SQL PRACTICE QUESTIONS BASED ON HOSPITAL_DATA

1. **Show unique birth years from patients and order them by ascending.**

```
SELECT
  DISTINCT YEAR(birth_date) AS birth_year
FROM patients
ORDER BY birth_year;
```

2. **Show unique first names from the patients table which only occurs once in the list.
For example, if two or more people are named 'John' in the first_name column then don't include their name in the output list. If only 1 person is named 'Leo' then include them in the output.**

```
SELECT first_name
FROM patients
GROUP BY first_name
HAVING COUNT(first_name) = 1
```

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.**

```
SELECT
  patient_id, first_name
FROM patients
WHERE first_name LIKE 's%s'
  AND len(first_name) >= 6;
```

4. **Show patient_id, first_name, last_name from patients whos diagnosis is 'Dementia'.
Primary diagnosis is stored in the admissions table.**

```
SELECT
  patients.patient_id, patients.first_name, patients.last_name
FROM patients
  JOIN admissions ON admissions.patient_id = patients.patient_id
WHERE diagnosis = 'Dementia';
```

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

```
SELECT first_name
FROM patients
order by
  len(first_name),
  first_name;
```

6. **Show the total amount of male patients and the total amount of female patients in the patients table.**
   Display the two results in the same row.

```
SELECT
  SUM(Gender = 'M') as male_count,
  SUM(Gender = 'F') AS female_count
FROM patients
```

7. **Show first and last name, allergies from patients which have allergies to either 'Penicillin' or 'Morphine'. Show results ordered ascending by allergies then by first_name then by last_name.**

```
SELECT
  first_name, last_name, allergies
FROM patients
WHERE
  allergies IN ('Penicillin', 'Morphine')
ORDER BY
  allergies, first_name, last_name;
```

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

```
SELECT
  patient_id, diagnosis FROM admissions
GROUP BY patient_id, diagnosis
HAVING COUNT(*) > 1;
```

**9. Show the city and the total number of patients in the city.
Order from most to least patients and then by city name ascending.**

```
SELECT
  city, count(patient_id) AS total_number_of_patients
from patients
group by city
order by total_number_of_patients DESC, city ASC;
```

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

```
SELECT first_name, last_name, 'Patient' AS role
FROM patients
   UNION ALL
SELECT first_name, last_name, 'Doctor' AS role
FROM doctors;
```

11. Show all allergies ordered by popularity. Remove NULL values from query.

```
SELECT
  allergies, COUNT(allergies) AS total_diagnosis
FROM patients
WHERE allergies IS NOT NULL
GROUP BY allergies
ORDER BY total_diagnosis DESC;
```

**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.**

```
SELECT
  first_name, last_name, birth_date
FROM patients
WHERE
  YEAR(birth_date) BETWEEN 1970 AND 1979
ORDER BY birth_date ASC;
```

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 the last_name and first_name with a comma. Order the list by the first_name in decending order**
    EX: SMITH,jane

```
 SELECT
 CONCAT(UPPER(last_name), ',', LOWER(first_name)) AS full_name
FROM patients
ORDER BY first_name DESC;
```

14. **Show the province_id(s), sum of height; where the total sum of its patient's height is greater than or equal to 7,000.**

```
SELECT
  province_id,
  SUM(height) AS sum_height
FROM patients
GROUP BY province_id
HAVING sum_height >= 7000;
```

15. **Show the difference between the largest weight and smallest weight for patients with the last name 'Maroni'.**

```
SELECT
  (MAX(weight) - MIN(weight)) AS weight_delta
FROM patients
WHERE last_name = 'Maroni';
```

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

```
SELECT
DAY(admission_date) AS day_number,
COUNT(*) AS number_of_admissions
FROM admissions
GROUP BY day_number
ORDER BY number_of_admissions  desc;
```

17. Show all columns for patient_id 542's most recent admission_date.

```
SELECT *
FROM admissions
WHERE patient_id = 542
GROUP BY patient_id
HAVING
  admission_date = MAX(admission_date);
```

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.

```
SELECT
  patient_id, attending_doctor_id, diagnosis
FROM admissions
WHERE
  (attending_doctor_id IN (1, 5, 19)
    AND patient_id % 2 != 0)
  OR
  (attending_doctor_id LIKE '%2%'
    AND len(patient_id) = 3);
```

19. Show first_name, last_name, and the total number of admissions attended for each doctor. Every admission has been attended by a doctor.

```
SELECT d.first_name, d.last_name,
  COUNT(a.patient_id) AS admissions_total
FROM doctors d
  JOIN admissions a
ON d.doctor_id = a.attending_doctor_id
GROUP BY d.first_name, d.last_name
ORDER BY admissions_total;
```

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

```
SELECT
    d.doctor_id,
   d.first_name || ' ' || d.last_name AS full_name,
   MIN(a.admission_date) AS first_admission_date,
   MAX(a.admission_date) AS last_admission_date
FROM doctors d
JOIN admissions a
   ON d.doctor_id = a.attending_doctor_id
GROUP BY d.doctor_id, d.first_name, d.last_name
ORDER BY d.doctor_id;
```

21. **Display the total amount of patients for each province. Order by descending.**

```
SELECT
  pr.province_name,
  COUNT(pa.patients_name) AS patient_count
FROM patients pa
 JOIN province_names pr
ON pr.province_id = pa.province_id
GROUP BY pr.province_id
ORDER BY patient_count DESC;
```

22. **For every admission, display the patient's full name, their admission diagnosis, and their doctor's full name who diagnosed their problem.**

```
SELECT concat(p.first_name, ' ' , p.last_name) AS patient_name, a.diagnosis,
        concat(d.first_name, ' ' , d.last_name) AS doctor_name
FROM patients p
JION admissions a
ON p.patient_id = a.patient_id
JION doctors d
ON d.doctor_id = a.attending_doctor_id
ORDER BY patient_name, diagnosis, doctor_name;
```

23. **display the first name, last name and number of duplicate patients based on their first name and last name.**
    Ex: A patient with an identical name can be considered a duplicate.

```
SELECT first_name, last_name,

  COUNT(*) AS num_of_duplicates

FROM patients

GROUP BY first_name, last_name

HAVING COUNT(*) > 1;
```

24. **Display patient's full name,**
    height in the units feet rounded to 1 decimal,
    weight in the unit pounds rounded to 0 decimals,
    birth_date, gender non abbreviated.
    Convert CM to feet by dividing by 30.48.
    Convert KG to pounds by multiplying by 2.205.

```
SELECT

    CONCAT(first_name, ' ', last_name) AS patient_name,

    ROUND(height / 30.48, 1) as height_Feet,

    ROUND(weight * 2.205, 0) AS weight_Pounds, birth_date,

CASE

        WHEN gender = 'M' THEN 'MALE'

        ELSE 'FEMALE'

 END AS gender_type

from patients;
```

25. **Show patient_id, first_name, last_name from patients whose does not have any records in the admissions table. (Their patient_id does not exist in any admissions.patient_id rows.)**

```
SELECT

  patients.patient_id, first_name, last_name

FROM patients

WHERE patients.patient_id NOT IN

    (SELECT admissions.patient_id

    FROM admissions);
```

26. Display a single row with max_visits, min_visits, average_visits where the maximum, minimum and average number of admissions per day is calculated. Average is rounded to 2 decimal places.

```
SELECT
        MAX(number_of_visits) AS max_visits,
        MIN(number_of_visits) AS min_visits,
 ROUND(AVG(number_of_visits),2) AS average_visits
FROM (
  SELECT admission_date, count(*) AS number_of_visits
  FROM admissions
  GROUP BY admission_date
);
```