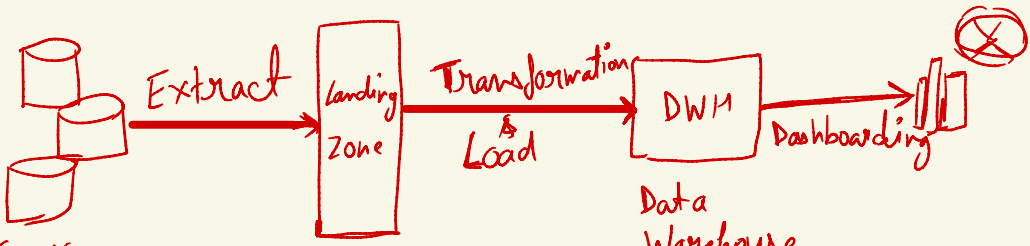
ETL Vs ELT

ETL → Extract transform Load

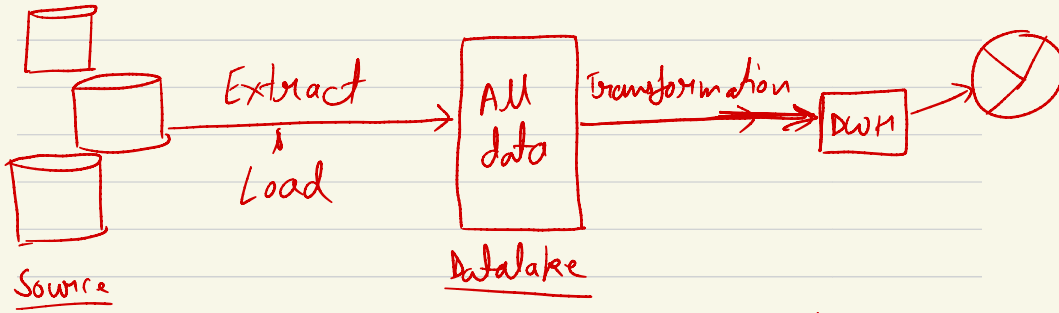


Source
- → databases
- → HDFS
- → API
- → NoSQL
- → IoT
- → Kafka

Temporary Area

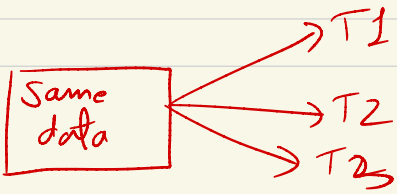
Data Warehouse
- → Hive
- → AWS Redshift
- → Snowflake

ELT → Extract Load Transform



Source

Datalake



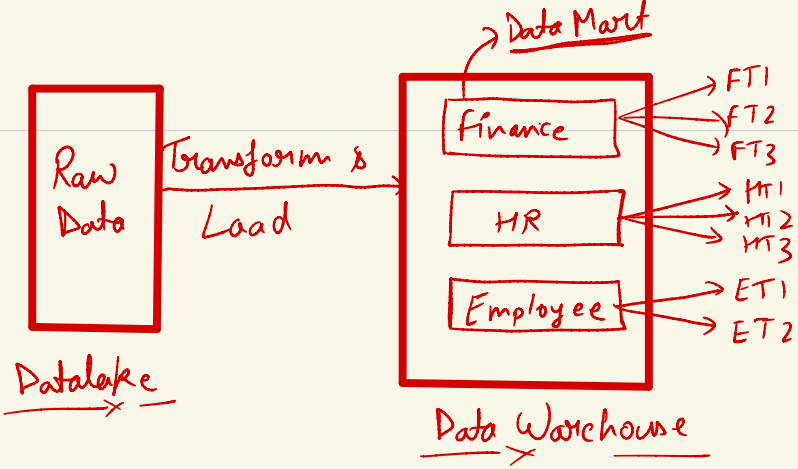
* Datalake: Any system/tool/file system where we keep all the data coming from different sources and different formats. Ex- structured, unstructured, semi-structured. Examples for datalakes:

- HDFS
- AWS S3
- Azure Blob
- GCP Storage

* Data warehouse: Any system/tools/framework which holds transformed data will be known as Data warehouse. It represents data in only structured form.

Example:
- Hive
- Aws Redshift
- Snowflake

* Data Marts: Category level separation inside a Data warehouse is known as Data Marts.



