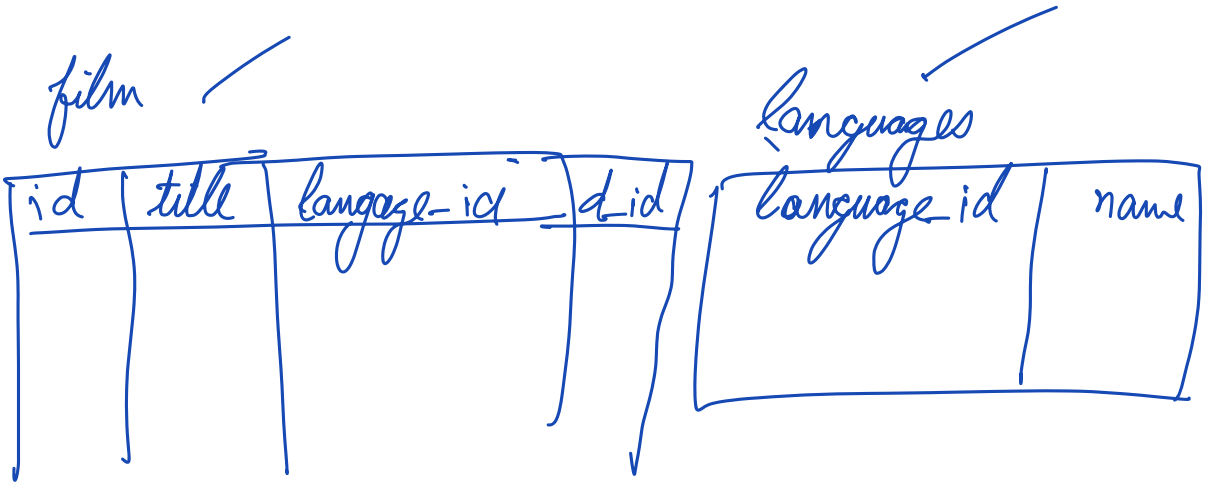
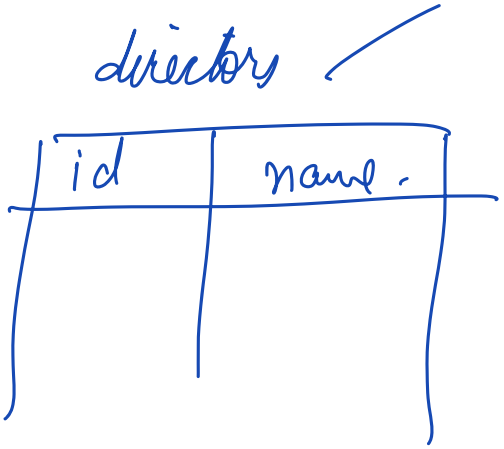
Class will start at 9:05 PM

Agenda

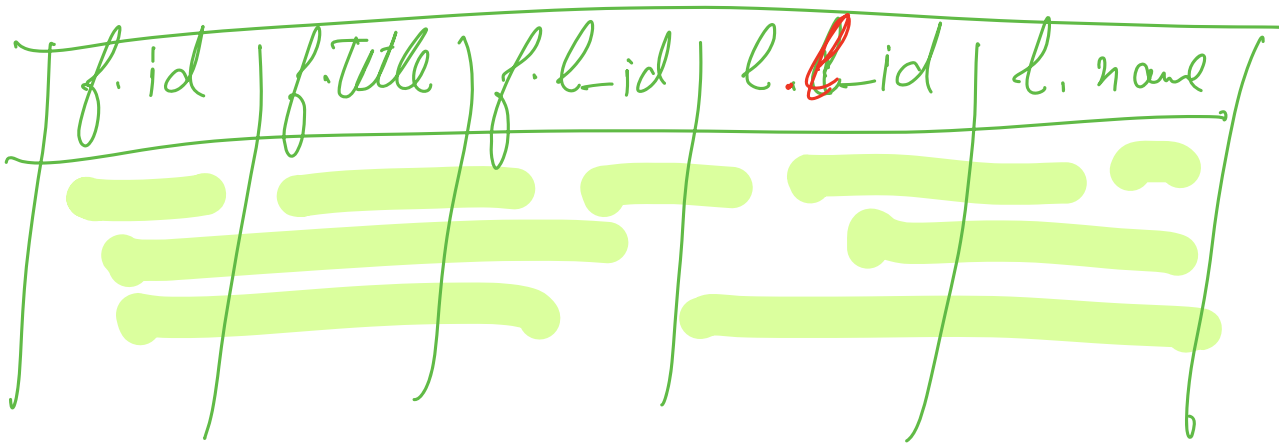
- Join with multiple table
- conditional join
- Type of Join
  - Cross Join
  - USING
  - NATURAL
  - IMPLICIT JOIN
  - Join with where vs ON
  - Union.





