##### Акостакіоає Флоріан 414 КН

##### Варіант 10

**Інформаційна система «Електронний староста»**

Ви розробляєте інформаційну систему для обліку відвідувань занять студентами певної академічної групи. В обов’язки старости входить фіксувати пропуски студентами занять з дисциплін, що їм викладаються. Слід пам’ятати, що існують різні види занять: лекції, лабораторні та практичні заняття. “Електронний староста” повинен фіксувати пропуски студентів групи з поважних та неповажних причин. Існує обмеження на пропуск 30 годин занять, після чого студент стає кандидатом на виключення.

**Варіант 10.1**

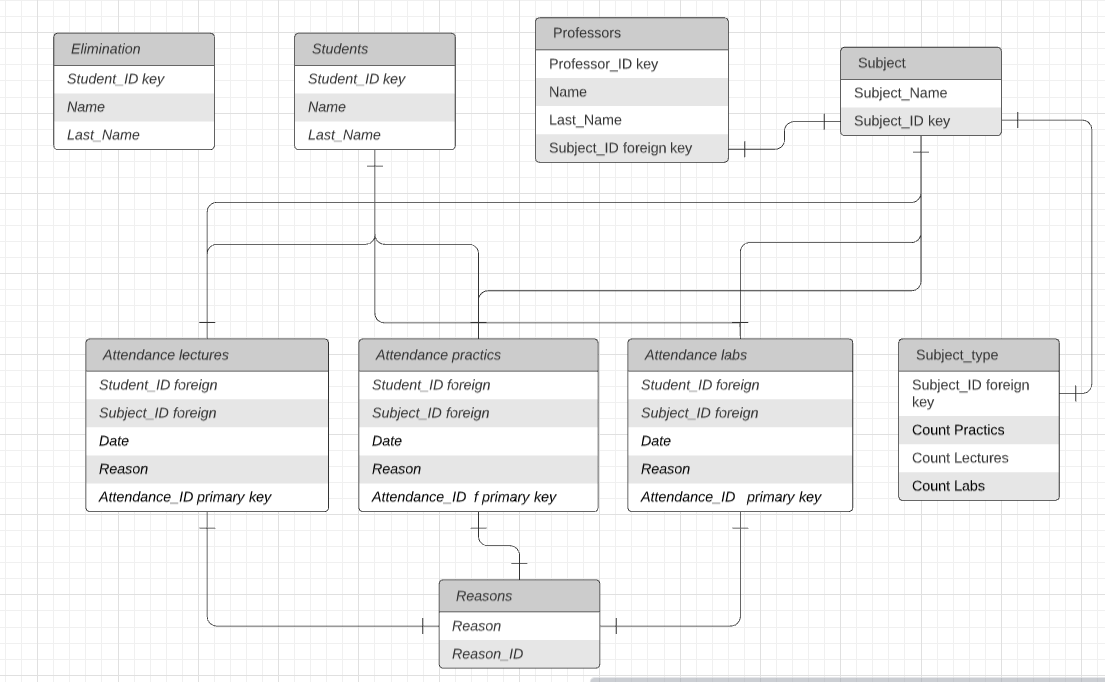
1. Знайти студентів, які не пропустили жодного заняття з початку семестру.
2. Знайти студентів, які пропустили більше 15% загальної кількості лабораторних з заданого предмету.
3. Додати запис про нового студента групи.
4. Підрахувати студентів, що пропустили 10 годин лекцій з поважних причин.
5. Видалити дані про студента, якщо кількість пропущених ним занять перевищує 30 годин.

