Data Science Tools
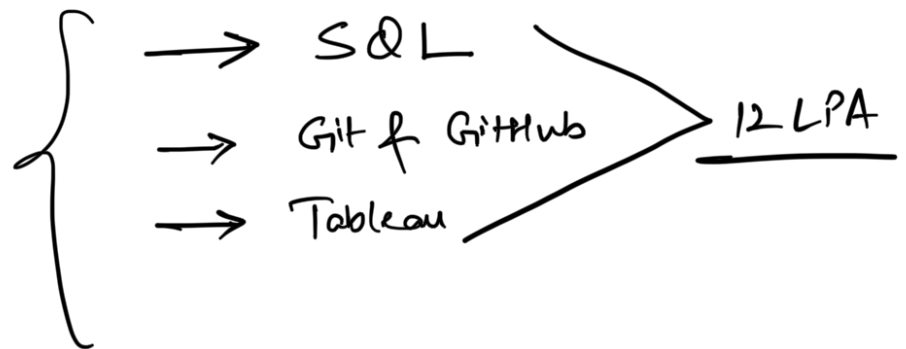
- → SQL
- → Git & GitHub   } 12 LPA
- → Tableau

Agenda : Intro. to DB & SQL

Data Scientist / Analyst at Reliance fresh

Manager

① ⟶ Database ✓ (DBMS) ?
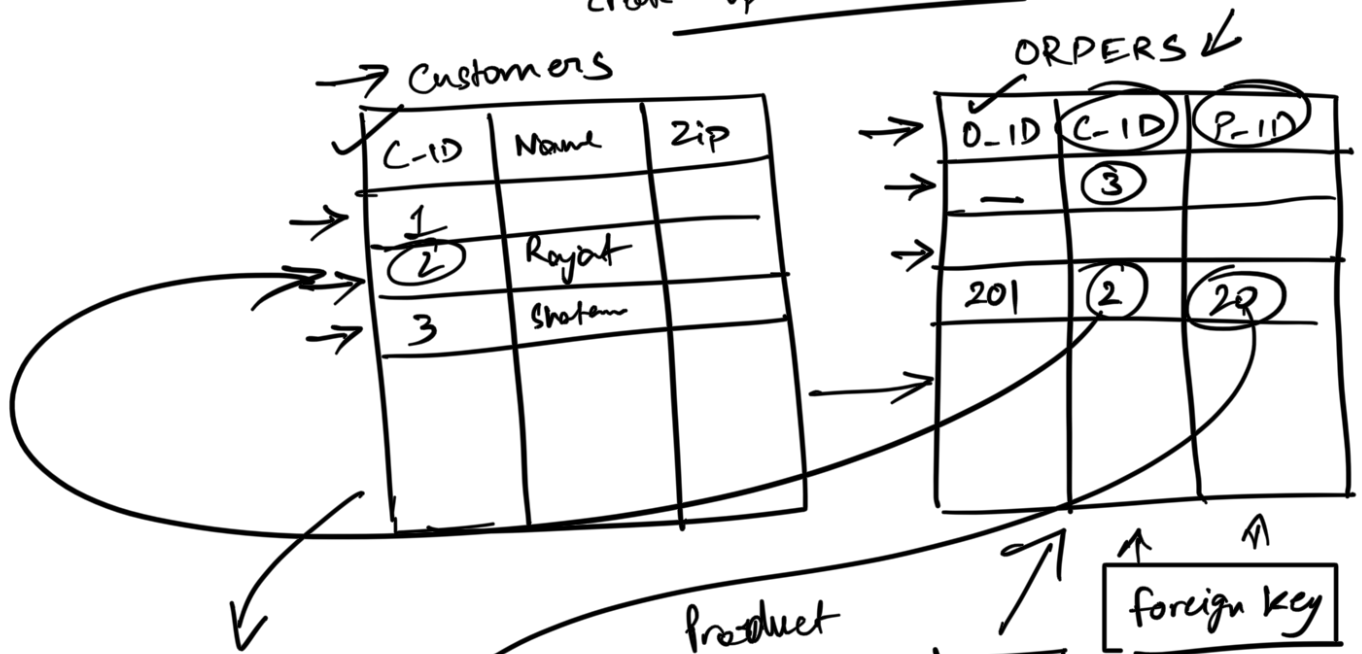
② ⟶ Schema ✓

D B , M S ,

→ Database — A collection of inter-related tables.

