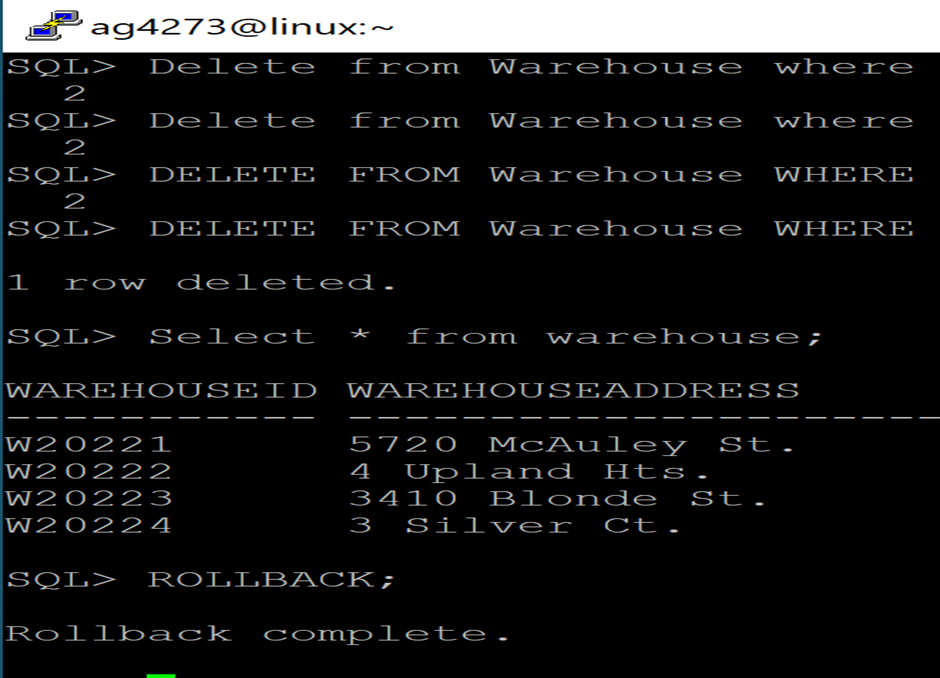
* + 1. Data after DELETE command

Select \* from warehouse;



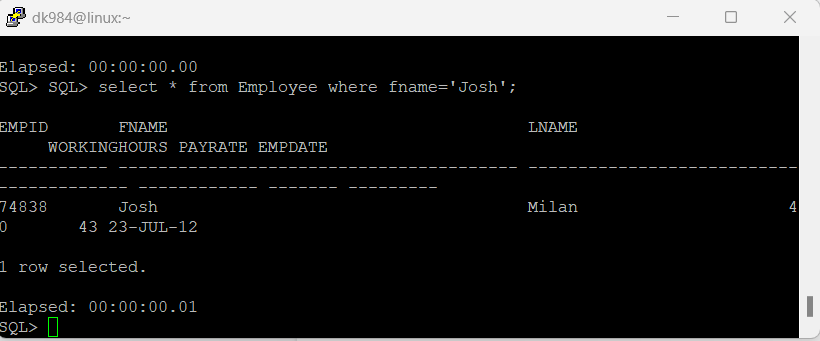
* + 1. ROLLBACK

ROLLBACK;



* 1. DML by Divyanshu Kumar
     1. Data before the UPDATE command

SELECT \* FROM Employee WHERE fName = ‘Josh’;



* + 1. UPDATE command

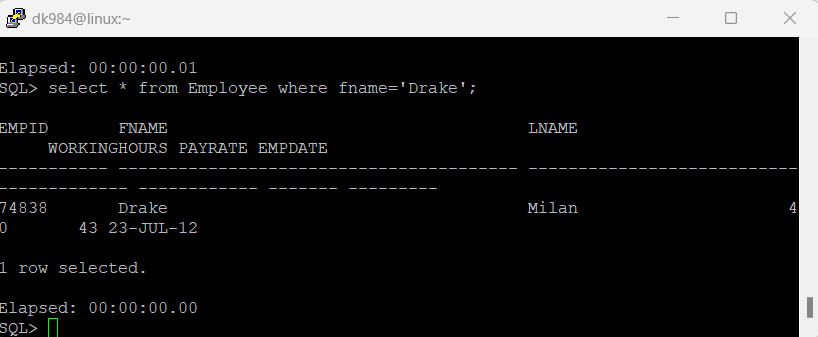
UPDATE Employee

SET fName = ‘Drake’

