SQL Practice

1. Create database NEW DATABASE

postgres=# create database homework;

CREATE DATABASE

postgres=# \l

List of databases

Name | Owner | Encoding | Collate | Ctype | Access privileges

-----------+----------+----------+----------------------------+----------------------------+-----------------------

blue | postgres | UTF8 | English\_United States.1252 | English\_United States.1252 |

homework | postgres | UTF8 | English\_United States.1252 | English\_United States.1252 |

postgres | postgres | UTF8 | English\_United States.1252 | English\_United States.1252 |

template0 | postgres | UTF8 | English\_United States.1252 | English\_United States.1252 | =c/postgres +

| | | | | postgres=CTc/postgres

template1 | postgres | UTF8 | English\_United States.1252 | English\_United States.1252 | =c/postgres +

| | | | | postgres=CTc/postgres

(5 rows)

1. \c postgres

You are now connected to database "postgres" as user "postgres".GOING INTO DATABASE USING

1. Create NEW table using command

CREATE TABLE shaiyan01 (id integer, name varchar(50), email varchar(50), address text)

1. Create schema
2. Create role
3. Create user
4. Insert records

#using this command insert data into a table

INSERT INTO shaiyan01 (id, name, email, address) VALUES (100, ‘john’, ‘sabuj987@gmail.com’, ‘8914 34th av’);

1. \dt # using \dt command we can see how many tables we have created

homework=# \dt

List of relations

Schema | Name | Type | Owner

--------+---------+-------+----------

public | shaiyan | table | postgres

(1 row)

1. Delete records
2. Update records
3. Drop table
4. Switch database / Moving current database to another database

postgres=# \c postgres blue

Password for user blue:

You are now connected to database "postgres" as user "blue".

1. Look into how many database I have created using

\list or l

You are now connected to database "postgres" as user "blue".

postgres=# \l

1. Select \du command see the how many user do I have in my database and what is the authority level

homework=# \du

List of roles

Role name | Attributes | Member of

-----------+------------------------------------------------------------+-----------

blue | Superuser, Create DB | {}

postgres | Superuser, Create role, Create DB, Replication, Bypass RLS | {}

1. We can create a new user using command : CREATE ROLE (name);

homework=# create user shaiyan;

CREATE ROLE

1. SELECT CLUSE # using this query I’m getting all actor f\_name and L\_name from actor table.

select first\_name, last\_name from actor;

-- from pullup any information from a database we must use SELECT command, It's A mandatory.

select last\_name from actor where actor\_id ='8'

--Using alias we can given a new column without changing the table or cloumn name

SELECT \*FROM ADDRESS

SELECT last\_name as Nick\_name from actor;

--Using this command we can organized records that we want to

SELECT first\_name, last\_name FROM actor ORDER BY last\_name ASC;

SELECT \* FROM FILM;

SELECT film\_id, title FROM film ORDER BY title desc

-- Distinct will give you all the value except repitation

SELECT DISTINCT last\_name from actor;

-- Limit column will gime the avility to limit my request.

SELECT address, district, postal\_code from address LIMIT '100'

SELECT address, district, postal\_code from address ORDER BY postal\_code desc LIMIT '100';

-- FETCH will give me the option to select it. we can get records from last to0...

SELECT \* FROM CATEGORY

SELECT name, last\_update FROM category ORDER BY name FETCH first 20 row only;

SELECT name, last\_update FROM category ORDER BY name

offset 10 rows

FETCH FIRST 5 row only;

-- IN will give the records that i want to pullup specifically.

SELECT \* FROM CUSTOMER

SELECT customer\_id, store\_id, last\_name, email, create\_date from customer where customer\_id in (20, 30);

SELECT customer\_id, store\_id, last\_name, email, create\_date from customer where customer\_id in (20, 24, 27, 28, 30)

SELECT customer\_id, store\_id, last\_name, email, create\_date from customer where customer\_id not in (20, 24, 27, 28, 30)

SELECT customer\_id, store\_id, last\_name, email, create\_date from customer where customer\_id in (20, 24, 27, 28, 30)

SELECT customer\_id, store\_id, last\_name, email, create\_date from customer where customer\_id < 4;

SELECT customer\_id, store\_id, last\_name, email, create\_date from customer where customer\_id <> 5

and customer\_id <> 10;

--BETWEEN will return me all the value from the figure i request

SELECT customer\_id, store\_id, last\_name, email, create\_date

from customer

where customer\_id between 20 and 30

SELECT customer\_id, store\_id, last\_name, email, create\_date

from customer

where customer\_id not between 20 and 30

--LIKE execute the query that i will provide the input

SELECT \* FROM COUNTRY

SELECT COUNTRY FROM COUNTRY where country like 'az%';

