SQL Practice Notes

-- what is SQL?

-- Structured Query Language

-- What is relational data base management system?

-- What is data base and why we are using it?

-- collection discrete or continous data

-- different types database languages

-- -----------------------------------------

-- 1.DDL Data Defination Language

-- CREATE, ALTER, DROP, TRUNCATE

-- -----------------------------------------

-- 2.DML Data Manipulation Language

-- INSERE, UPDATE, DELETE

-- -----------------------------------------

-- 3.DQL Data Query Language

-- SELECT

-- ------------------------------------------

-- 4.DCL Data Control Language

-- GRANT, REVOKE

-- -------------------------------------------

-- 5.TCL Trancation Control Language

-- COMMIT, ROLLBACK, SAVEPOINT, SET TRANSACTION

-- --------------------------------------------

-- 1.DDL

-- How to create a database?

-- Is SQL programming language is case sensitive? NO

create database charan\_db;

use charan\_db;

drop database charan\_db;

-- create a database sqlnotes and create a table customer

create database sql\_notes;

use sql\_notes;

create table customers(

cust\_id int(15),

first\_name varchar(20),

last\_name varchar(20),

country varchar(20),

contact\_num int(10));

select \* from customers;

drop table customers;

create table customers(

cust\_id int(15),

first\_name varchar(20),

last\_name varchar(20),

country varchar(20),

contact\_num int(10));

select \* from customers;

-- what is the diff btw drop and truncate?

-- drop deletes the entire table

-- truncate -> columns and rows will be their but no data

-- DML ----> insert,update,delete

insert into customers(cust\_id,first\_name,last\_name,country,contact\_num)

values(1,'charan','shetty','India',955014355),

(2,'sai','rggt','nepal',345466);

-- visualise the table --

select \* from customers;

-- insert 2 more elements into table

insert into customers(cust\_id,first\_name,last\_name,country,contact\_num)

values(3,'rahul','jvvg','usa',134566),

(4,'pasha','shaik','pakistan',85436);

-- if you want to display particular column of the table

select first\_name, last\_name from customers;

-- what is varchar?

-- VARCHAR is probably the most widely used data type for strings and it stores 65,000+ chharacters

-- what is int and bigint?

-- int allows less than 10 numbers

-- big int allows more than 10 numbers

-- I want to see the 1st entire row of the table

select \* from customers

where cust\_id = 1;

use sql\_notes;

-- create a table students sid sname and marks

create table students(

sid int,

sname varchar(20),

marks int);

insert into students values

(100, 'Ram', 71 ),

(101, 'Raj', 75 ),

(102, 'Rajesh', 65 ),

(103, 'Deepa', 79 ),

(104, 'Deepak', 91 ),

(107, 'Animesh', 82 ),

(108, 'Harish', 63 ),

(110, 'Hitesh', 55 ),

(120, 'Sam', 50 ),

(130, 'Shreya', 49 );

select \* from students;

-- only first row from the table

select \* from students

where sid=100;

-- filter out the students who got above 75 marks

select \* from students

where marks>=75;

-- get the data about shreya

select \* from students

where sname='shreya';

-- relational operators also called comparision operators

-- =,>,<,>=,<=,!=,<>

-- retrive the data less than 75 marks

select \* from students

where marks<75;

-- retrive the data more than 60

select \* from students

where marks>=60;

-- list out the students who got marks btw 60 and 75

select \* from students

where marks between 60 and 75;

-- logical operators ---> and ,or

select \* from students

where marks >60 and marks <75;

select \* from students

where marks <> 60 and 75;

-- i want the records of the students whose name starts with letter 's'

-- % represents the upcoming letters in any name

select \* from students

where sname like 's%'; -- % used after a letter means the names starts with that letter

select \* from students

where sname like '%s'; -- % used before a letter means the names ends with that letter

select \* from students

where sname like '%s%'; -- % used before and after then the letter may present at any position of the word

-- i don't know the 1st letter and i know the 2nd letter 'aj'

select \* from students

where sname like '\_aj%';

