use sakila;

-- 1 Compound Joins

drop table students;

create table students (

id int PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(50),

birth\_year int,

psp int

);

insert into students(

name, birth\_year, psp

) VALUES

("Naman", 1996, 80),

("Deepak", 1998, 85),

("Mohit", 1997, 70),

("Dilip", 1996, 90);

-- for every student, give me those students,

-- who are 2 years older/younger to this student,

-- and their psp is more than this student.

select s1.name as s1\_student, s2.name as s2\_student

FROM students s1

JOIN students s2

ON s2.birth\_year >= s1.birth\_year-2

AND s2.birth\_year <= s1.birth\_year+2

AND s2.psp > s1.psp;

-- 2 how many rows will be here, with IS NULL condition.

drop table students;

drop table batches;

create table batches(

id int PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(50)

);

insert into batches(

name

) VALUES

("MAY22"),

("NOV22"),

("APR22"),

("DEC22");

select \* from batches;

create table students (

id int PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(50),

batch\_id int,

FOREIGN KEY (batch\_id)

REFERENCEs batches(id)

);

insert into students(

name, batch\_id

) VALUES

("Naman", 1),

("Deepak", 1),

("Mohit", null),

("Dilip", null),

("Jack", 2);

select \* from students;

select \*

FROM students

JOIN batches

ON students.batch\_id = batches.id OR students.batch\_id IS NULL;

-- 3 inner join

select \*

from film\_actor fa

join film f

on fa.film\_id=f.film\_id;

select fa.film\_id, f.film\_id

from film\_actor fa

join film f

on fa.film\_id=f.film\_id;

-- left join

-- you will get those film\_actors who dont have any films.

select \*

from film\_actor fa

left outer join film f

on fa.film\_id=f.film\_id;

-- right join

-- you will get those films who dont have any film\_actors.

select \*

from film\_actor fa

right outer join film f

on fa.film\_id=f.film\_id;

-- full outer join

-- not supported in mysql 8.

-- select \*

-- from film\_actor fa

-- full outer join film f

-- on fa.film\_id=f.film\_id;

-- distinct of left join and right join

-- union of left join and right join - inner join

-- quizes

-- 4. cross join

select count(\*) from film cross join language;

-- film has 1006 rows, language has 6 rows,

-- so total 6036

-- 5. using

select \*

from film

join film\_actor

using (film\_id);

-- behind the scenes its the same as film.film\_id = film\_actor.film\_id

select \*

from film

join film\_actor

using (film\_id, xyz);

-- behind the scenes its the same as

-- film.film\_id = film\_actor.film\_id and film.xyz = film\_actor.xyz

-- using ON and WHERE in same query.

drop table students;

drop table batches;

create table batches(

id int PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(50)

);

insert into batches(

name

) VALUES

("MAY22"),

("NOV22"),

("APR22"),

("DEC22");

select \* from batches;

drop table students;

create table students (

id int PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(50),

batch\_id int,

psp int,

FOREIGN KEY (batch\_id)

REFERENCEs batches(id)

);

insert into students(

name, batch\_id, psp

) VALUES

("Naman", 1, 80),

("Deepak", 1, 90),

("Mohit", null, 100),

("Dilip", null, 80),

("Jack", 2, 90),

("Tushar", 2, 100);

select \* from students;

EXPLAIN select \*

from students s

JOIN batches b

ON s.batch\_id = b.id

AND s.psp > 90;