DBMS - Introduction to SQL

Agenda

-> 3NF

-> SQL

L-> DDL, DML etc

L-> CRUD

-> String | pattern matching

-> Joins

(c_group_id , user_id) -> name

(GROUP ID, USER ID) → NAME

GROUP ID → GROUP name

1NF

→ no collections — atomic

id | name | friends ×1NF

[ ]

→ separate columns ×

→ separate rows

↳ anomalies

→ separate table

user_friends

id	user_id	friend_id
	1	2
	2	1

↓

Mapping table

PK — id

— composite — user_id, friend_id

2NF

→ 1NF compliant

→ no partial dependency.

(user_id | batch_id) | batch_name

x

user_id, batch_id → *

batch_id → batch_name

part of pk

→ non pk attribute

user_id | batch_id |*

|

batch_id | batch_name

3NF

→ 1NF, 2NF

2NF → PK ✓

→ Composite

→ no transitive dependency

a = b b = c

a = c

a → b b → c

a → c

a → b → c

USC_ID | NAME | BATCHID | BATCH_NAME

C_ID → NAME

→ BATCH ID

BATCH ID → BATCH_NAME

U ID → BATCHID → 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

S – Structured

Q – Query

L – Language

C – Create

R – Read

U – Update

D – Delete

DML

Database Language

SQL

DDL | DML | DCL | TCL

Definition | Manipulation | control | Transaction control

→ CREATE

→ ALTER

→ DROP

INSERT UPDATE REVOKE

SELECT

DELETE

DELETE TABLE X

DDL

DDL + DML

C — Create — CREATE

R — Read — DESCRIBE | EXPLAIN

U — Update — ALTER

D — Delete — DROP

CRUD → DML

C — Insert

R

U

D

Insert INTO TABLE_NAME

VALUES

(, , , ,)

INSERT INTO

(id, name)

VALUES

(1, T)

NOT NUL

3 <

NOT NULL

DEFAULT

|

ID

Update

→ UPDATE TABLE_NAME

SET

COLUMN = VALUE

UPDATE USER

SET

FNAME = "Sherlock"
LNAME = "Holmes"

WHERE columnName <condition>

WHERE

id = 1

id > 1

id < 1

SET

COLUMN_1 = VALUE_1,

Delete

↳ DDL → D ROP

↳ 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;

③ DELETE
→ FROM
--→ WHERE
↓
Clauses

SELECT
'o*'o; → los

SELECT
① { column_01,
column_02, → *

FROM
② TABLE_NAME

WHERE

(3) id = 1 or id = 2

ORDER BY

ID DESC

ASC ;

Top 5 shortest

①	②
200	10
180	20
150	30
	40

LIMIT 5 ;

=, >, <, >=, =<

!= or <>

OR, AND

IN