PROJECT

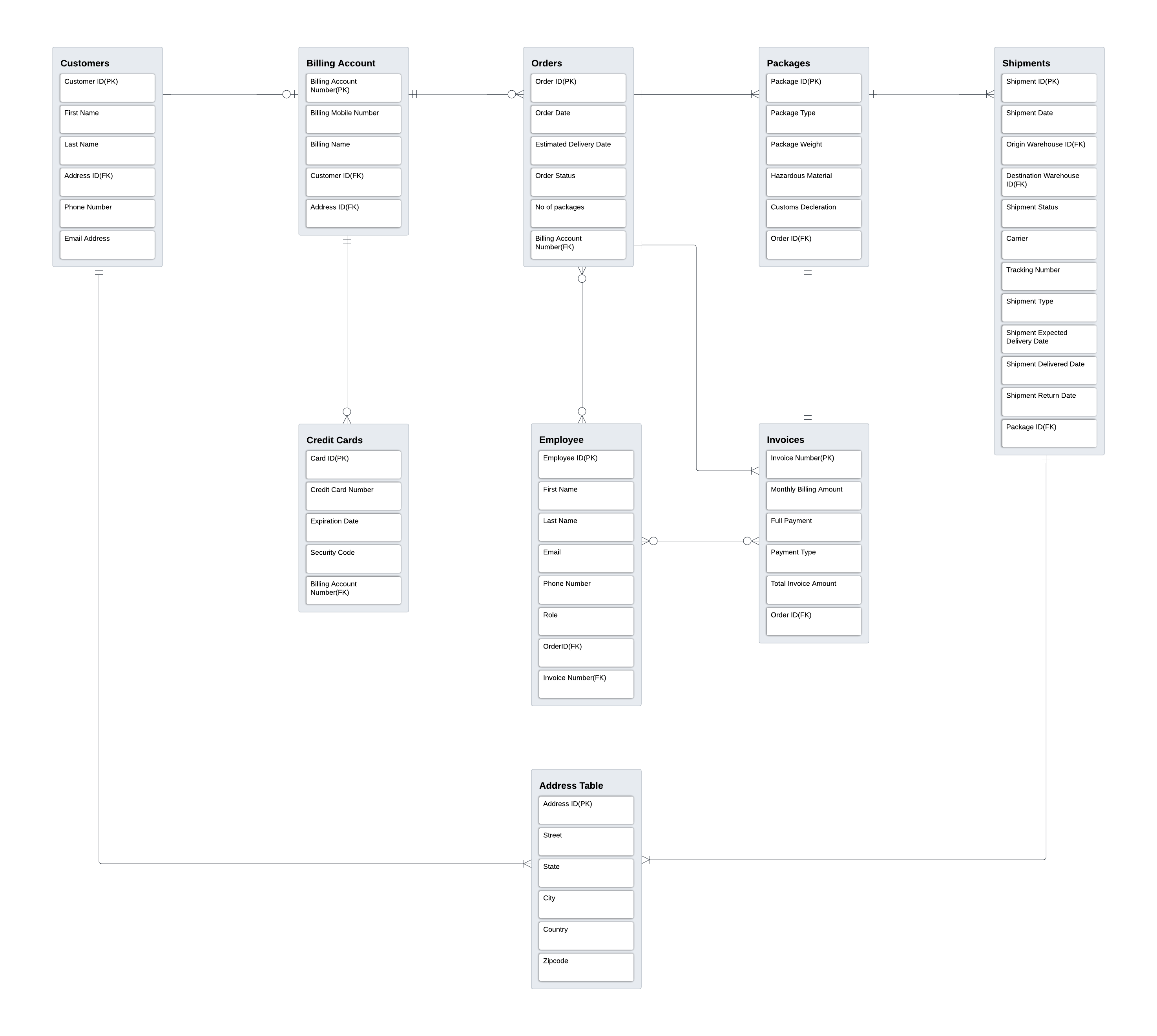
Database Management System

Developing New Application to support the activities of package company

Rup Kiran Boyana

1**. E-R Model**

**E-R diagram representing the conceptual design of the database.**



Based on the information provided, here is a proposal for the database design:

I have considered a total of 9 entities and each entity has a primary key and foreign key, below are the list of primary keys.

