1. Create a table called person that records a person's id, name, age, height (in cm), city, favorite_color.
    - id should be an auto-incrementing id/primary key - Use type: SERIAL

```
create table person (
  id serial PRIMARY KEY,
  name varchar(255),
  age int,
  height int,
  city char(255),
  favorite_color char(255));
```

2. Add 5 different people into the person database.
    - Remember to not include the person_id because it should auto-increment.

```
insert into person (name, age, height, city, favorite_color) values
  ('Mike', 21, 182, 'Dallas', 'red'),
  ('Fred', 21, 152, 'Dallas', 'green'),
  ('John', 21, 162, 'Dallas', 'blue'),
  ('Molly', 21, 172, 'Dallas', 'orange'),
  ('Mary', 21, 180, 'Dallas', 'red');
```

3. List all the people in the person table by height from tallest to shortest.

```
select * from person order by height desc
```

4. List all the people in the person table by height from shortest to tallest.

```
select * from person order by height
```

5. List all the people in the person table by age from oldest to youngest.

```
select * from person order by age desc
```

6. List all the people in the person table older than age 20.

```
select * from person where age > 20
```

7. List all the people in the person table that are exactly 18.

```
select * from person where age = 18
```

Query ran successfully. 0 rows to display.

8. List all the people in the person table that are less than 20 and older than 30.

```
select * from person where age < 20 OR age > 30
```

Query ran successfully. 0 rows to display.

9. List all the people in the person table that are not 27 (Use not equals).

   select * from person where age < > 27

10. List all the people in the person table where their favorite color is not red.

    select * from person where favorite_color != 'red'

11. List all the people in the person table where their favorite color is not red and is not blue.

    select * from person where favorite_color not in ('red', 'blue')

12. List all the people in the person table where their favorite color is orange or green.

    select * from person where favorite_color in ('orange', 'green')

13. List all the people in the person table where their favorite color is orange, green or blue (use IN).

    select * from person where favorite_color in ('orange', 'green', 'blue')

14. List all the people in the person table where their favorite color is yellow or purple (use IN).

    select * from person where favorite_color in ('yellow', 'purple')

    Query ran successfully. 0 rows to display.

Table - orders

Instructions

1. Create a table called orders that records: person_id, product_name, product_price, quantity.

```
create table orders (
  person_id integer,
  product_name varchar(255),
  product_price float,
  quantity int);
```

2. Add 5 orders to the orders table.
   - Make orders for at least two different people.
   - person_id should be different for different people.

```
insert into orders (person_id, product_name, product_price, quantity)
values
(1, 'apple', 0.25, 3),
(1, 'banana', 0.5, 1),
(2, 'grapes', 1.59, 1),
(2, 'orange', 0.25, 3),
(3, 'pineapple', 2.35, 1);
```

3. Select all the records from the orders table.

```
Select * from orders
```

4. Calculate the total number of products ordered.

```
Select sum(quantity) from orders
```

5. Calculate the total order price.

```
Select sum(product_price * quantity) from orders;
```

6. Calculate the total order price by a single person_id.

```
Select person_id, sum(product_price * quantity)
from orders
where person_id = 1
group by person_id;
```

Table - artist

Instructions

1. Add 3 new artists to the artist table. (It's already created)

```
INSERT INTO ARTIST (name)
VALUES ('Fred'),
('Wilma'),
('Pebbles');
```

2. Select 10 artists in reverse alphabetical order.

```
SELECT *
FROM ARTIST
WHERE ARTIST_ID < 11
```

3. Select 5 artists in alphabetical order.

```
SELECT *
FROM ARTIST
LIMIT 10;
```

4. Select all artists that start with the word 'Black'.

```
SELECT *
FROM ARTIST
WHERE NAME LIKE 'Black%';
```

5. Select all artists that contain the word 'Black'.

```
SELECT *
FROM ARTIST
WHERE NAME LIKE '%Black%';
```

EMPLOYEE TABLE

1. List all employee first and last names only that live in Calgary.

   ```
   SELECT first_name, last_name
   FROM employee
   WHERE city = 'Calgary';
   ```

2. Find the birthdate for the youngest employee.

   ```
   SELECT MAX(birth_date)
   FROM employee
   ```

3. Find the birthdate for the oldest employee.

   ```
   SELECT MIN(birth_date)
   FROM employee
   ```

4. Find everyone that reports to Nancy Edwards (Use the ReportsTo column).
   - You will need to query the employee table to find the Id for Nancy Edwards

   ```
   SELECT *
   FROM employee
   WHERE reports_to = 2
   ```

5. Count how many people live in Lethbridge.

   ```
   SELECT count(*)
   FROM employee
   WHERE city = 'Lethbridge'
   ```

INVOICE TABLE

1. Count how many orders were made from the USA.
   ```
   SELECT count(*)
   FROM invoice
   WHERE billing_country = 'USA'
   ```

2. Find the largest order total amount.
   ```
   SELECT Max(total)
   FROM invoice
   ```

3. Find the smallest order total amount.
   ```
   SELECT Min(total)
   FROM invoice
   ```

4. Find all orders bigger than $5.
   ```
   SELECT *
   FROM invoice
   WHERE total > 5
   ```

5. Count how many orders were smaller than $5.
   ```
   SELECT count(*)
   FROM invoice
   WHERE total < 5
   ```

6. Count how many orders were in CA, TX, or AZ (use IN).
```
SELECT count(*)
FROM invoice
WHERE billing_state IN ('CA', 'TX', 'AZ')
```

7. Get the average total of the orders.
```
SELECT AVG(total)
FROM invoice
```

8. Get the total sum of the orders.
```
SELECT sum(total)
FROM invoice
```