

|  |
| --- |
| **NYC taxi trip** |
|  |



Contents

[1 Business Description 3](#_Toc155614186)

[1.1 Business background 3](#_Toc155614187)

[1.2 Problems because of poor data management 3](#_Toc155614188)

[1.3 Benefits from implementing a Data Warehouse 3](#_Toc155614189)

[1.4 DATASETS DESCRIPTION 3](#_Toc155614190)

[1.5 GRAIN / DIM / FACT 4](#_Toc155614191)

[2 Business Layer 3NF 4](#_Toc155614192)

[2.1 Definitions & Acronyms 3](#_Toc62212635)

[2.2 Logical Scheme 3](#_Toc62212636)

[2.3 Objects 3](#_Toc62212637)

[3 Business Layer Dimensional Model 4](#_Toc155614193)

[4 Logical Scheme 4](#_Toc155614194)

[5 Data Flow 4](#_Toc155614195)

[6 Fact Table Partitioning Strategy 4](#_Toc155614196)

# 

# Business Description

## Business background

Taxis are a vital part of the transportation infrastructure in any large city, including New York. In this industry, there are many operators offering taxi services for various types of trips, from short city rides to long trips to airports. Competition in this sector is high, so successful companies must effectively manage their data to improve service and meet customer needs.

## Problems because of poor data management

Poor data management hampers effective business operations due to insufficient information for decision-making. Without using data analysis tools, companies cannot develop a strategy focused on customer needs and operational optimization. This can lead to decreased market competitiveness and lower quality of services provided.

## Benefits from implementing a Data Warehouse

Implementing a data warehouse can help solve the described problems and answer the following questions:

* What times of day are most in demand for taxi rides?
* Which areas of the city (by longitude and latitude) are most frequently served?
* How does demand for taxis change depending on the day of the week or month?
* Which types of trips (business, private) are more popular?

## DATASETS DESCRIPTION

In the dataset, Yellow Taxi is only associated with the Street method, so it will not have a booking\_type column but can be inferred by the method of booking. Booking type hierarchical structure can be useful for understanding the relationship between taxi types and booking methods.

Booking Type:

This represents the type of taxi service.

* Values: Green Taxi (Yellow Taxi is only available through the Street method).
* Description: The category of taxi associated with the booking. In this case, Yellow Taxi is always associated with street bookings, while Green Taxi can have multiple booking methods.

Booking Method (Subcategory):

This indicates the method or channel used to book the taxi.

* Values: Street, Phone.
* Description: The specific booking method used for each taxi type. For instance, booking via the street or through a phone call.

Example Hierarchy:

Booking Type = Yellow Taxi

* Street — Booking made through the street (Yellow Taxi can only be booked this way).

Booking Type = Green Taxi

* Street — Booking made through the street.
* Phone — Booking made through a phone call.

Datasets:

**The First Dataset: Yellow Taxi Data** **(SRC\_GREEN\_TAXI\_DATA.csv)**

Information about Yellow Taxi trips:

1. **Vendor Information**:

vendor\_name: Vendor name

vendor\_street: Vendor street, part of the vendor’s address

vendor\_house: Vendor house, part of the vendor’s address

vendor\_city: Vendor city, part of the vendor’s address

vendor\_house: Vendor house, part of the vendor’s address

vendor\_postal\_code: Vendor postal code, part of the vendor’s address

vendor\_telephone: Vendor telephone

1. **Payment Information**:

payment\_type: card or cash

payment\_time: payment time

1. **Rate Information**:

base\_fare: base fare

rate\_per\_mile: rate per mile

1. **Trip Information**:

id: Trip identifier

pickup\_longitude: pickup longitude

pickup\_latitude: pickup latitude

dropoff\_longitude: dropoff longitude

dropoff\_latitude: dropoff latitude

pickup\_datetime: Pickup date and time

dropoff\_datetime: Dropoff date and time

1. **Trip details Information**:

distance\_miles: distance\_miles

trip\_duration: trip duration

**The Second Dataset: Green Taxi Data (SRC\_YELLOW\_TAXI\_DATA.csv)**

Information about Green Taxi trips:

1. **Vendor Information**:

vendor: Vendor name

vendor\_street: Vendor street, part of the vendor’s address

vendor\_house: Vendor house, part of the vendor’s address

vendor\_city: Vendor city, part of the vendor’s address

vendor\_house: Vendor house, part of the vendor’s address

vendor\_postal\_code: Vendor postal code, part of the vendor’s address

vendor\_telephone: Vendor telephone

1. **Customer Information**:

customer\_type: individual, or business

passenger\_count: how many people on board

customer telephone: telephone number used to place the ordrer

1. **Booking**:

booking\_type: phone, or street

booking\_time: booking time

1. **Payment Information**:

payment\_type: card or cash

payment\_time: payment time

1. **Rate Information**:

base\_fare: base fare

rate\_per\_mile: rate per mile

1. **Promotion Information**:

promo code: promo code

discount\_percentage: 10%, 20%, none

1. **Trip Information**:

id: Trip identifier

vendor\_id: Vendor identifier

vendor\_name: Vendor name

pickup\_datetime: Pickup date and time

dropoff\_datetime: Dropoff date and time

passenger\_count: Number of passengers

trip\_duration: Duration of the trip

1. **Trip details Information**:

distance\_miles: distance\_miles

trip\_duration: trip duration

## **GRAIN / DIM / FACT**

#### Grain

Each record represents one taxi trip. This will be the finest level of detail. We will be analyzing this in this assignment.

#### Dimensions and Facts

The granularity of the datasets is defined at the level of individual trips. Each row in the fact table represents one taxi trip: vendor\_id, customer\_id, booking\_id, payment\_id, transmission\_id, rate\_id, promo\_id, trip\_id

* **DIM\_VENDORS**

The taxi company transports people from point A to point B

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| vendor\_id | unique identifier for each row (PK) | bigint |
| vendor\_name | vendor name | varchar |
| vendor\_street | part of full address | varchar |
| vendor\_house | part of full address | varchar |
| vendor\_city | part of full address | varchar |
| vendor\_country | part of full address | varchar |
| vendor\_postal code | address postal code | varchar |
| vendor\_telephone | telephone number | varchar |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| vendor\_id | vendor\_name | street | house | city | country | postal code | vendor\_telephone |
| 3 | Green Taxi | Beaver Street | 33 | New York | USA | NY 10004 | +1 212-639-9675 |

Example with filled data

* **DIM\_CUSTOMERS**

A Customer is a client who can be either an individual or a business.

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| customer\_id | unique identifier for each row (PK) | bigint |
| customer\_type | Business/Individual | varchar |
| customer\_telephone | telephone number | varchar |

|  |  |  |  |
| --- | --- | --- | --- |
| customer\_id | customer\_type | passenger\_count | customer\_telephone |
| 101 | Business | 3 | +1 212-639-9675 |
| 1 | Individual | 1 | NULL |

Example with filled data

* **DIM\_BOOKINGS**

Green Taxi can be ordered from the street and via phone. Yellow Taxi can only be ordered from the street, therefore, the call is only made on the street.

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| booking\_id | unique identifier for each row (PK) | bigint |
| booking\_type | Phone/Street | varchar |

|  |  |  |
| --- | --- | --- |
| booking\_id | booking\_type | booking\_time |
| 52 | Phone | 3/26/2016 13:18 |

Example with filled data

* **DIM\_PAYMENTS**

For ride can by paid card or cash

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| payment\_id | unique identifier for each row (PK) | bigint |
| payment\_type | Card/Cash | varchar |

|  |  |  |
| --- | --- | --- |
| payment\_id | payment\_type | payment\_time |
| 1015 | Card | 6/12/2016 0:57 |
| 1158 | Cash | NULL |

Example with filled data

* **DIM\_RATES**

Taxi rates base fare plus rate per mile.

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| rate\_id | unique identifier for each row (PK) | bigint |
| base\_fare | base fare | decimal(10,2) |
| rate\_per\_mile | rate per mile | decimal(10,2) |

|  |  |  |
| --- | --- | --- |
| rate\_id | base\_fare | rate\_per\_mile |
| 888 | 2.50 | 1.50 |

Example with filled data

* **DIM\_PROMOTIONS**

Taxi promo codes. With discount: 10%, 20%.