→ Management System — A set of operations to manage the data in the DB

search

→ CRUD ⟶ Reset

Create Update Delete

Customers

C-ID	Name	Zip
1		
2	Royat	
3	Shatam	

ORPERS

O-ID	C-ID	P-ID
—	3	
201	2	20

foreign key

Tables
↓
Relations

Product

P-ID	Name	
20	iphone	

Relational DBMS

RDBMS (✓)

→ Tabular Format

{ → MySQL
  → PostgreSQL
  → SqLite }

NRDBMS

→ Document
→ Graph }

NoSQL

Mongo DB
Cassandra

market_date_info
- 🔑 market_date DATE
- ○ market_day VARCHAR(45)
- ○ market_week VARCHAR(45)
- ○ market_year VARCHAR(45)
- ○ market_start_time VARCHAR(45)
- ○ market_end_time VARCHAR(45)
- ○ special_notes BLOB
- ○ market_season VARCHAR(45)
- ○ market_min_temp VARCHAR(200)
- ○ market_max_temp VARCHAR(45)
- ○ market_rain_flag INT(11)
- ○ market_snow_flag INT(11)

Indexes

vendor_booth_assignments
- vendor_id INT(11)
- booth_number INT(11)
- market_date DATE

Indexes

customer
- 🔑 customer_id INT(11)
- ○ customer_first_name VARCHAR(45)
- ○ customer_last_name VARCHAR(45)
- ○ customer_zip VARCHAR(45)

Indexes

customer_purchases
- product_id INT(11)
- vendor_id INT(11)
- market_date DATE
- customer_id INT(11)
- ○ quantity DECIMAL(16,2)
- ○ cost_to_customer_per_qty DECIMAL(16,2)
- 🔑 transaction_time TIME

Indexes

product_category
- 🔑 product_category_id INT(11)
- ○ product_category_name VARCHAR(45)

Indexes

booth
- 🔑 booth_number INT(11)
- ○ booth_price_level VARCHAR(45)
- ○ booth_description VARCHAR(255)
- ○ booth_type VARCHAR(45)

Indexes

vendor
- 🔑 vendor_id INT(11)
- ○ vendor_name VARCHAR(45)
- ○ vendor_type VARCHAR(45)
- ○ vendor_owner_first_name VARCHAR(45)
- ○ vendor_owner_last_name VARCHAR(45)

Indexes

vendor_inventory
- 🔑 market_date DATE
- ○ quantity DECIMAL(10,0)
- vendor_id INT(11)
- product_id INT(11)
- ○ original_price DECIMAL(10,0)

Indexes

product
- 🔑 product_id INT(11)
- ○ product_name VARCHAR(45)
- ○ product_size VARCHAR(45)
- product_category_id INT(11)
- ○ product_qty_type VARCHAR(45)

Indexes

1.4 B $\longrightarrow$ 1 B

400M ?

→ Population

ID	S.ID	A.ID
	1	
	2	
	3	
	4	NULL
	5	

1.4 B

Unique / Primary

Types of Relationships (Cardinality)

One — to — One

One — to — Many | Many — to — One

Many — to — Many

① **1 : 1**

Husbands

H-ID	—	—

Wives

W-ID	—	—

1 $\longrightarrow$ 1

1 N $\longleftarrow$ 1 N

1:1

1 : 1

Customer

C-ID	Adre	Email

C-profile

A-ID	PAN-D	C-ID

$$\left\{ \begin{array}{ccc} 1 & \longrightarrow & 1 \\ 1 & \longleftarrow & 1 \end{array} \right\}$$

2.