-- get the records of student whose name ends with 'm'

select \* from students

where sname like '%m';

-- list out the students whose name contains 'e' letter

select \* from students

where sname like '%e%';

-- i don't know the first 2 letters and i know 2nd 2 letters

select \* from students

where sname like '\_\_aj';

-- length function

-- get the record whose name lengh = 5

select \* from students

where length(sname)=5;

-- what can i write if whose number is 104 and 107

select \* from students

where sid=104 or sid=107 or sid=102;

-- 'in' function --> its an operator where you can give multiple values

select \* from students

where sid in(104,107,102);

select \* from students

where sid not in(104,107,102);

-- --------------------------------------

use sql\_notes;

create table customer(

cust\_id int primary key,

first\_name varchar(10) not null,

last\_name varchar(10)not null,

dob date,

phone varchar(12));

insert into customer values

(1,'John','Brown','1965-01-01','800-555-1211'),

(2,'Cynthia','Green','1968-02-05','800-555-1212'),

(3,'Steve','White','1971-03-16','800-555-1213'),

(4,'Gail','Black',NULL,'800-555-1214'),

(5,'Doreen','Blue','1970-05-20',NULL);

-- 1.last name starts with 'Bl'

select \* from customer

where last\_name like 'bl%';

-- 2.whose first name and last name contains 'ee'

select \* from customer

where first\_name like '%ee%' or last\_name like '%ee%';

-- 3.whose phone num ends in 12

select \* from customer

where phone like '%12';

-- 4.whose doesnt have the phone number

select \* from customer

where phone is null;

-- 5.who do not have a dob

select \* from customer

where dob is null;

-- 6.lets merge the first\_name and last\_n ame together and print customer name

select first\_name,last\_name,concat(first\_name,last\_name)

as full\_name from customer;

select \*,concat(first\_name,' ',last\_name)

as full\_name from customer;

select \* from students;

-- add new column to the table

-- DDL --> create, alter, drop, truncate

alter table students add city varchar(20);

describe students; -- describe command defines or gives all the datatypes of the table

-- how to fill or update the null values

set sql\_safe\_updates=0; -- 0 means turn off saftey in db

-- DML -->update

-- update [table name] set [column name] = ['values']

update students set city = 'chennai';

set sql\_safe\_updates=1; -- 1 means turn off saftey in db

-- what is commit and roll back?

-- is is used for DML commands and should execute immediately

insert into students values(11,'test',80,'banglore');

commit;

-- when you execute the commit you cannot rollback

select \* from students;

insert into students values(11,'ramu',80,'hyd');

rollback;

insert into students values(122,'charan',100,'hyd');

rollback;

select \* from students;

-- how to delete some rows from the table

-- DML --> insert, update, delete

delete from students where sid=11;

set sql\_safe\_updates=0;

delete from students where sid=122;

select \* from students;

set sql\_safe\_updates=1;

-- how to delete a column from a table

-- alter and drop use it to delete column from table

alter table students drop city;

-- can you add multiple columns to the table by using alter

alter table students add(city varchar(20),country varchar(20));

describe students;

-- update the null values of marks

set sql\_safe\_updates=0;

update students set marks=75;

rollback;

-- how to update the null values of the rows and columns without effecting the actual values

use sql\_notes;

create table studentss(

sid int,

sname varchar(20),

marks int);

insert into studentss values

(100, 'Ram', 71 ),

(101, 'Raj', 75 ),

(102, 'Rajesh', 65 ),

(103, 'Deepa', 79 ),

(104, 'Deepak', 91 ),

(107, 'Animesh', 82 ),

(108, 'Harish', 63 ),

(110, 'Hitesh', 55 ),

(120, 'Sam', 50 ),

(130, 'Shreya', 49 );

select \* from studentss;

alter table studentss add city varchar(20);

set sql\_safe\_updates=0;

update studentss set city = 'chennai'

where sid=100;

update studentss set city = 'hyd'

where sid=104;

update studentss set city = 'blr'

where sid=102;

update studentss set city = 'kmm'

where sid=110;