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| promo\_id | unique identifier for each row (PK) | int |
| promo\_code | promo code with discount | varchar |
| discount\_percentage | discount percentage | decimal(5,2) |

|  |  |  |
| --- | --- | --- |
| promo\_id | promo\_code | discount\_percentage |
| 666 | Promo10 | 10 |

Example with filled data

* **DIM\_TRIP\_LOCATIONS**

Information about taxi trip locations point A and point B

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| location\_id | unique identifier for each row ((PK)) | varchar |
| pickup\_longitude | pickup longitude | decimal (10, 6) |
| pickup\_latitude | pickup latitude | decimal (10, 6) |
| dropoff\_longitude | dropoff longitude | decimal (10, 6) |
| dropoff\_latitude | dropoff latitude | decimal (10, 6) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| trip\_id | pickup\_longitude | pickup\_latitude | dropoff\_longitude | dropoff\_latitude | pickup\_datetime | dropoff\_datetime |
| id18133569 | -73.98041534 | 40.73856354 | -73.9994812 | 40.73115158 | 6/12/2016 0:43 | 6/12/2016 0:54 |

Example with filled data

* **FCT\_TAXI\_TRIPS**

Fact table with references on all dimensional tables. Plus measurements: distance, duration, amount

|  |  |  |
| --- | --- | --- |
| Column name | Description | Data Type |
| vendor\_id | vendor id ((FK)) | bigint |
| customer\_id | customer id ((FK)) | bigint |
| booking\_id | booking id ((FK)) | bigint |
| payment\_id | payment id ((FK)) | bigint |
| transmision\_id | transmission id ((FK)) | bigint |
| rate\_id | rate id ((FK)) | bigint |
| promo\_id | promotion id ((FK)) | bigint |
| location\_id | location id pickup and dropoff ((FK)) | bigint |
| passenger\_count | count of passengers | int |
| booking\_time | time of booking | datetime |
| pickup\_time | time of pickup | datetime |
| dropoff\_time | time of dropoff | datetime |
| payment\_time | payment time | datetime |
| distance\_miles | measure; trip distance | decimal(10, 2) |
| amount | measure; trip amount | decimal(10,2) |

Example with filled data

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| vendor\_id | customer\_id | booking\_id | payment\_id | transmision\_id | rate\_id | promo\_id | trip\_id | passenger\_count | booking\_time | pickup\_time | dropoff\_datetime | payment\_time | distance\_miles | amount |
| 1 | 101 | 5001 | 2001 | 3001 | 5 | 7001 | 9001 | 2 | 6/12/2016 0:38 | 6/12/2016 0:48 | 6/12/2016 0:57 | 6/12/2016 0:58 | 12.5 | 25.75 |

# Business Layer 3NF

## Definitions & Acronyms

SQL: Structured Query Language, used for managing and querying data in the database.

• (PK) (Primary Key): A unique identifier for a record in a table.

• (FK) (Foreign Key): A field in one table that uniquely identifies a row of another table, creating a relationship between the two tables.

• SQL (Structured Query Language): The standard language used to communicate with and manipulate databases.

• PL/pgSQL: A procedural language supported by PostgreSQL that allows for more complex operations and control structures in SQL scripts.

m:m

: Many-to-many relationship in the database, where multiple records from one table can be associated with multiple records from another table.

1:m (m:1)

: One-to-many (many-to-one) relationship in the database, where one record from a table is associated with many records from another table.