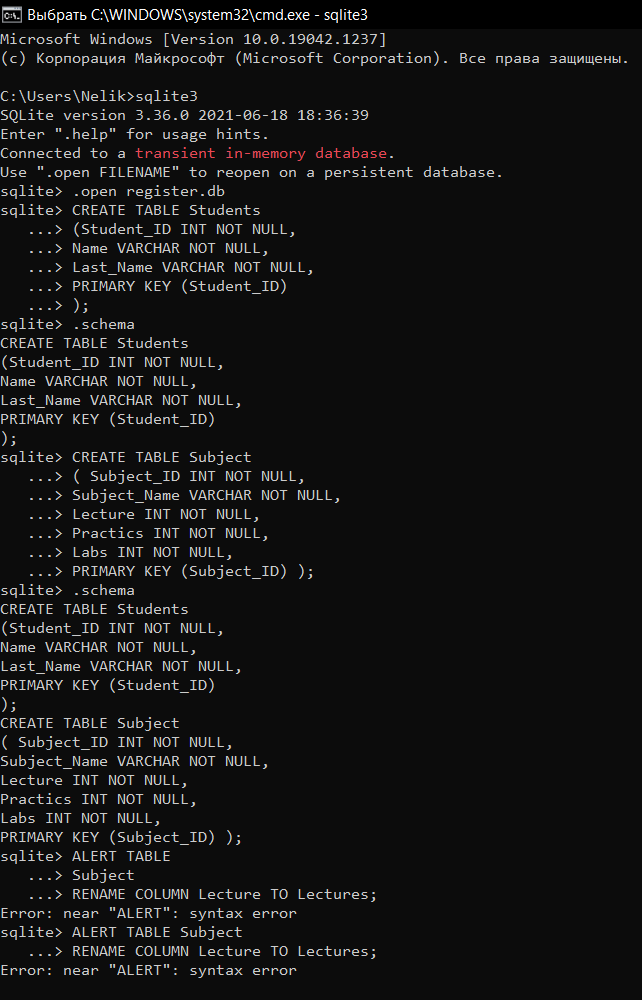
**Варіант 10.2**

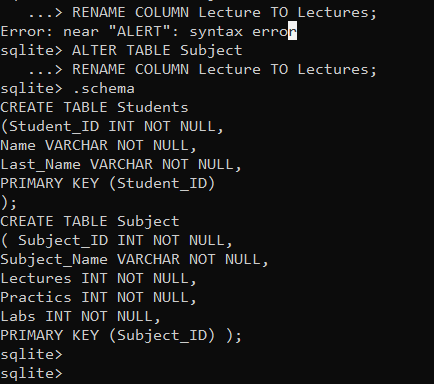
1. Знайти студентів, які пропустили більше 20 годин занять від початку семестру.
2. Знайти дисципліни, практичні заняття з яких були пропущені студентами.
3. Підрахувати загальну кількість пропусків, які підкріплені довідками.
4. Знайти інформацію про студентів, які відвідують лекції з певних предметів, проте не відвідують лабораторні заняття з них.
5. Знайти детальну інформацію по предмету з зазначенням скільки кожен студент пропустив лекційних, практичних та лабораторних занять.