update studentss set city = 'dmp'

where city is null;

set sql\_safe\_updates=1;

use sql\_notes;

set sql\_safe\_updates=0;

update students set marks=71

where sid=100;

select \* from students;

update students set marks=75

where sid=101;

update students set marks=65

where sid=102;

update students set marks=79

where sid=103;

update students set marks=91

where sid=104;

update students set marks=82

where sid=107;

update students set marks=63

where sid=108;

update students set marks=55

where sid=110;

update students set marks=50

where sid=120;

update students set marks=49

where sid=130;

update students set city = 'hyd'

where sid in(104,107,101,110);

update students set country = 'India';

update students set city = 'kmm'

where sid in(103,100,120);

update students set city = 'dmp'

where city is null;

-- string functions in sql

select \* from customer;

-- change first\_name into lower cases

select first\_name,lcase(first\_name) from customer;

-- change first\_name into upper cases

select first\_name,ucase(first\_name) from customer;

-- concat the first\_name and the last\_name

select first\_name,last\_name,concat(first\_name,' ',last\_name) as full\_name from customer;

-- chnage this full name into upper case

select first\_name,last\_name,ucase(concat(first\_name,' ',last\_name)) as full\_name from customer;

-- trim

-- dual --> it is a default table to practice the string functions

select trim(' charan ') from dual;

-- ltrim

select ltrim(' charan ') from dual;

-- rtrim

select rtrim(' charan ') from dual;

create table products(

product\_id integer primary key,

product\_type\_id integer references product\_types(product\_type\_id),

name varchar(30) not null,

description varchar(50),

price decimal(5,2)

);

INSERT INTO products ( product\_id, product\_type\_id, name, description, price) VALUES (1, 1,

'Modern Science', 'A description of modern science', 19.95);

INSERT INTO products ( product\_id, product\_type\_id, name, description, price) VALUES (2, 1,

'Chemistry', 'Introduction to Chemistry', 30.00);

INSERT INTO products ( product\_id, product\_type\_id, name, description, price) VALUES (3, 2,

'Supernova', 'A star explodes', 25.99);

INSERT INTO products ( product\_id, product\_type\_id, name, description, price)

VALUES (4, 2,'Tank War', 'Action movie about a future war', 13.95);

INSERT INTO products ( product\_id, product\_type\_id, name, description, price)

VALUES (5, 2,'Z Files', 'Series on mysterious activities', 49.99);

INSERT INTO products ( product\_id, product\_type\_id, name, description, price)

VALUES (6, 2,'2412: The Return', 'Aliens return', 14.95);

INSERT INTO products ( product\_id, product\_type\_id, name, description, price)

VALUES (7, 3,'Space Force 9', 'Adventures of heroes', 13.49);

INSERT INTO products ( product\_id, product\_type\_id, name, description, price)

VALUES (8, 3,'From Another Planet', 'Alien from another planet lands on Earth', 12.99);

INSERT INTO products ( product\_id, product\_type\_id, name, description, price)

VALUES (9, 4,'Classical Music', 'The best classical music', 10.99);

INSERT INTO products ( product\_id, product\_type\_id, name, description, price)

VALUES (10,4, 'Pop 3', 'The best popular music', 15.99);

INSERT INTO products ( product\_id, product\_type\_id, name, description, price)

VALUES (11,4, 'Creative Yell', 'Debut album', 14.99);

INSERT INTO products ( product\_id, product\_type\_id, name, description, price)

VALUES (12,NULL, 'My Front Line', 'Their greatest hits', 13.49);

select \* from products;

-- what is substring?

-- it is a part of slicing we are going to get one value from the table

-- what is slicing?