Code Vs NO-Code

Spark (ETL)

Informatica

OLTP vs OLAP

x) OLTP (Online Transaction Processing) : Systems designed to support transaction level operations like Insert - Update - Delete etc. They support ACID properties and main goal is to capture data in real time. In OLTP data resides in it's normalized form.

Example: RDBMS
- MySQL
- Oracle
- PostgreSQL

x) OLAP (Online Analytical Processing)

It is designed to support Analytical Queries. It keep data in denormalized form and doesn't care about ACID properties. Main goal is to help users to query historical data & insights out of it.

Ex: Redshift
Hive
NoSQL
Snowflake

Important Terminologies in Data Warehousing.

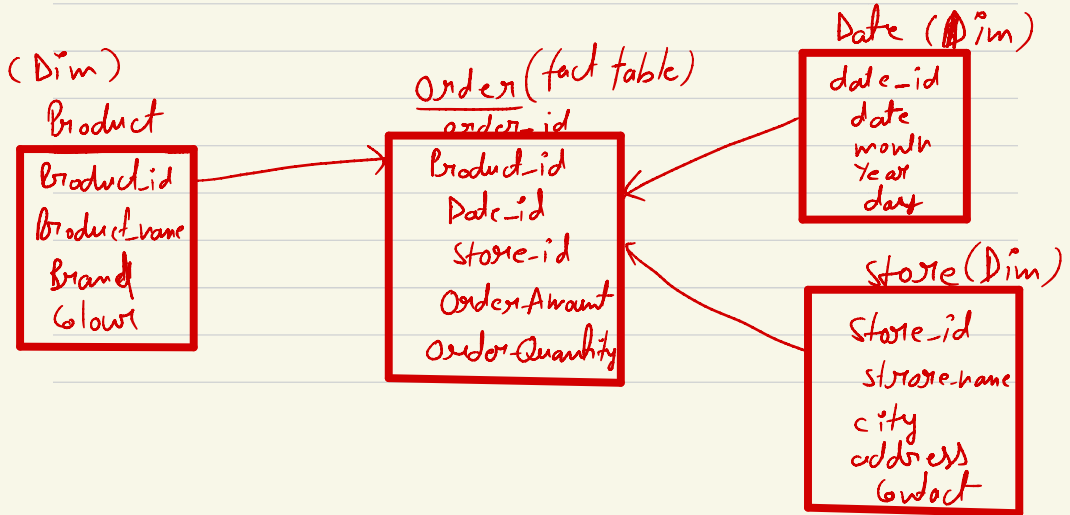
1. Dimension Table
2. Fact Table
3. Relationship
4. Star Schema
5. Snowflake Schema

* Dimension Table :

- It contains dimensions of a fact.
- They are joined with fact tables using foreign keys
- Dimension tables are de-normalized
- It offers descriptive characteristic of the facts with the help of attributes

* Fact Tables

- Contain those numerical attributes which helps to derive meaningful values.
- It holds foreign key of dimension table.



Employee_detail

emp_id	emp_name	dept_name
1	abc	Software
2	xyz	Software
3	klm	HR
4	mno	HR
6	efg	Software

Engineering Department

De-normalized

For Heavy analytical Queries it is good fit.

