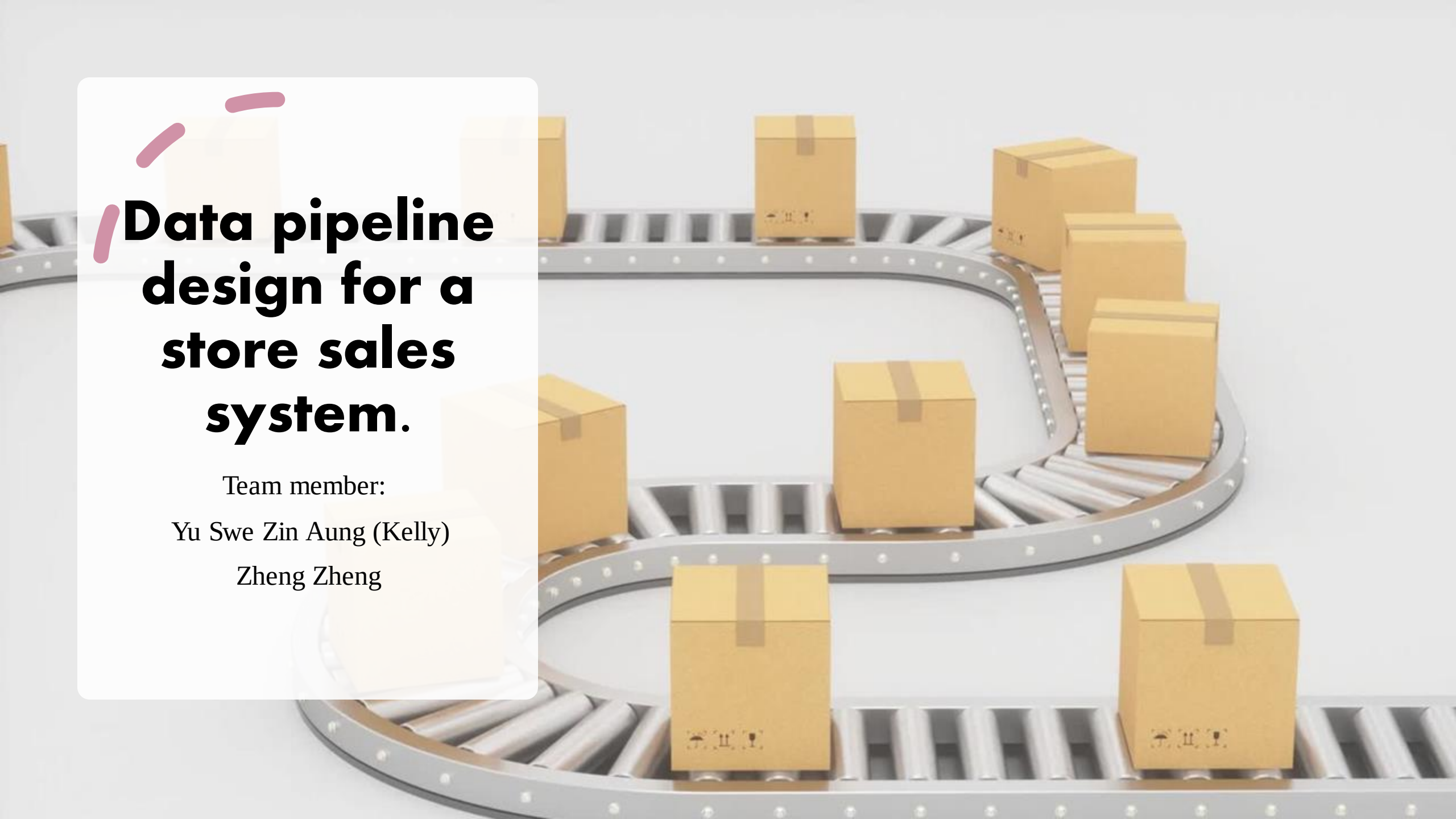
Data pipeline design for a store sales system.