-- getting a range of characters from the string

-- i want to print 1st letter from the name

-- syntax: substring(column name, start value , end value

select substring(name,1,1) from products;

select substring(name,1,2) from products;

select substring(name,4,2), name from products;

select substring(name,-5,2), name from products;

-- get data of last character from name of table products

select substring(name,-1), name from products;

-- use length function and print last character

select substring(name,length(name),1),name from products;

-- How to replace the values?

select name, replace(name,substring(name,1,1), lower(substr(name,1,1))) from products;

-- make the above code in a new column

select name, replace(name,substring(name,1,1), lower(substr(name,1,1))) as replaced\_name from products;

-- use same syntax and replace 2 letters with upper case

select name, replace(name,substr(name,1,2), upper(substr(name,1,2))) as replaced\_name from products;

-- instring --> instring defines the index part of a particular letter

select name, instr(name,'e') from products;

select name, instr(name,'a') from products;

select description, instr(description,'o') from products;

-- Aggregate function

-- count,max,min,avg,sum

use sql\_notes;

select \* from products;

-- print the min price from the products

select min(price) from products;

-- get the name of the product along with min price

select name, min(price) from products; -- error

select name, min(price) from products

group by name;

-- whenever you're using agg functions you have to use groupby function

-- i want to know min,max,avg,count

select min(price),max(price),count(price),avg(price) from products;

select product\_type\_id,avg(price) from products

group by product\_type\_id;

-- the product\_type\_id shouldn't be null here

select product\_type\_id,avg(price) from products

where product\_type\_id is not null

group by product\_type\_id;

-- or

select product\_type\_id,avg(price) from products

group by product\_type\_id

having product\_type\_id is not null;

-- i want avg price of product\_type\_id is not null and i want avg of product\_type\_id above 20

select product\_type\_id,avg(price) from products

where product\_type\_id is not null

group by product\_type\_id

having avg(price)>20;

-- or

select product\_type\_id,avg(price) from products

group by product\_type\_id

having avg(price)>20 and product\_type\_id is not null;

-- order when you're using groupby and aggregate function

-- select,where,groupby,having

-- arrange the above query data in decending order -- avg prices

select product\_type\_id,avg(price) from products

group by product\_type\_id

having avg(price)>20 and product\_type\_id is not null

order by avg(price) desc;

-- i want see the highest avg price of product\_type\_id only one value

select product\_type\_id,avg(price) from products

group by product\_type\_id

having avg(price)>20 and product\_type\_id is not null

order by avg(price) desc limit 1;

create table product\_types(

product\_type\_id integer primary key,

name varchar(10) not null);

insert into product\_types(product\_type\_id,name) values(1,'Book');

insert into product\_types(product\_type\_id,name) values(2,'Video');

insert into product\_types(product\_type\_id,name) values(3,'DVD');

insert into product\_types(product\_type\_id,name) values(4,'CD');

insert into product\_types(product\_type\_id,name) values(5,'Magazine');

CREATE TABLE purchases (

product\_id INTEGER

REFERENCES products(product\_id),

customer\_id INTEGER

REFERENCES customers(customer\_id),

quantity INTEGER NOT NULL,

CONSTRAINT purchases\_pk PRIMARY KEY (product\_id, customer\_id));

INSERT INTO purchases ( product\_id, customer\_id, quantity) VALUES ( 1, 1, 1);

INSERT INTO purchases ( product\_id, customer\_id, quantity) VALUES ( 2, 1, 3);

INSERT INTO purchases ( product\_id, customer\_id, quantity) VALUES ( 1, 4, 1);

INSERT INTO purchases ( product\_id, customer\_id, quantity) VALUES ( 2, 2, 1);

INSERT INTO purchases ( product\_id, customer\_id, quantity) VALUES ( 1, 3, 1);

INSERT INTO purchases ( product\_id, customer\_id, quantity) VALUES ( 1, 2, 2);

INSERT INTO purchases ( product\_id, customer\_id, quantity) VALUES ( 2, 3, 1);

INSERT INTO purchases ( product\_id, customer\_id, quantity) VALUES ( 2, 4, 1);

INSERT INTO purchases ( product\_id, customer\_id, quantity) VALUES ( 3, 3, 1);

INSERT INTO purchases ( product\_id, customer\_id, quantity) VALUES ( 4, 4, 6);