another approach

emp_id	name	dept_id
1	xyz	100
2	abc	100
3	klm	200
4	mno	200
5	efg	100

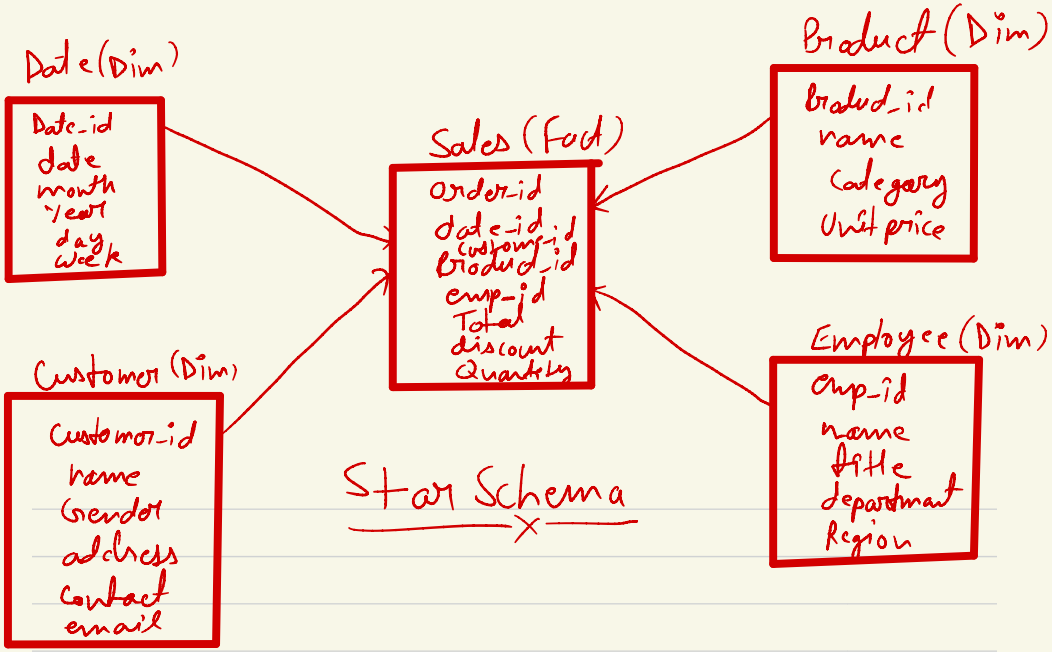
Department

dept_id	dept_name
100	Software
200	HR

Normalized

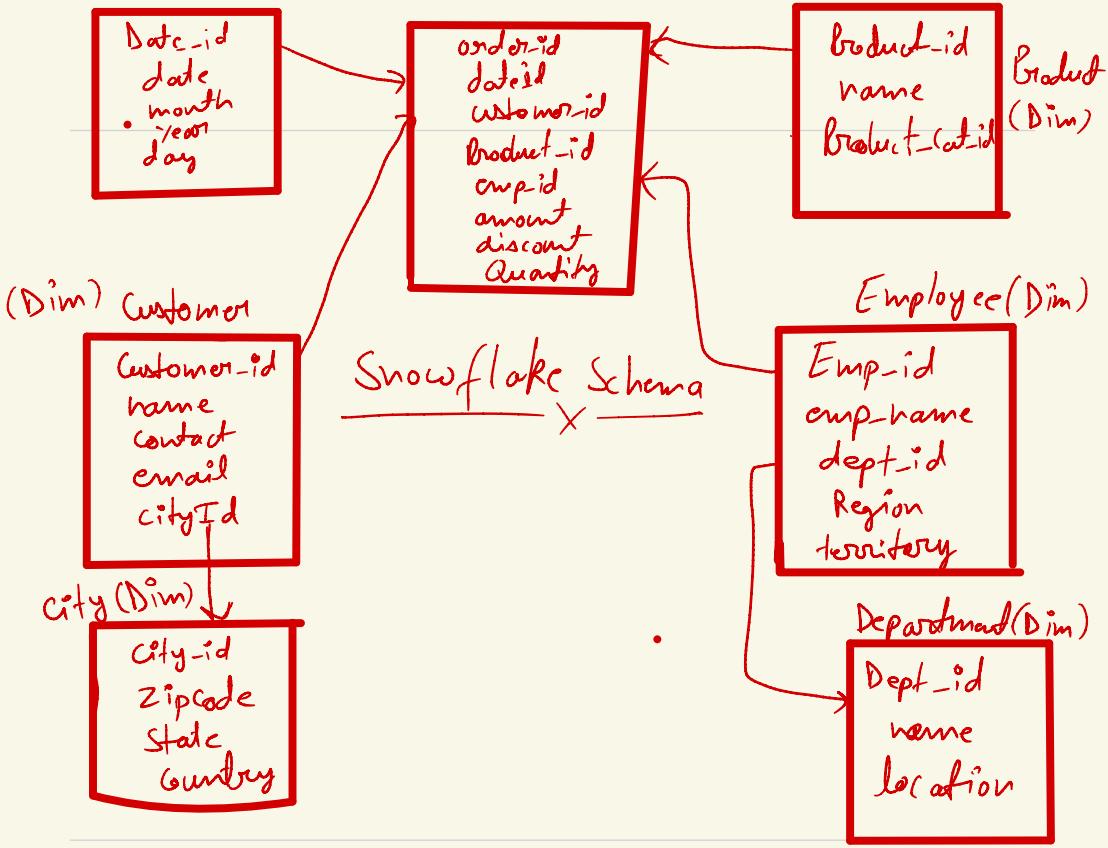
We need to perform Joins which is bad for analytical Queries

Star Schema: In this kind of schema design one or more fact tables will there and they can be connect with any number of dimension tables.