Team member:

Yu Swe Zin Aung (Kelly)

Zheng Zheng



Structure

Introduction

Background information

Business problem

Requirements

Data Pipeline Design

Pipeline Structure

Data Implementation

Normalization

Data Visualization

Connect Tableau with MySQL

Tableau Dashboad

Business background

- Favorita is the largest retail chain in Ecuador with a nationwide network of supermarkets and hypermarkets.
- The stores offer a wide range of products across thousands of product families, including groceries, fresh produce, household items, electronics, and clothing.
- Favorita sources products from both local and international suppliers, ensuring a wide variety of choices for customers.
- The company's business strategy is focused on providing high-quality products at affordable prices to customers across Ecuador.

Business problems

Retailers need to manage inventory levels to avoid waste and stockouts.

Favorita tracks popular products and effective marketing to allocate resources.

Adjustments are made for seasonal and holiday demand.

Accurate sales trend analysis helps Favorita optimize operations and improve customer experience.

Requirements

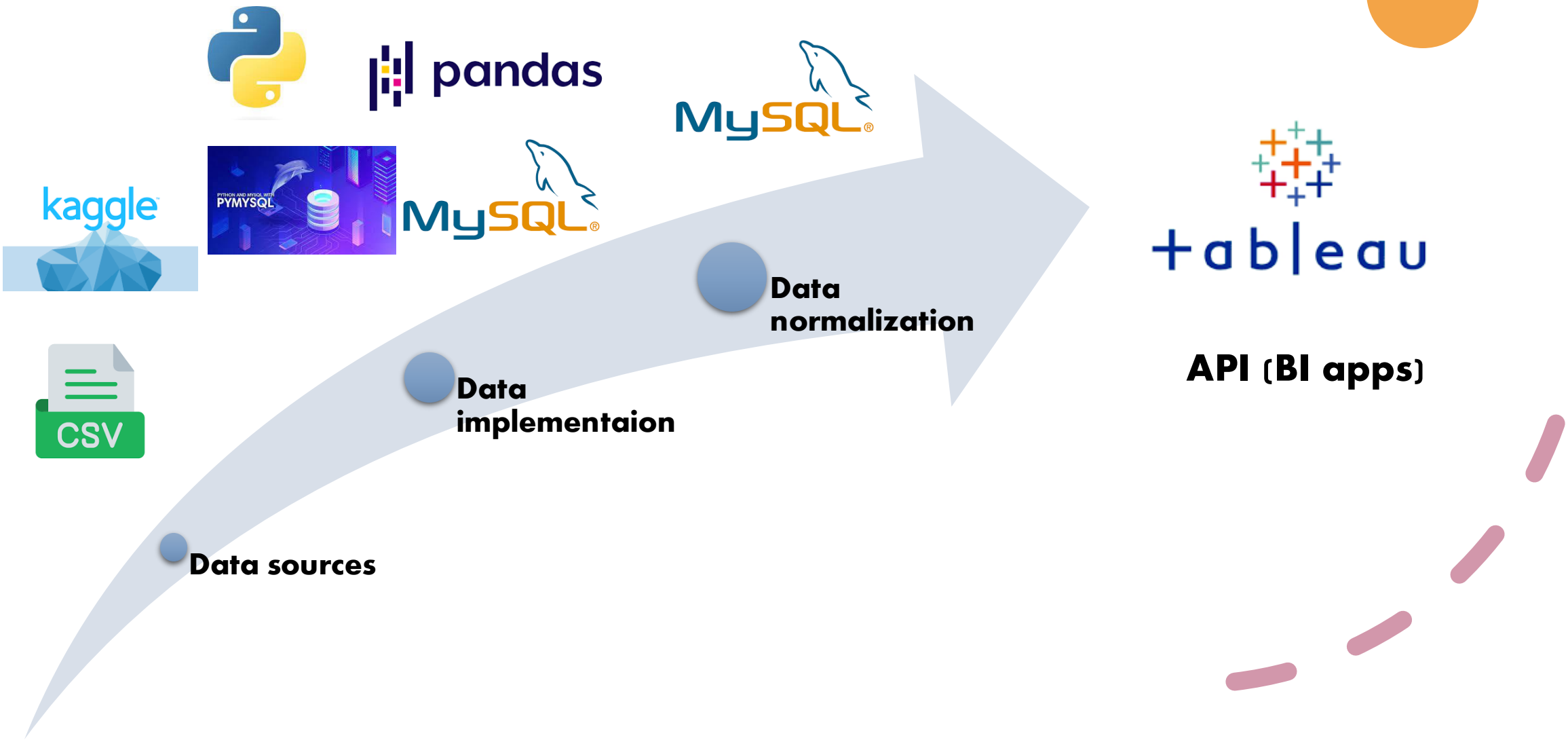
Accurate sales analysis is crucial for optimizing inventory levels and reducing waste.

