

# Data Warehouse Modeling

Hijir Della Wirasti

Google Drive:

[https://drive.google.com/drive/folders/1frZL0E9G8tOs8RVBlg0o7q8hqgwI\\_PWX?usp=sharing](https://drive.google.com/drive/folders/1frZL0E9G8tOs8RVBlg0o7q8hqgwI_PWX?usp=sharing)

3 query sql: [Create Source Table.sql](#) | [Insert Source Table.sql](#) | [star\\_schema.sql](#)

## Question 1

Identify the main entities to be included in the data warehouse. List at least three relevant entities for this project.

For this travel booking system project, the main entities to be included in the data warehouse are those that will provide valuable insights and analytics for the business. Here are three relevant entities:

### 1. Customer:

Attributes:

CustomerID

FirstName

LastName

Email

Phone

Address

#### Reasons for Inclusion:

1. Analyzing customer demographics and behavior.
2. Tracking customer purchases and preferences.
3. Personalizing marketing campaigns and offers.

### 2. TravelOrders:

Attributes:

OrderID

CustomerID

OrderDate

OrderStatus

TotalAmount

#### Reasons for Inclusion:

1. Monitoring sales performance and trends over time.
2. Evaluating the effectiveness of promotional campaigns.
3. Identifying peak booking periods and order patterns.

### 3. Package:

Attributes:

PackageID  
 PackageName  
 Destination  
 DurationDays  
 Price

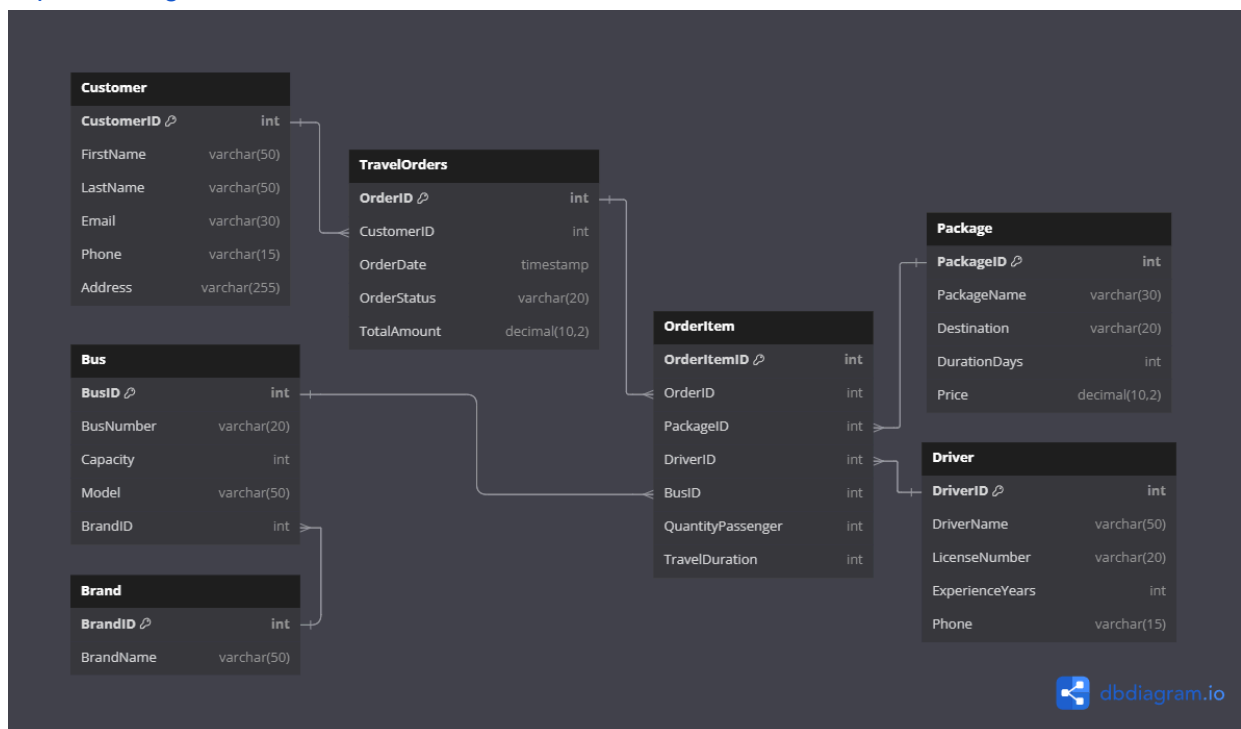
**Reasons for Inclusion:**

1. Analyzing the popularity of different packages and destinations.
2. Assessing package performance and profitability.
3. Optimizing package offerings based on customer demand and feedback.

