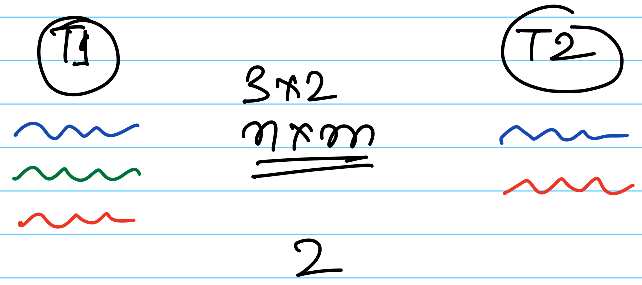
13/05/24

SQL 05: JOINS-1

AGENDA

1. JOINS
2. JOINS on multiple tables
3. Compound join
4. Self join
5. USING
6. Natural join



eg

table: students

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

5 X 3
= 15

table: batches

b_id	b_name
1	X
2	Y
3	Z

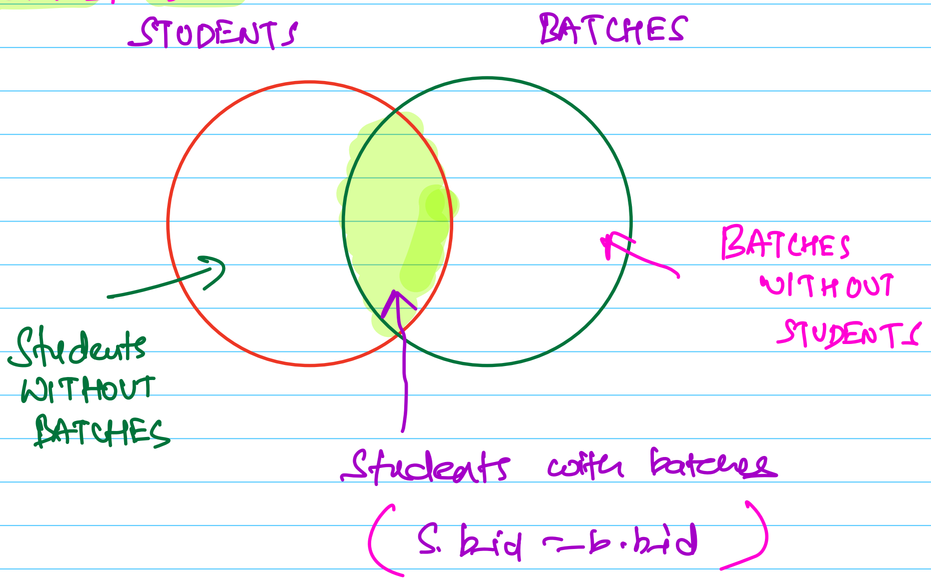
S_name	b_name
A	X
B	Z
C	Y
D	Y
E	X

Condition
b_id must be same.

students			batches	
S_id	s_name	bid	bid	b_name
1	A	1	1	X
1	A	1	2	Y
1	A	1	3	Z
2	B	2	1	X
2	B	2	2	Y
2	B	2	3	Z

```
SELECT  S.S_NAME , B.B_NAME
 FROM   STUDENTS AS S
 JOIN    BATCHES  AS B
  ON    S.B_ID = B.BID;
```

INNER JOIN



NOTE: DEFAULT IS INNER JOIN

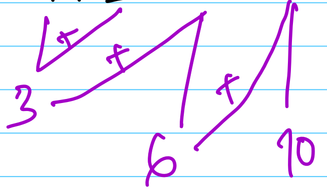
⇒ "INNER" keyword is OPTIONAL!

PSEUDO CODE

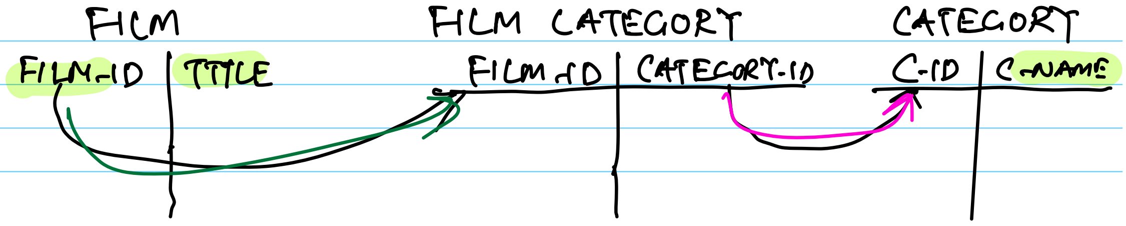
```
students, batches = [], []
ans = []

for each row1 in students :                              → FROM
    for each row2 in batches :                           → JOIN
        if ( (row1 + row2) Matches ON Condn.) :          → ON
            ans.add (row1 + row2);


filt_ans = []
for each row in ans:                                     → SELECT
    filt_ans.add (row1['s.name'],
                  row2['b_name'] )


print (filt_ans)
```

TC: $(n \times m)$

1+2+3+4+5+ . . . .