Historical sales data will be analyzed to identify patterns and trends.

Descriptive analytics will be used to analyze sales for thousands of product families sold at Favorita stores in Ecuador.

Data Pipeline Design

-Pipeline structure



Data Pipeline Design
-data implementation

Connect MySQL server in Python

- PYMYSQL library
  - Connect() function
  - Cursor() function

```python
import pymysql
import pandas as pd
```

Python

```python
def create_database():
    # connect to default database
    conn = pymysql.connect(host='127.0.0.1',port=int(3306),user='root', passwd='')
    cur = conn.cursor()

    #create sparkify database with UTF8 encoding
    cur.execute("DROP DATABASE IF Exists original_db")
    cur.execute("CREATE DATABASE original_db")

    #close connection to default database
    conn.close()

    #connec                    base
    conn = pymysql.connect(host='127.0.0.1',port=int(3306),user='root', passwd='', db = 'original_db')
    cur = conn.cursor()

    return cur, conn
```

(module) pymysql

Python

Import data

- Pandas library
  - pd.read_csv
  - Drop_na() function
  - Drop_duplicate() function

```python
holidays_events = pd.read_csv('/Users/zhengzheng/Desktop/IE 6700 SQL/final_project/data_table/holidays_events.csv'
oil = pd.read_csv('/Users/zhengzheng/Desktop/IE 6700 SQL/final_project/data_table/oil.csv')
stores = pd.read_csv('/Users/zhengzheng/Desktop/IE 6700 SQL/final_project/data_table/stores.csv')
train = pd.read_csv('/Users/zhengzheng/Desktop/IE 6700 SQL/final_project/data_table/train.csv')
transactions = pd.read_csv('/Users/zhengzheng/Desktop/IE 6700 SQL/final_project/data_table/transactions.csv')
```

Python

```python
holidays_events = holidays_events.drop_duplicates().dropna()
oil = oil[['date', 'dcoilwtico']].drop_duplicates().dropna()
stores = stores[['store_nbr', 'city', 'state', 'type', 'cluster']].drop_duplicates().dropna()
train = train[['id', 'date', 'store_nbr', 'family', 'sales', 'onpromotion']].drop_duplicates().dropna()
transactions = transactions[['date', 'store_nbr', 'transactions']].drop_duplicates().dropna()
```

Python

+ Code + Markdown

Data Pipeline Design
-data implementation

Create database and table in MySQL

```python
holidays_events_table_create = ("""
CREATE TABLE IF NOT EXISTS holidays_events(
date DATE,
type VARCHAR(225),
locale VARCHAR(225),
locale_name VARCHAR(225),
description VARCHAR(225),
transferred TINYINT(1) NOT NULL DEFAULT 0
)""")
cur.execute(holidays_events_table_create)
conn.commit()
```

Python

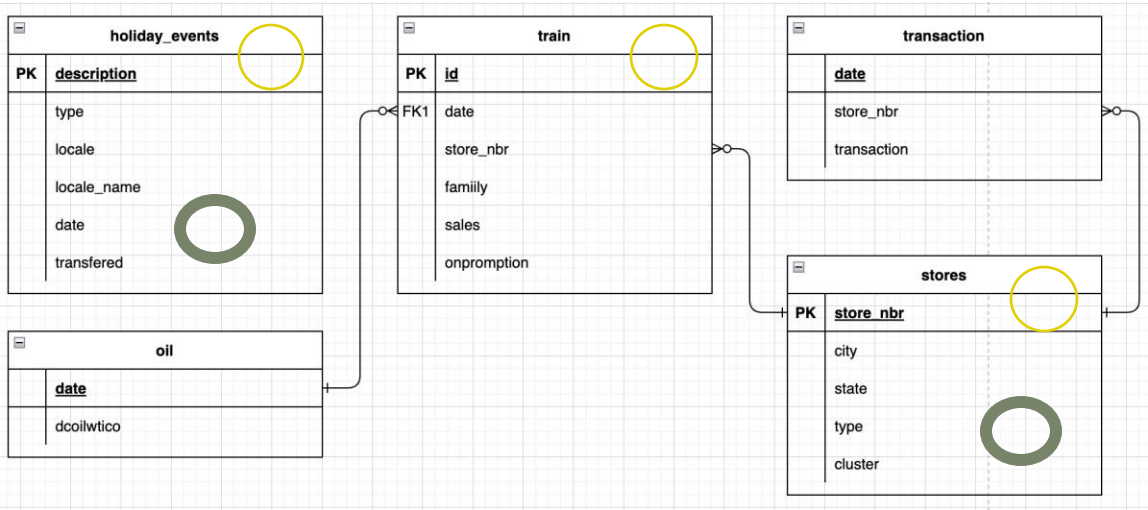
Insert data into tables

```python
holidays_events_table_insert = ("""INSERT INTO holidays_events(
date,
type,
locale,
locale_name,
description,
transferred)
VALUES(%s, %s, %s, %s, %s, %s)
""")

for i, row in holidays_events.iterrows():
    cur.execute(holidays_events_table_insert, tuple(row))

conn.commit()
```

