Business Analytics and Data Driven Decision Making

End – 2 – End Analysis using Operational and Analytical Databases

Abid Hussain,
Associate Professor
Center for Business Data Analytics
Department of Digitalization, Copenhagen Business School, Copenhagen, Denmark.

Contact: ah.digi@cbs.dk

AGENDA

Agenda

- Putting it altogether
- End to end Demo Operational to Analytical Dbs to Report

Learning Objective

- Understand the structure of relations/tables suitable for (Visual) analysis
- Understand use of SQL for reporting
- **Strong understanding of GROUP BY concept for aggregations**
- **I don’t expect that you will remember syntax !**

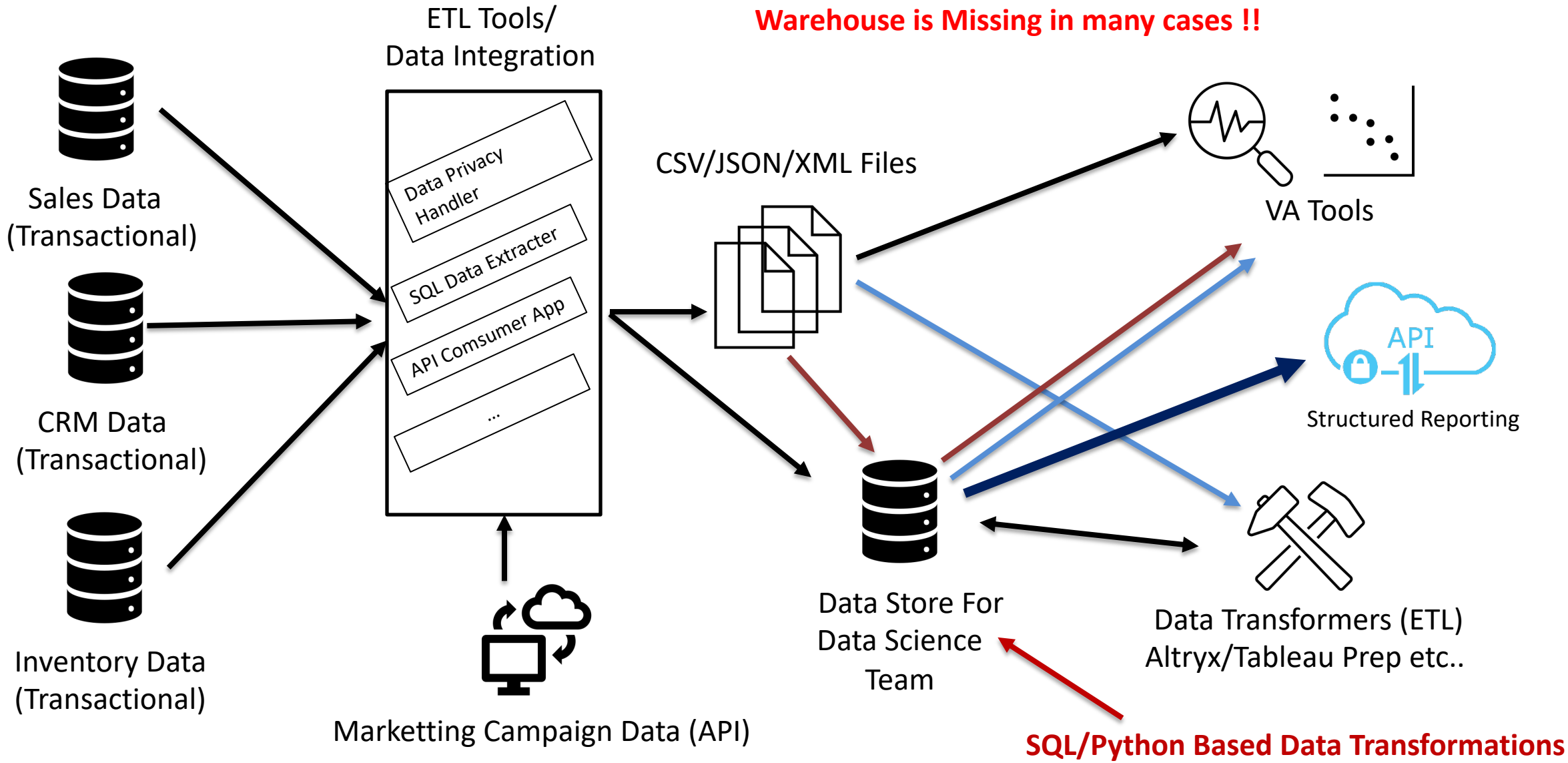
Literature

- This lecture will use examples and illustrations for the following books.

