SP 24: Advanced Projects III, Section BK36

PRESENTED BY : Jiayu Liu(jl13683,N13584659),Harshini Dayanand (hd2379, N12186579)
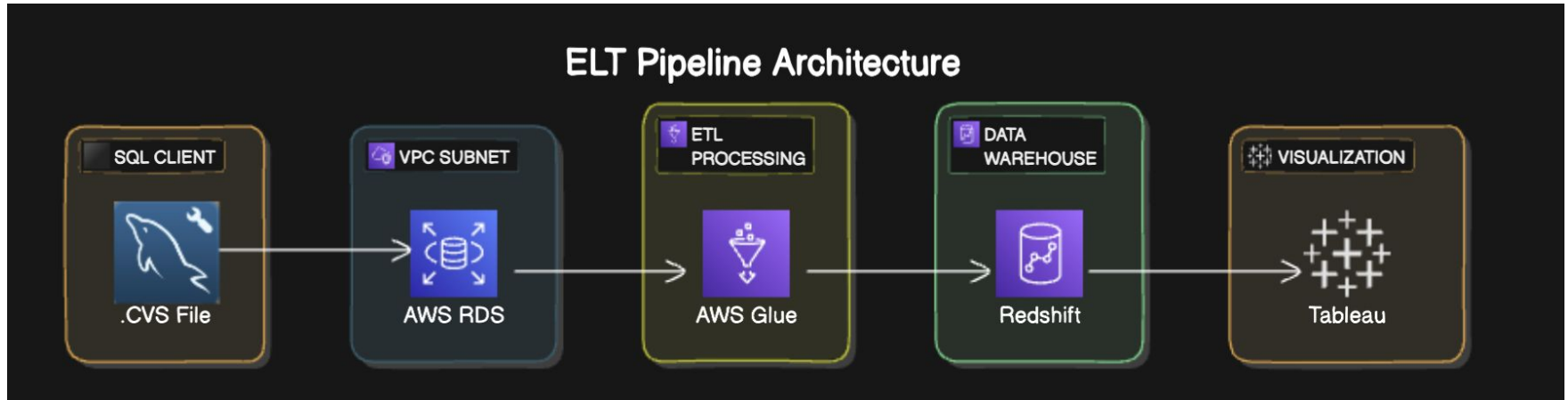
Satya Vamsi Gudimella (sg7401,N14501266)