INSERT INTO purchases ( product\_id, customer\_id, quantity) VALUES ( 6, 2, 3);

select \* from product\_types;

select \* from purchases;

select \* from products;

select \* from customer;

-- what is join?

select pt.procuct\_type\_id,pt.name,p.name

from products p inner join product\_types pt

where pt.product\_type\_id = p.product\_id;

-- join the purchase table and product table based on product\_id and i want product\_id,product\_name and customer id from purchase table

select p.product\_id,p.name,ps.customer\_id

from products p join purchases ps

where p.product\_id = ps.product\_id;

-- combine purchase table and customer table based on customer\_id

select c.first\_name,c.last\_name,c.cust\_id,p.quantity,c.dob

from customer c inner join purchases p

where c.cust\_id = p.customer\_id;

use sql\_notes;

create table student

(id integer primary key,

course varchar(10),

location varchar(10),

batch integer,

marks integer);

insert into student values(1,'Mtech','Bangalore',1,70);

insert into student values(2,'Mtech','Bangalore',1,85);

insert into student values(3,'Mtech','Bangalore',1,71);

insert into student values(4,'Mtech','Bangalore',1,64);

insert into student values(5,'Mtech','Bangalore',2,55);

insert into student values(6,'Mtech','Bangalore',2,74);

insert into student values(7,'Mtech','Bangalore',2,84);

insert into student values(8,'Mtech','Bangalore',2,63);

insert into student values(9,'Mtech','Chennai',1,50);

insert into student values(10,'Mtech','Chennai',1,98);

insert into student values(11,'Mtech','Chennai',1,67);

insert into student values(12,'Mtech','Chennai',1,20);

insert into student values(13,'Dse','Bangalore',1,64);

insert into student values(14,'Dse','Bangalore',1,88);

insert into student values(15,'Dse','Bangalore',1,79);

insert into student values(16,'Dse','Bangalore',1,69);

insert into student values(17,'Dse','Chennai',1,80);

insert into student values(18,'Dse','Chennai',1,61);

insert into student values(19,'Dse','Chennai',1,39);

insert into student values(20,'Dse','Chennai',1,48);

insert into student values(21,'Dse','Chennai',1,96);

insert into student values(22,'Dse','Chennai',2,78);

insert into student values(23,'Dse','Chennai',2,69);

insert into student values(24,'Dse','Chennai',2,86);

insert into student values(25,'Dse','Chennai',2,93);

insert into student values(26,'Dse','Chennai',2,66);

insert into student values(27,'Dse','Chennai',1,80);

insert into student values(28,'Mtech','Bangalore',1,85);

select \* from student;

-- 1. List the average marks of dse students

select avg(marks) from student

where course = 'dse';

-- 2. List the average marks of dse students for each location

select id, course,avg(marks) from student

where course = 'dse'

group by id;

-- 3. list the average marks of students in bangalore for different courses

select id,location,course,avg(marks) from student

where location = 'bangalore'

group by id,location,course;

-- 4. List the average marks of dse students for each location,batch

select location,batch,course,avg(marks) from student

where course = 'dse'

group by location,batch;

-- 5. list the highest mark in each course, location wise

select location,course,max(marks) from student

group by location,course;

-- 6. list the highest mark in each course, location,batch wise

select location,course,batch,max(marks) from student

group by location,course,batch;

-- 7. List the highest mark in each location

select location,max(marks) from student

group by location;

-- 8. List students who have got more than 75 marks

select \* from student

where marks>75;

-- 9 List the number of mtech student in Chennai

select location,course,count(course) from student

where course = 'Mtech' and location = 'chennai' ;

-- 10. List students who have got less than 40 in Chennai

select \* from student

where marks < 40 and location = 'Chennai';

-- 11. List the top 3 performers in DSE and Mtech

select id,location,course,max(marks) from student

where course = 'Mtech' or course = 'Dse'

group by id,location,course

order by max(marks) limit 3;

(select id, location, course, batch, marks

from student

where course = 'DSE'

order by marks desc limit 3)

union

(

select id, location, course, batch, marks

from student

where course = 'Mtech'

order by marks desc limit 3

);