Silberschatz, A., Korth, H., Sudarshan, S. Database System Concepts. 7th Edition. McGraw-Hill Education. ISBN13: 9780078022159

Jukic, N. (2019). Database systems: Introduction to databases and data warehouses.

BA Project Work Flow And Tools



END - END DEMO

Relational Database – Operational DB ER

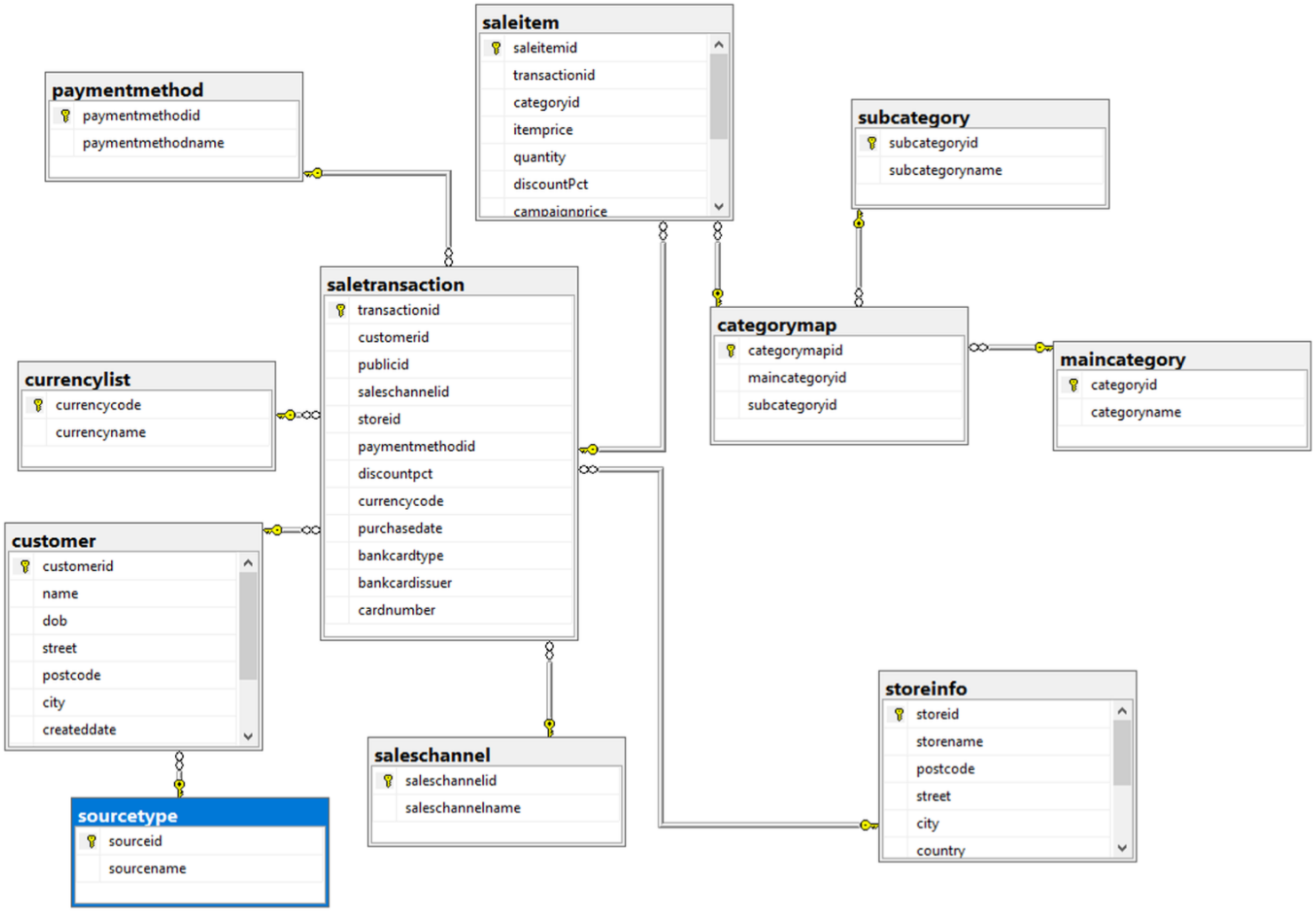
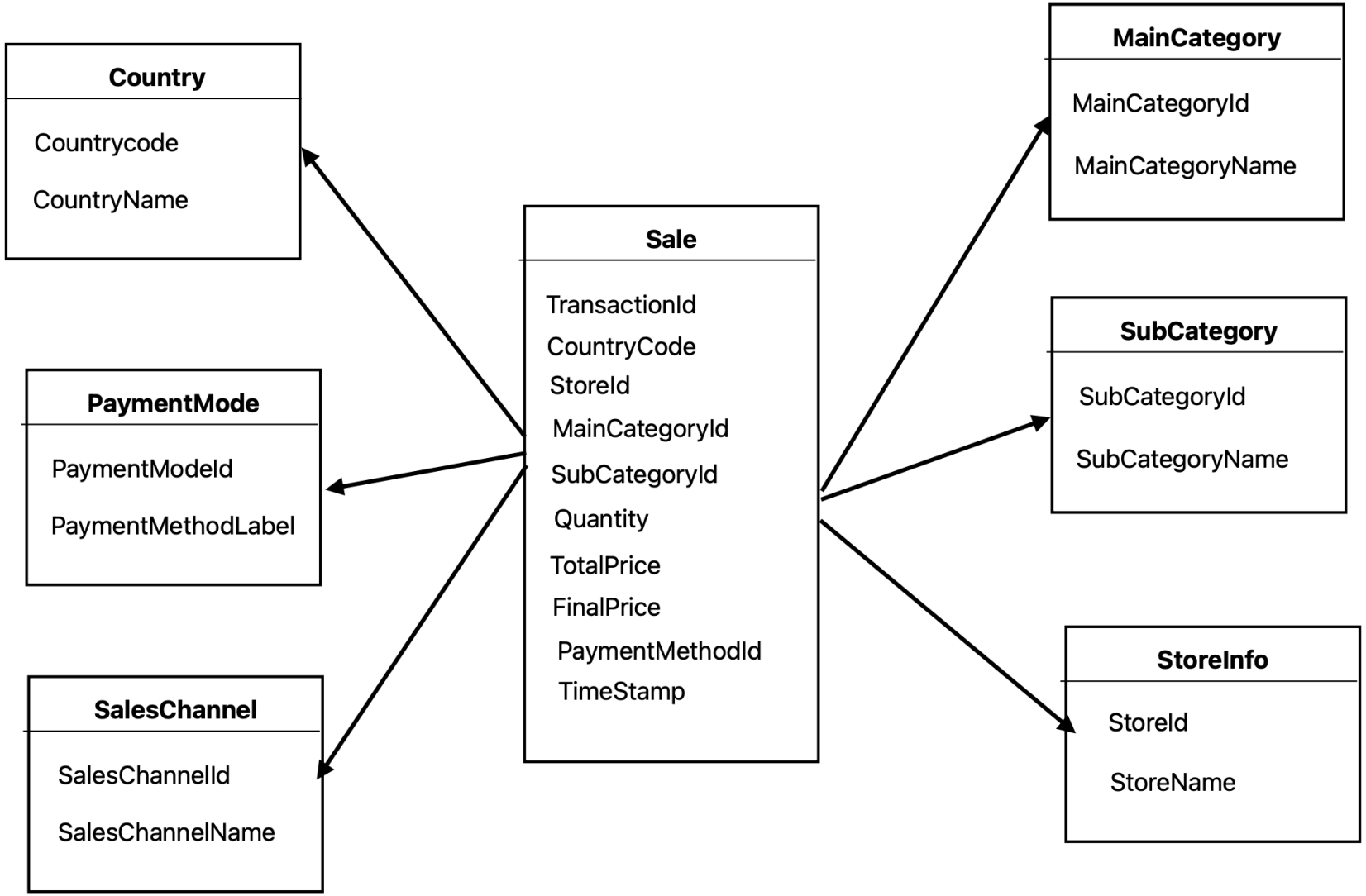