1:1

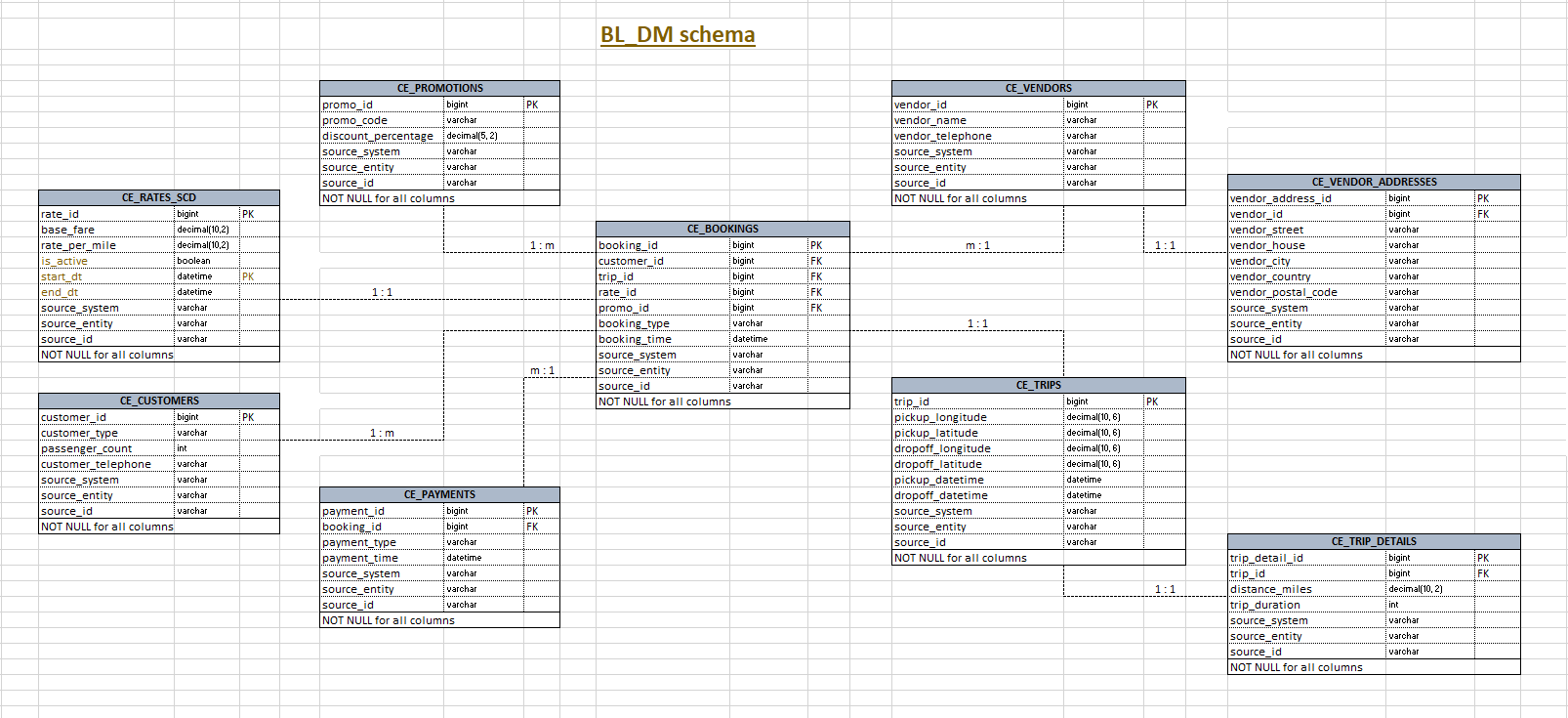
: One-to-one relationship in the database, where one record from a table is associated with one record from another table.

(PK): Primary Key, a unique identifier for each record in a database table.

(FK): Foreign Key, a reference to a Primary Key in another table to establish a relationship between two tables.

NOT NULL: attribute in a table must have a value.

## Logical Scheme



## Objects

1. **CE\_VENDORS table description**

This table stores information about taxi companies. It also includes metadata for tracking the source of the data and the entity within the source system.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| CE\_VENDORS | vendor\_id | unique identifier for each vendor (PK) | bigint |
| vendor\_name | vendor name | varchar |
| vendor\_telephone | vendor telephone number | varchar |
| source\_system | system where the data originates from (datasets) | varchar |
| source\_entity | entity within the source system | varchar |
| source\_id | unique identifier within the source system | varchar |

Comments on table relationships:

* CE\_VENDORS ↔ CE\_BOOKINGS: one-to-many (1:m) relationship;
* CE\_VENDORS ↔ CE\_VENDOR\_ADDRESSES: one-to-one (1:1) relationship.

