use sakila;

-- 1

select \* from film;

-- \* all columns

-- 2 give me only title

select title from film;

-- 3 give me title and film\_id

select title, film\_id from film;

-- 4 case insensitive

select TITLe, FILm\_iD from film;

-- do you see the heading here^ ?

-- prefer writing everything in lowercase

-- case insensitive when it comes to columns.

-- 5 call title as film\_name, film\_id as id. give alias.

select title as film\_name, film\_id as id from film;

-- 6 I want to print the rating of every film. [Q] how will I do it?

select rating from film;

-- do you see a problem. I am getting a lot of duplicates.

-- 7 print DISTINCT for the ratings.

select distinct rating from film;

-- it will go through all the rows, and get the distinct out of them.

-- 8. now I want to know the different ratings for every year.

-- in 2008 there were PG-13 and R films. in 2010 there were PG and PG-13 films. etc

-- it should print release year and the different types of rating for each year.

-- [Q] how will I do this?

-- okay so lets try to keep this simple first.

select rating, release\_year from film;

-- there are many duplicates for 2006.

-- I want only distinct ratings for each year.

-- 2006 PG should come only once.

-- so basically each of the combination should come only once.

select distinct rating, release\_year from film;

-- [Q] does this give distinct values of only ratings ...

-- ... or does it give distinct values of all the columns together?

-- its not giving distinct ratings. its not giving distinct year.

-- BUT its giving unique combination.

-- 9. is this allowed.

select distinct(rating, release\_year) from film;

-- it does not use parantheses.

-- it is not a function.

-- 10 [Q] will this query work? No.

select rating, distinct release\_year from film;

-- important

-- distinct must always be the first keyword after SELECT.

-- it cannot come anywhere else.

-- 11. its creating unique for the entire result of ..

select rating, release\_year from film;

select distinct rating, release\_year from film;

-- 12

select 1;

select "hello";

-- 13

select title, "hello" from film;

-- 14

select title, "hello" as greeting from film;

-- 15

-- 16

select title, length/60 from film;

-- 17

select title, length/60 as duration\_in\_hours from film;

-- print rounded off hour, built-in functions

select title, round(length/60) as duration\_in\_hours from film;

-- 18 can you do operations on multiple columns together?

-- first, lets look at the film table

select \* from film;

-- [Q] how many times a person can watch a movie?

-- movie is rented for a particular duration.

select title, length, rental\_duration from film;

-- [Q] will it be round?

-- rental\_duration is 11.5 hrs. movie length of two 2hrs. round will give 6. floor will give 5

select title, length, length/60 as duration\_in\_hours, rental\_duration, floor(rental\_duration/(length/60)) as times\_a\_person\_can\_watch from film;

-- I combined 2 columns, length and duration, to get the answer for a new column.

-- after all the theory, then here.

-- 19 put all city in new table

select \* from city;

drop table randomNames;

create table randomNames(

name VARCHAR(100)

);

-- [Q] how will I do this?

-- how will the sql query look which will take all the city column into the randomNames column;

Insert into randomNames (name)

select city from city;

select \* from randomNames;

-- 20. now I also want to add the title of film in the name column

-- I want to append the title of film in the randomName table.

insert into randomNames (name)

select title from film;

-- [Q] is there any mandate that the column names should match?

-- only the data types should match!

-- no need to have constraints. just get the values and put them.

-- if length is more than 100, it will truncate.

-- launch 3 QUIZES.

-- after theory of AND/OR/NOT

-- 21

select \* from film where rating = "PG-13";

-- 22

select \* from film where release\_year = 2006;

-- !=

select \* from film where release\_year != 2006;

select \* from film where release\_year <> 2006;

-- NOT

select \* from film where NOT release\_year = 2006;

-- try this

select \* from film where NOT release\_year <> 2006;

-- 23 release year 2006 and rating = PG-13

select \* from film where release\_year = 2006 AND rating = 'PG-13';

-- 24 SQL supports comparison operators.

-- in programming you do equal to by ==, in SQL its =

-- == vs =

-- < vs <

-- > vs >

-- <= vs <=

-- >= vs >=

-- != vs (!= OR <>)

-- 25 IN

select \* from film where rating = 'PG-13' OR rating = 'R';

-- I am comparing the same column, with multiple ORs.

-- so its better to use IN

select \* from film where rating in ('PG-13', 'R');

-- 26 [Q] what will this query do?

select \* from film where rating in ('PG-13', 'R') AND release\_year=2006;

-- [Q] what will this query do

select \* from film where rating in ('PG-13', 'R') AND NOT release\_year=2006;

-- 27 [Q] give me movies that have rating neither PG-13 nor R

select \* from film where (rating != 'PG-13') and (rating != 'R');

-- which query will give me rating = PG-13 OR rating = R

select \* from film where rating in ('PG-13', 'R');

-- we want to do just the opposite

select \* from film where not (rating in ('PG-13', 'R'));

-- we write NOT A = B as A != B

-- so NOT A in B can also be written as A not in B

select \* from film where rating not in ('PG-13', 'R');

-- 28 [Q] uasge of between

select \* from film where release\_year between 2006 and 2010;

-- 29 like

select \* from film where title like '%love%';

-- 30 just the word love

select \* from film where title like '% love' OR 'love %' OR '% love %';

-- students between

drop table students;

create table students(

name VARCHAR(100),

psp INT

);

insert into students (name, psp) values

("Saharsh", 55),

("Rohan", 65),

("Mohit", 75),

("Naman", 85),

("Deepak", 95);

select \* from students;

select \* from students where (psp between 50 and 70) OR (psp between 80 and 85);