One — to — May

Product

P-ID	Nam	Desc	PC-ID

P Category

PC-ID	PC-M	

1 P $\longrightarrow$ 1 PC

Many $\longleftarrow$ 1 PC

| Many — to — One | ✓

May - to - Many

Student | Teachers

1S → May

May ← 1T

May - to- May Relationship

1V → May Products

May Vendors ← 1 Product

May - to- May

1

May
↓

One

One & only one

zero or One

zero or many

Entity - Relationship Diagram

ER Diagram
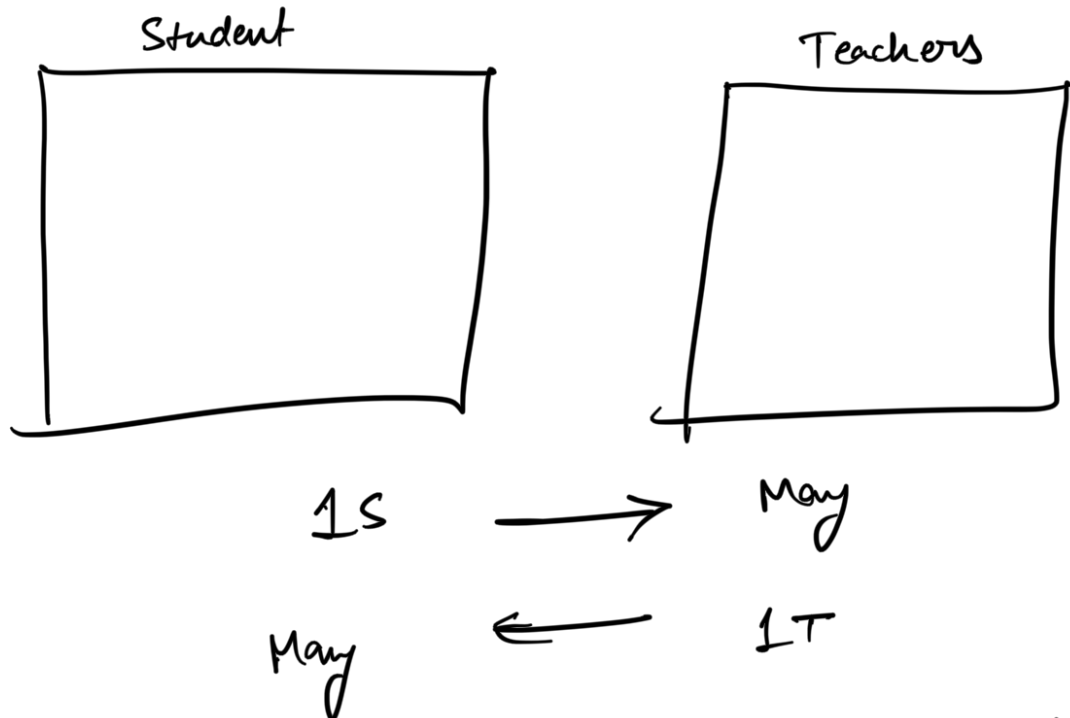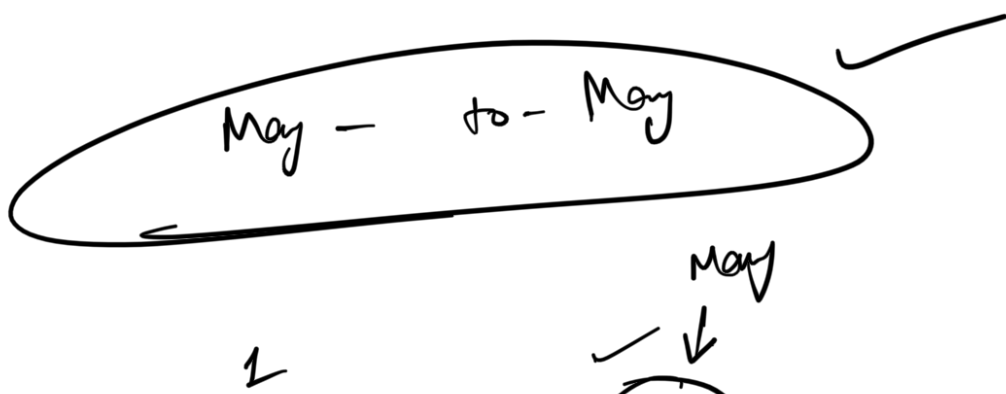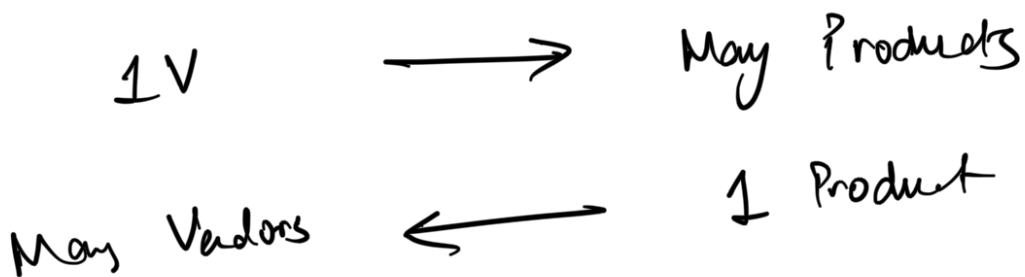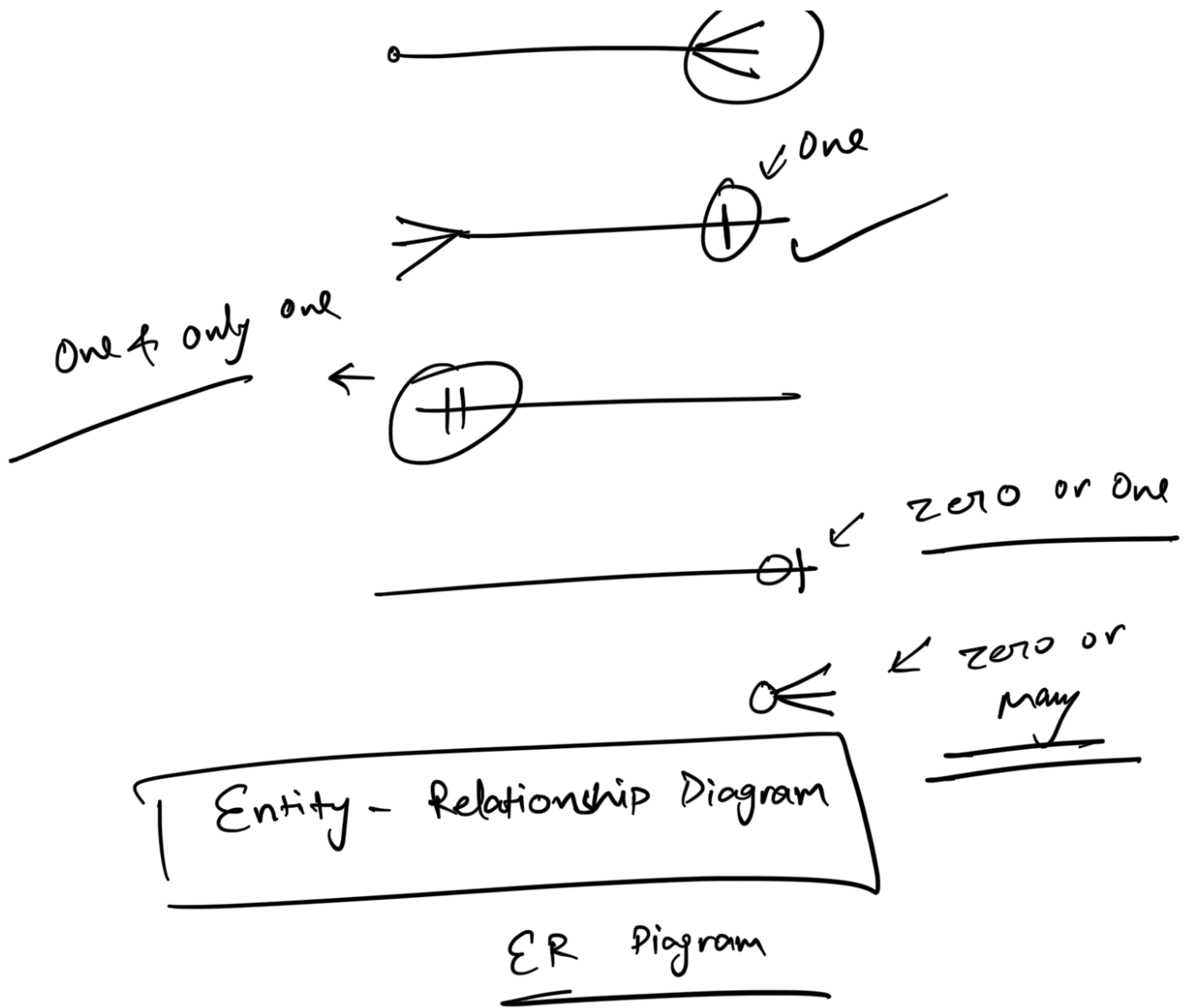
SQL → Structured Query Language