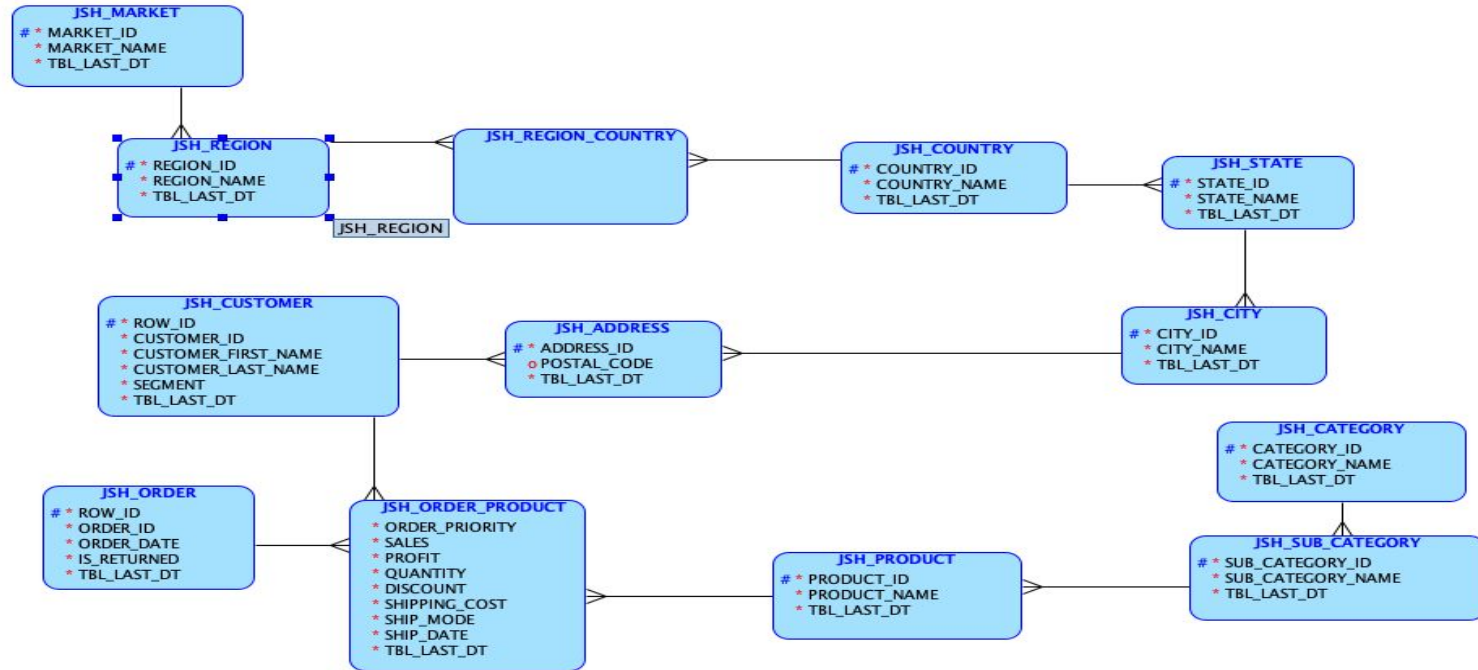# An Awesome Inc Case Study

Business Case

Awesome Inc. seeks to enhance operational efficiency and customer service through the development of an online transaction processing (OLTP) database and data analytics system, alongside a data warehouse (DW) solution on a cloud platform like AWS.