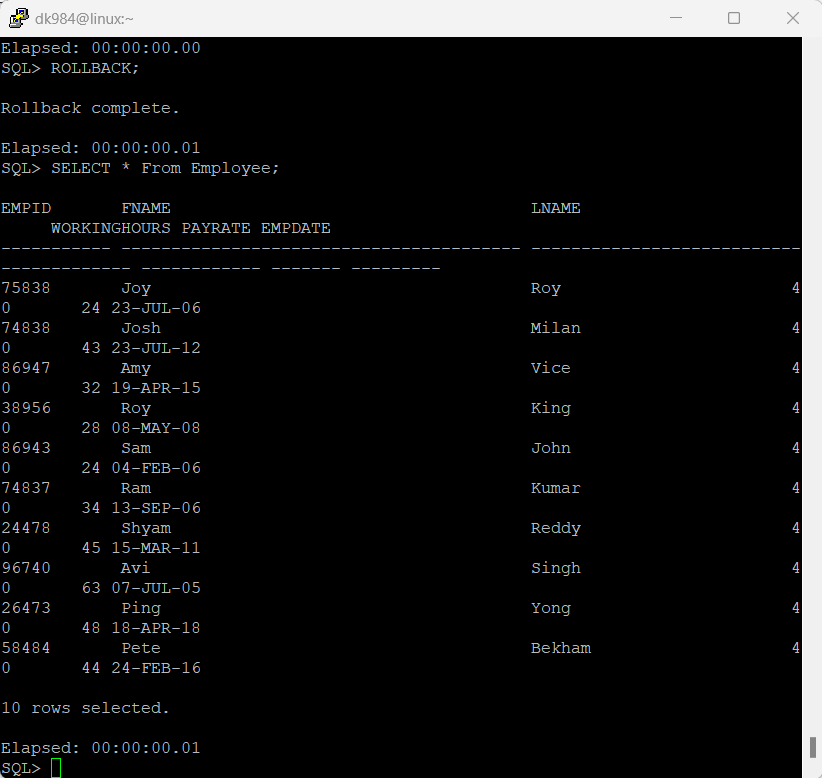
WHERE fName = ‘Josh’;

* + 1. Data after the UPDATE command

SELECT \* FROM Employee WHERE fName = ‘Drake’;



* + 1. ROLLBACK



* + 1. Data before the DELETE command

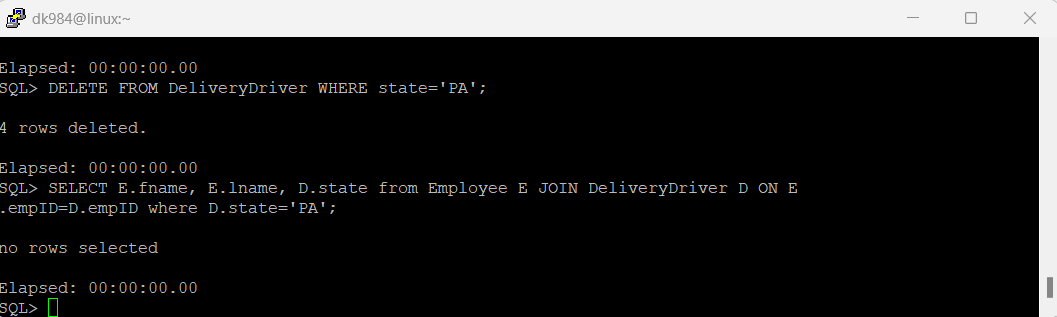
SELECT E.fname, E.lname, D.state from Employee E JOIN DeliveryDriver D ON E.empID=D.empID where D.state='PA';