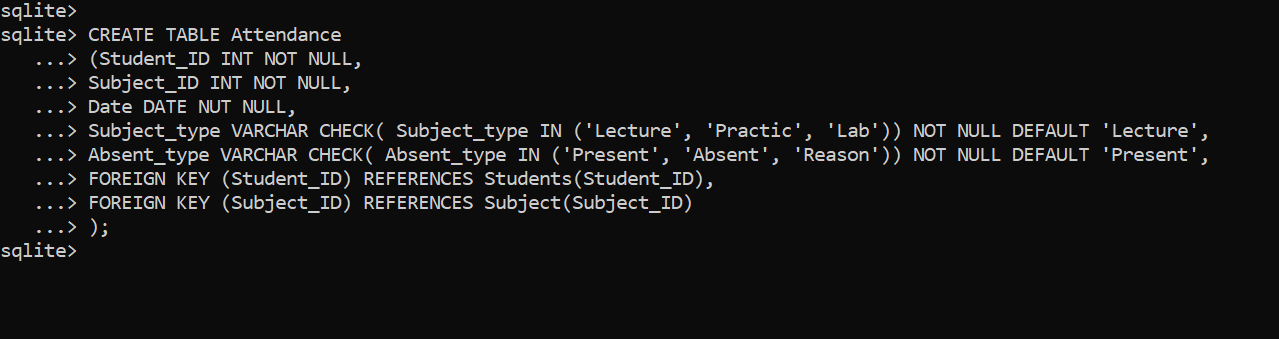
**Варіант 10.3**

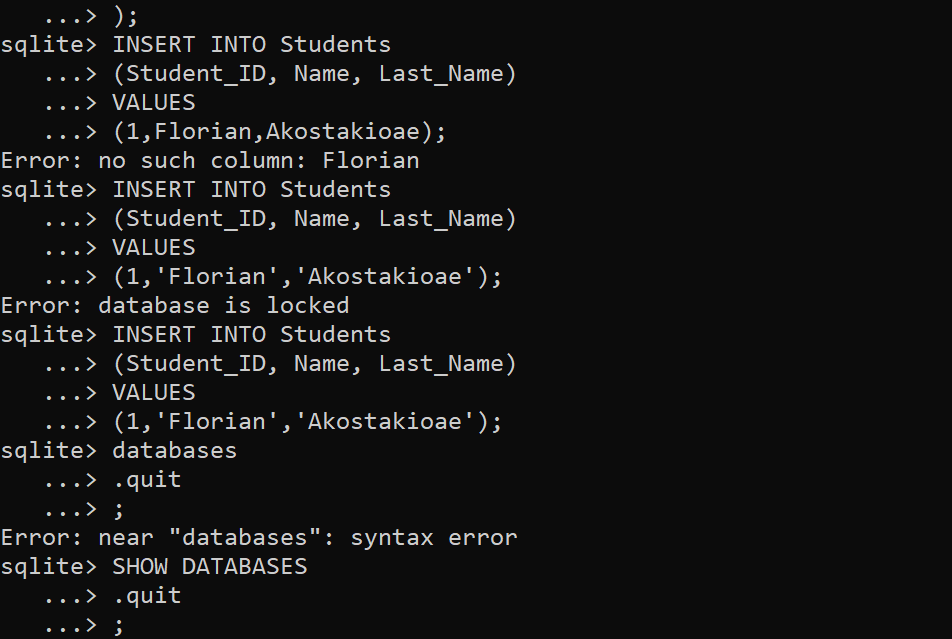
1. Знайти предмет, заняття з якого не пропускав з неповажних причин жоден студент.
2. Знайти кількість пропущених практичних занять з певного предмету студентами у жовтні.
3. Створити звіт з зазначенням загальної кількості пропущених занять (вказати види) студентами групи.
4. Визначити студента, що пропустив максимальну кількість занять з неповажних причин.
5. Скласти рейтинг предметів за рівнем їх відвідування студентами.

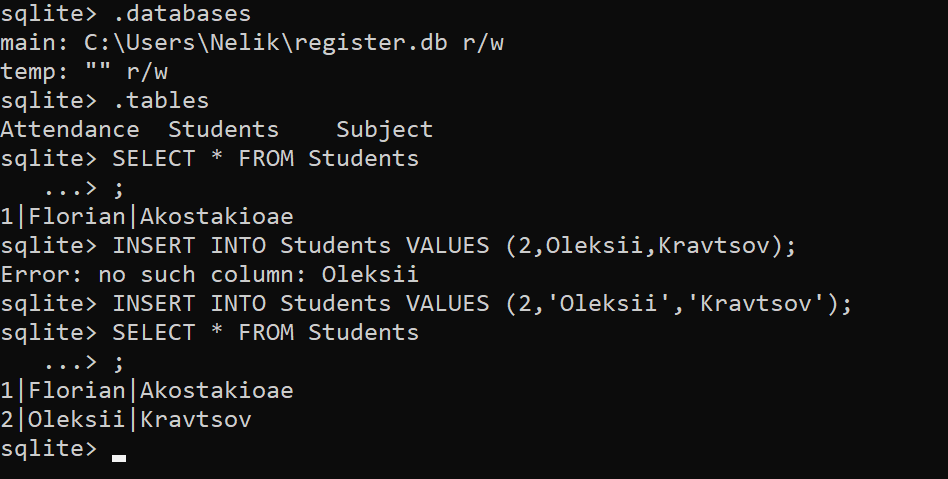
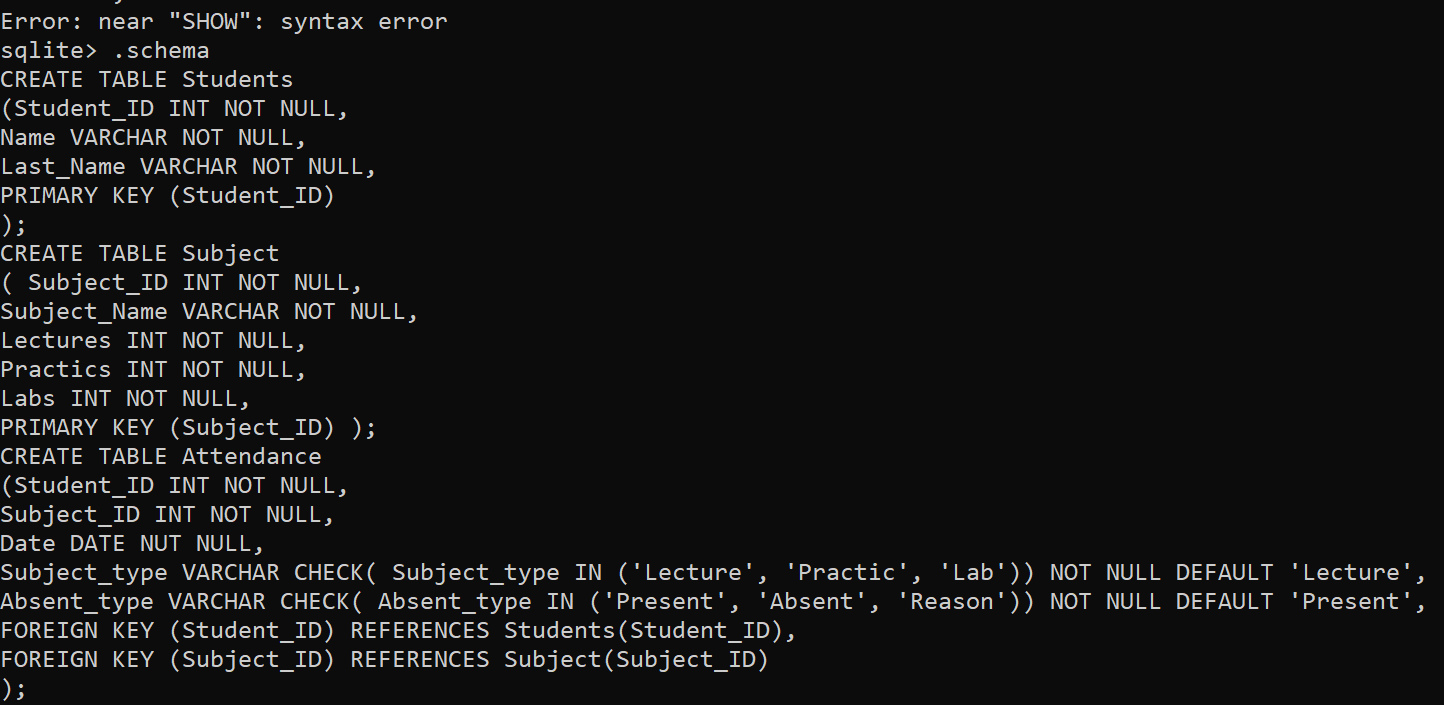


