Agenda

JOINS

→ important concept

Instructor

iid	Name	LinkedIn	
7	Prateek		

[classic models DB]

↑ Download .sql Script & execute it

Example

Student + Batch

sid	first	last	batch_id
1			
2			
3	Rohan	Singh	5

bid	batchname	iid
1	DSA	
2	ML	
5	SQL	7

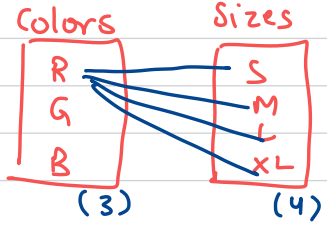
Welcome Rohan, current doing SQL

Combining data from 2 or more tables

→ Query 1 table
→ Query combine data from multiple tables.

O(N)
O(N)
O(N²)

```
SELECT *
FROM students
JOIN batches
ON students.batch_id
   = batches.batch_id
```




```
for s in students:
    for b in batches:
        if ( s.batch-id = b.batch-id )
            ans.append(s,b);
return ans;
```

CROSS JOIN

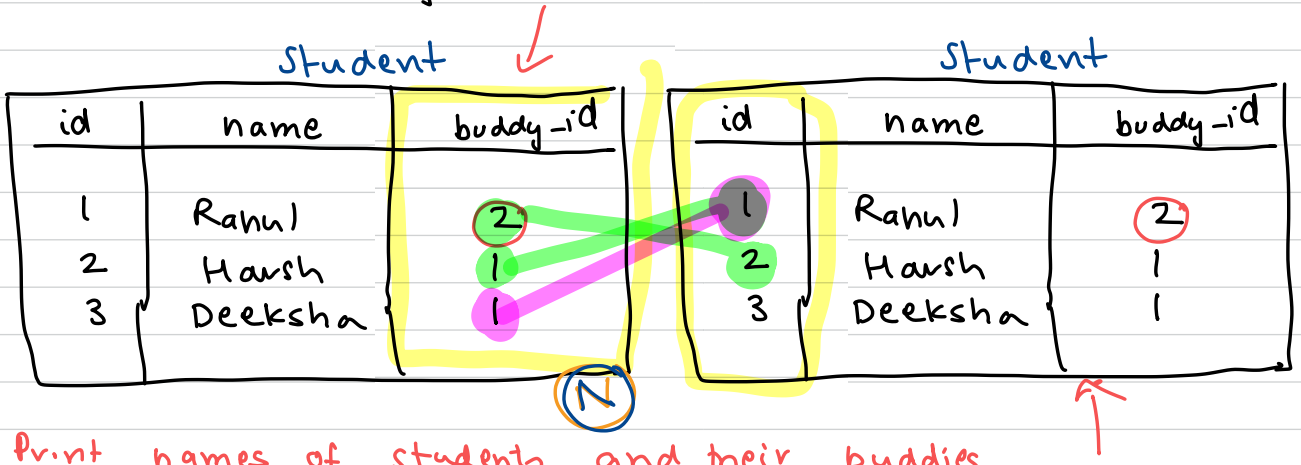
```
SELECT *
FROM colors
JOIN sizes;
```

12 Rows

R-S	G-S	B-S
R-M	G-M	B-M
R-L	G-L	B-L
R-XL	G-XL	B-XL

SELF-JOIN

→ Join a table itself.



Print names of students and their buddies

⇒

Student	Student
Rahul →	Harsh
Hash →	Rahul
Deeksha →	Rahul

AS studentName

```
SELECT   s1.name , s2.name as buddyName
FROM Student s1
JOIN Student s2
ON       s1.buddy_id = s2.id
WHERE    s2.name = "Prateek"
```

O(N)

A

O(N²)

B

JOIN

```
ans = []
for row1 in student s1:
    for row2 in student s2:
        if row1.buddy_id = row2.id
            ans.append(row1 + row2)
```

ON

```
output = []
```

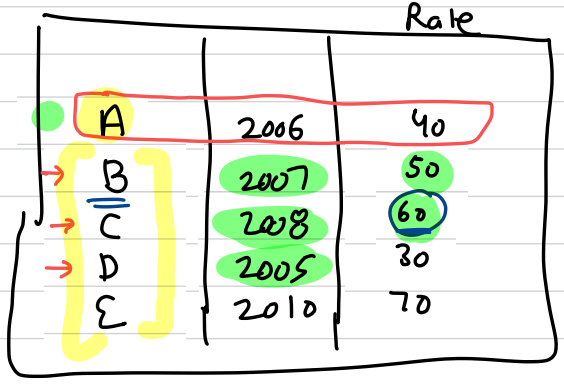
WHERE

SELECT

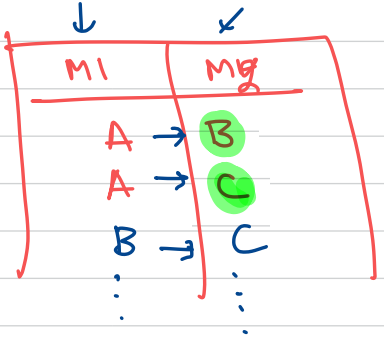
```
for row in ans:
    if (row.buddyname = Prateek)      # WHERE
        output.append(row)

print(output)                         # SELECT
```

-- TODO: Sakila Database
-- For every film, name all the films that were release
-- in the range of 2 years +- of that film
-- and there rental rate was more than the rate of the
-- of given film



B → (2005 – 2009) ~~A, C, D~~
A → (2004 to 2008)



Self-Join

```
SELECT  *
FROM    film  f1
JOIN    film  f2
ON      f2.release_year BETWEEN f1.year_rel - 2 AND f1.year_rel + 2
AND     f2.rental_rate >= f1.rental_rate
AND     f1.film_id != f2.film_id,
```