

Agenda

→ 3NF

→ SQL

↳ DDL, DML etc

↳ CRUD

→ String/pattern matching

→ Joins

(C → DML, DDL, UPDATE TO) → DML, DDL

GROUP ID, USER ID) → VALUE

GROUP ID → GROUP
N/A

INF

→ no collections

- atomic

id | name | friends

× INF

USER - friends

id	user_id	friend_id
	1	2
	2	1

→ separate columns []

→ separate rows

↳ anomalies?

→ separate table

↓
Mapping table

PK - id

- Composite - user_id, friend_id

2NF

→ 1NF compliant

→ ^{no} partial dependencies.

<u>user_id</u>	<u>batch_id</u>	batch_name
----------------	-----------------	------------

x

u ser_id, batchid → *

batchid → batch_name

part of PK

→ non PK attribute

u ser_id | batch_id | *

↓

batch_id | batch_name

3NF

→ 1NF, 2NF

2NF → PK ✓
Composite

→ no transitive dependencies

a = b b = c

a = c

$a \rightarrow b$ $b \rightarrow c$

$a \rightarrow c$



`CTID` \rightarrow `NAME`

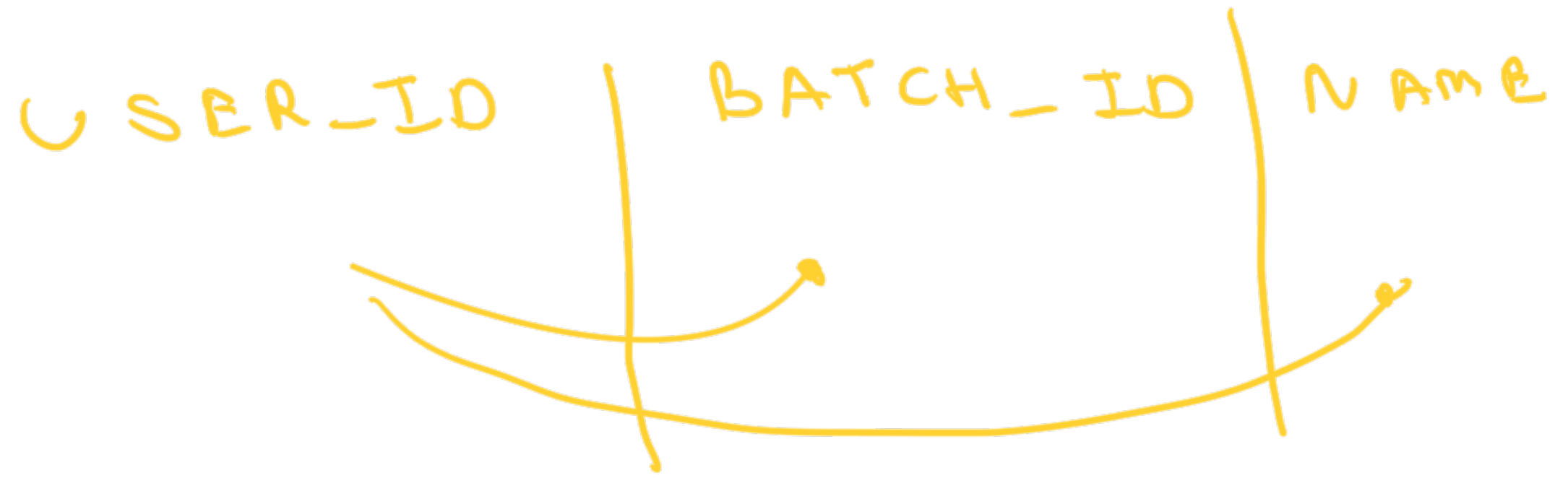
USER ID

→ BATCH ID

BATCH ID → BATCH_NAME



TFD



Batch-id

Batch-name

1NF, 2NF, 3NF

3.5NF — Boyce Codd Normal

4NF
5NF

1NF — atomic → Create a new table

1NF - atomic →
 2NF - no PD → "
 3NF - no TD → "

SQL

↳ Language to interact with
 a database (DBMS)