The resulting system will support self-service capabilities for customers, streamline retail processes, enable advanced analytics, reporting, and data visualization, and facilitate data-driven decision-making across marketing, manufacturing, and financial domains.
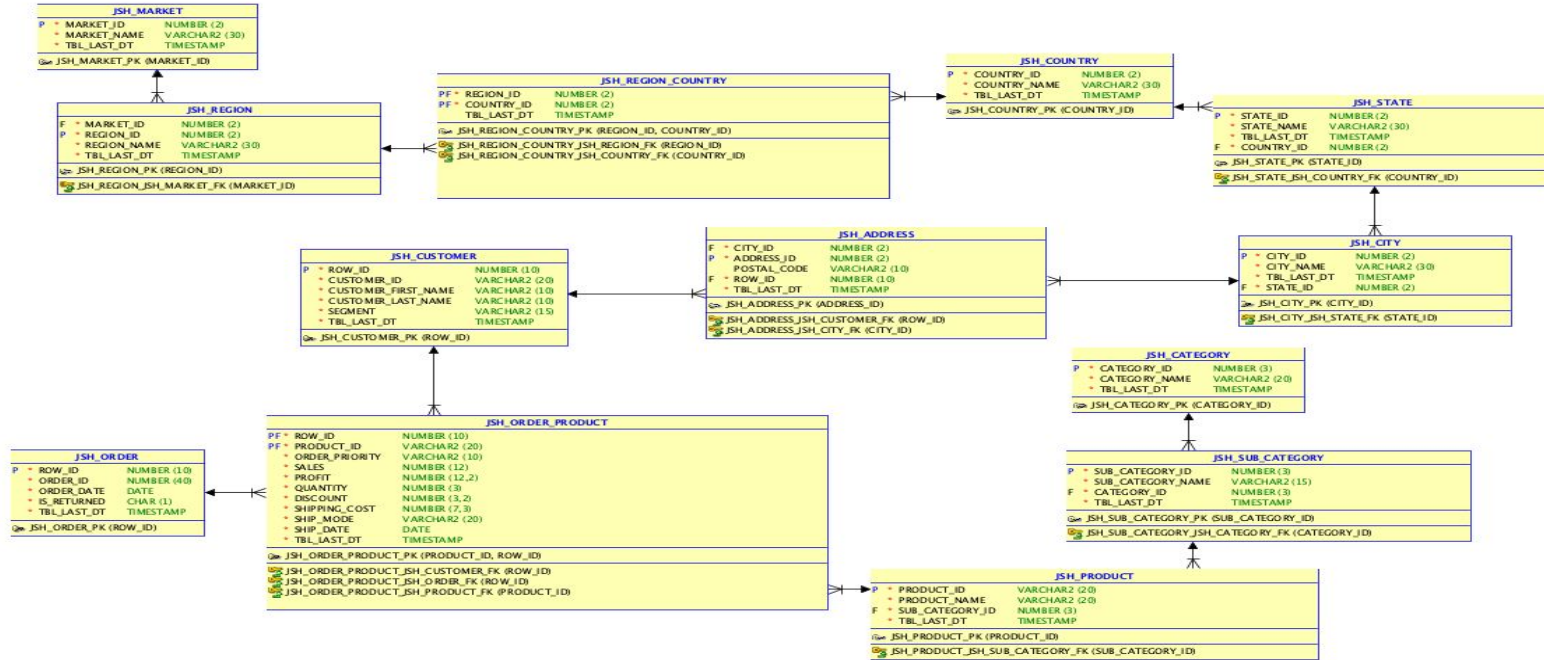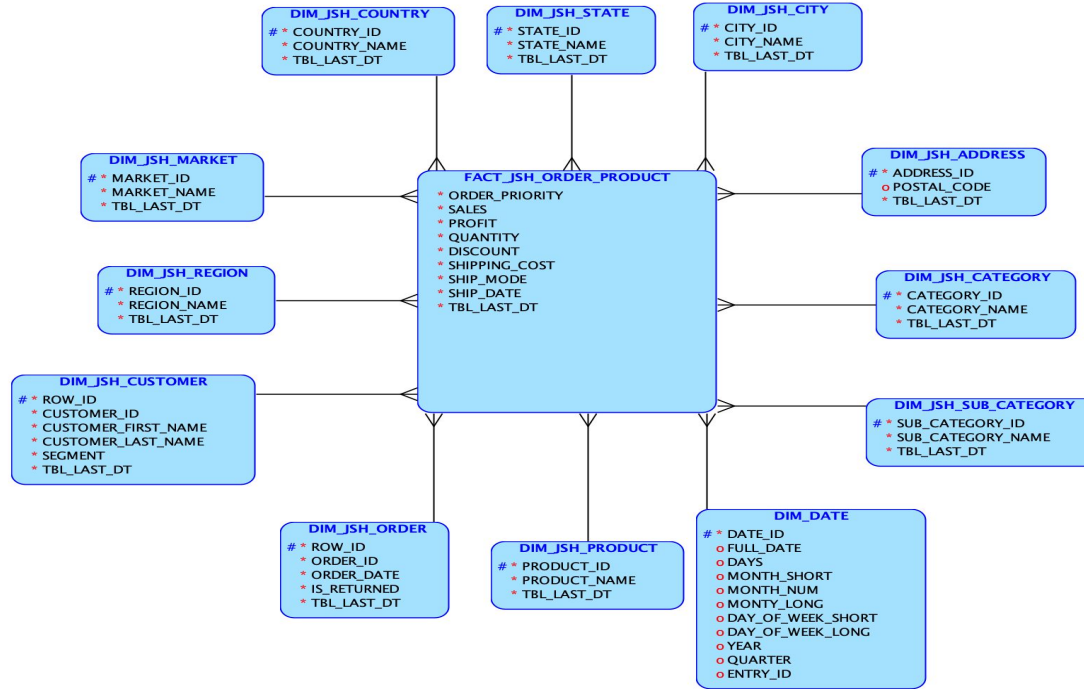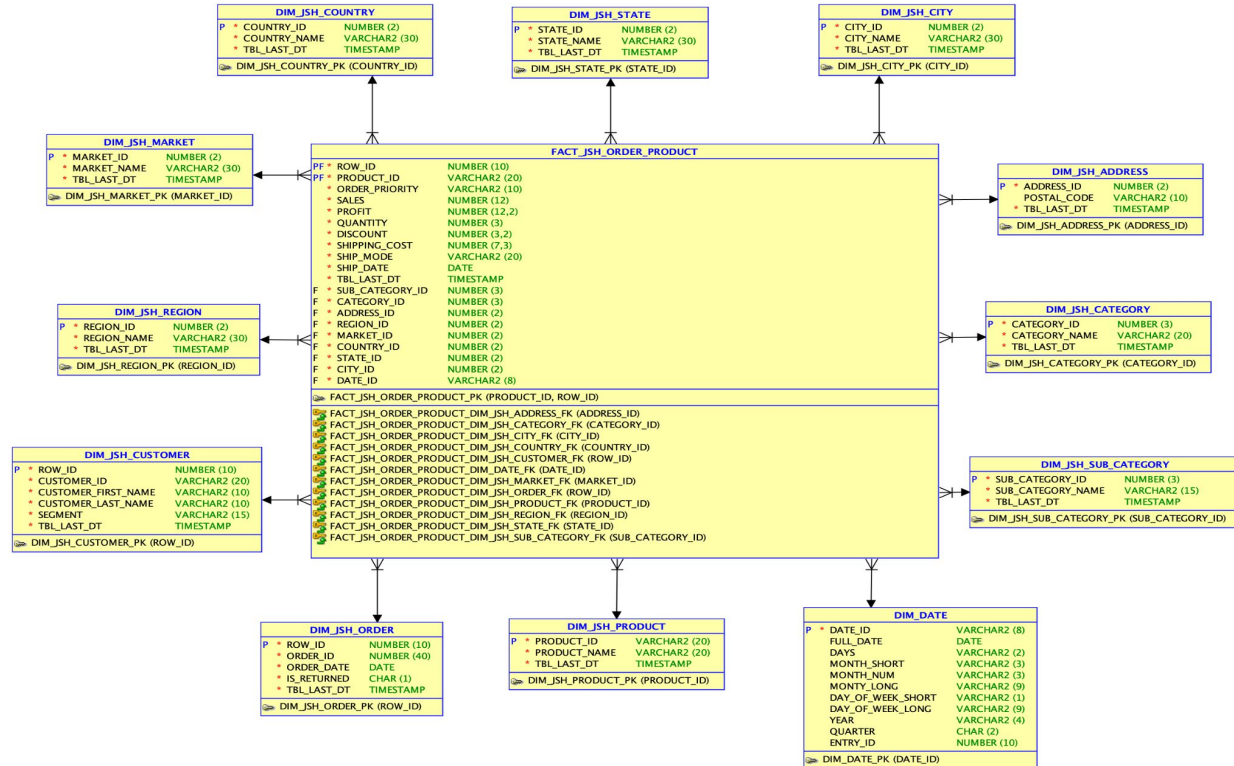
System Topology