* + 1. DELETE command

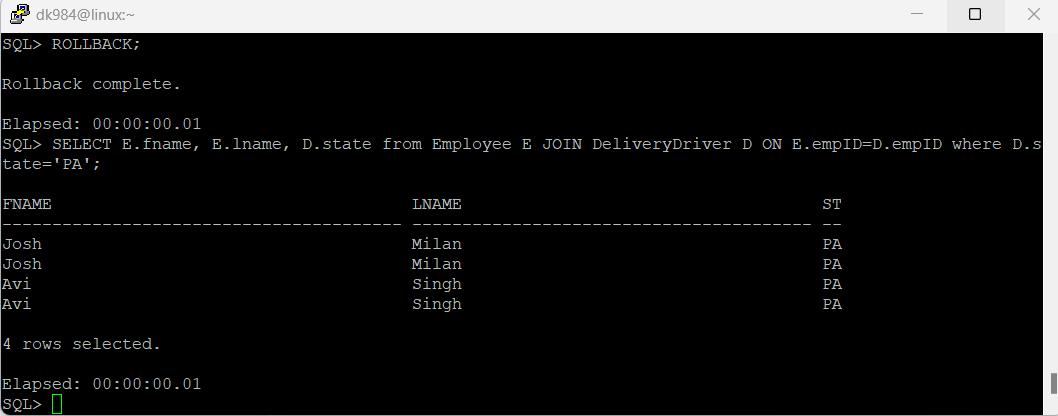
Delete From DeliveryDriver

WHERE state = ‘PA’;

* + 1. Data after DELETE command



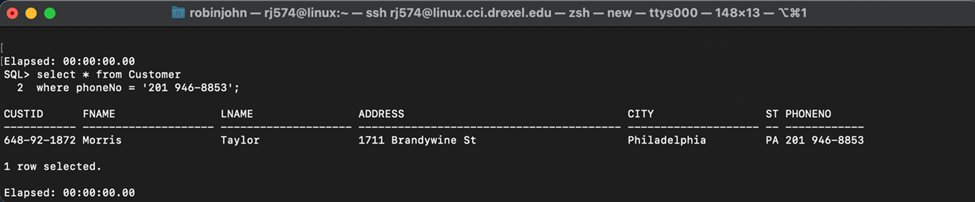
* + 1. ROLLBACK



* 1. DML by Robin John
* Data before the UPDATE command

select \* from Customer

where phoneNo = '201 946-8853';

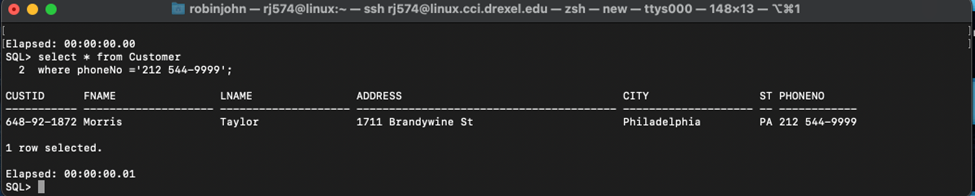


* UPDATE command

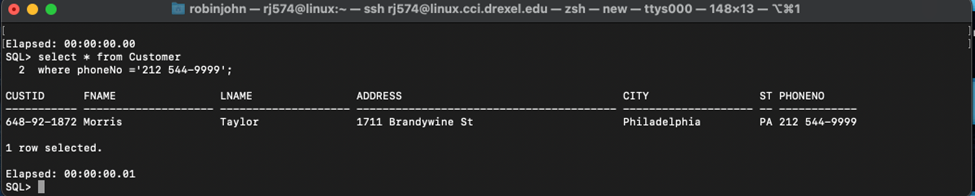
UPDATE Customer

SET phoneNo = '212 544-9999'