intermediate

```
select f.title, l.name, d.name
from film f
join languages l
on f.language_id = l.language_id
join directors d
on f.d_id = d.id;
```



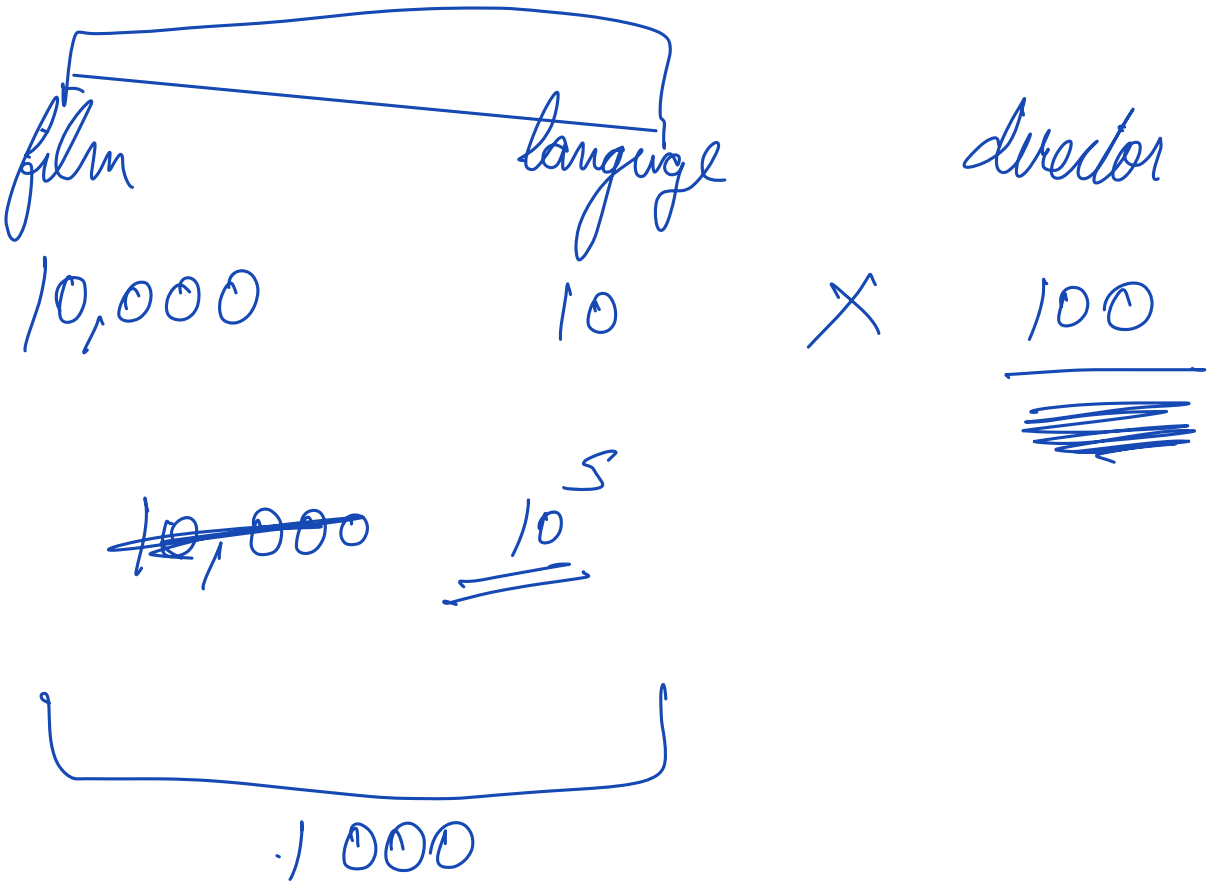
Pseudo code

ans = []

1000 [for row1 in film
  for row2 in languages
    if (film.l_id = languages.l_id)
      ans.add(row1 + row2)

for row in ans
  [for row3 in directors
    if (row.f.d_id = row3.d_id)
      ans.add(row + row3);

```
for row in ans
    print (row.f.title , row.l.name, row.d.name)
```



In case of multiple tables in join, the order of tables matters.
Try to put the join which reduces the no of rows first.

→ In practice SQL engine does this optimization for you.

Compound joins

Multiple condition in ON clause.