Python

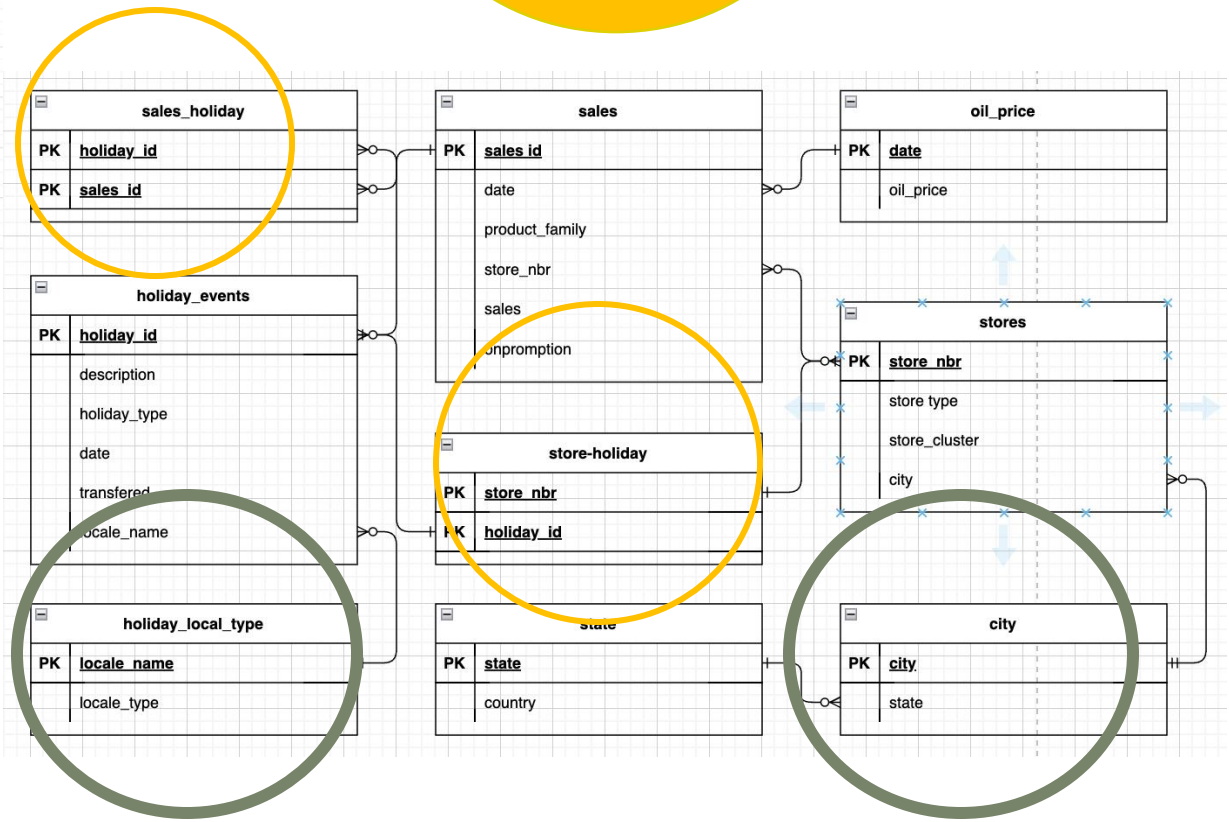
Data Pipeline Design

-data normalization



Many-Many tables: **Transaction table**

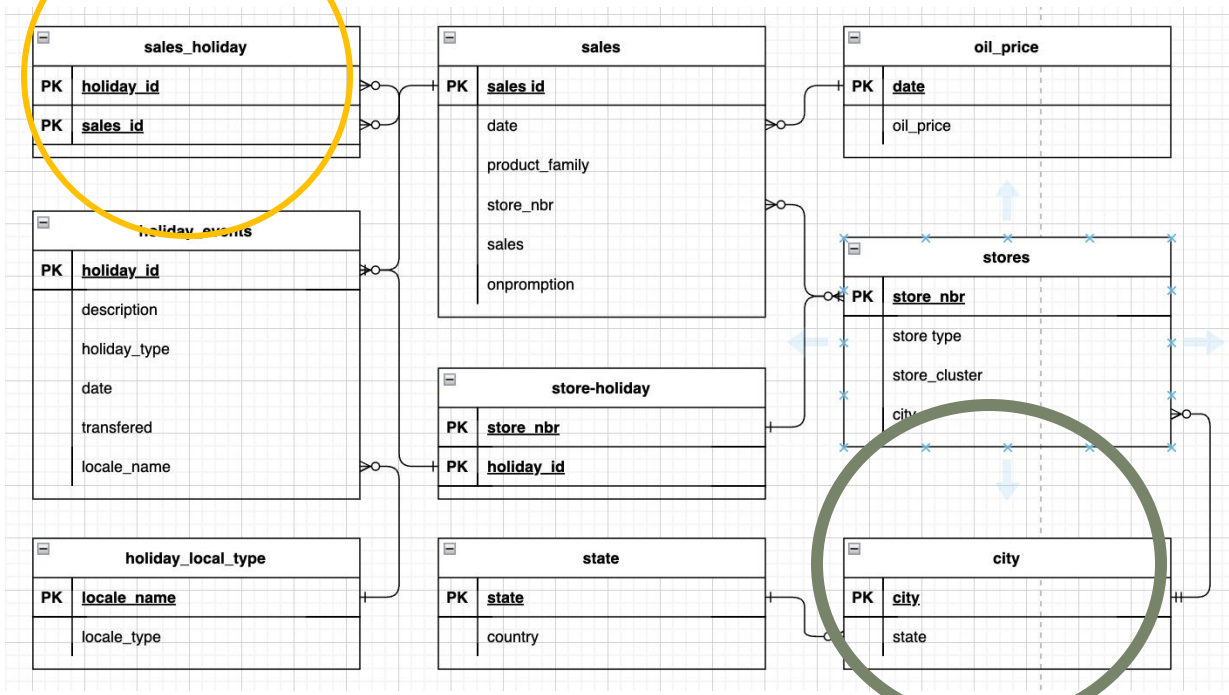
Transitive dependency e.g. store_nbr -> city -> state -> country



Data Pipeline Design

-data normalization

```
INSERT INTO store_holiday (holiday_id, store_nbr)
WITH combine AS
(
SELECT s.store_nbr, h.holiday_id
FROM original_db.train t
LEFT JOIN original_db.stores s
ON t.store_nbr = s.store_nbr
LEFT JOIN original_db.holidays_events h
ON s.country = h.locale_name AND t.date = h.date
UNION DISTINCT
SELECT s.store_nbr, h.holiday_id
FROM original_db.train t
LEFT JOIN original_db.stores s
ON t.store_nbr = s.store_nbr
LEFT JOIN original_db.holidays_events h
ON s.state = h.locale_name AND t.date = h.date
UNION DISTINCT
SELECT s.store_nbr, h.holiday_id
FROM original_db.train t
LEFT JOIN original_db.stores s
ON t.store_nbr = s.store_nbr
LEFT JOIN original_db.holidays_events h
ON s.city = h.locale_name AND t.date = h.date
)
```

