15/05/24

SQL 06 : JOINS - 2

AGENDA

1. OUTER JOIN → LEFT, RIGHT
2. FULL JOIN
3. CROSS JOIN , IMPLICIT
4. UNION (ALL)
5. ON VS WHERE.

Revision

JOIN (INNER JOIN)
SELF JOIN
COMPOUND JOIN ⟶ >1 ON Condition
JOIN ON MULTIPLE TABLES
USING
NATURAL JOIN

ON
^ bid = bid
[2, 7]

data type - matches - we can join.

Ques1

Table: Students

Sid	s-name	bid
1	A	1
2	B	3
3	C	NULL
4	D	NULL
5	E	2

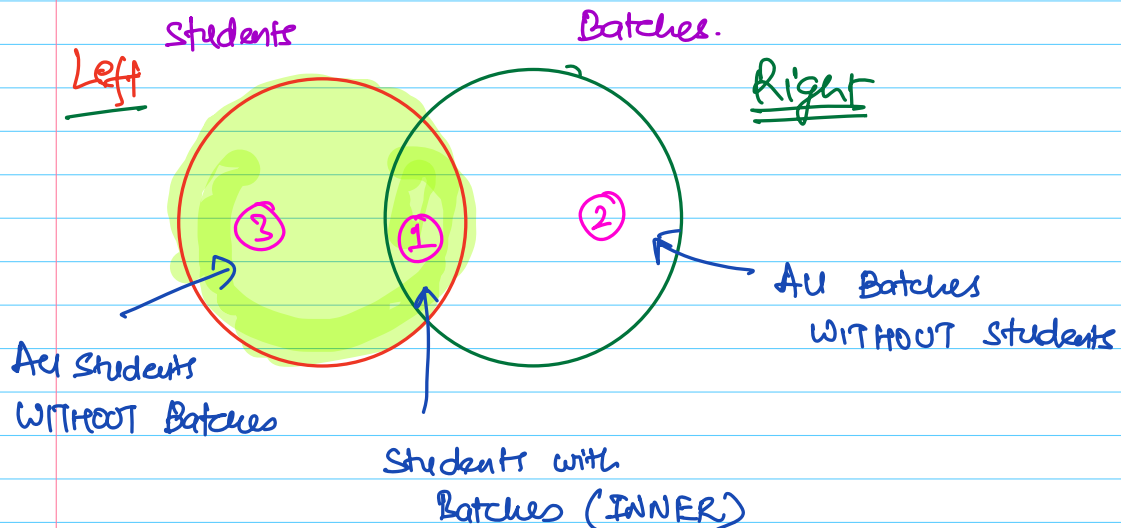
Table: Batches

bid	b-name
1	X
2	Y
3	Z

O/P

S_name	b_name
A	X
B	Z
C	NULL
D	NULL
E	Y

(join b_id)



O/P = (1) + (3)

LEFT OUTER JOIN

```
SELECT S.s_name, b.b_name.
FROM STUDENT S          <-- LEFT
LEFT JOIN BATCHES B     <-- RIGHT
  ON S.b_id = B.b_id ;
```