Additional constraints: NOT NULL for all columns

1. **CE\_VENDOR\_ADDRESSES table description**

This table stores information about taxi companies. It also includes metadata for tracking the source of the data and the entity within the source system.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| CE\_VENDOR\_ADDRESSES | vendor\_address\_id | unique identifier for each vendor address (PK) | bigint |
| vendor\_id | foreign key referencing to the ce\_vendors (FK) | bigint |
| vendor\_street | vendor street | varchar |
| vendor\_house | vendor telephone house | varchar |
| vendor\_city | vendor city | varchar |
| vendor\_country | vendor country | varchar |
| vendor\_postal\_code | vendor postal code | varchar |
| source\_system | system where the data originates from (datasets) | varchar |
| source\_entity | entity within the source system | varchar |
| source\_id | unique identifier within the source system | varchar |

Comments on table relationships:

* CE\_VENDORS ↔ CE\_VENDOR\_ADDRESSES: one-to-one (1:1) relationship.

Additional constraints: NOT NULL for all columns

1. CE\_CUSTOMERS table description

The CE\_CUSTOMERS table stores information about customers, including a unique identifier, customer type, number of passengers, and contact details. It also includes metadata for tracking the source of the data and the entity within the source system.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| CE\_CUSTOMERS | customer\_id | unique identifier for each customer (PK) | bigint |
| customer\_type | Individual/Business | varchar |
| passenger\_count | passenger count | int |
| customer\_telephone | customer telephone | varchar |
| source\_system | system where the data originates from (datasets) | varchar |
| source\_entity | entity within the source system | varchar |
| source\_id | unique identifier within the source system | varchar |

Comments on table relationships:

* CE\_CUSTOMERS ↔ CE\_BOOKINGS: one-to-many (1:m) relationship.

Additional constraints: NOT NULL for all columns

1. CE\_BOOKINGS table description

The CE\_BOOKINGS table contains information about customer bookings, including a unique identifier, associated customer, trip, rate and promo IDs, booking type, and the time of booking. It also tracks the source of the data and the entity within the source system.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| CE\_BOOKINGS | booking\_id | unique identifier for each booking (PK) | bigint |
| customer\_id | unique identifier for each customer (FK) | bigint |
| trip\_id | unique identifier for each trip (FK) | bigint |
| rate\_id | unique identifier for each rate (FK) | bigint |
| promo\_id | unique identifier for each promotion (FK) | bigint |
| booking\_type | booking type | varchar |
| booking\_time | booking time | datetime |
| source\_system | system where the data originates from (datasets) | varchar |
| source\_entity | entity within the source system | varchar |
| source\_id | unique identifier within the source system | varchar |

Comments on table relationships:

* CE\_CUSTOMERS ↔ CE\_BOOKINGS: one-to-many (1:m) relationship;
* CE\_BOOKINGS ↔ CE\_RATES\_SCD: one-to-one (1:1) relationship;
* CE\_BOOKINGS ↔ CE\_PROMOTIONS: many-to-one (m:1) relationship;
* CE\_BOOKINGS ↔ CE\_VENDORS: many-to-one (m:1) relationship;
* CE\_BOOKINGS ↔ CE\_TRIPS: one-to-one (1:1) relationship;
* CE\_BOOKINGS ↔ CE\_PAYMENTS: many-to-one (m:1) relationship.

Additional constraints: NOT NULL for all columns

1. CE\_PAYMENTS table description

The **CE\_PAYMENTS** table stores information about payments made for bookings, including a unique payment identifier, the booking ID it relates to, payment type, and the time of payment. It also includes metadata for tracking the source of the data and the entity within the source system.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| CE\_PAYMENTS | payment\_id | unique identifier for each payment (PK) | bigint |
| booking\_id | unique identifier for each booking (FK) | bigint |
| payment\_type | payment type | varchar |
| payment\_time | payment time | datetime |
| source\_system | system where the data originates from (datasets) | varchar |
| source\_entity | entity within the source system | varchar |
| source\_id | unique identifier within the source system | varchar |

Comments on table relationships:

* CE\_CUSTOMERS ↔ CE\_PAYMENTS: one-to-many (1:m) relationship;
* CE\_BOOKINGS ↔ CE\_PAYMENTS: many-to-one (m:1) relationship.