JOIN MULTIPLE TABLES



Ques table: students

s_id	s_name	buddy_id
1	A	3
2	B	5
3	C	4
4	D	1
5	E	2

S_name	Buddy_name
A	C
B	E

S1

table: students

s_id	s_name	buddy_id
1	A	3
2	B	5
3	C	4
4	D	1
5	E	2

S2

table: students

s_id	s_name	buddy_id
1	A	3
2	B	5
3	C	4
4	D	1
5	E	2

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

SELF JOIN

(INNER , LEFT , RIGHT)

↑ DEFAULT.

USING

→ Syntactic sugar.

→ If the column name is same in 2 table we can use "USING" instead of "ON" condition

eg ON s.bid = b.bid

↳ USING(b_id)

SAME.

NATURAL JOIN

→ Syntactic sugar

→ Joins on all columns with same name.

```
ON  t1.c1 = t2.c1
AND t1.c2 = t2.c2
AND t1.c3 = t2.c3
        .
        .
        .
```

Revision

1. JOIN (INNER)
2. Multiple Tables
3. Self Join
4. using
5. Natural Join

NEXT CLASS

1. OUTER → LEFT
   → RIGHT
2. FULL JOIN
3. CROSS JOIN,
4. Implicit cross join
5. UNION – (UNION ALL)
6. ON vs WHERE .

QUERIES RUN IN CLASS

```sql
-- SQL 05 JOINS 1

USE SQL_030524;

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

-- NEXT CLASS ON VS WHERE DIFF
SELECT S.S_NAME, B.B_NAME
FROM STUDENTS S
JOIN BATCHES B
WHERE S.B_ID = B.B_ID;

--

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

-- BATCH NAMES OF ALL STUDENTS WITH S_ID [2,7]
-- COMPOUND JOIN (>1 CONDITION)
SELECT S.S_ID, S.S_NAME, B.B_NAME
FROM STUDENTS S
JOIN BATCHES B
ON S.B_ID = B.B_ID
AND S.S_ID BETWEEN 2 AND 7
ORDER BY S.S_ID;


-- Q PRINT STUDENTS NAME, BATCH NAME,
INSTRUCTOR NAME
SELECT S.S_ID, S.S_NAME, B.B_NAME, I.I_NAME
```

```
-- Q PRINT STUDENTS NAME, BATCH NAME, INSTRUCTOR NAME
SELECT S.S_ID, S.S_NAME, B.B_NAME, I.I_NAME
FROM STUDENTS S
JOIN BATCHES B
ON S.B_ID = B.B_ID
JOIN INSTRUCTORS I
ON B.I_ID = I.I_ID;


SELECT *
FROM STUDENTS S
JOIN BATCHES B
JOIN INSTRUCTORS I;

-- PRINT THE FILM ID , TITLE, CATEGORY, S0RTED BY FILM ID
-- ANS -
-- 1 ACADEMY DINOSAUR DOCUMENTARY
-- 2 ACE GOLDFINGER HORROR

USE SAKILA;

SELECT F.FILM_ID, F.TITLE, C.NAME
FROM FILM F
JOIN FILM_CATEGORY FC
ON F.FILM_ID = FC.FILM_ID
JOIN CATEGORY C
ON C.CATEGORY_ID = FC.CATEGORY_ID
ORDER BY F.FILM_ID;

-- BREAK TILL 8:08 AM

-- SELF JOIN

USE SQL_030524;

SELECT S1.S_ID, S1.S_NAME, S2.S_NAME AS BUDDY
FROM STUDENTS S1
JOIN STUDENTS S2
```

```
ON S2.BUDDY_ID = S1.S_ID;

-- USING
SELECT S.S_ID, S.S_NAME, B.B_NAME, I.I_NAME
FROM STUDENTS S
JOIN BATCHES B
USING(B_ID)
JOIN INSTRUCTORS I
USING (I_ID);

-- NATURAL JOIN
-- IT WILL JOIN ON ALL COL WITH SAME NAME (AND)
SELECT S.S_ID, S.S_NAME, B.B_NAME
FROM STUDENTS S
NATURAL JOIN BATCHES B;

SELECT S.S_ID, S.S_NAME, B.B_NAME, I.I_NAME
FROM STUDENTS S
NATURAL JOIN BATCHES B
NATURAL JOIN INSTRUCTORS I;

-- PRINT THE TITLE OF MOVIE, ALL THE ACTORS IN MOVIE,
-- RELEASE YEAR, LANGUAGE

-- ANS -
-- FILM -- FILM ACTOR - ACTOR
-- FILM -- LANGUAGE (LANGUAGE_ID)

-- 1 ACADEMY DINOSAUR PENELOPE 2006 ENGLISH
-- 1 ACADEMY DINOSAUR CHRISTIAN 2006 ENGLISH

USE SAKILA;
SELECT F.TITLE, A.FIRST_NAME, A.LAST_NAME,
L.NAME, F.RELEASE_YEAR
FROM FILM F
JOIN FILM_ACTOR FA
USING (FILM_ID)
JOIN ACTOR A
USING (ACTOR_ID)
JOIN LANGUAGE L
USING(LANGUAGE_ID);


-- doubts
-- NATURAL JOIN
-- JOIN ON ALL MATCHING COLUMN NAMES (AND)

-- FILM AND FILM_ACTOR
-- MATCHING COL NAMES
-- FILM ID
ON F.FILM_ID = FA.FILM_ID AND
F.LAST_UPDATE = FA.LAST_UPDATE
```