MySQL

Postgres

SQL

Struct

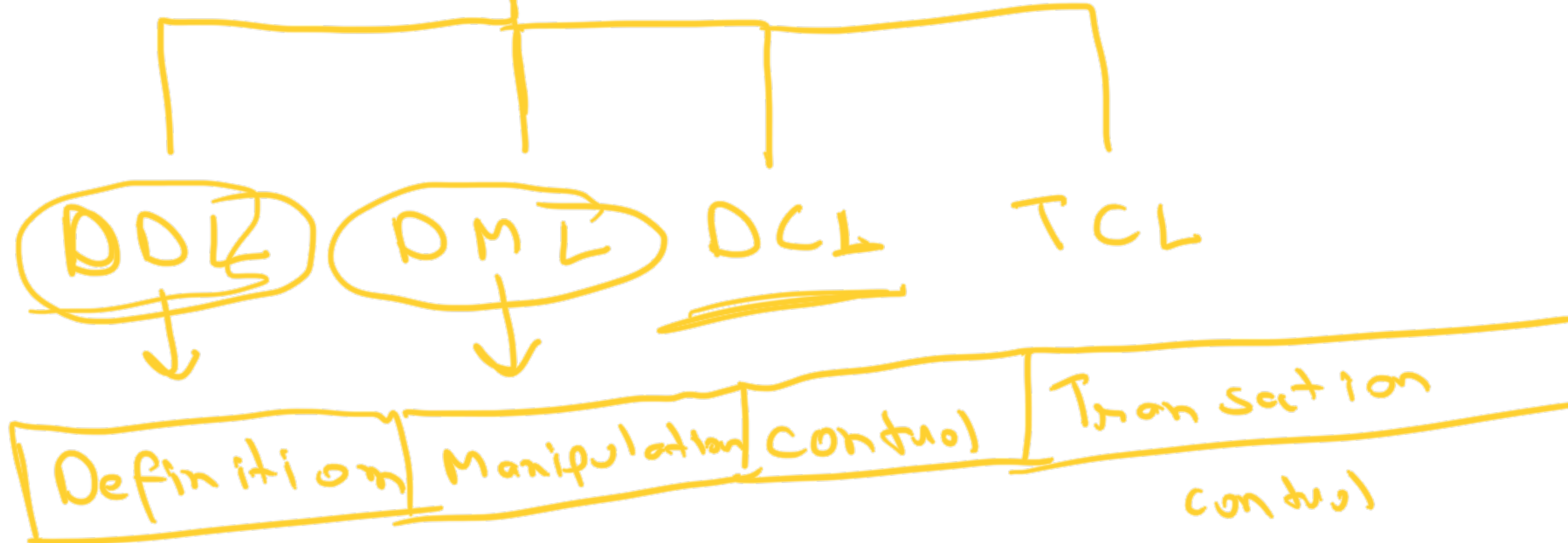
Query

Language

C - Create
 R - Read
 U - Update
 D - Delete

D X L
 ↓ ↓
 Database Language

SQL



TRANSACTION CONTROL

→ CREATE

INSERT UPDATE

→ ALTER

UPDATE REVOKE

→ DROP

SELECT

DELETE



DELETE TABLE X

DPL

DOL + DML

C	- <u>Create</u>	- <u>CREATE</u>
R	- <u>Read</u>	- <u>DESCRIBE</u> EXPLAIN

C	Update	ALTER
D	Delete	DROP

CRUD → DML

C — Insert

R /
D /
C

Insert INTO TABLE_NAME

VALUES

(' ' ' ')
↑ ↑ ↑ ↑

INSERT INTO TABLE_NAME

(id, name)

VALUES

(1, T)

NOT NULL

|

3 < NOT NULL
DEFAULT
|
ID

Update

→ UPDATE TABLE_NAME

SET

COLUMN = VALUE

UPDATE USER
SET

FNAME = "Sherlock",
LNAME = "Patel"

WHERE columnname <condition>

WHERE

id = 1

id > 1

id < 1

32

SET

COLUMN_1 = VALUE_1,

" " " "

" 2 = " ,

Delete

↳ DDL → DROP

↳ DML → DELETE

DELETE FROM

TABLE_NAME

DELETE FROM

USER
WHERE id = 1

Read - DML



SELECT ✓

FROM?

WHERE;

① INSERT

→ INTO

→ VALUES

② UPDATE

→ SET

→ WHERE (opt.)

SELECT (1);

SELECT → los
(10 * 10);

(3)

DELETE

→ FROM

--> WHERE

↓
clauses

SELECT

(1) { column_01,
column_02,

→ (*)

FROM

(2) TABLE_NAME

WHERE

③ id = 1 or id = 2

ORDER BY

IQ

DESC

ASC

Top 5 shortest

LIMIT

5;

①	②
200	10
180	20
150	30
	40

$\neq, >, <, =, \neq$

↓
! = on \Leftarrow
.

OR, AND

IN