Q.2

Table: Students

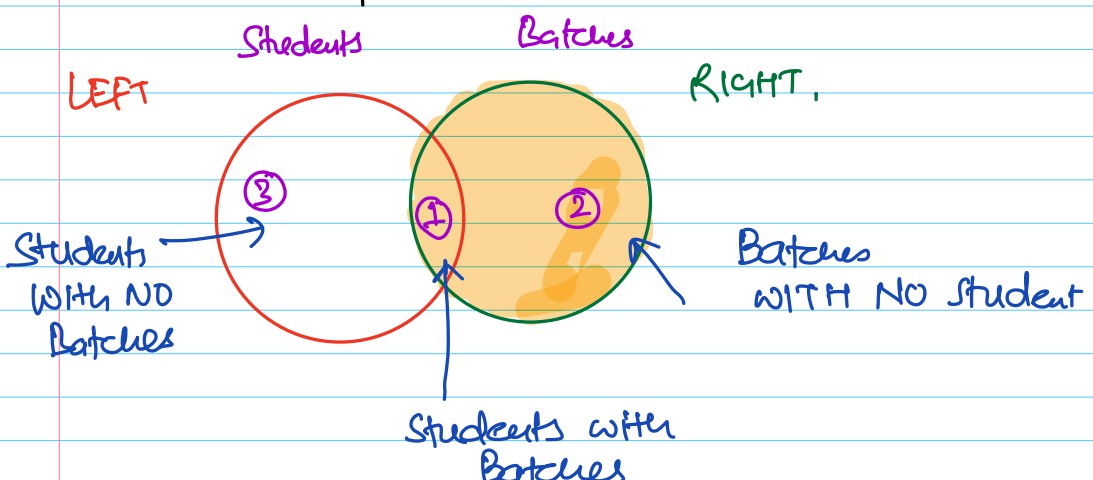
S.id	s-name	b.id
1	A	1
2	B	1
3	C	NULL
4	D	NULL
5	E	2

Table: Batches

b.id	b-name
1	X
2	Y
3	Z

O/P

S-name	b-name
A	X
B	X
E	Y
NULL	Z



O/p → 1+2

RIGHT OUTER JOIN

```
SELECT  S.s_name , b.b_name.
FROM  STUDENT S          <-- LEFT
RIGHT JOIN  BATCHES B    <-- RIGHT
  ON    S.b_id = B.b_id ;
```

```
SELECT  S.s_name , b. b_name
 FROM  BATCHES  B          <-- LEFT
 LEFT  JOIN  STUDENTS  S   <-- RIGHT.
   ON    S.b_id = B.b_id ;
```

Q.3

Table: Students

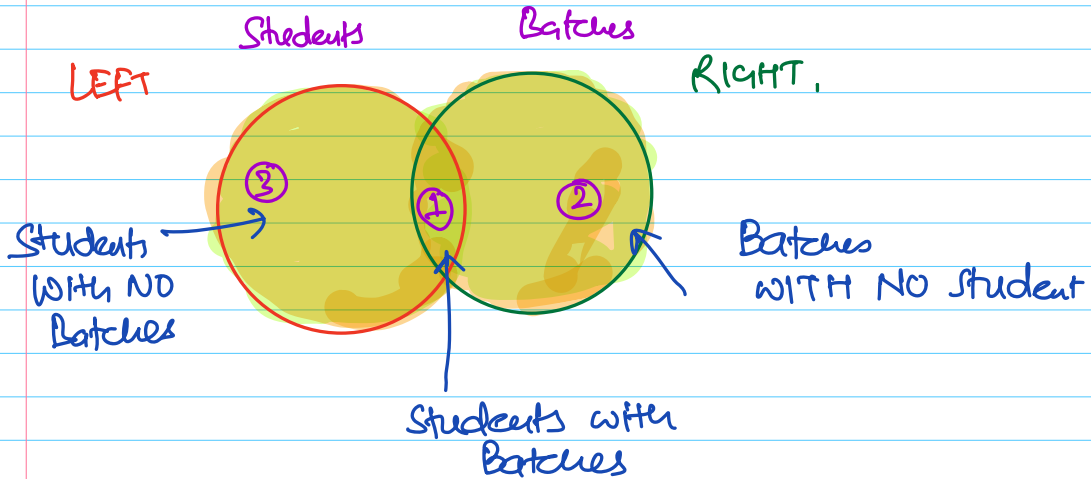
S_id	s_name	b_id
1	A	1
2	B	1
3	C	NULL
4	D	NULL
5	E	2

Table: Batches

b_id	b_name
1	X
2	Y
3	Z

O/P

S_name	b_name
A	X
B	X
C	NULL
D	NULL
E	Y
NULL	Z



① + ② + ③

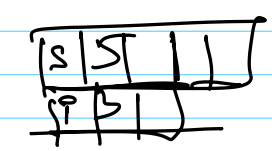
NOTE: MySQL does NOT support FULL JOIN.