SQL

DDL	DML	TCL	DCL	DQL
Data Definition Lang.	(Manipulation)	Transaction Control	Data Control	Query
- Create	- Insert	- Commit	- Grant	SELECT
- Drop	- Update	- Rollback	- Revoke	
	- Delete			
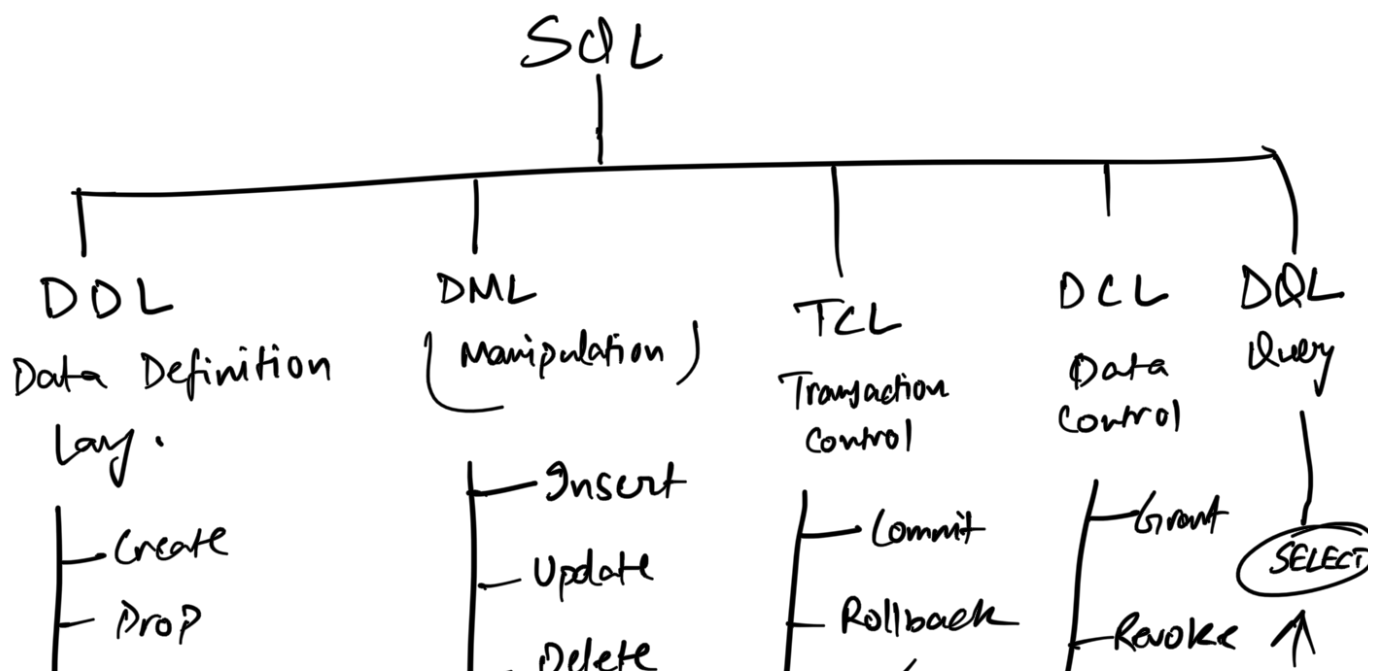
— Alter

