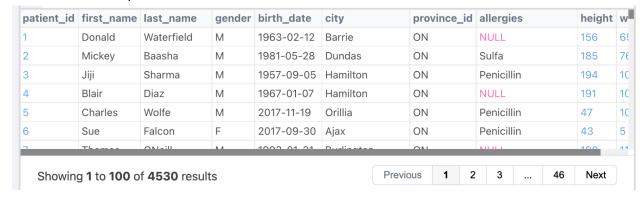
# Практика SQL

Виконується на сайті <a href="https://www.sql-practice.com/">https://www.sql-practice.com/</a> використовуючи базу, яка є там.

0. вивести всю інформацію про пацієнтів.

## SELECT \* FROM patients;



### Завдання

- 1. Виведіть мінімальну і максимальну вагу в наборі даних.
  - -- Min weight SELECT weight FROM patients ORDER BY weight ASC LIMIT 1;

SELECT MIN(weight) FROM patients;

-- MAX weight SELECT weight FROM patients ORDER BY weight DESC LIMIT 1;

SELECT MAX(weight) FROM patients;

```
84 --Hw: 1. Виведіть мінімальну і максимальну вагу в наборі даних.
85 -- Min weight
86 SELECT weight
87 FROM patients
88 ORDER BY weight ASC
89 LIMIT 1;
90
91 SELECT MIN(weight)
92 FROM patients;
93
94
95
```

```
94 -- MAX weight
95 SELECT weight
96 FROM patients
97 ORDER BY weight DESC
98 LIMIT 1;
99
100 SELECT MAX(weight)
101 FROM patients;
102
103
MAX(weight)
142
```

2. Виведіть ім'я, прізвище та вагу пацієнта з найбільшою вагою.

SELECT first name, last name, weight

FROM patients

ORDER BY weight DESC

#### LIMIT 1;

```
103 /*HW: 2. Виведіть ім'я, прізвище та вагу пацієнта з найбільшою вагою.

104 Результатом має бути лише один рядок. */

105 SELECT first_name, last_name, weight

106 FROM patients
107 ORDER BY weight DESC

108 LIMIT 1;

first_name last_name weight

Jerry Camonte 142
```

3. Виведіть ідентифікатор пацієнта, вагу, зріст та день народження 5ти пацієнтів з найменшою вагою в наборі даних.

SELECT patient\_id, weight, height, birth\_date

FROM patients

ORDER BY weight ASC

LIMIT 5;

```
112 SELECT patient id, weight, height, birth date
113 FROM patients
114 ORDER BY weight ASC
115 LIMIT 5;
110
                             weight
                                                  height
                                                                      birth_date
patient_id
                             1
                                                  31
                                                                       2018-02-06
3233
3411
                             1
                                                  28
                                                                       2018-06-16
3532
                             1
                                                  29
                                                                       2018-01-24
                             2
1856
                                                  26
                                                                       2018-07-09
                             2
2989
                                                  34
                                                                       2018-02-24
```

4. Додайте до попереднього результату кількість років пацієнта, округлену до 2 цифр після коми, на момент його звернення в клініку (admission\_date з таблиці admissions). Може вийти так, що вік на момент звернення негативний. Або це баг в даних, або специфіка даних, наприклад, звернулась вагітна жінка - цього ми не дізнаємось.

```
SELECT p.patient_id
,p.weight
,p.height
,p.birth date
```

,a.admission\_date

,round(

(JULIANDAY(a.admission\_date) - JULIANDAY(p.birth\_date)) / 365, 2 ) AS age years

FROM patients AS p JOIN admissions AS a ON a.patient\_id = p.patient\_id ORDER BY weight ASC