ELT Pipeline Architecture

SQL CLIENT — .CVS File → VPC SUBNET — AWS RDS → ETL PROCESSING — AWS Glue → DATA WAREHOUSE — Redshift → VISUALIZATION — Tableau

Logical Model of OLTP

JSH_MARKET
- # * MARKET_ID
- * MARKET_NAME
- * TBL_LAST_DT

JSH_REGION
- # * REGION_ID
- * REGION_NAME
- * TBL_LAST_DT

JSH_REGION

JSH_REGION_COUNTRY

JSH_COUNTRY
- # * COUNTRY_ID
- * COUNTRY_NAME
- * TBL_LAST_DT

JSH_STATE
- # * STATE_ID
- * STATE_NAME
- * TBL_LAST_DT

JSH_CUSTOMER
- # * ROW_ID
- * CUSTOMER_ID
- * CUSTOMER_FIRST_NAME
- * CUSTOMER_LAST_NAME
- * SEGMENT
- * TBL_LAST_DT

JSH_ADDRESS
- # * ADDRESS_ID
- o POSTAL_CODE
- * TBL_LAST_DT

JSH_CITY
- # * CITY_ID
- * CITY_NAME
- * TBL_LAST_DT

JSH_CATEGORY
- # * CATEGORY_ID
- * CATEGORY_NAME
- * TBL_LAST_DT

JSH_ORDER
- # * ROW_ID
- * ORDER_ID
- * ORDER_DATE
- * IS_RETURNED
- * TBL_LAST_DT

JSH_ORDER_PRODUCT
- * ORDER_PRIORITY
- * SALES
- * PROFIT
- * QUANTITY
- * DISCOUNT
- * SHIPPING_COST
- * SHIP_MODE
- * SHIP_DATE
- * TBL_LAST_DT