Star Schema

Snowflake schema: In this schema design dimension tables will be in normalized form



Star Schema Vs Snowflake Schema difference

1.) Star schema dim tables are not normalized, Snowflake schemas are normalized.

2) Snowflake schemas will use less space to store dimension data but it is complicated

3) Star schema will only join the fact table with dim tables and that is why queries are faster in star schema than snowflake.

4) Snowflake schemas have no redundant data that's easy to maintain.

5) Snowflake schemas are good for data warehouse, star schemas are better for data marts with simple relationship.

Data Warehousing > Case Study.

1) Four Step Dimensional Design Process

1) Select the Business Process
2) Declare the grain ("How do you describe a single row in the fact table?")
3) Identify Dimensions – Who, What, Where, When, Why and How?
4) Identify facts

Solution

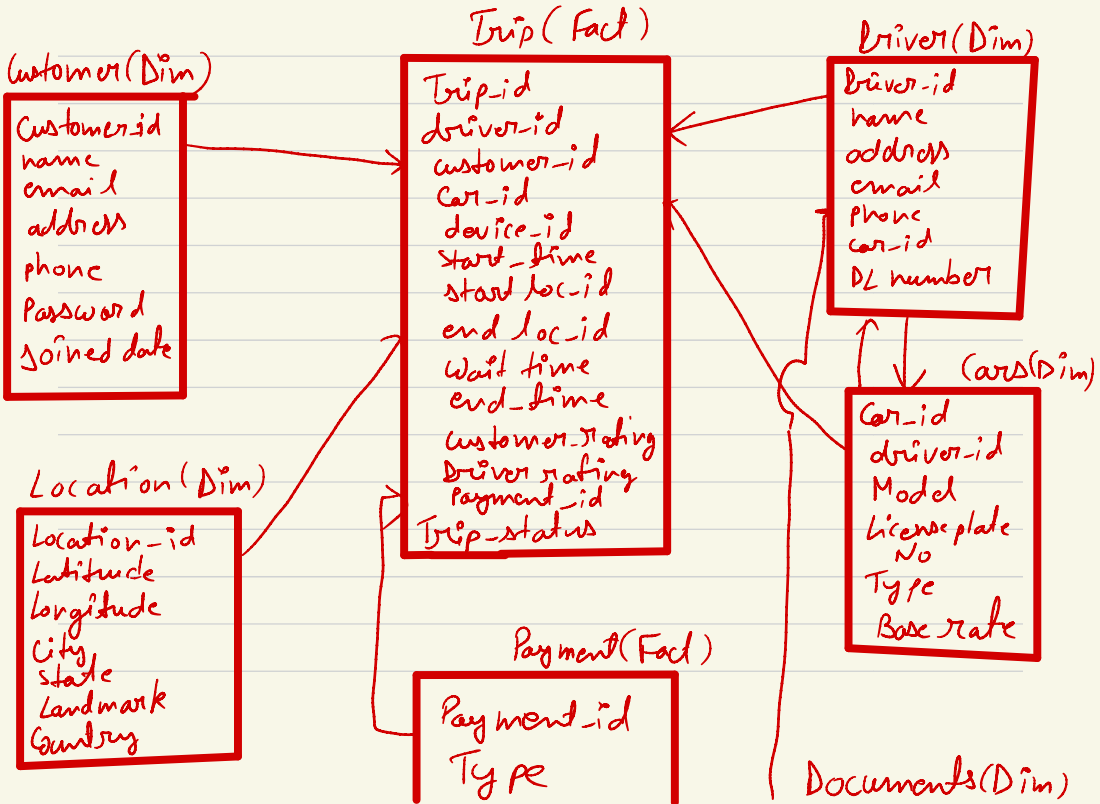
1) Business Process:- Taxi company would like to design a data model to capture all critical data elements.
   - Track rides done by driver & their performance
   - How many rides are happening to a common or famous destination each day

- How many trips cancelled per day
- How many rides and the average price during the peak hour per day

② Grain - Individual trip on each transaction level

③ Dimensions - Date, Customers, Drivers, Cars, Documents, Devices, locations

④ Facts - Trips, Payments



Base rate
Surge rate
Trip Amount
Total Amount
Transaction_id

Doc_id
Name
driver_id
Category
Code
Country
expiry-date