— Truncate

V. V. Imp

Q. Get all the products available in the market.

Syntax:

DQL ✓

SELECT

```
SELECT  [ columns to return ]
FROM    [ table ]
WHERE   [ condition ]
→ ORDER BY  [____]
↓ LIMIT   [ _x_ ]
            ↑
       Top n  rows
```

SQL code that retrieves data from the selected table in the database

→ All the cols

```
SELECT  ⊛
FROM    product;
```

SQL   0.1 $
           # 10

100MB

1M

SELECT (*)

LIMIT (5)

(*)

SQL

SELECT (*)
From transaction Data

Python

(7)

(X)

(5) ← Preprocess

Dashboard

ML
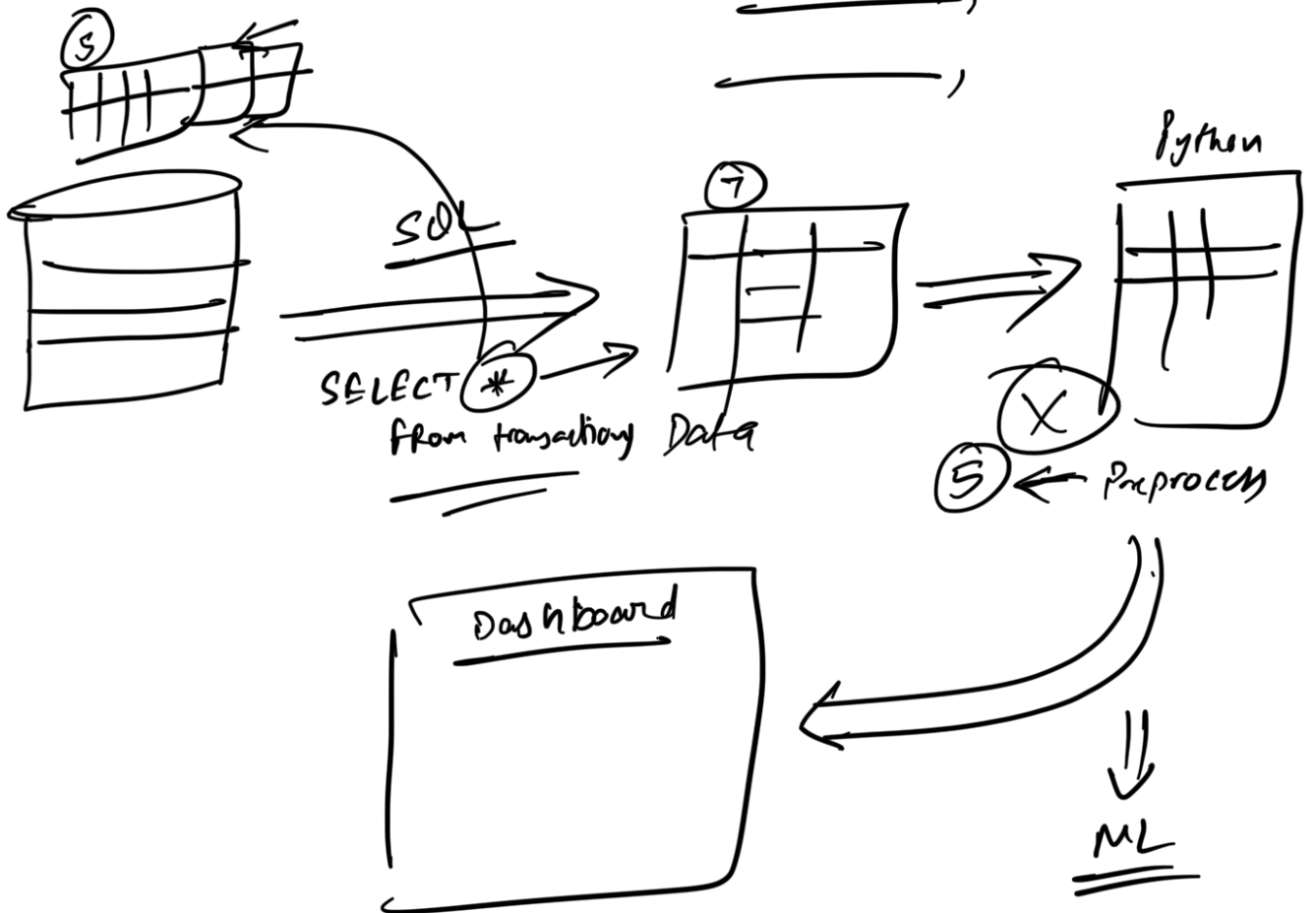
SELECT (*)