JSH_PRODUCT
- # * PRODUCT_ID
- * PRODUCT_NAME
- * TBL_LAST_DT

JSH_SUB_CATEGORY
- # * SUB_CATEGORY_ID
- * SUB_CATEGORY_NAME
- * TBL_LAST_DT

Relational Model of OLTP

Logical Model of Data Warehouse

DIM_JSH_COUNTRY
* COUNTRY_ID
* COUNTRY_NAME
* TBL_LAST_DT

DIM_JSH_STATE
* STATE_ID
* STATE_NAME
* TBL_LAST_DT

DIM_JSH_CITY
* CITY_ID
* CITY_NAME
* TBL_LAST_DT

DIM_JSH_MARKET
* MARKET_ID
* MARKET_NAME
* TBL_LAST_DT

DIM_JSH_ADDRESS
* ADDRESS_ID
o POSTAL_CODE
* TBL_LAST_DT

FACT_JSH_ORDER_PRODUCT
* ORDER_PRIORITY
* SALES
* PROFIT
* QUANTITY
* DISCOUNT
* SHIPPING_COST
* SHIP_MODE
* SHIP_DATE
* TBL_LAST_DT

DIM_JSH_REGION
* REGION_ID
* REGION_NAME
* TBL_LAST_DT

DIM_JSH_CATEGORY
* CATEGORY_ID
* CATEGORY_NAME
* TBL_LAST_DT

DIM_JSH_CUSTOMER
* ROW_ID
* CUSTOMER_ID
* CUSTOMER_FIRST_NAME
* CUSTOMER_LAST_NAME
* SEGMENT
* TBL_LAST_DT

DIM_JSH_SUB_CATEGORY
* SUB_CATEGORY_ID
* SUB_CATEGORY_NAME
* TBL_LAST_DT

DIM_JSH_ORDER
* ROW_ID
* ORDER_ID
* ORDER_DATE
* IS_RETURNED
* TBL_LAST_DT

DIM_JSH_PRODUCT
* PRODUCT_ID
* PRODUCT_NAME
* TBL_LAST_DT

DIM_DATE
* DATE_ID
o FULL_DATE
o DAYS
o MONTH_SHORT
o MONTH_NUM
o MONTY_LONG
o DAY_OF_WEEK_SHORT
o DAY_OF_WEEK_LONG
o YEAR
o QUARTER
o ENTRY_ID

NYU

Relational Model of Data Warehouse

ETL approach

- Trial and Error on different approaches

  - Original approach:File permission issue

    with AWS

- AWS Glue with dimension tables

ETL approach

● AWS Glue with the fact table


```
SQL query
Enter a SQL statement to add to your job.

1  select op.*, subcate.sub_category_id,cate.category_id, address.address_id, reg.re
2  join reg on reg.market_id = mar.market_id
3  join regcon on regcon.region_id = reg.region_id
4  join country on regcon.country_id = country.country_id
5  join state on country.country_id = state.country_id
6  join city on state.state_id = city.state_id
7  join address on city.city_id = address.city_id
8  join customer on address.row_id = customer.row_id
9  join op on customer.row_id = op.row_id
10 join o on op.row_id = o.row_id
11 join p on op.product_id = p.product_id
12 join subcate on p.sub_category_id = subcate.sub_category_id
13 join cate on subcate.category_id = cate.category_id
```

NYU

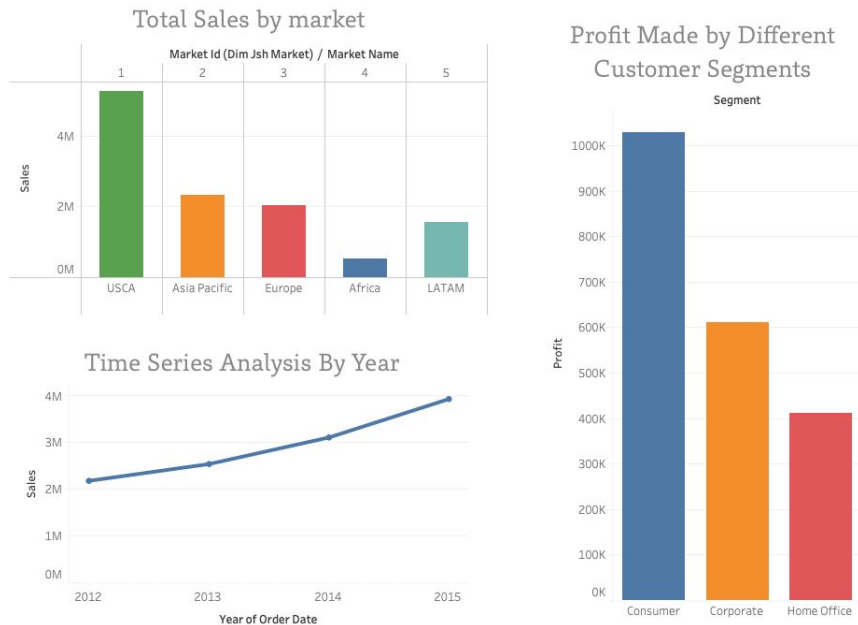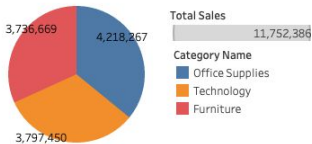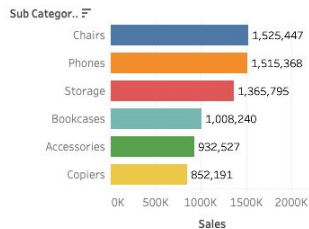# Tableau Dashboards