Figure 1: ER Model of actual database

Relational Database – Analytical DB ER for Sales Subject



Views

- create view **dim_country**

```
select countrycode,
case when countrycode = 'N' then 'Norway'
when countrycode = 'S' then 'Sweden'
when countrycode = 'DK' then 'Denmark'
end countryname
from customer
group by countrycode
-- PK: countrycode
```

- create view **dim_paymentmethod** as

```
select paymentmethodid methodid, paymentmethodname methodname from paymentmethod
-- PK: methodid
```

- create view **dim_saleschannel** as

```
select saleschannelid channelid, saleschannelname channelname from saleschannel
-- PK: channelid
```

- create view **dim_maincategory** as

```
select categoryid maincategoryid, categoryname maincategoryname from maincategory
--PK: maincategoryid
```

Views

- create view **dim_subcategory** as

```
select * from subcategory
-- PK: subcategoryid
```

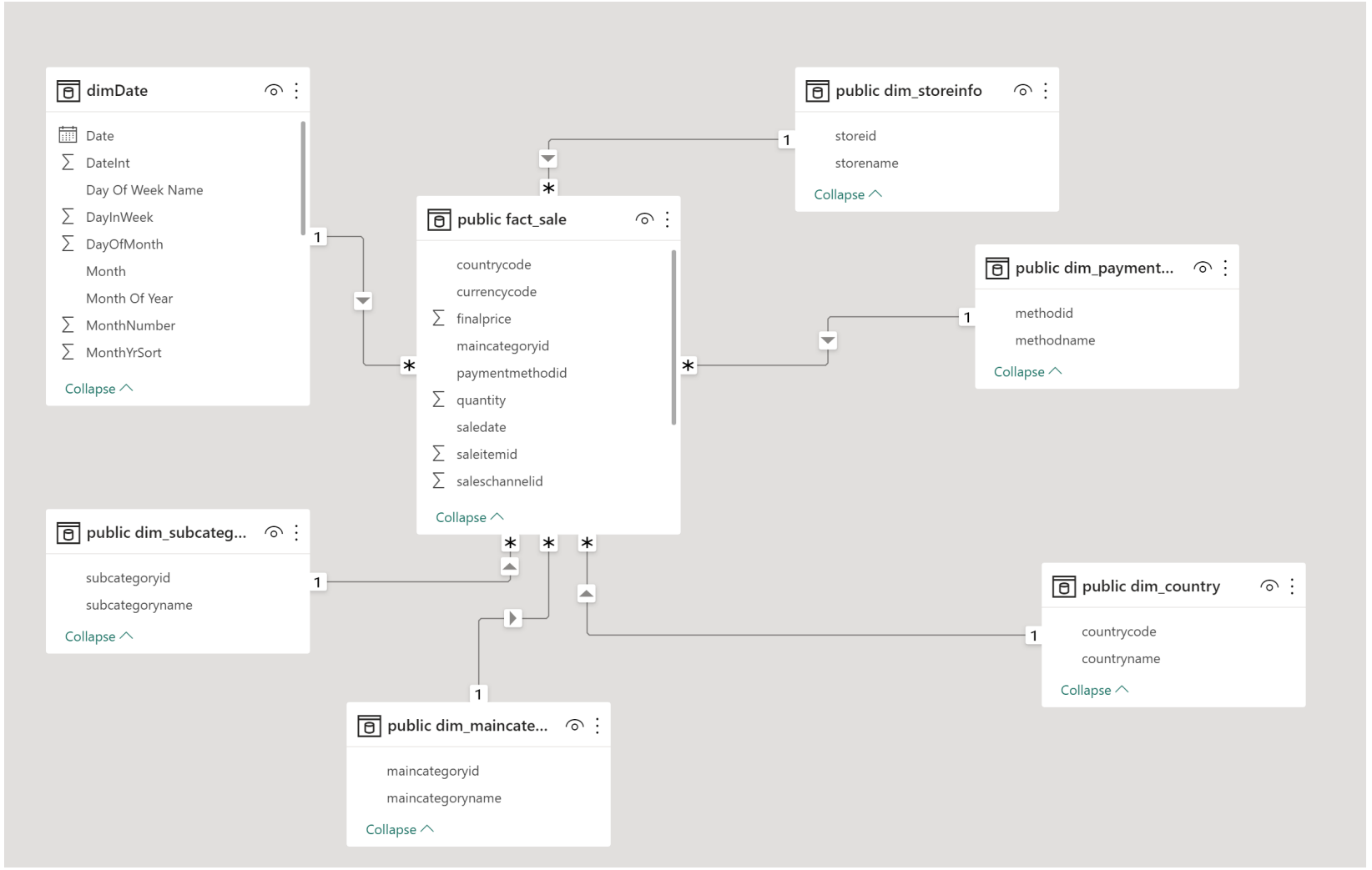
- create view **dim_storeinfo** as

```
select storeid,storename from storeinfo
-- PK: storeid
```

