SQL (Medium)
25+2 bonus questions

Practice here: sql-practice

Question 1:

Show unique birth years from patients and sort them in ascending order

Query:

```
SELECT
/* Step 1: YEAR() gets only year part from birth_date column */
/* Step 2: DISTINCT() gets only unique birth years */
DISTINCT(YEAR(birth_date)) as [unique birth years]
FROM patients
/* Step 3: ORDER BY to sort in ascending order */
ORDER BY [unique birth years];
```

Question 2:

Show unique first names from patients table which only occurs once.

For example,
if 2 or more people are named 'John' in the first_name column then don't include 'John' in output. If only 1 person is named 'Leo' then include them in output.

Query:

```
/* Step 1: use a CTE to get first_name and their no.of occurrences */
WITH cte_occurrence AS
(
SELECT first_name,
COUNT(*) as occurrence
FROM patients
GROUP BY first_name
)

/* Step 2: get first_name that occur only once from CTE */
SELECT first_name
FROM cte_occurrence
WHERE occurrence=1;
```

Question 3:

Show patient_id and first_name from patients where their first_name start and ends with 's' and is at least 6 characters long

Query:

```
SELECT patient_id, first_name
FROM patients
/* Step 1: LIKE to filter only first_name which start and ends with 's' */
WHERE first_name LIKE 's%s' AND
/* Step 2: LEN() to filter only first_name which are at least 6 characters long */
LEN(first_name)>=6;
```

Question 4:

Show patient_id, first_name, last_name from patients whose diagnosis is 'Dementia'.

Diagnosis is stored in admissions table.

Query:

```
SELECT patients.patient_id, first_name, last_name
FROM patients
/* Step 1: JOIN patients and admissions table */
JOIN admissions
ON patients.patient_id=admissions.patient_id
/* Step 2: Filter only patients whose diagnosis is 'Dementia' */
WHERE diagnosis='Dementia';
```

Question 5:

Display every patient's first_name. Order the list by length of each name and then alphabetically

Query:

```
SELECT first_name
FROM patients
/* Step 1: LEN to calculate length of first_name */
/* Step 2: ORDER BY to sort alphabetically */
ORDER BY LEN(first_name),first_name;
```

Question 6:

Show total number of male patients and total number of female patients from patients table.

Display the two results in same row.

Query:

```
/* Step 3: main query to display the 2 subqueries output in same row */
SELECT
(
/* Step 1: subquery to calculate total no.of male patients */
SELECT COUNT(*) FROM patients
WHERE gender='M'
) AS [total no.of male patients],
/* Step 2: subquery to calculate total no.of female patients */
(
SELECT COUNT(*) FROM patients
WHERE gender='F'
) AS [total no.of female patients];
```

Question 7:

Show first_name, last_name and allergies from patients table who have allergies to either 'Penicillin' or 'Morphine'.

Show results in ascending order of allergies then by first_name and finally by last_name.

Query:

```
SELECT first_name,last_name,allergies
FROM patients
/* Step 1: WHERE and IN to filter allergies that are either 'Penicillin' or 'Morphine' */
WHERE allergies IN('Penicillin','Morphine')
/* Step 2: ORDER BY to sort multiple columns in ascending order */
ORDER BY allergies,first_name,last_name;
```

Question 8:

Show patient_id, diagnosis from admissions table. Find patients admitted multiple times for the same diagnosis.

Query:

```
/* Step 2: main query to get patients who've been admitted for the same diagnosis more than once */
SELECT patient_id,diagnosis
FROM
(
/* Step 1: subquery to get how many times each patient has been admitted for the same diagnosis */
SELECT patient_id,diagnosis,COUNT(*) AS [no.of times admitted]
FROM admissions
GROUP BY patient_id,diagnosis
) AS [subquery_no_of_times_admitted]
WHERE [no.of times admitted]>1;
```

Question 9:

Show city along with number of patients in each city.

Order from most to least patients and then by city name ascending.

Query:

```
SELECT city,COUNT(*) AS [no.of patients]
FROM patients
/* Step 1: GROUP BY to get city-wise patients count */
GROUP BY city
/* Step 2: ORDER BY most to least no.of patients and then by city name in ascending order */
ORDER BY [no.of patients] DESC, city ASC;
```

Question 10:

Show first name, last name and role of every person that is either patient or doctor. The roles are either "Patient" or "Doctor".

Query:

```
/* Step 1: set role as 'Patient' for entries from patients table */
SELECT first_name,last_name,'Patient' as role FROM patients
/* Step 3: UNION ALL to combine both outputs from individual SELECT statements */
UNION ALL
/* Step 2: set role as 'Doctor' for entries from doctors table */
SELECT first_name,last_name,'Doctor' as role FROM doctors;
```

Question 11:

Show all allergies ordered by most to least popular. Don’t consider NULL values into account.