## Question 2

Create an ERD for the source tables.

<https://dbdiagram.io/d/source-table-669a661f8b4bb5230ecfdf46>



## Question 3

Are there slowly changing dimensions in this project? Identify them.

Yes, there are likely Slowly Changing Dimensions (SCDs) in this project. Slowly Changing Dimensions are dimensions that change over time, but not frequently. In a data warehouse, it's important to track these changes to maintain historical accuracy and for better analysis. Here are the potential SCDs in this project:

**1. Customer:**

Attributes:

FirstName  
LastName  
Email  
Phone  
Address

**Reason:**

Customer information such as address, phone number, and email can change over time. Tracking these changes helps in maintaining the history of customer details for accurate reporting and analysis.

**2. Package:**

Attributes:  
PackageName  
Destination  
DurationDays  
Price

**Reason:**

Package details such as name, destination, duration, and price may change over time due to updates or new offerings. Keeping a record of these changes is important for historical pricing analysis and tracking the evolution of travel packages.

**3. Driver:**

Attributes:  
DriverName  
LicenseNumber  
ExperienceYears  
Phone

**Reason:**

Driver details such as phone number and years of experience can change. It is useful to track these changes, especially the experience years, which will increment annually.

## Question 4

Identify at least one fact table that will exist in your data warehouse.

Based on the Star Schema design for the bus travel agent business dataset, the primary fact table that will exist in the data warehouse is the **Travel Order Fact** Table. This table captures the detailed transactional data related to travel orders and associated items.

### **Travel Order Fact Table**

This fact table contains metrics and measures related to travel orders and the services provided by the travel agent. It combines the travel orders and order items into a single table for a streamlined analysis.

#### **Attributes of the Travel Order Fact Table:**

OrderItemID: Unique identifier for each order item (Primary Key).

OrderID: Identifier for the travel order.  
CustomerID: Identifier for the customer who placed the order.  
OrderDate: Date when the order was placed.  
OrderStatus: Status of the order (e.g., completed, pending, canceled).  
TotalAmount: Total amount for the order.  
PackageID: Identifier for the travel package booked.  
DriverID: Identifier for the driver assigned to the travel order.  
BusID: Identifier for the bus used in the travel order.  
QuantityPassenger: Number of passengers for the travel order.  
TravelDuration: Duration of the travel in days.

