SQL (Easy)
25+1 bonus questions

Practice here: sql-practice

Question 1:

Show first name, last name and gender of patients who's gender is 'M'

Query:

```
/* Step 1: Get required columns */
SELECT first_name,last_name,gender FROM patients
/* Step 2: use WHERE to get gender 'M' */
WHERE gender='M';
```

Question 2:

Show first name and last name of patients who does not have allergies (i.e. NULL)

Query:

```
/* Step 1: Get required columns */
SELECT first_name,last_name FROM patients
/* Step 2: use WHERE to get NULL entries in allergies column */
WHERE allergies IS NULL;
```

Question 3:

Show first name of patients that start with the letter 'C'

Query:

```
/* Step 1: Get required column */
SELECT first_name FROM patients
/* Step 2: use WHERE to get first names that start with 'C' */
WHERE first_name LIKE 'C%';
```

Question 4:

Show first name and last name of patients that weight within the range of 100 to 120 (both inclusive)

Query:

```
/* Step 1: Get required columns */
SELECT first_name,last_name FROM patients
/* Step 2: use WHERE to get weight between 100 and 120 */
WHERE weight BETWEEN 100 and 120;
```

Question 5:

Update the patients table for the allergies column. If the patient's allergies is null then replace it with 'NKA'

Query:

```
UPDATE patients
/* Step 2: replace NULL entries with 'NKA' */
SET allergies='NKA'
/* Step 1: get NULL entries from allergies column*/
where allergies IS NULL;
```

Question 6:

Show first name and last name concatenated into one column to show their full name

Query:

```
/* Step 1: use CONCAT() to concatenate required columns */
SELECT CONCAT(first_name,' ',last_name) as full_name FROM patients;
```

Question 7:

Show first name, last name and the full province name of each patient
Example: 'Ontario' instead of 'ON'

Query:

```
/* Step 1: Get required columns from both tables */
SELECT pat.first_name, pat.last_name, prv.province_name
/* Step 2: JOIN both tables and use alias for better readability */
FROM patients AS pat
JOIN province_names AS prv
ON pat.province_id=prv.province_id;
```

Question 8:

Show how many patients have a birth_date with 2010 as the birth year

Query:

```
/* Step 1: use COUNT to get their count */
SELECT COUNT(birth_date) AS total_patients FROM patients
/* Step 2: use YEAR() to get year */
/* Step 3: use WHERE to filter patients who have their birth year in 2010 */
WHERE YEAR(birth_date)=2010;
```

Question 9:

Show first_name, last_name and height of the patient with greatest height

Query:

```
/* Step 1: use MAX() to get greatest height */
SELECT first_name,last_name,MAX(height) as height
FROM patients;
```

Question 10:

Show all columns for patients who have one of the following patient_ids:
1, 45, 534, 879, 1000

Query:

```
/* Step 1: Get all columns */
SELECT * FROM patients
/* Step 2: use WHERE and IN to filter details from given list of patient_ids */
WHERE patient_id IN (1,45,534,879,1000);
```

Question 11:

Show the total number of admissions

Query:

```
/* Step 1: use COUNT() to get total number of admissions */
SELECT COUNT(*) as total_no_of_admissions FROM admissions;
```

Question 12:

Show all columns from admissions table where patient was admitted and discharged on the same day

Query:

```
/* Step 1: get all columns */
SELECT * FROM admissions
/* Step 2: use WHERE to filter patients who were admitted and discharged on the same day */
WHERE admission_date=discharge_date;
```

Question 13:

Show patient id and the total number of admissions for patient_id: 579

Query:

```
/* Step 2: use COUNT() to get total number of admissions */
SELECT patient_id,COUNT(*) AS total_number_of_admissions
FROM admissions
/* Step 1: use WHERE to filter for patient_id: 579 */
WHERE patient_id=579
GROUP BY patient_id;
```

Question 14:

Based on cities that our patients live in, show unique cities that are in province_id 'NS'

Query:

```
/* Step 2: use DISTINCT() to get unique cities */
SELECT *
FROM patients
/* Step 1: use WHERE to filter province_id 'NS' */
WHERE province_id='NS';
```

Question 15:

Write a query to find the first name, last name and birth date of patients who has height greater than 160 and weight greater than 70

Query:

```
/* Step 1: Get required columns */
SELECT first_name,last_name,birth_date FROM patients
/* Step 2: use WHERE and AND to filter based on given conditions */
WHERE height>160 AND weight>70;
```

Question 16:

Write a query to find list of patients first_name, last_name and allergies from Hamilton city where allergies are not null

Query:

```
/* Step 1: get required columns */
SELECT first_name,last_name,allergies FROM patients
/* Step 2: use WHERE and AND to filter based on given conditions */
WHERE city = 'Hamilton' AND allergies IS NOT NULL;
```

Question 17:

Based on cities where our patient lives in, write a query to display the list of unique city starting with a vowel (a, e, i, o, u).