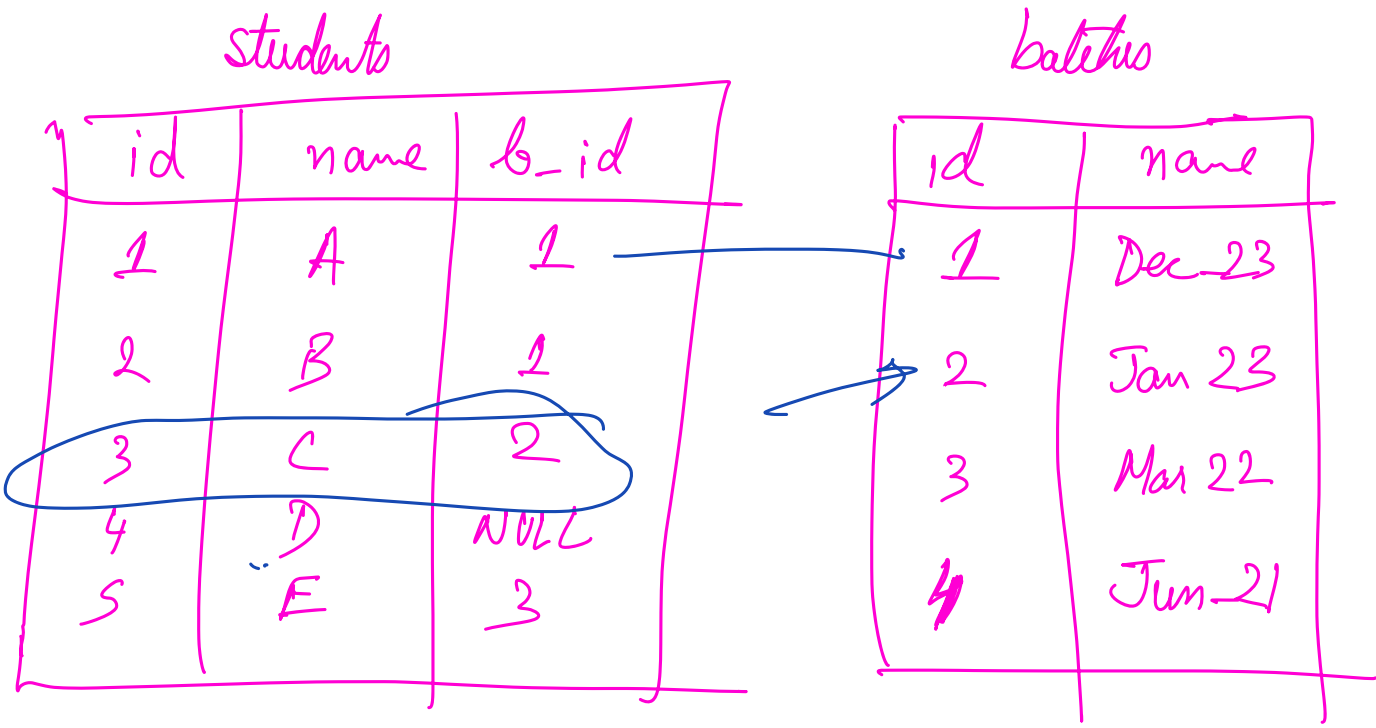
Q) for every film, name all the films that are released in range of 2 years from that film and their rental rate was more than the rate of movies.

film

title	release_year	rental
A	2006	10
B	2008	11
C	2008	9
D	209	10

A B

```
select f1.name, f2.name
from film f1
join film f2
on (f2.year - f1.year = 2)
and f2.rental > f1.rental;
```




```
select s.name, b.name
from students s
join batches b
on s.b_id = b.id;
```

s.name	b.name
A	Dec 23
B	Dec 23
C	Jan 23
E	Mar 22

```
select s.name, b.name
from students s
left join batches b
on s.b_id = b.id.
```

```
select s.name, b.name
from batches b
right join students s
on s.b_id = b.id.
```

Students join Batches.

s.id	s.name	s.b_id	b.id	b.name
1	A	1	1	Dec 23
2	B	1	1	Dec 23
3	C	2	2	Jan 23
5	E	3	3	Mar 22
4	D	NULL	NULL	NULL

Students

id	name	b_id
1	A	1
2	B	1
3	C	2
4	D	NULL
5	E	3

Batches

id	name
1	Dec 23
2	Jan 23
3	Mar 22
4	Jun 21

Left :→ Join
Join + all rows of left table

Right → Join
Join + all rows of right side table.