SELECT COUNTRY

FROM COUNTRY

where country like '%n%';

SELECT COUNTRY

FROM COUNTRY

where country like '%%az%';

--is null mean there is no value but space available that we can later insert any value.

create table show (

id int,

first\_name varchar(40) not null,

last\_name varchar(50) not null,

primary key (id));

-- Table aliases / INNER JOIN

SELECT c.customer\_id,

first\_name,

amount, payment\_date from customer c

INNER JOIN payment p

on p.customer\_id = c.customer\_id

order by payment\_date desc;

select \* from INVENTORY

-- INNER JOIN We are getting common vlaue different value using relations

SELECT CUSTOMER.CUSTOMER\_ID, INVENTORY.INVENTORY\_ID

from CUSTOMER

INNER JOIN INVENTORY

on CUSTOMER\_ID =INVENTORY\_ID;

-- right join

SELECT CUSTOMER.CUSTOMER\_ID, INVENTORY.INVENTORY\_ID

from CUSTOMER

RIGHT JOIN INVENTORY

on CUSTOMER\_ID =INVENTORY\_ID;

-- LEFT JOIN WILL GIVE ME ALL VALUE FROM LEFT TABLE AND COMMON VALUE FROM RIGHT TABLE

SELECT CUSTOMER.CUSTOMER\_ID, INVENTORY.INVENTORY\_ID

from CUSTOMER

LEFT JOIN INVENTORY

on CUSTOMER\_ID =INVENTORY\_ID;

-- LEFT OUTER JOIN

SELECT CUSTOMER.CUSTOMER\_ID, INVENTORY.INVENTORY\_ID

from CUSTOMER

LEFT OUTER JOIN INVENTORY

on CUSTOMER\_ID =INVENTORY\_ID;

-- RIGHT OUTER JOIN

SELECT CUSTOMER.CUSTOMER\_ID, INVENTORY.INVENTORY\_ID

from CUSTOMER

RIGHT OUTER JOIN INVENTORY

on CUSTOMER\_ID =INVENTORY\_ID;

-- CROSS JOIN deliver only first name from both table that are common

SELECT first\_name

from actor

CROSS JOIN film;

-- natural join

select first\_name

from actor

natural join film;

-- GROUP BY

SELECT \* FROM INVENTORY

SELECT inventory\_id, store\_id

from inventory

group by inventory\_id, store\_id;

-- HAVING

SELECT inventory\_id, store\_id

from inventory

group by inventory\_id

HAVING sum(inventory\_id) >2

-- UNION

SELECT inventory\_id

from inventory

UNION

SELECT store\_id from store;

--EXCEPT

SELECT DISTINCT INVENTORY.FILM\_ID,

TITLE FROM INVENTORY

INNER JOIN FILM ON FILM.FILM\_ID = INVENTORY.FILM\_ID

ORDER BY TITLE;

SELECT \* FROM INVENTORY

-- ROLLUP THIS IS RETURN WITH THE SERIAL NUMBER REQUESTED

SELECT FILM\_ID, STORE\_ID

FROM INVENTORY GROUP BY

ROLLUP (FILM\_ID, STORE\_ID)

ORDER BY FILM\_ID, STORE\_ID;

-- AVRAGE

SELECT AVG (AMOUNT) FROM PAYMENT;

1. Alter database/ Rename Database # using this command we can change the database name but before we need to exit from the data base that will be renamed.

postgres=# ALTER DATABASE homework RENAME TO newclasswork;

ALTER DATABASE

1. Drop database ## using this command we can drop the database

postgres=# DROP DATABASE newclasswork;

DROP DATABASE

-- Subquery # is query inside another query. any execuation subquery execute first and then the main query

SELECT \* FROM RENTAL

SELECT film\_id, title, rental\_rate from film

WHERE rental\_rate > (SELECT AVG (rental\_rate)

FROM film);

SELECT first\_name, last\_name FROM customer

WHERE EXISTS (

SELECT FROM PAYMENT WHERE PAYMENT.CUSTOMER\_ID= CUSTOMER.CUSTOMER\_ID);

--ANY #

SELECT MAX( length ) FROM film

INNER JOIN film\_category

USING (film\_id) GROUP BY category\_id;

-- UPDATE # WILL PROVIDE USER TO ENTER OR MODIFY THE THE EXISTING RECORDS

select \* from actor

UPDATE actor

set first\_name ='F\_Name'

WHERE actor\_id ='1';

--