## LIMIT 5;

```
126 SELECT p.patient_id
127
        ,p.weight
128
        ,p.height
129
        ,p.birth_date
        ,a.admission_date
131
        , ROUND (
          (JULIANDAY(a.admission_date) - JULIANDAY(p.birth_date)) / 365, 2
132
133
             ) AS age years
134 FROM patients AS p JOIN admissions AS a ON a.patient_id = p.patient_id
135 ORDER BY weight ASC
136 LIMIT 5;
patient_id
                           height
                                      birth_date
                                                        admission_date
                                                                                age_years
                           31
                                      2018-02-06
                                                        2018-08-09
3411
                           28
                                                        2019-05-14
                                                                               0.91
                                      2018-06-16
                                                                               1.28
3532
                           29
                                      2018-01-24
                                                       2019-05-07
1856
                           26
                                      2018-07-09
                                                       2018-06-06
                                                                               -0.09
2989
                           34
                                      2018-02-24
                                                       2018-09-03
```

5. Виведіть ТОР 5 діагнозів в базі за кількістю випадків прийомів з ними.

SELECT COUNT(\*) AS count\_cases, diagnosis FROM admissions group by diagnosis ORDER BY count\_cases DESC LIMIT 5;

```
130 SELECT COUNT(*) AS count_cases, diagnosis
131 FROM admissions
132 GROUP BY diagnosis
133 ORDER BY count_cases DESC
134 LIMIT 5;
135
count_cases
                                     diagnosis
277
                                     Congestive Heart Failure
170
                                     Pregnancy
142
                                     Appendicitis
105
                                     Myocardial Infarction
83
                                     Chest Pain
```

6. Виведіть всю інформацію з таблиці admissions для останнього в базі прийому (визначаємо за admission\_date) з найчастішим діагнозом.

```
--STEP#1

SELECT diagnosis
    ,COUNT(*) AS count_cases

FROM admissions
group by diagnosis
order BY count_cases DESC

LIMIT 1;
--STEP#2

SELECT *

FROM admissions

WHERE diagnosis = 'Congestive Heart Failure'

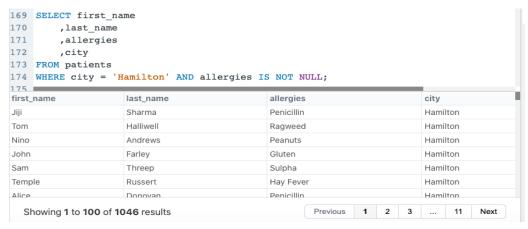
ORDER BY admission_date DESC

LIMIT 1;
```

```
142 --STEP#1
143 SELECT diagnosis
144
           ,COUNT(*) AS count cases
145 FROM admissions
146 GROUP BY diagnosis
147 ORDER BY count_cases DESC
148 LIMIT 1;
149 -- STEP#2
150 SELECT *
151 FROM admissions
152 WHERE diagnosis = 'Congestive Heart Failure'
153 ORDER BY admission date DESC
154 LIMIT 1;
155
patient_id
            admission_date
                              discharge_date
                                                diagnosis
                                                                          attending doctor id
            2019-06-05
                              1971-01-05
                                                Congestive Heart Failure
```

7. Напишіть запит, щоб знайти список імен, прізвищ та алергій пацієнтів з міста Hamiltion, у яких є алергії (тобто значення allergies не є пустим).

```
SELECT first_name
    ,last_name
    ,allergies
    ,city
FROM patients
WHERE city = 'Hamilton' AND allergies IS NOT NULL;
```



8. Кожен прийом коштує 50 доларів для пацієнтів без страхування та 10 доларів для пацієнтів зі страхуванням. Усі пацієнти з **парним** ідентифікатором пацієнта мають страховку.