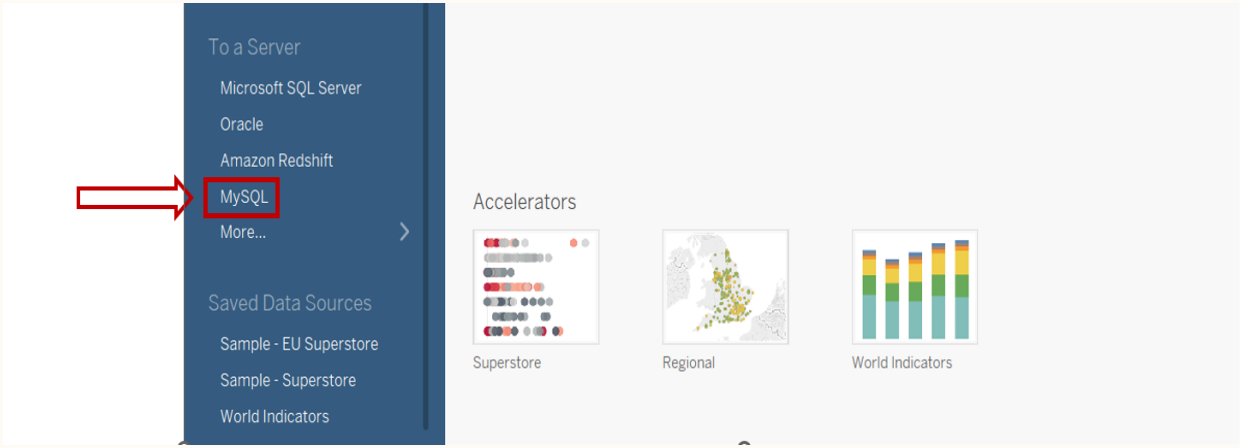


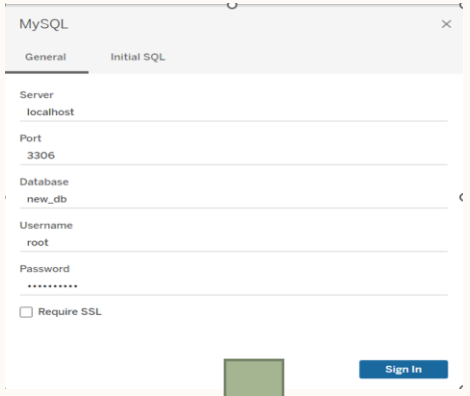
Transaction table:
LEFT JOIN
&
DISTINCT
UNION

```
INSERT INTO city (city, state)
SELECT
    s.city,
    s.state
FROM original_db.stores s
GROUP BY s.city, s.state
;

INSERT INTO state (state, country)
SELECT
    s.state,
    s.country
FROM original_db.stores s
GROUP BY s.state, s.country
;
```

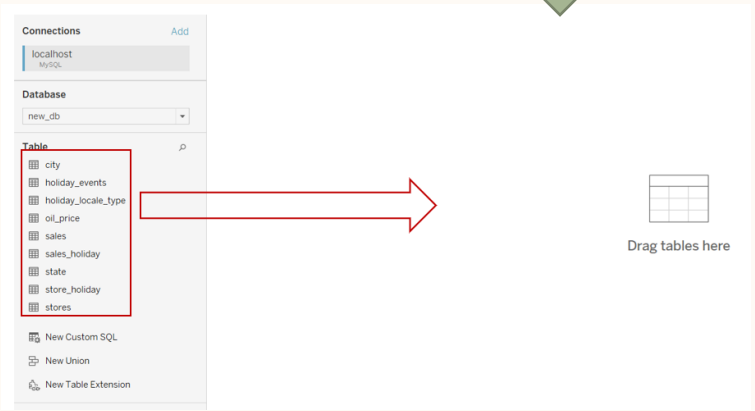
Transitive dependency
SELECT

Data Visualization - Connect Tableau with MySQL





Drag and drop all six tables into the designated area labeled "Drag tables"



Data Visualization - Connect Tableau with MySQL

- Due to large size of the data, generating an analytic report in Tableau was too slow
- View table [sale_detail] was created by joining all the tables required for the analysis

```
CREATE VIEW Sale_Detail AS
select s.sales_id, s.date,  Year(s.date) as Sale_Year
, MONTHNAME(s.date) as Sale_Month
, DAYNAME(s.date) as Sale_Day
, s.product_family
, s.sales
, s.onpromotion
, o.oil_price
, s.store_nbr
, s1.store_type
, s1.store_cluster
, s1.state
, s1.city
, s2.holiday_type
from Sales s
left join oil_price o on s.date=o.date
left join (select st.store_nbr, st.store_type, st.store_cluster, c.state,c.city from stores st inner join city c on c.city=st.city) s1 on s.store_nbr=s1.store_nbr
left join (select  sh.store_nbr, he.date  , max(he.holiday_type) as holiday_type from holiday_events he  inner join store_holiday sh on he.holiday_id=sh.holiday_id
group by sh.store_nbr , he.holiday_type, he.date  ) s2 on s.store_nbr=s2.store_nbr and s.date = s2.date
```