**Primary Key Attributes -**

Customer\_ID\_PK is the primary key for the Customers Table

Billing\_Account\_number\_PK is the primary key for the BillingAccounts Table

Order\_ID\_PK is the primary key for the Orders Table

Invoice\_number\_PK is the primary key for the Invoices Table

Package\_ID\_PK is the primary key for the Packages Table

Shipment\_ID\_PK is the primary key for the Shipments Table

Card\_ID\_PK is the primary key for the CreditCards Table

Address\_ID\_PK is the primary key for the Address Table

Employee\_ID\_PK is the primary key for the Employee Table

**Relationships:**

- A customer can have only one billing account or no billing account and one billing account is

related to only one customer. (ONE TO ONE RELATION)

A screenshot of a computer

Description automatically generated with medium confidence

- A customer can have many orders and each order is related to only one billing account

(ONE TO MANY RELATION)

A picture containing text, screenshot, number, font

Description automatically generated

- An order can have multiple packages, multiple Invoices, and a single package or a unique

Invoice is related to only one order. (ONE TO MANY RELATION), BINARY RELATIONSHIP.

A picture containing text, screenshot, diagram, font

Description automatically generated

- A package can have only one Invoice (ONE TO ONE RELATION)

A screenshot of a computer

Description automatically generated with medium confidence

- A package can be in many shipments, but each shipment is still linked to a single package.

(ONE TO ONE) OR (ONE TO MANY RELATION)

A screenshot of a logistic

Description automatically generated with low confidence

- A Billing Account can have multiple credit cards, but each card is linked to a single account

(ONE TO MANY RELATION)

A picture containing text, screenshot, font, number

Description automatically generated

- A customer can have a single address or multiple addresses and multiple shipments can have

same address(ONE TO ONE) OR (ONE TO MANY RELATION)

A picture containing text, screenshot, diagram, parallel

Description automatically generated

- An employee or multiple employees can work on a single order, multiple order or no orders

- An employee or multiple employees can work on a single Invoice, multiple Invoice or no

Invoice. (ONE TO ONE), (ONE TO MANY), OR (MANY TO ONE RELATION)

A picture containing text, screenshot, diagram, parallel

Description automatically generated

**Weak Entities-**

Here tables Orders, Packages, shipments, credit cards and Invoices can be considered to weak entities as these are all dependent on the Customer table and the Customers ID.

If there are no customers, then there will be no orders which leads no packages and shipments.

**Normalizations –**

**1 NF –**

****

Rules – 1. Each table cell should contain only a single value.

2. Each record needs to be unique.

Following the above rules, I have created a single table with 16 rows and 58 columns which has all the data related to 9 entities and there attributes. Please refer to the above attached excel.

**2 NF –**

Rules – 1. Be in 1NF

2. Single column primary should not be functionally dependent on any subset of candidate-

key relation.

Here we have divided 1NF into eight tables and assigned primary keys and foreign keys for all the below listed tables to avoid partial dependencies. All the below listed tables are linked in a hierarchical structure. Please refer to the above attached excel.

Table 1 contains Customers Information

Table 2 contains Customers Billing Accounts Information

Table 3 contains Customer Order Information

Table 4 contains Invoice Information

Table 5 contains Packages Information

Table 6 contains Shipments Information

Table 7 contains Credit Cards Information

Table 8 contains Employee Information

* Customer ID is the primary key in the Customers table and foreign key in the Billing Accounts table.
* Billing Account number is the primary key in Billing Accounts table and foreign key in the Orders table and Credit Cards table.
* Order ID is the primary key in the Orders table and foreign key in the Invoice table, Packages Table and Employee table
* Invoice number is the primary key in the Invoices table and foreign key in the Employee table
* Package ID is the primary key in the Packages table and foreign key in the Shipments table
* Card ID is the primary key in the Credit Cards table
* Employee ID is the primary key in the Employees table.

**3 NF –** Rules – 1. Be in 2NF

2. Has no transitive functional dependency.

Here in the third normal form we have created Address table and linked to the Customers table, Billing Accounts table and Shipments table to avoid the transitive dependency’s. Please refer to the above attached excel.

Fields – Street, City, State, Country have been remove from the tables Customers, Billing Accounts and Shipments Tables.

Address ID is the primary key in the Address table and foreign key in the Customers, Billing Accounts and Shipments Tables.

About our Script File –



1. We have create a Database package named PackageDBBB.
2. In the PackageDBBB database we have created 9 tables and assigned primary keys to the attributes.
3. Later we have Inserted data into the created tables.
4. After the data has been loaded Constraints have been added and assigned foreign keys to the attributes.
5. Finally to check the database we have run the below test queries. Please refer to the above attached script file.