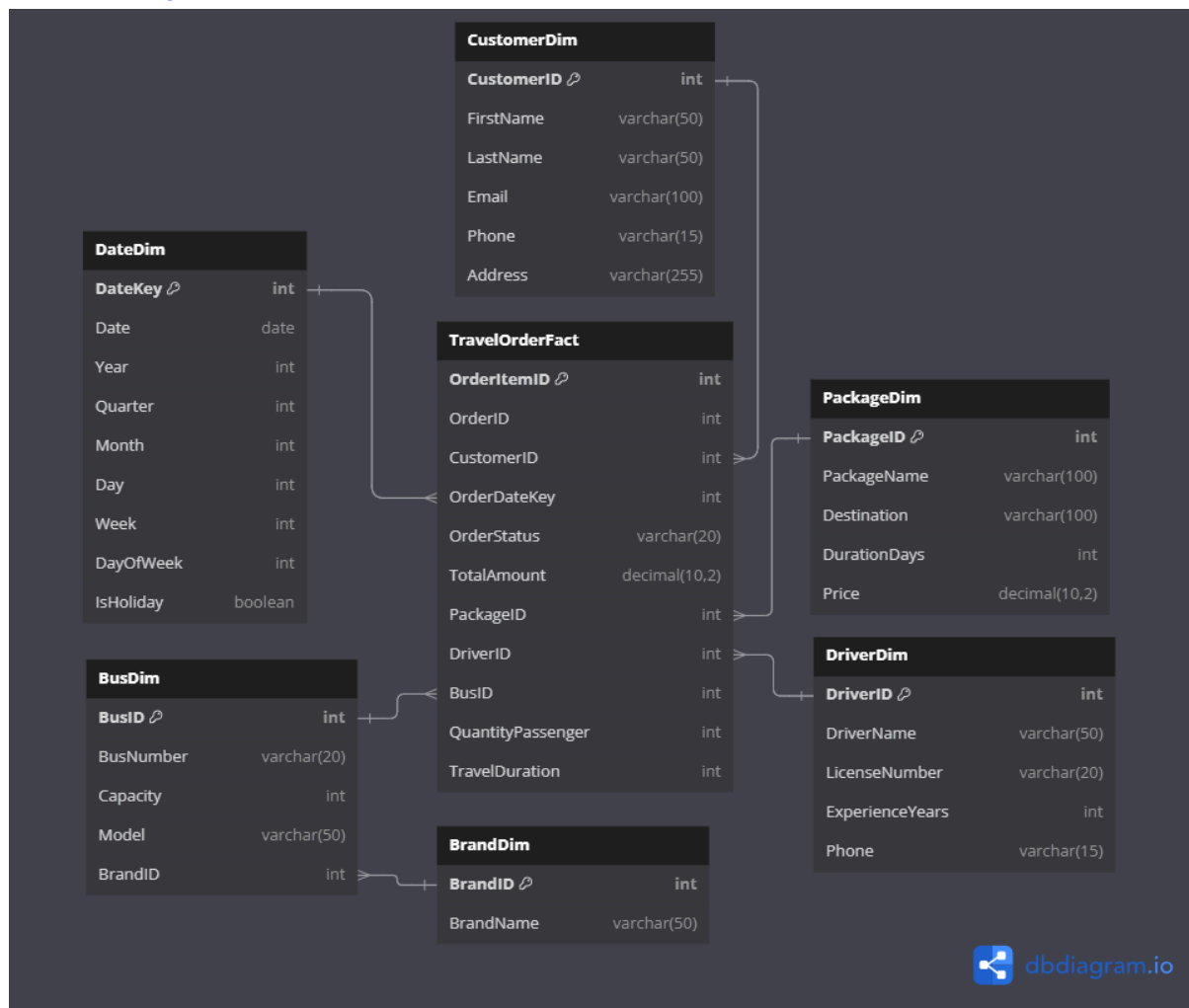
**Reasons for Inclusion:**

1. Order Analysis:  
Track the total number of orders and analyze order trends over time.
2. Revenue Analysis:  
Monitor total revenue and revenue trends by different dimensions such as time, package, and customer.
3. Customer Behavior:  
Analyze customer purchasing patterns and preferences.
4. Package Performance:  
Assess the popularity and performance of different travel packages.
5. Operational Efficiency:  
Evaluate driver and bus utilization to optimize resource allocation.

## Question 5

Illustrate a Star Schema diagram for your data model.

[https://dbdiagram.io/d/star\\_schema-669a62ab8b4bb5230ecf8b15](https://dbdiagram.io/d/star_schema-669a62ab8b4bb5230ecf8b15)



## Question 6

Create the Star Schema on postgresql based on dummy data tables.

Query:

```
CREATE SCHEMA star_schema;

-- Create Dimension Tables in the star_schema

-- Create BrandDim table first
CREATE TABLE star_schema.BrandDim (
    BrandID SERIAL PRIMARY KEY,
    BrandName VARCHAR(50)
);
```

```

-- Create CustomerDim table
CREATE TABLE star_schema.CustomerDim (
    CustomerID SERIAL PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Email VARCHAR(100),
    Phone VARCHAR(15),
    Address VARCHAR(255)
);

-- Create PackageDim table
CREATE TABLE star_schema.PackageDim (
    PackageID SERIAL PRIMARY KEY,
    PackageName VARCHAR(100),
    Destination VARCHAR(100),
    DurationDays INT,
    Price DECIMAL(10, 2)
);

-- Create DriverDim table
CREATE TABLE star_schema.DriverDim (
    DriverID SERIAL PRIMARY KEY,
    DriverName VARCHAR(50),
    LicenseNumber VARCHAR(20),
    ExperienceYears INT,
    Phone VARCHAR(15)
);

-- Create BusDim table, with reference to BrandDim
CREATE TABLE star_schema.BusDim (
    BusID SERIAL PRIMARY KEY,
    BusNumber VARCHAR(20),
    Capacity INT,
    Model VARCHAR(50),
    BrandID INT,
    FOREIGN KEY (BrandID) REFERENCES star_schema.BrandDim(BrandID)
);

-- Create DateDim table
CREATE TABLE star_schema.DateDim (
    DateKey INT PRIMARY KEY,
    Date DATE,