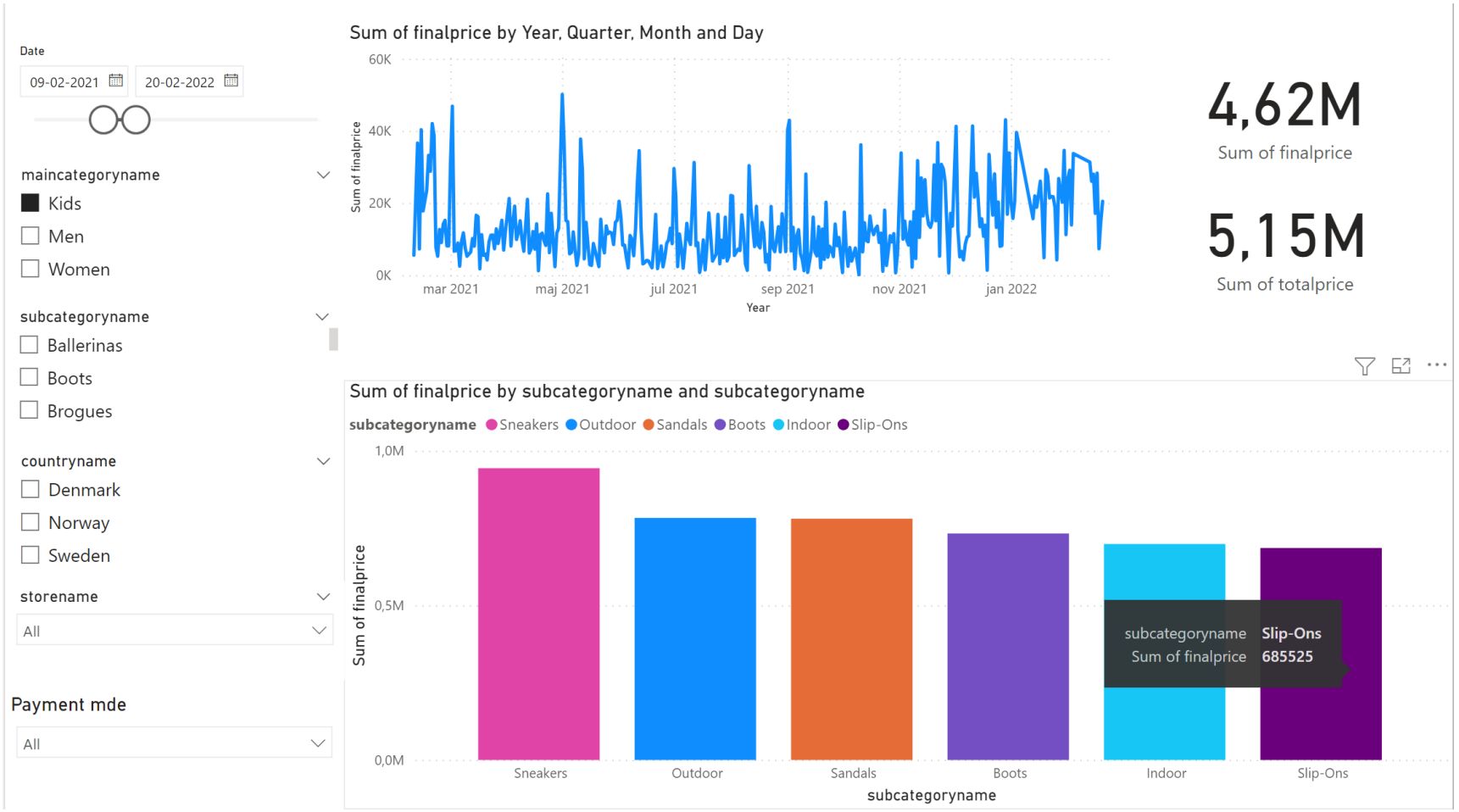
- create view **fact_sale** as

```
select t.transactionid, saleitemid, t.storeid,
cm.subcategoryid,cm.maincategoryid,t.saleschannelid,t.paymentmethodid,
c.countrycode, i.quantity,i.totalprice,i.finalprice, to_char(purchasedate, 'dd-MM-YYYY') saledate,
currencycode
from saleitem i join saletransaction t
on t.transactionid = i.transactionid join categorymap cm
on i.categoryid = cm.categorymapid join customer c
on t.customerid = c.customerid
--PK: SaleItemid and is not relevant
```

Import in Visual Tool > Connect > Design > Deploy



Import in Visual Tool > Connect > Design > Deploy



THANK YOU