Additional constraints: NOT NULL for all columns

1. CE\_RATES\_SCD table description

This table stores historical rate information for transportation services. It is structured to track changes over time using Slowly Changing Dimensions (SCD) Type 2, capturing the historical data of each rate, including start and end dates, status of the rate (active or inactive), and additional metadata to trace when and why a rate change occurred.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| CE\_RATES\_SCD | rate\_id | unique identifier for each rate (PK) | bigint |
| base\_fare | base fare for the transportation service | decimal(10,2) |
| rate\_per\_mile | Rate per mile for the transportation service | decimal(10,2) |
| is\_active | Indicates if the rate is currently active | boolean |
| start\_dt | Start date of the rate's validity (SCD Type 2 field) | datetime |
| end\_dt | End date of the rate's validity (SCD Type 2 field) | datetime |
| source\_system | system where the data originates from (datasets) | varchar |
| source\_entity | entity within the source system | varchar |
| source\_id | unique identifier within the source system | varchar |

Comments on table relationships:

• CE\_BOOKINGS ↔ CE\_RATES\_SCD: one-to-one (1:1) relationship.

Additional constraints: NOT NULL for all columns

1. CE\_PROMOTIONS table description

This table stores promotional information related to transportation services. It tracks active and past promotions, including the promotional code, the discount offered, and metadata about the source system from which the data originates.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| CE\_PROMOTIONS | promo\_id | unique identifier for each promotion (PK) | bigint |
| promo\_code | promotion code | varchar |
| discount\_percentage | discount percentage | decimal(5,2) |
| source\_system | system where the data originates from (datasets) | varchar |
| source\_entity | entity within the source system | varchar |
| source\_id | unique identifier within the source system | varchar |

Comments on table relationships:

• CE\_BOOKINGS ↔ CE\_PROMOTIONS: many-to-one (m:1) relationship.

Additional constraints: NOT NULL for all columns

1. CE\_TRIPS table description

This table stores information about individual trips, including geographic data related to pickup and drop-off locations, as well as timestamps for the trip's start and end. It serves as a core component for tracking the details of transportation journeys. It also includes metadata for tracking the source of the data and the entity within the source system.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| CE\_TRIPS | trip\_id | unique identifier for each trip (PK) | bigint |
| pickup\_longitude | Longitude of the pickup location | decimal(10,6) |
| pickup\_latitude | Latitude of the pickup location | decimal(10,6) |
| dropoff\_longitude | Longitude of the drop-off location | decimal(10,6) |
| dropoff\_latitude | Latitude of the drop-off location | decimal(10,6) |
| pickup\_datetime | Date and time when the trip was picked up | datetime |
| dropoff\_datetime | Date and time when the trip was completed (dropped off) | datetime |
| source\_system | system where the data originates from (datasets) | varchar |
| source\_entity | entity within the source system | varchar |
| source\_id | unique identifier within the source system | varchar |

Comments on table relationships:

* CE\_TRIPS ↔ CE\_TRIP\_DETAILS: one-to-one (1:1) relationship;
* CE\_BOOKINGS ↔ CE\_TRIPS: one-to-one (1:1) relationship.

Additional constraints: NOT NULL for all columns

1. CE\_TRIP\_DETAILS table description

This table stores detailed information about each trip, including metrics such as distance traveled and trip duration. It provides additional context to the CE\_TRIPS table by linking a trip with its operational details, such as the distance in miles and the total duration. It also includes metadata for tracking the source of the data and the entity within the source system.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| CE\_TRIP\_DETAILS | trip\_detail\_id | unique identifier for each trip details (PK) | bigint |
| trip\_id | Unique identifier for each trip (FK) | bigint |
| distance\_miles | Total distance of the trip in miles | decimal(10,2) |
| trip\_duration | Duration of the trip in minutes | int |
| source\_system | system where the data originates from (datasets) | varchar |
| source\_entity | entity within the source system | varchar |
| source\_id | unique identifier within the source system | varchar |

Comments on table relationships:

* CE\_TRIPS ↔ CE\_TRIPS\_DETAILS: one-to-one (1:1) relationship.

Additional constraints: NOT NULL for all columns

# Business Layer Dimensional Model

# Logical Scheme

# Data Flow

# Fact Table Partitioning Strategy