Show the result order in ascending by city

Query:

```
/* Step 2: use DISTINCT to get unique cities only */
SELECT DISTINCT(city) FROM patients
/* Step 1: use WHERE and OR to filter based on given conditions */
WHERE city LIKE 'A%'
OR city LIKE 'E%'
OR city LIKE 'I%'
OR city LIKE 'O%'
OR city LIKE 'U%'
/* Step 3: use ORDER BY to arrange cities in ascending order */
ORDER BY city;
```

Question 18:

Show the category_name and description from the categories table sorted by category_name

Query:

```
/* Step 1: get required columns */
SELECT category_name,description FROM categories
/* Step 2: use ORDER BY to arrange category_name in ascending order */
ORDER BY category_name;
```

Question 19:

Show contact_name, address, city of all customers who are not from the following countries: 'Germany', 'Mexico', 'Spain'

Query:

```
/* Step 1: get required columns */
SELECT contact_name,address,city FROM customers
/* Step 2: use WHERE and NOT IN to filter based on given countries */
WHERE country NOT IN ('Germany', 'Mexico', 'Spain');
```

Question 20:

Show order_date, shipped_date, customer_id, freight of all orders placed on 2018 Feb 26

Query:

```
/* Step 1: get required columns */
SELECT order_date, shipped_date, customer_id, freight
FROM orders
/* Step 2: use WHERE to filter only orders placed on given date */
WHERE order_date='2018-02-26';
```

Question 21:

Show employee_id, order_id, customer_id, required_date, shipped_date from all orders shipped later than required date

Query:

```
/* Step 1: get required columns */
SELECT employee_id, order_id, customer_id, required_date, shipped_date
FROM orders
/* Step 2: use WHERE to filter only orders that are shipped later than required date */
WHERE shipped_date>required_date;
```

Question 22:

Show all even numbered order_id from the orders table

Query:

```
/* Step 1: get order_id column */
SELECT order_id
FROM orders
/* Step 2: use WHERE and % to filter only even numbered order_id's */
WHERE (order_id%2)=0;
```

Question 23:

Show city, company_name, contact_name of all customers from cities which contains the letter 'L' in the city name, sorted by contact_name

Query:

```
/* Step 1: get required columns */
SELECT city, company_name, contact_name FROM customers
/* Step 2: use WHERE and LIKE to filter cities which contains 'L' in their name */
WHERE city LIKE('%L%')
/* Step 3: use ORDER BY to sort by contact_name */
ORDER BY contact_name;
```

Question 24:

Show company_name, contact_name, fax of all customers who have a fax number
(i.e. fax isn't NULL)

Query:

```
/* Step 1: get required columns */
SELECT company_name, contact_name, fax FROM customers
/* Step 2: use WHERE and IS NOT NULL to filter customers whose fax isn't NULL */
WHERE fax IS NOT NULL;
```

Question 25:

Show first_name, last_name and hire_date of the most recently hired employee

Query:

```
/* Step 1: get required columns */
SELECT first_name, last_name, hire_date FROM employees
/* Step 2: use ORDER BY and DESC to sort based on descending order of hire_date */
ORDER BY hire_date DESC
/* Step 3: use LIMIT to display only certain no.of rows */
LIMIT 1;
```

Bonus question 1:

Show average unit price rounded to 2 decimal places, total units in stock, total discontinued products from products table

Query:

```
SELECT
/* Step 1: use ROUND(), AVG() to calculate average unit price */
ROUND(AVG(unit_price),2) as [average unit price],
/* Step 2: use SUM() to calculate total units in stock */
SUM(units_in_stock) as [total units in stock],
/* Step 3: create a subquery and use COUNT() to calculate total discontinued products */
(SELECT COUNT(discontinued) FROM products WHERE discontinued=1) as [total
discontinued products]
FROM products;
```