Query:

```
SELECT allergies,COUNT(*) as popularity
FROM patients
/* Step 1: WHERE to ignore allergies that have NULL values */
WHERE allergies IS NOT NULL
GROUP BY allergies
/* Step 2: ORDER BY to sort from most to least popular allergies */
ORDER BY popularity DESC;
```

Question 12:

Show all patient's first_name, last_name and birth_date who were born in the 1970s decade.

Sort the list starting from the earliest birth_date.

Query:

```
SELECT first_name, last_name, birth_date
FROM patients
/* Step 1: WHERE to filter patients who were born in 1970s decade i.e. 1970-1979 */
WHERE YEAR(birth_date) BETWEEN 1970 AND 1979
/* Step 2: ORDER BY to sort list starting from earliest birth_date */
ORDER BY birth_date;
```

Question 13:

We want to display each patient's full name in a single column. Their last_name in all upper letters must appear first, then first_name in all lower case letters. Separate last_name and first_name with a comma. Order the list by first_name in descending order.

Example: SMITH,jane

Query:

```
SELECT
/* Step 1: UPPER() and LOWER() to convert letters to upper and lower case */
/* Step 2: CONCAT() to concatenate upper and lower case with ',' in-between */
CONCAT(UPPER(last_name),',',LOWER(first_name)) as [full_name]
FROM patients
/* Step 3: ORDER BY to sort using first_name in descending order */
ORDER BY first_name DESC;
```

Question 14:

Show province_id(s) and sum of height where sum of its patients height are greater than or equal to 7000

Query:

```
/* Step 1: SUM() to get sum of patients height */
SELECT province_id, SUM(height) as [sum_of_height]
FROM patients
GROUP BY province_id
/* Step 2: HAVING to filter province_id that has sum of patients height greater than or equal to
7000 */
HAVING SUM(height)>=7000;
```

Question 15:

Show difference between largest weight and smallest weight for patients with last name as 'Maroni'

Query:

```
SELECT
/* Step 1: MAX() and MIN() to find largest weight and smallest weight */
/* Step 2: Subtract previous found 2 values */
MAX(weight)-MIN(weight) AS difference
FROM patients
/* Step 3: WHERE to filter only patients with last name as 'Maroni' */
WHERE last_name IS 'Maroni';
```

Question 16:

Show all days of the month (1-31) and how many admission_dates occurred on that day. Sort by the day with most to least admissions.

Query:

```
SELECT
/* Step 1: DAY() to get day number from admission_date column */
DAY(admission_date) as [admission date],COUNT(*) AS [no.of admissions]
FROM admissions
GROUP BY [admission date]
/* Step 2: ORDER BY to sort admission dates from most to least admissions */
ORDER BY [no.of admissions] DESC;
```

Question 17:

Show all columns for patient_id: 542's most recent admission_date

Query:

```
SELECT * FROM admissions
/* Step 1: WHERE to filter only patient_id: 542 */
WHERE patient_id=542
/* Step 2: ORDER BY to sort based on most recent admission_date to first admission_date */
ORDER BY admission_date DESC
/* Step 3: LIMIT to show only first entry which will be the most recent admission_date */
LIMIT 1;
```

Question 18:

Show patient_id, attending_doctor_id, and diagnosis for admissions that match one of the two criteria:

1. patient_id is an odd number and attending_doctor_id is either 1, 5, or 19

2. attending_doctor_id contains a 2 and the length of patient_id is 3 characters

Query:

```
SELECT patient_id,attending_doctor_id,diagnosis
FROM admissions
-- Step 1: WHERE to filter based on two given criteria
WHERE
-- Step 2: %2 to check if remainder is not 0 i.e. odd number and IN to filter attending_doctor_id of 1,5,19
(patient_id%2!=0) AND attending_doctor_id IN (1,5,19)
-- Step 3: OR to match one of the two given criteria
OR
-- Step 4: LIKE to identify if attending_doctor_id contains 2 and LEN to get length of patient_id
attending_doctor_id LIKE '%2%' AND LEN(patient_id)=3;
```

Question 19:

Show first_name, last_name, and total number of admissions attended for each doctor

Every admission has been attended by a doctor

Query:

```
-- Step 3: COUNT to get total number of admissions
SELECT first_name,last_name,COUNT(*) as [number of admissions]
FROM admissions
-- Step 1: JOIN (i.e. inner join) to match admissions and doctors table based on attending_doctor_id and doctor_id
JOIN doctors
ON attending_doctor_id=doctor_id
-- Step 2: GROUP BY to group based on doctor_id
GROUP BY doctor_id;
```

Question 20:

For each doctor, display their id, full name, first and last admission date they attended

Query:

```
SELECT doctor_id,
-- Step 1: CONCAT() to concatenate first_name and last_name as full_name
CONCAT(first_name,' ',last_name) AS full_name,
-- Step 2: MIN() to get first admission date
MIN(admission_date) AS first_admission_date,
-- Step 3: MAX() to get most recent admission date
MAX(admission_date) AS last_admission_date
FROM doctors, admissions
-- Step 4: WHERE to match tables doctors and admissions based on common column
WHERE doctor_id=attending_doctor_id
GROUP BY doctor_id,full_name;
```

Question 21:

Display the number of patients for each province. Order by patient count in descending order.