```

```

        Year INT,
        Quarter INT,
        Month INT,
        Day INT,
        Week INT,
        DayOfWeek INT,
        IsHoliday BOOLEAN
    );

-- Create TravelOrderFact table with references to dimension tables
CREATE TABLE star_schema.TravelOrderFact (
    OrderItemID SERIAL PRIMARY KEY,
    OrderID INT,
    CustomerID INT,
    OrderDateKey INT,
    OrderStatus VARCHAR(20),
    TotalAmount DECIMAL(10, 2),
    PackageID INT,
    DriverID INT,
    BusID INT,
    QuantityPassenger INT,
    TravelDuration INT,
    FOREIGN KEY (CustomerID) REFERENCES
star_schema.CustomerDim(CustomerID),
    FOREIGN KEY (PackageID) REFERENCES
star_schema.PackageDim(PackageID),
    FOREIGN KEY (DriverID) REFERENCES star_schema.DriverDim(DriverID),
    FOREIGN KEY (BusID) REFERENCES star_schema.BusDim(BusID),
    FOREIGN KEY (OrderDateKey) REFERENCES star_schema.DateDim(DateKey)
);

-- Populate the Date Dimension
INSERT INTO star_schema.DateDim (DateKey, Date, Year, Quarter, Month,
Day, Week, DayOfWeek, IsHoliday)
SELECT
    to_char(d, 'YYYYMMDD')::int AS DateKey,
    d::date AS Date,
    extract(year from d) AS Year,
    extract(quarter from d) AS Quarter,
    extract(month from d) AS Month,
    extract(day from d) AS Day,
    extract(week from d) AS Week,
    extract(isodow from d) AS DayOfWeek,

```

```

        CASE WHEN d = date '2024-01-01' THEN true ELSE false END AS
IsHoliday
FROM
    generate_series('2023-01-01'::date, '2024-12-31'::date, '1
day'::interval) d;

-- Populate the Dimension Tables

-- Populate BrandDim
INSERT INTO star_schema.BrandDim (BrandID, BrandName)
SELECT BrandID, BrandName FROM public.Brand;

-- Populate BusDim
INSERT INTO star_schema.BusDim (BusID, BusNumber, Capacity, Model,
BrandID)
SELECT BusID, BusNumber, Capacity, Model, BrandID FROM public.Bus;

-- Populate CustomerDim
INSERT INTO star_schema.CustomerDim (CustomerID, FirstName, LastName,
Email, Phone, Address)
SELECT CustomerID, FirstName, LastName, Email, Phone, Address FROM
public.Customer;

-- Populate PackageDim
INSERT INTO star_schema.PackageDim (PackageID, PackageName,
Destination, DurationDays, Price)
SELECT PackageID, PackageName, Destination, DurationDays, Price FROM
public.Package;

-- Populate DriverDim
INSERT INTO star_schema.DriverDim (DriverID, DriverName, LicenseNumber,
ExperienceYears, Phone)
SELECT DriverID, DriverName, LicenseNumber, ExperienceYears, Phone FROM
public.Driver;

-- Populate the Fact Table
INSERT INTO star_schema.TravelOrderFact (
    OrderID,
    CustomerID,
    OrderDateKey,
    OrderStatus,
    TotalAmount,
    PackageID,

```



```

        DriverID,
        BusID,
        QuantityPassenger,
        TravelDuration
    )
SELECT
    oi.OrderID,
    o.CustomerID,
    to_char(o.OrderDate, 'YYYYMMDD')::int AS OrderDateKey,
    o.OrderStatus,
    o.TotalAmount,
    oi.PackageID,
    oi.DriverID,
    oi.BusID,
    oi.QuantityPassenger,
    oi.TravelDuration
FROM
    public.OrderItem oi
JOIN
    public.TravelOrders o ON oi.OrderID = o.OrderID;

```

## branddim

pgAdmin 4

File Object Tools Help

Object Explorer

- Schemas (2)
  - public
  - star\_schema
    - Aggregates
    - Collations
    - Domains
    - FTS Configurations
    - FTS Dictionaries
    - FTS Parsers
    - FTS Templates
    - Foreign Tables
    - Functions
    - Materialized Views
    - Operators
    - Procedures
    - Sequences
    - Tables (7)
      - branddim**
      - busdim
      - customerdim
      - datedim
      - driverdim
      - packagedim
      - travelorderfact
    - Trigger Functions