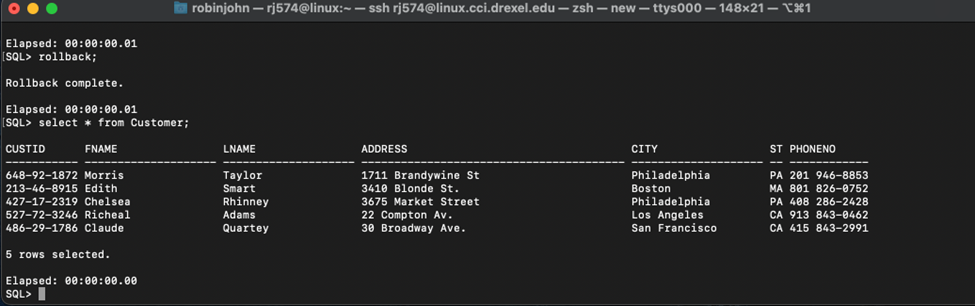
WHERE phoneNo = '201 946-8853';



* Data after the UPDATE command



* ROLLBACK



* Data before the DELETE command

Text

Description automatically generated with medium confidence

* DELETE command

delete from employee

WHERE empID = 75838;

* Data after DELETE command

Timeline

Description automatically generated with medium confidence

* ROLLBACK

Text

Description automatically generated

10.SUMMARY

* 1. Summary by David Osei-Poku

This course has given me a solid foundation in database creation and management using SQL which I was relatively new to. In my previous work experience, I had the opportunity to develop databases using Microsoft Excel but always felt like my skills were limited and could use some improvement. With the knowledge and techniques acquired over the last couple of months I believe I am well equipped to tackle the world of database management. I am now confident that the fundamental knowledge gained in the principles of relational databases and entity-relationship diagram development throughout this period and in the completion of my term project has put me on the right path to achieve my goals in becoming an effective database administrator regardless of my environment. In conclusion, I have enjoyed learning and using SQL to solve problems and look forward to continuing in that stead in the years to come.

* 1. Summary Harshitha Thippegundi Nagaraj

I had a great exposure to all the database concepts. Previously I have worked in the IT industry and I was involved in creating databases and tables. In this project, I was involved in the planning phase of listing out entities, establishing relationships among them and creating an ER diagram. With the conceptual diagram, I learnt about how the UML diagram can be used to establish constraints and create foreign keys relations between tables. Upon creating relationships between tables, I was able to write DDL and DML queries for various joins and aggregation functions. I also picked up a few special SQL tricks like subqueries, views and grouping the dimensions in a table. With the knowledge I have gained I would like to further work on learning advanced SQL queries like CTE’s and temporary tables.

* 1. Summary by Arnav Goel

This course provided me with an opportunity to learn about databases which will lay the foundation for my further courses in my major data science. I’ve had some past experience with the databases but this course has filled the gaps in my knowledge to the point where I’m confident in handling sql databases in the future. This course will help me complete certain future projects that I have in my mind for a long time that require some serious data handling and data manipulation. I’m planning to take some further database courses to gain more knowledge in this field, but haven’t decided on which ones to take yet but that is what I look for in the future.

* 1. Summary by Divyanshu Kumar

I have a lot of experience with data manipulation with pandas and other python libraries but I had never worked on SQL before. I know that SQL is very important to become a data scientist and that's why I took this course. I am really happy as I not only got enough experience with SQL but concepts of database management as well. I am sure that this will help me a lot in my career and also give me an edge over others in the interviews. I can now confidently put up SQL and database management on my resume thanks to INFO-605(DBMS).

* 1. Summary by Robin John

I was always fascinated by the data world. I have very much interest in data science, database management, machine learning, etc. Even though I am from a cyber security background I took up this course to quench my thirst for working in the data field. I feel empowered after completing this course as I am now able to qualify for multidisciplinary roles. I am going to be taking more courses in this direction with my cyber security courses so that I can add more skills to my skill set and solve real-life problems as when you are a working professional you can never rely only on one of the skills. I am thankful that I took this course.

11.Appendix

Work was distributed equally and everyone contributed to the best of their abilities.