Query:

```
-- Step 1: COUNT() to get patients count
SELECT province_name,COUNT(*) AS number_of_patients
FROM patients AS pat,province_names AS pvn
-- Step 2: WHERE to match tables patients and province_names based on common column
WHERE pat.province_id=pvn.province_id
GROUP BY province_name
-- Step 3: ORDER BY to sort based on patients count in descending order
ORDER BY number_of_patients DESC;
```

Question 22:

Display the patient's full name, their admission diagnosis and their doctor's full name who diagnosed their problem

Query:

```
SELECT
-- Step 3: CONCAT() to concatenate first_name and last_name of patient as patient_full_name
CONCAT(pat.first_name,' ',pat.last_name) AS patient_full_name,
diagnosis,
-- Step 4: CONCAT() to concatenate first_name and last_name of doctor as doctor_full_name
CONCAT(doc.first_name,' ',doc.last_name) AS doctor_full_name
FROM patients AS pat
-- Step 1: JOIN (i.e. inner join) to match patients and admissions table based on patient_id
JOIN admissions AS adm
ON pat.patient_id=adm.patient_id
-- Step 2: JOIN (i.e. inner join) to match admissions and doctors table based on attending_doctor_id and doctor_id
JOIN doctors AS doc
ON adm.attending_doctor_id=doc.doctor_id;
```

Question 23:

Display the first_name, last_name and number of duplicate patients based on their first_name and last_name

Query:

```
SELECT first_name,last_name,
-- Step 3: COUNT() to get no.of occurrences
COUNT(*) AS no_of_duplicate_patients
FROM patients
-- Step 1: GROUP BY to consider first_name and last_name as a single entity
GROUP BY first_name,last_name
-- Step 2: HAVING to filter first_name and last_name that occur more than once
HAVING COUNT(*)>1;
```

Question 24:

Display patient's full name, height in feet rounded to 1 decimal, weight in pounds rounded to whole number, birth_date, gender in non abbreviated capital letters.

Note:
Given units in patients table for height: cm and weight: kg

Hint:
Convert cm to feet: divide by 30.48
Convert kg to pounds: multiply by 2.205

Query:

```
SELECT
-- Step 1: CONCAT() to concatenate first_name and last_name as full_name
CONCAT(first_name,' ',last_name) AS full_name,
-- Step 2: ROUND() to round height to 1 decimal in feet
ROUND(height/30.48,1) as height_feet,
-- Step 3: ROUND() to round weight to 0 decimal i.e. whole number in pounds
ROUND(weight*2.205,0) as weight_pounds,
birth_date,
-- Step 4: CASE() to show non abbreviated form of gender i.e. 'MALE' for 'M'
CASE(gender)
WHEN 'M' THEN 'MALE'
ELSE 'FEMALE'
END AS gender
FROM patients;
```

Question 25:

Show product_name,company_name and category_name from the products, suppliers and categories table

Query:

```
SELECT
product_name,company_name,category_name
-- Step 1: get all 3 tables and set aliases
FROM products AS prd,suppliers AS sup,categories AS cat
-- Step 2: WHERE to combine all 3 tables based on common columns
WHERE prd.category_id=cat.category_id AND prd.supplier_id=sup.supplier_id;
```

Bonus question 1:

Show category_name and average product unit price for each category rounded to 2 decimal places

Query:

```
-- Step 3: AVG() and ROUND() to calculate average and round to 2 decimal places
SELECT category_name,ROUND(AVG(unit_price),2) AS average_unit_price
-- Step 1: get categories and products tables and set aliases
FROM categories AS cat,products AS prd
-- Step 2: WHERE to combine categories and products tables based on common column i.e. category_id
WHERE cat.category_id=prd.category_id
GROUP BY category_name;
```

Bonus question 2:

Show city, company_name, contact_name from customers and suppliers table merged together.

Create a column which contains 'customers' or 'suppliers' depending on the table it came from.

Query:

```
-- Step 1: get required columns and add a new column 'source' from customers table
SELECT city, company_name, contact_name,'customers' AS source FROM customers
-- Step 3: UNION ALL to combine all records from both tables
UNION ALL
-- Step 2: get required columns and add a new column 'source' from suppliers table
SELECT city, company_name, contact_name,'suppliers' AS source FROM suppliers;
```