Dashboard Properties SQL Statistics Dependencies D

star\_schema.branddim/HW DWH/postgres@PostgreSQL 1

Query Query History

```

1 SELECT * FROM star_schema.branddim
2 ORDER BY brandid ASC

```

Data Output Messages Notifications

	brandid [PK] integer	brandname character varying (50)
1	1	Mercedes-Benz
2	2	Volvo
3	3	Scania
4	4	MAN
5	5	Iveco
6	6	Renault
7	7	DAF
8	8	Setra
9	9	Van Hool
10	10	Neoplan

## busdim

star\_schema.busdim/HW DWH/postgres@PostgreSQL 16

Query Query History

```
1 SELECT * FROM star_schema.busdim
2 ORDER BY busid ASC
```

Data Output Messages Notifications

	busid [PK] integer	busnumber character varying (20)	capacity integer	model character varying (50)	brandid integer
1	1	BUS123	50	Model X	1
2	2	BUS124	60	Model Y	2
3	3	BUS125	55	Model Z	3
4	4	BUS126	45	Model A	4
5	5	BUS127	70	Model B	5
6	6	BUS128	65	Model C	6
7	7	BUS129	75	Model D	7
8	8	BUS130	40	Model E	8
9	9	BUS131	80	Model F	9
10	10	BUS132	90	Model G	10

## customerdim

star\_schema.customerdim/HW DWH/postgres@PostgreSQL 16

Query Query History

```
1 SELECT * FROM star_schema.customerdim
2 ORDER BY customerid ASC
```

Data Output Messages Notifications

	customerid [PK] integer	firstname character varying (50)	lastname character varying (50)	email character varying (100)	phone character varying (15)	address character varying (255)
1	1	John	Doe	john.doe@example.com	123-456-7890	123 Elm St
2	2	Jane	Smith	jane.smith@example.com	987-654-3210	456 Oak St
3	3	Alice	Johnson	alice.johnson@example.com	555-123-4567	789 Pine St
4	4	Bob	Brown	bob.brown@example.com	555-987-6543	101 Maple St
5	5	Charlie	Davis	charlie.davis@example.com	555-567-8901	102 Birch St
6	6	Eva	Wilson	eva.wilson@example.com	555-890-1234	103 Cedar St
7	7	Frank	Garcia	frank.garcia@example.com	555-456-7890	104 Willow St
8	8	Grace	Martinez	grace.martinez@example.com	555-321-6549	105 Ash St
9	9	Hannah	Taylor	hannah.taylor@example.com	555-654-3210	106 Cherry St
10	10	Ian	Anderson	ian.anderson@example.com	555-789-0123	107 Spruce St

**datedim**

star\_schema.datedim/HW DWH/postgres@PostgreSQL 16

No limit

Query Query History

```

1 SELECT * FROM star_schema.datedim
2 ORDER BY datekey ASC

```

Data Output Messages Notifications

	datekey [PK] integer	date date	year integer	quarter integer	month integer	day integer	week integer	dayofweek integer	isholiday boolean
1	20230101	2023-01-01	2023	1	1	1	52	7	false
2	20230102	2023-01-02	2023	1	1	2	1	1	false
3	20230103	2023-01-03	2023	1	1	3	1	2	false
4	20230104	2023-01-04	2023	1	1	4	1	3	false
5	20230105	2023-01-05	2023	1	1	5	1	4	false
6	20230106	2023-01-06	2023	1	1	6	1	5	false
7	20230107	2023-01-07	2023	1	1	7	1	6	false
8	20230108	2023-01-08	2023	1	1	8	1	7	false
9	20230109	2023-01-09	2023	1	1	9	2	1	false
10	20230110	2023-01-10	2023	1	1	10	2	2	false
11	20230111	2023-01-11	2023	1	1	11	2	3	false
12	20230112	2023-01-12	2023	1	1	12	2	4	false
13	20230113	2023-01-13	2023	1	1	13	2	5	false
14	20230114	2023-01-14	2023	1	1	14	2	6	false

Total rows: 731 of 731    Query complete 00:00:00.337

**driverdim**

Schemas (2)

