



Maven Toys Database Management and Analysis Report

[Github link](#)

Authors:
Ferdawes Haoualaa
Imen Kenza Chouaieb

IT-300 Business Intelligence and Database Management Systems
Pr.Abdelkader & Pr.Azzouz
January 2024

Table of contents:

1.Introduction

2.Database Management and analysis:

2.1.Data Gathering

2.2.ETL Process

2.3.Modeling and Datawarehouse Creation

2.4.Data Analysis

3.Conclusion

1.Introduction:

The Maven Toys Database project analyzes this business management of inventory and sales based on different factors such as the products' categories and the stores' locations all over Mexico.

These valuable insights will introduce us to the target customers' location and preferences, boosting the company's future expansion and inventory management strategies.

2.Implementation:

2.1.Data Gathering:

The Maven Toys dataset was extracted from Maven Analytics.

2.2.ETL Process:

We first started by cleaning data by manipulating:

-Columns' names to facilitate the process:

```
#define cleaning 1 : change all columns names to lower case
# code :
sales.columns= sales.columns.str.lower()
inventory.columns= inventory.columns.str.lower()
product.columns= product.columns.str.lower()
stores.columns= stores.columns.str.lower()
```

-Columns' type (eg: the currency and date ones):

```
#define : converting datatype :in stores table: Store_Open_Date to date
#code
stores['store_open_date']=pd.to_datetime(stores['store_open_date'])
#test
stores.info()
```

```
#define : in product table : product_price and product_cost to float
product['product_price'] = product['product_price'].str.replace('$', '').astype(float)
product['product_cost'] = product['product_cost'].str.replace('$', '').astype(float)
```

-New columns needed for analysis (eg: profit):

```
#define cleaning 3 : add a profit column
# code
product["profit"]=product["product_price"] - product["product_cost"]
#test
product
```

2.3.Modeling and Data Warehouse Creation:

The datasets contains 4 tables: Sales, Inventory, Products and Stores. The core of our data warehousing infrastructure comprised PostgreSQL as the database management system, SQLAlchemy, Python library, for seamless table mapping within our code, and psycopg2, a Python library, enabling smooth interactions between them.

The final tables contain the following information:

- Stores: store_id, store_name, store_city, store_location, store_open_date and year_operation
- Products: product_id, product_name, product_category, product_cost, product_price and profit
- Inventory: store_id, product_id, stock_on_hand
- Sales: sale_id, date, store_id, product_id, units, year, quarter, month

The next line codes represent how we connected data to PostgreSQL using Python (full code in data warehouse file)

```
In [9]: from sqlalchemy import create_engine, Column, Date, Float, Integer, String, ForeignKey
from sqlalchemy.orm import relationship
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy.orm import sessionmaker

SQLALCHEMY_DATABASE_URL = f'postgresql://{{"postgres"}:{"root"}@{"localhost"}/{{"postgres"}}'

engine = create_engine(SQLALCHEMY_DATABASE_URL)

SessionLocal = sessionmaker(autocommit=False, autoflush=False, bind=engine)

Base = declarative_base()

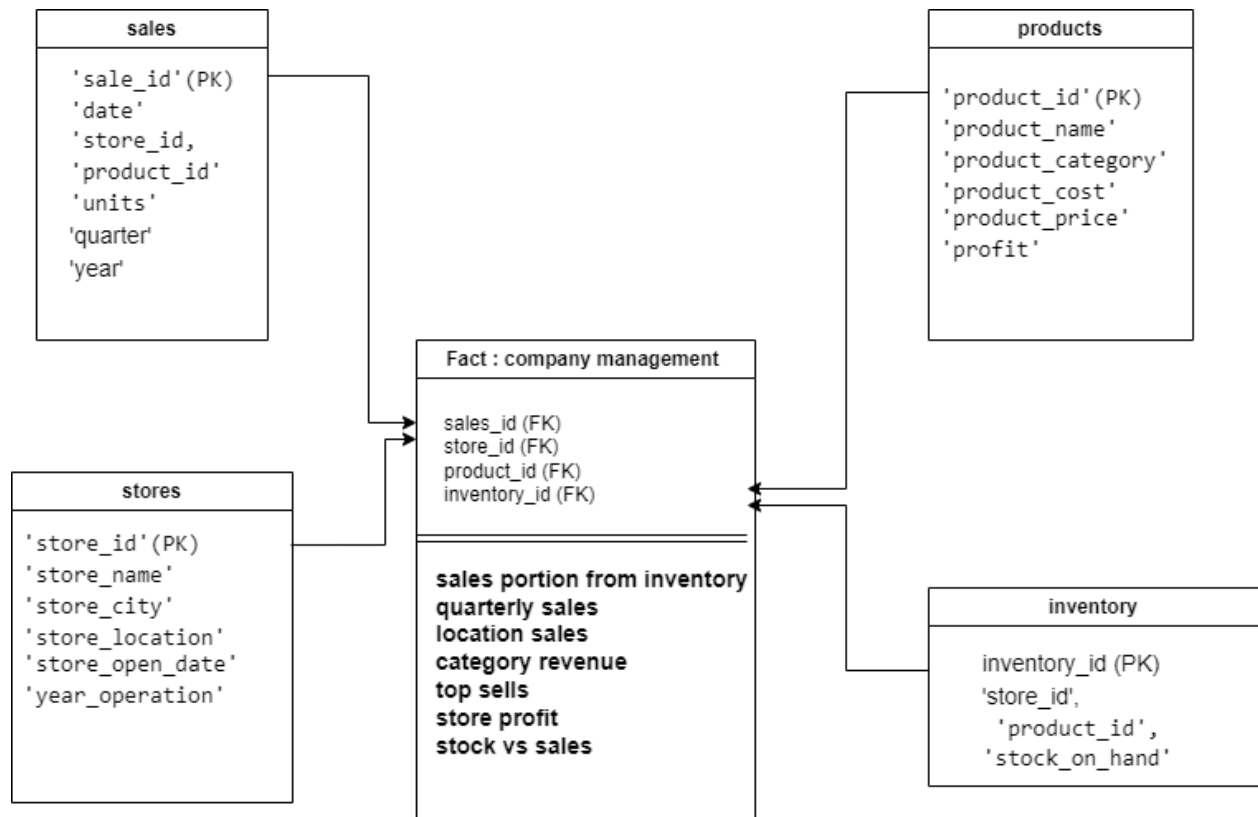
def get_db():
    db = SessionLocal()
    try:
        yield db
    finally:
        db.close()

class sales(Base):
    __tablename__ = 'sales'
    sale_id = Column(Integer, ForeignKey('products.product_id', ondelete='CASCADE'), ForeignKey('stores.
    date = Column(Date)
    units = Column(Integer)
    year = Column(Integer)
    quarter = Column(Integer)
    month = Column(Integer)

class products(Base):
    __tablename__ = 'products'
    product_id = Column(Integer, ForeignKey('inventory.inventory_id', ondelete='CASCADE'), primary_key=
    product_name = Column(String)
    product_category = Column(String)
    product_cost = Column(Float)
    product_price = Column(Float)
    profit = Column(Float)
    sales = relationship("sales", backref="products")

class inventory(Base):
    __tablename__ = 'inventory'
    inventory_id = Column(Integer, primary_key=True)
    stock_on_hand = Column(Integer)
    products = relationship("products", backref="inventory")
    stores = relationship("stores", backref="sales")
```