OUTER JOINS

- Left Join a Left outer Join
- Right Join, or Right " "
- FULL Join, Full " "

Students

id	name	b_id
1	A	1
2	B	1
3	C	2
4	D	NULL
5	E	3

Batches

id	name
1	Dec_23
2	Jan 23
3	Mar 22
4	Jun_21

```
select *
from students s
right join batches b
on s.b_id = b.id.
```

s.id	s.name	s.b_id	b.id	b.name
1	A	1	1	Dec23
2	B	1	1	Dec 23
3	C	2	2	Jan23
5	E	3	3	Mar22
NULL	NULL	NULL	4	Jun21

FULL OUTER Join

S. id	S. name	S. b_id	b. id	b. name
1	A	1	1	Dec 23
2	B	1	1	Dec 23
3	C	2	2	Jan 23
5	E	3	3	Mar 22
4	D	NULL	NULL	NULL
NULL	NULL	NULL	4	Jun 22

```
select *
from student S
full join Batches b
on s.b_id = b.id;
```

batches

batch_id	batch_name
1	Batch A
2	Batch B
3	Batch C

students

student_id	f_name	l_name	b_id
1	John	Doe	1
2	Jane	Doe	1
3	Jim	Brown	NULL
4	Jenny	Smith	NULL
5	Jack	Johnson	3

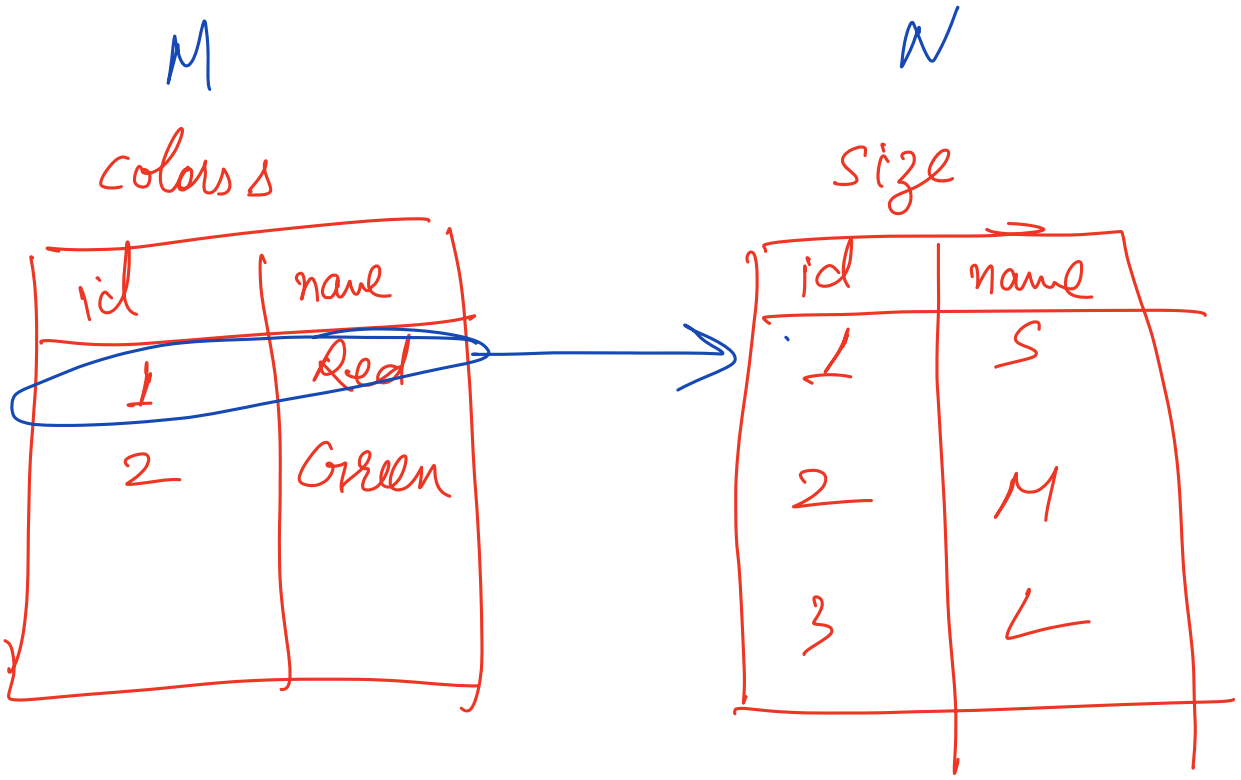
1, John, Doe, 1

3, Jim, Brow, null

5, Jack, Johnson, 2

None of the above

Class will resume at 10:30 PM



Red S

Red M

Red L

Green S

Green M

Green L

m x n

```
select c.name, s.name
from colors c
cross join sizes s;
```

C. Id	C. name	S. id	S. name
1	Red	1	S
1	Red	2	M
1	Red	3	L
2	Green	1	S
2	Green	2	M
2	Green	3	L