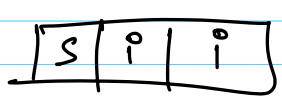
UNION

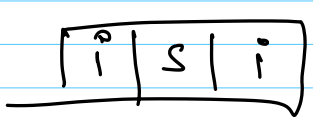
→ Combine the o/p of 2 or more SELECT stmts.

Condn: on SELECT stmts:-

1. Should have same no. of cols.
2. Data type of cols must be same.
3. Cols must be in same order (of datatype).







* Col names need NOT be same
* Cannot use ORDER BY * each select executed independently.
* DEFAULT → UNION o/ps only DISTINCT VALUE
* DUPLICATE VALUES → UNION ALL.

batches

batch_id	batch_name
1	Batch A
2	Batch B
3	Batch C

students

student_id	first_name	last_name	batch_id
1	John	Doe	1
2	Jane	Doe	1
3	Jim	Brown	null
4	Jenny	Smith	null
5	Jack	Johnson	2

L 2

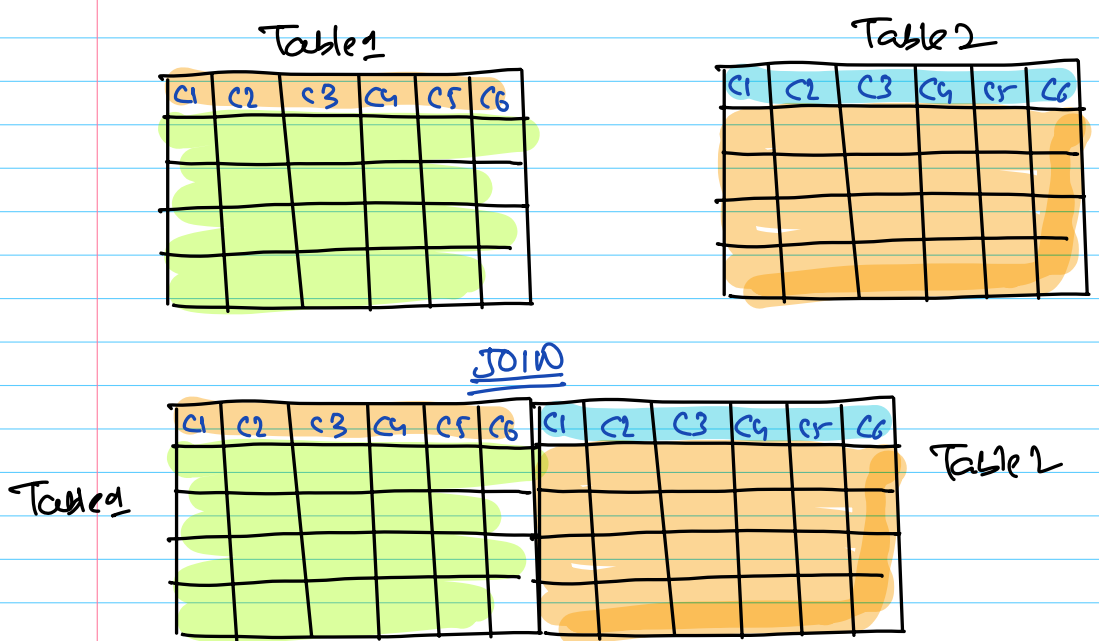
Students Batches.

JIM BROWN
JENNY SMITH

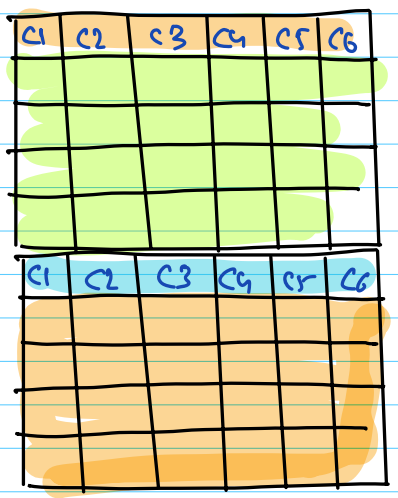
BREAK TILL – 8:17 AM.