- public
  - star\_schema
    - Aggregates
    - Collations
    - Domains
    - FTS Configurations
    - FTS Dictionaries
    - FTS Parsers
    - FTS Templates
    - Foreign Tables
    - Functions
    - Materialized Views
    - Operators
    - Procedures
    - Sequences
    - Tables (7)
      - branddim
      - busdim
      - customerdim
      - datedim
      - driverdim
      - packagedim
      - travelorderfact

star\_schema.driverdim/HW DWH/postgres@PostgreSQL 16

No limit

Query

Query History

1 **SELECT \*** FROM star\_schema.driverdim  
 2 **ORDER BY** driverid **ASC**

Data Output

Messages

Notifications

	driverid [PK] integer	drivername character varying (50)	license number character varying (20)	experience years integer	phone character varying (15)
1	1	Jack Thompson	DL123456789	10	555-101-2020
2	2	Lily Evans	DL987654321	8	555-202-3030
3	3	Mia Adams	DL555555555	12	555-303-4040
4	4	Noah Robinson	DL444444444	15	555-404-5050
5	5	Olivia Martinez	DL333333333	7	555-505-6060
6	6	Paul Walker	DL222222222	9	555-606-7070
7	7	Quinn White	DL111111111	11	555-707-8080
8	8	Rachel Green	DL666666666	14	555-808-9090
9	9	Sam Harris	DL777777777	6	555-909-1010
10	10	Tina Brooks	DL888888888	13	555-101-2021

packagedim

Schemas (2)

public

star\_schema

Aggregates

Collations

Domains

FTS Configurations

FTS Dictionaries

FTS Parsers

FTS Templates

Foreign Tables

Functions

Materialized Views

Operators

Procedures

1.3 Sequences

Tables (7)

branddim

busdim

customerdim

datedim

driverdim

packagedim

travelorderfact

star\_schema.packagedim/HW DWH/postgres@PostgreSQL 16

No limit

Query

Query History

1 SELECT \* FROM star\_schema.packagedim

2 ORDER BY packageid ASC

Data Output

Messages

Notifications

packageid [PK] integer

packagename character varying (100)

destination character varying (100)

durationdays integer

price numeric (10,2)

1	1	Holiday Package A	Paris	7	999.99
2	2	Holiday Package B	London	5	799.99
3	3	Holiday Package C	Rome	6	899.99
4	4	Holiday Package D	Berlin	4	699.99
5	5	Holiday Package E	Madrid	8	1099.99
6	6	Holiday Package F	Lisbon	5	749.99
7	7	Holiday Package G	Prague	7	949.99
8	8	Holiday Package H	Vienna	6	849.99
9	9	Holiday Package I	Budapest	4	649.99
10	10	Holiday Package J	Amsterdam	5	799.99

travelorderfact

star\_schema.travelorderfact/HW DWH/postgres@PostgreSQL 16

No limit

Query

Query History

1 SELECT \* FROM star\_schema.travelorderfact

2 ORDER BY orderitemid ASC

Data Output

Messages

Notifications

orderitemid [PK] integer

orderid integer

customerid integer

orderdatekey integer

orderstatus character varying (20)

totalamount numeric (10,2)

packageid integer

driverid integer

busid integer

quantitypassenger integer

travelduration integer

1	1	22	9	20230814	pending	357.91	9	5	9	7	8
2	2	94	8	20240519	confirmed	63.71	7	9	6	10	4
3	3	66	7	20240629	confirmed	768.80	4	10	1	2	7
4	4	47	7	20231003	pending	721.02	4	7	1	6	14
5	5	66	7	20240629	confirmed	768.80	9	4	9	9	19
6	6	34	1	20240617	pending	714.43	4	6	8	7	24
7	7	18	3	20231130	pending	270.31	6	5	1	5	23
8	8	78	8	20240326	pending	973.24	1	5	2	10	1
9	9	94	8	20240519	confirmed	63.71	10	1	4	6	2
10	10	61	4	20230801	pending	7.91	7	8	5	7	14
11	11	24	5	20240320	confirmed	300.73	1	7	1	1	13
12	12	29	9	20230809	confirmed	481.24	6	10	6	4	6
13	13	7	7	20240612	confirmed	287.20	10	8	4	4	19
14	14	90	10	20231108	pending	164.67	4	3	3	7	10

Total rows: 200 of 200    Query complete 00:00:00.476    Ln 1,