Test queries and their outputs-

--Assume truck 1721 is destroyed in a crash. Find all customers who had a package on that truck at the time of the crash. Write a query to list all recipients who had a

--package on that truck at the time of the crash. Find the last successful delivery by that truck prior to the crash.

USE PackageDBBB

SELECT Customers.Customer\_ID\_PK, Customers.First\_name, Customers.Last\_name, Packages.Package\_ID\_PK, Shipments.Shipment\_ID\_PK, Packages.Package\_type AS Service, Packages.Package\_weight\_lb, Invoices.Total\_Invoice\_Amount,Shipments.Shipment\_Type, Shipments.Shipment\_status

FROM Customers

INNER JOIN BillingAccounts ON Customers.Customer\_ID\_PK = BillingAccounts.Customer\_ID\_FK

INNER JOIN Orders ON BillingAccounts.Billing\_Account\_number\_PK = Orders.Billing\_Account\_number\_FK

INNER JOIN Invoices ON Orders.Order\_ID\_PK = Invoices.Order\_ID\_FK

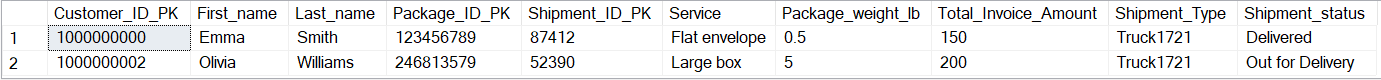
INNER JOIN Packages ON Orders.Order\_ID\_PK = Packages.Order\_ID\_1

INNER JOIN Shipments ON Packages.Package\_ID\_PK = Shipments.Package\_ID\_FK

WHERE Shipments.Shipment\_Type='Truck1721'

GROUP BY Customers.Customer\_ID\_PK, Customers.First\_name, Customers.Last\_name, Invoices.Total\_Invoice\_Amount, Packages.Package\_type, Packages.Package\_ID\_PK, Shipments.Shipment\_ID\_PK, Packages.Package\_weight\_lb, Shipments.Shipment\_status, Shipments.Shipment\_Type

Output –



--Write a query to list the customer who has shipped the most packages in the past year

SELECT Customers.Customer\_ID\_PK, Customers.First\_name, Customers.Last\_name,Orders.Order\_Date, Orders.No\_of\_packages

FROM Customers

INNER JOIN BillingAccounts ON Customers.Customer\_ID\_PK = BillingAccounts.Customer\_ID\_FK

INNER JOIN Orders ON BillingAccounts.Billing\_Account\_number\_PK = Orders.Billing\_Account\_number\_FK

WHERE Orders.Order\_Date <= DATEADD(year, -1, GETDATE()) and Orders.No\_of\_packages > 1

GROUP BY Customers.Customer\_ID\_PK, Customers.First\_name, Customers.Last\_name, Orders.Order\_Date, Orders.No\_of\_packages

Output –

A screenshot of a computer

Description automatically generated with low confidence

--Write a query to list the customer who has spent the most money on shipping in the past year.

SELECT Customers.Customer\_ID\_PK, Customers.First\_name, Customers.Last\_name, SUM(Total\_Invoice\_Amount) AS Highest\_Invoice\_Amount

FROM Customers

INNER JOIN BillingAccounts ON Customers.Customer\_ID\_PK = BillingAccounts.Customer\_ID\_FK

INNER JOIN Orders ON BillingAccounts.Billing\_Account\_number\_PK = Orders.Billing\_Account\_number\_FK

INNER JOIN Invoices ON Orders.Order\_ID\_PK = Invoices.Order\_ID\_FK

WHERE Orders.Order\_Date <= DATEADD(year, -1, GETDATE())

GROUP BY Customers.Customer\_ID\_PK, Customers.First\_name, Customers.Last\_name

Output-

A screenshot of a computer

Description automatically generated with medium confidence

--Write a query to list the state with the most customers.

SELECT State, COUNT(\*) AS Repetitions

FROM Customers

INNER JOIN AddressTable ON Customers.Address\_ID\_FK = AddressTable.Address\_ID\_PK

GROUP BY State

HAVING COUNT(\*) > 1;

Output-

A screenshot of a computer

Description automatically generated with low confidence

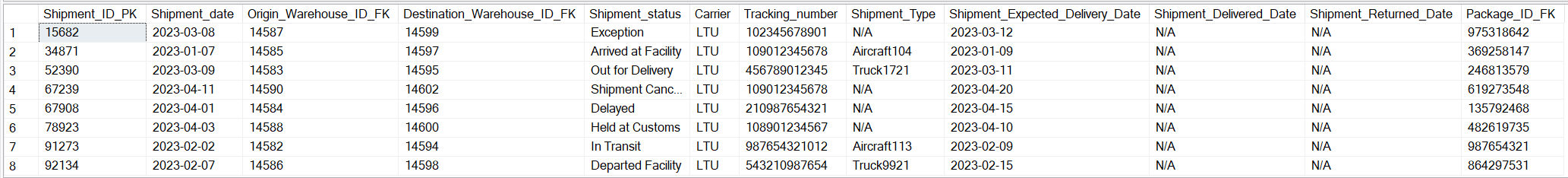
--Write a query to list those packages that were not delivered within the promised time.