The following star schema figure represents the fact table: company management and the dimensions:



The ROLAP phase is presented using psycopg2 library in python: (due to some connection problems, we worked with a second method (SQL file))

```

cursor = conn.cursor()
cursor.execute("""
SELECT stores.store_id,
       100.0 * SUM(sales.units) / (SUM(sales.units) + SUM(inventory.stock_on_hand)) AS sales_proportion
FROM sales
JOIN stores ON sales.store_id = stores.store_id
JOIN inventory ON stores.store_id = inventory.store_id AND sales.product_id = inventory.product_id
GROUP BY stores.store_id;
""")
sales_proportions = cursor.fetchall()

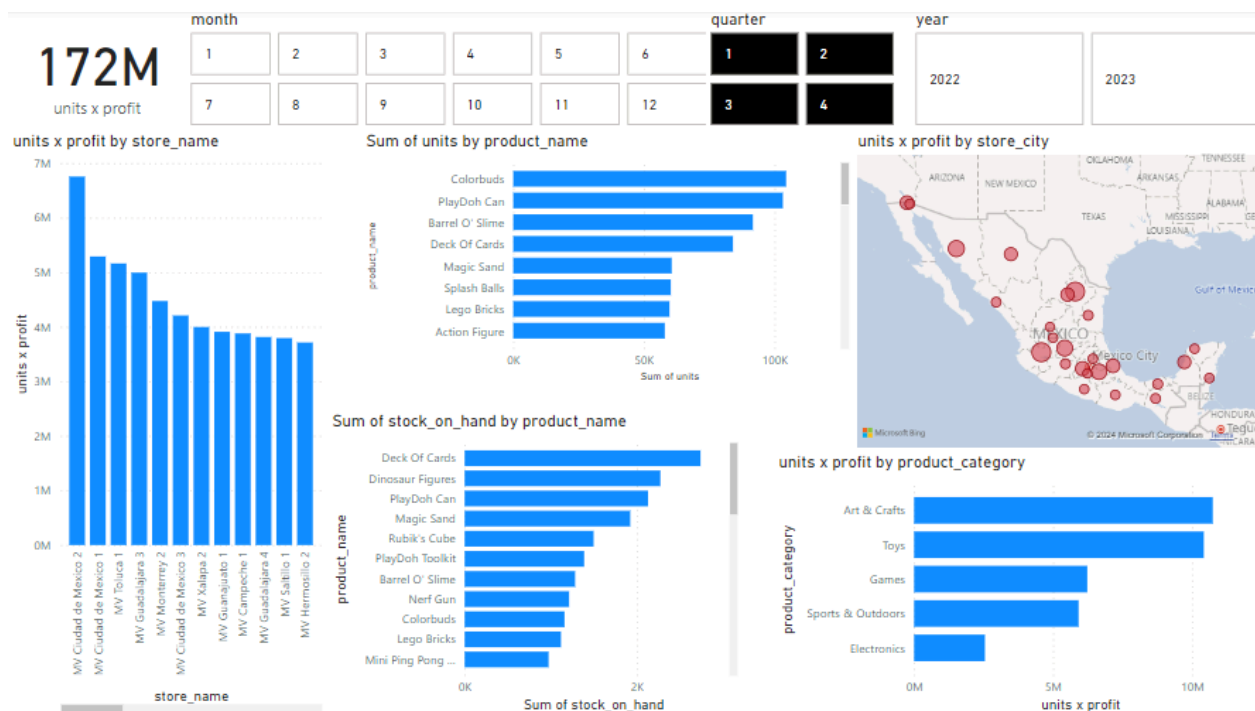
print("Proportion of sales per inventory:")
for store_id, proportion in sales_proportions:
    print(f"Store name: {store_name}, Proportion: {proportion:.2f}%")

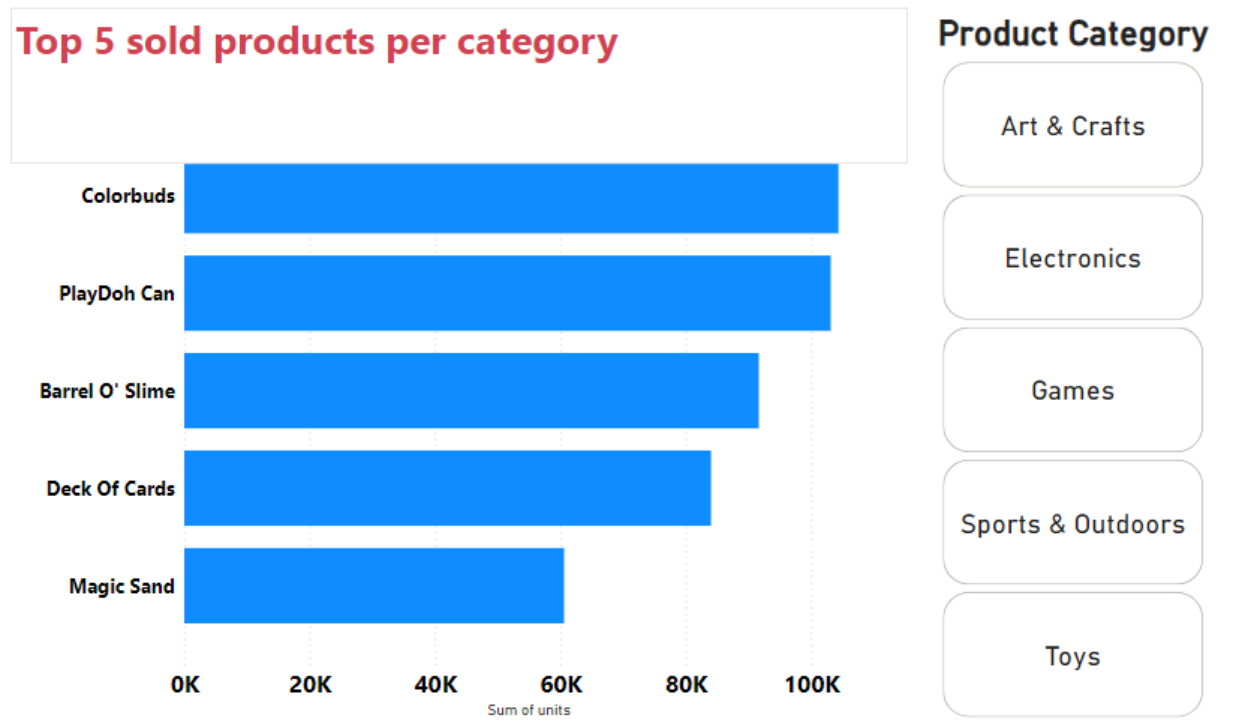
```

2.4.Data Analysis

For the data analysis we started by asking important questions about the business, creating new measures and presenting answers in plots filtered through slicers, usually location,date or product category ones, we got the following insights(PBIX file, we presented 2 of the 5 pages):

- The most profitable product category is **art & crafts** then **toys**
- The store with the highest total profit is **Ciudad de Mexico 2**
- The products that were sold more than 100K times are: in the first-place Colorbuds and the second place PlayDoh Can
- The Deck of Cards product has the highest stock in inventory thus the company should reconsider selling this product
- Ciudad city stores have the highest profitability rate with a profit higher than 5 million for each





3.Conclusion:

In conclusion, the analysis provided that the location of the percentage of target customers is in Ciudad City, Mexico. Their favorite games are Colorbuds and PlayDoh Can. Moreover, to attract more people to new games, Maven Toys must enlarge its products under the arts and crafts category. These insights are the ones that will serve the most in the growth of the company by boosting its strategies and decision-making.