(1)
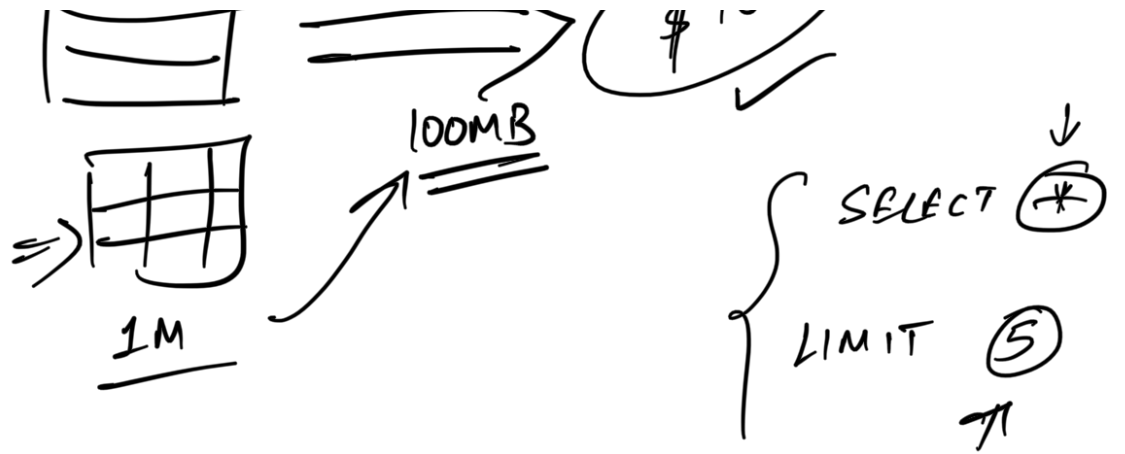
SKIP fr

custone-ID
custone-age

(2)

Explicit

Q. Extract (10) rows from VBA, display
the data → market-date,
vendor-ID, booth-number,

skip first 2 records

OFFSET

Q. Sort the o/p in prev. query by date.
from the latest date on top to
the oldest at the bottom.

↑ ↓

ORDER BY market_date DESC ↓

By default ASC

↓ ↓
ORDER BY ( ) vendor-ID, ASC

ORDER

↗ DESC   or   ↗

Left - to - Right

Employees

E-ID	Salary	

↓

ORDER BY salary

DESC

3rd highest

Q 3rd highest ←

Given that every emp
has a different salary,
find the emp with
3rd highest salary.

① | | |
② | | |
③ | | |

SKIP 2
rows

| | | |

LIMIT 1 OFSET 2

SELECT   emp-id,
          salary
FROM   employees
ORDER BY salary DESC

LIMIT 1 OFFSET 2;

SQL-02 → WHERE, fns, Subqueries
→ SQL-03 - JOINS, Correlated subquery

SQL-04 — GROUP By Aggr.   PS
SQL-05 — Window fns
SQL-06 — Date & Time / FAANG
SQL-07 — CTE, View / PS.

⇒ PS