INNER, LEFT, RIGHT ⊂ only 1.

JOIN V/S UNION



UNION



Q CLOTHING STORE

Table: SHIRTS (Design)

1
2
3
4
5
.
.
.
1000

Table: SIZES

S
M
L
XL
XXL

(5)

n x m

1000 x 5 = 5000

CROSS JOIN (Cross X product)

```
SELECT ( )
FROM STUDENT S
JOIN BATCHES B;
```

ON VS WHERE

ON

```
aus = []                                   (Sample Space)
for row1 in table1:
    for row2 in table2:                    ← JOIN ON
        if (matches ON condn):
            aus.add (row1 + row2)

filt-aus = []   (< n x m)
for row in aus:                            ← SELECT
    filt-aus.add ( row1['s.name'],
                   row2['b.name'])

print (filt-aus)                           <= n x m
```

WHERE

```
aus = []                                   ← CROSS JOIN TABLE = SAMPLE SPACE
for row1 in table1:
    for row2 in table2:
        aus.add (row1 + row2)

filt-aus = []   = u x m                    = n x m
for row in aus:
    if (row matches WHERE condn):
        filt-aus.add [row[s-name],
                      row[b-name]);

print (filt-aus)
```

WHERE:

Space - more: size of intermediary table bigger always = n x m (cross join)

Time : No. of iterations - larger - slower!

Queues →

```sql
-- SQL 06 JOINS 2
-- INNER, LEFT, RIGHT
USE SQL_030524;

SELECT S.S_ID, S.S_NAME, B.B_NAME
FROM STUDENTS S
INNER JOIN BATCHES B
ON S.B_ID = B.B_ID;

SELECT S.S_ID, S.S_NAME, B.B_NAME
FROM STUDENTS S
LEFT JOIN BATCHES B
ON S.B_ID = B.B_ID;

SELECT S.S_ID, S.S_NAME, B.B_NAME
FROM STUDENTS S
RIGHT OUTER JOIN BATCHES B
ON S.B_ID = B.B_ID;

-- SELF JOIN
SELECT S1.S_ID, S1.S_NAME, S2.S_NAME AS BUDDY
FROM STUDENTS S1
INNER JOIN STUDENTS S2
ON S2.BUDDY_ID = S1.S_ID;

SELECT S1.S_ID, S1.S_NAME, S2.S_NAME AS BUDDY
FROM STUDENTS S1
LEFT JOIN STUDENTS S2
ON S2.BUDDY_ID = S1.S_ID;


SELECT S1.S_ID, S1.S_NAME, S2.S_NAME AS BUDDY
FROM STUDENTS S1
RIGHT JOIN STUDENTS S2
ON S2.BUDDY_ID = S1.S_ID;
```

TABLES SCHEMA

```
CREATE TABLE `BATCHES` (
  `B_ID` int NOT NULL AUTO_INCREMENT,
  `B_NAME` varchar(20) NOT NULL,
  `I_ID` int DEFAULT NULL,
  PRIMARY KEY (`B_ID`)
)
```

```
CREATE TABLE `STUDENTS` (
  `S_ID` int NOT NULL AUTO_INCREMENT,
  `S_NAME` varchar(20) NOT NULL,
  `B_ID` int DEFAULT NULL,
  `BUDDY_ID` int DEFAULT NULL,
  `PSP` int DEFAULT '0',
  `IS_STUD` tinyint(1) DEFAULT NULL,
  `IS_TA` tinyint(1) DEFAULT NULL,
  PRIMARY KEY (`S_ID`)
)
```

```
CREATE TABLE `INSTRUCTORS` (
  `I_ID` int NOT NULL AUTO_INCREMENT,
  `I_NAME` varchar(20) NOT NULL,
  PRIMARY KEY (`I_ID`)
)
```