SELECT \* FROM Shipments

WHERE Shipment\_Status != 'Delivered' and Shipment\_Status != 'Returned to Sender'

AND Shipment\_Delivered\_Date = 'N/A';

Output –



--Write queries that can be used to generate the following bills. These forms will be sent to each customer in the past month:

-- A simple bill: customer, address, and amount owed.

SELECT Customers.Customer\_ID\_PK, Customers.First\_name, Customers.Last\_name,Invoices.Total\_Invoice\_Amount- Invoices.Monthly\_billing\_amount AS Amount\_Owed

FROM Customers

INNER JOIN BillingAccounts ON Customers.Customer\_ID\_PK = BillingAccounts.Customer\_ID\_FK

INNER JOIN Orders ON BillingAccounts.Billing\_Account\_number\_PK = Orders.Billing\_Account\_number\_FK

INNER JOIN Invoices ON Orders.Order\_ID\_PK = Invoices.Order\_ID\_FK

WHERE Invoices.Monthly\_billing\_amount <> 'N/A' OR Invoices.Monthly\_billing\_amount IS NULL

GROUP BY Customers.Customer\_ID\_PK, Customers.First\_name, Customers.Last\_name, Invoices.Total\_Invoice\_Amount, Invoices.Monthly\_billing\_amount

Output –

A screenshot of a computer

Description automatically generated with low confidence

-- A bill listing charges by type of service.

SELECT Customers.Customer\_ID\_PK, Customers.First\_name, Customers.Last\_name, Packages.Package\_type AS Service, Invoices.Total\_Invoice\_Amount

FROM Customers

INNER JOIN BillingAccounts ON Customers.Customer\_ID\_PK = BillingAccounts.Customer\_ID\_FK

INNER JOIN Orders ON BillingAccounts.Billing\_Account\_number\_PK = Orders.Billing\_Account\_number\_FK

INNER JOIN Invoices ON Orders.Order\_ID\_PK = Invoices.Order\_ID\_FK

INNER JOIN Packages ON Orders.Order\_ID\_PK = Packages.Order\_ID\_1

WHERE Packages.Package\_type <> 'N/A' OR Packages.Package\_type IS NULL

GROUP BY Customers.Customer\_ID\_PK, Customers.First\_name, Customers.Last\_name, Invoices.Total\_Invoice\_Amount, Packages.Package\_type

Output –

A screenshot of a computer

Description automatically generated with low confidence

-- An itemize billing listing each individual shipment and the charges for it.

SELECT Customers.Customer\_ID\_PK, Customers.First\_name, Customers.Last\_name, Packages.Package\_ID\_PK, Shipments.Shipment\_ID\_PK, Packages.Package\_type AS Service, Packages.Package\_weight\_lb, Invoices.Total\_Invoice\_Amount

FROM Customers

INNER JOIN BillingAccounts ON Customers.Customer\_ID\_PK = BillingAccounts.Customer\_ID\_FK

INNER JOIN Orders ON BillingAccounts.Billing\_Account\_number\_PK = Orders.Billing\_Account\_number\_FK

INNER JOIN Invoices ON Orders.Order\_ID\_PK = Invoices.Order\_ID\_FK

INNER JOIN Packages ON Orders.Order\_ID\_PK = Packages.Order\_ID\_1

INNER JOIN Shipments ON Packages.Package\_ID\_PK = Shipments.Package\_ID\_FK

WHERE Packages.Package\_type <> 'N/A' OR Packages.Package\_type IS NULL

GROUP BY Customers.Customer\_ID\_PK, Customers.First\_name, Customers.Last\_name, Invoices.Total\_Invoice\_Amount, Packages.Package\_type, Packages.Package\_ID\_PK, Shipments.Shipment\_ID\_PK, Packages.Package\_weight\_lb

Output –

A screenshot of a computer

Description automatically generated with medium confidence