Поставте кожному пацієнту «Yes», якщо він має страхування, і «No», якщо він не має страхування, назвіть цю колонку has\_insurance. Загрегуйте дані за значенням has\_insurance та додайте загальну вартість прийомів (admissions) для кожної групи has insurance.

```
SELECT has insurance
       ,SUM(cost_admission) AS sum_cost_admission
FROM(
      SELECT a.patient id
             (case when mod(p.patient id,2) <> 0 then 'NO' else 'YES' end) as has insurance
             ,SUM(case when mod(a.patient id,2) <> 0 then 50 else 10 end) as cost admission
      FROM patients AS p JOIN admissions AS a ON a patient id = p.patient id
      GROUP BY a.patient id
      ) TEMP
```

GROUP BY has insurance;

```
185 SELECT has_insurance
186
        ,SUM(cost admission) AS sum cost admission
187 FROM(
188
        SELECT a.patient id
           ,(CASE WHEN mod(p.patient_id,2) <> 0 THEN 'NO' ELSE 'YES' END) AS has_insurance
189
190
            ,SUM(CASE WHEN mod(a.patient id,2) <> 0 THEN 50 ELSE 10 END) AS cost admission
191
        FROM patients AS p JOIN admissions AS a ON a.patient_id = p.patient_id
        GROUP BY a.patient_id
192
        ) TEMP
193
194 GROUP BY has insurance;
195
has_insurance
                                             sum_cost_admission
NO
                                             127800
YES
                                             25110
```

9. Вивести унікальні імена пацієнтів, які починаються і закінчуються на «s», разом з кількістю таких пацієнтів. Кількість носіїв імен відсортувати за спаданням (від найбільшої кількості, до найменшої).

```
SELECT distinct first name
       ,COUNT(patient id)
FROM patients
WHERE first name like 'S%s'
GROUP BY first_name
```

ORDER BY COUNT(patient id) DESC;

```
196 /* НW: 9. Вивести унікальні імена пацієнтів, які починаються і закінчуються на «s», разом з кількістю таких пацієнтів.
197 Кількість носіїв імен відсортувати за спаданням (від найбільшої кількості, до найменшої). */
198 SELECT DISTINCT first_name
199
       ,COUNT(patient_id)
200 FROM patients
201 WHERE first_name LIKE 'S%s'
202 GROUP BY first_name
203 ORDER BY COUNT(patient_id) DESC;
204
first name
                                            COUNT(patient id)
Stanislaus
Spiros
Spats
Seras
Seamus
Samus
```

10. Виведіть кількість пацієнтів чоловіків та пацієнток жінок за провінціями (province\_name) за спаданням кількості чоловіків та лише для тих провінцій, за якими є більше 20ти пацієнтів всього (і чоловіків, і жінок).

```
SELECT pn.province_name

"COUNT(CASE WHEN gender = 'M' THEN 1 ELSE 0 END) AS count_M

"COUNT(CASE WHEN gender = 'F' THEN 1 ELSE 0 END) AS count_F

FROM patients AS p JOIN province_names AS pn ON p.province_id = pn.province_id

GROUP BY pn.province_name

HAVING (count_M + count_F)>20

ORDER BY count_M DESC;
```

Saskatchewan

```
205 /* HW: 10. Виведіть кількість пацієнтів чоловіків та пацієнток жінок за провінціями (province_name)
206 за спаданням кількості чоловіків та лише для тих провінцій, за якими є більше 20ти пацієнтів всього
207 (і чоловіків, і жінок). */
208 SELECT pn.province_name
209
         ,COUNT(CASE WHEN gender = 'M' THEN 1 ELSE 0 END) AS count_M
210
         ,COUNT(CASE WHEN gender = 'F' THEN 1 ELSE 0 END) AS count_F
211 FROM patients AS p JOIN province_names AS pn ON p.province_id = pn.province_id
212 GROUP BY pn.province_name
213 HAVING (count_M + count_F)>20
214 ORDER BY count M DESC;
215
province_name
                                                      count_M
                                                                                      count_F
Ontario
                                                      4298
                                                                                      4298
Nova Scotia
                                                      60
                                                                                      60
British Columbia
                                                      49
                                                                                      49
Alberta
                                                      47
                                                                                      47
Manitoba
                                                      31
                                                                                      31
Quebec
                                                      23
                                                                                      23
```

13

13