Data Visualization - Connect Tableau with MySQL

- Since [Sale_Detail view table contains all the necessary attributes, it will be the only table used in the Tableau data box

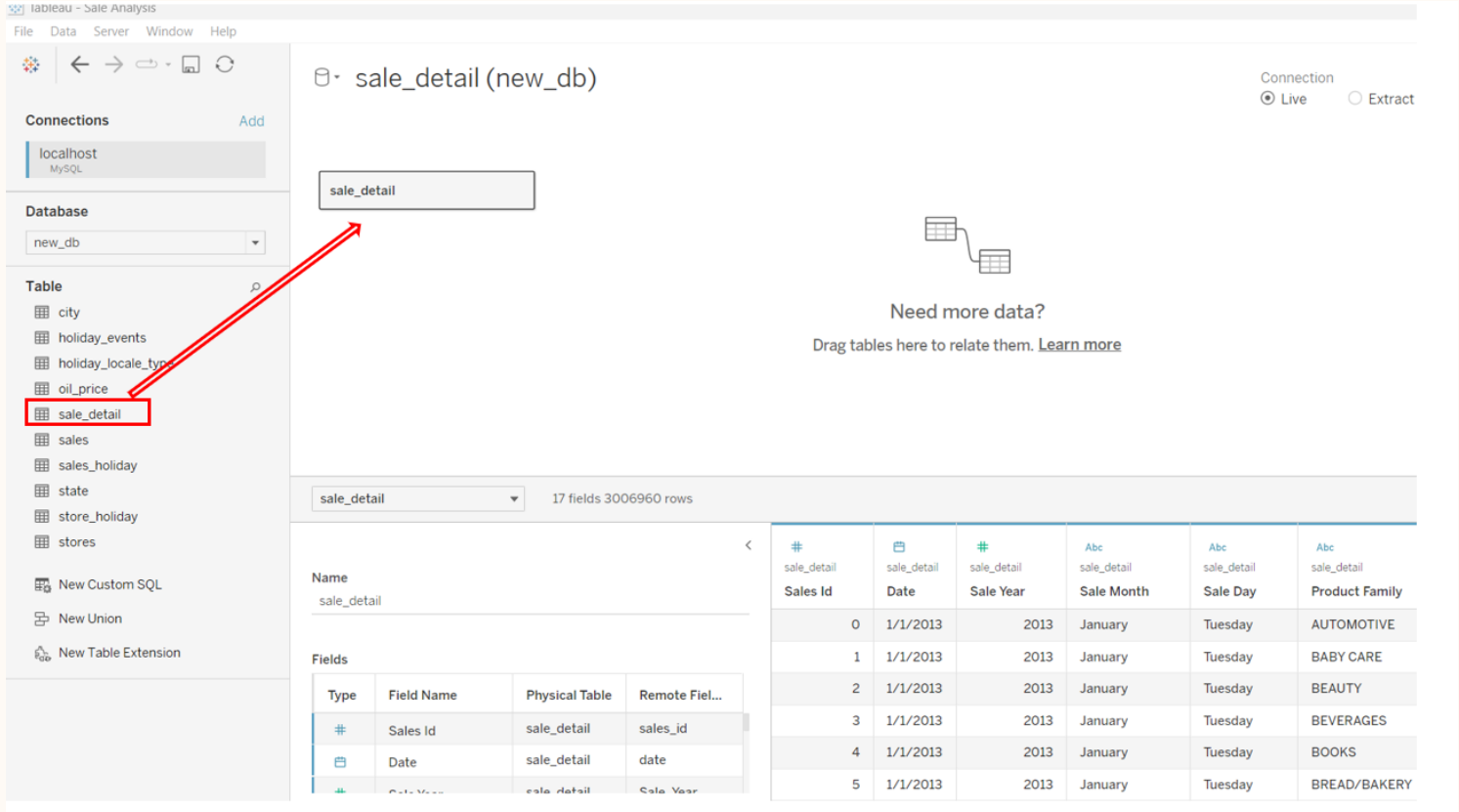
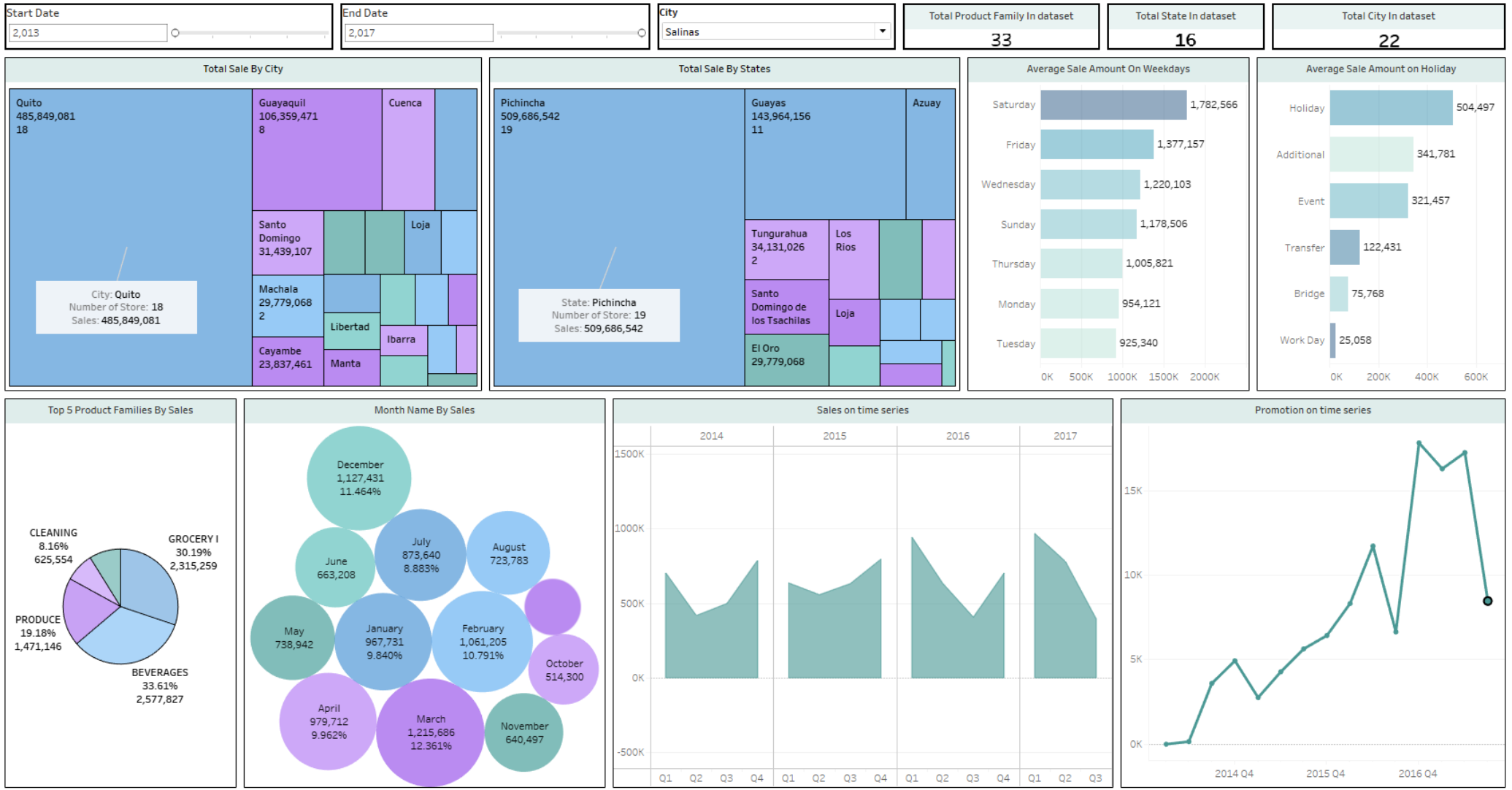


Tableau Dashboard – sale & promotion visualization



Thank You!

Any question ?