Tableau Dashboards

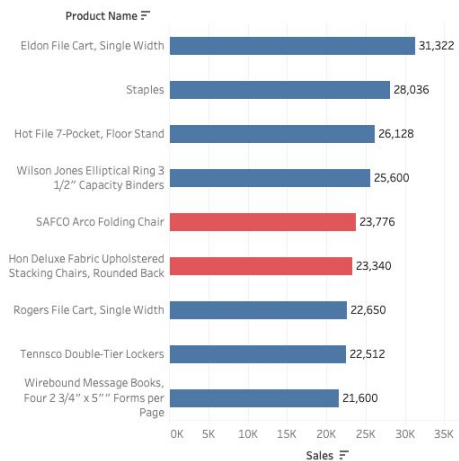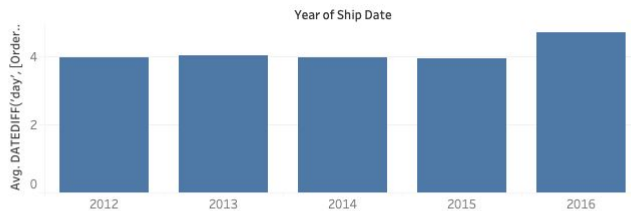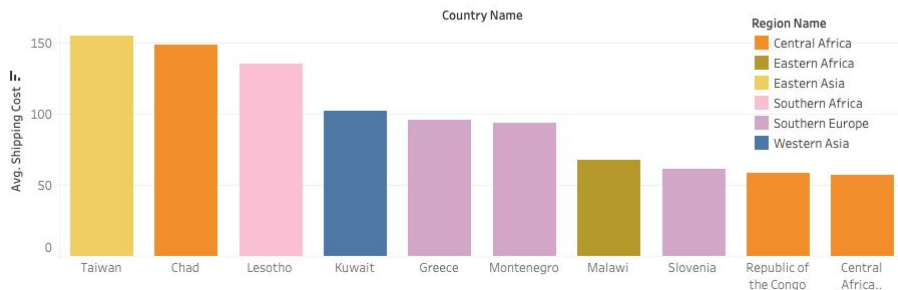Category, Sub-Category and Product Performance

Tableau Dashboards

Order Processing

Average days to Ship by Year

Top 10 Countries with Highest Average Shipping Cost

Partitioned Table

```
-- before partition, query and the time it takes
SELECT * FROM jsh_order_copy
WHERE order_date >= '2012-01-01' AND order_date < '2013-01-01'; -- 551ms
SELECT * FROM jsh_order_copy
WHERE order_date >= '2013-01-01' AND order_date < '2014-01-01'; -- 601ms
```

```
-- after partition, query and the time it takes
SELECT * FROM jsh_order_copy
WHERE order_date >= '2012-01-01' AND order_date < '2013-01-01'; -- 518ms
SELECT * FROM jsh_order_copy
WHERE order_date >= '2013-01-01' AND order_date < '2014-01-01'; -- 579ms
```

```
ALTER TABLE jsh_order_copy
PARTITION BY RANGE(YEAR(order_date)) (
    PARTITION p2018 VALUES LESS THAN (2012),
    PARTITION p2019 VALUES LESS THAN (2013),
    PARTITION p2020 VALUES LESS THAN (2014),
    PARTITION p2021 VALUES LESS THAN (2015),
    PARTITION pLater VALUES LESS THAN MAXVALUE
);
```

NYU

Demo

–Link: Video Link

NYU

Lesson Learned

- ETL pipeline and Data warehouse creation
- AWS
- Data visualization
- DB design and implementation

Thank you!

NYU