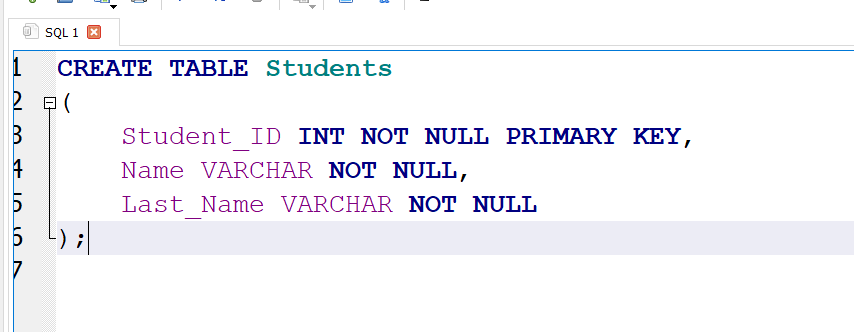












CREATE TABLE Students

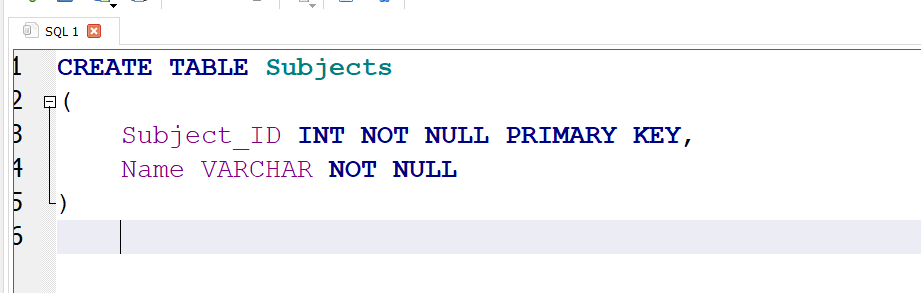
(

Student\_ID INT NOT NULL PRIMARY KEY,

Name VARCHAR NOT NULL,

Last\_Name VARCHAR NOT NULL

);



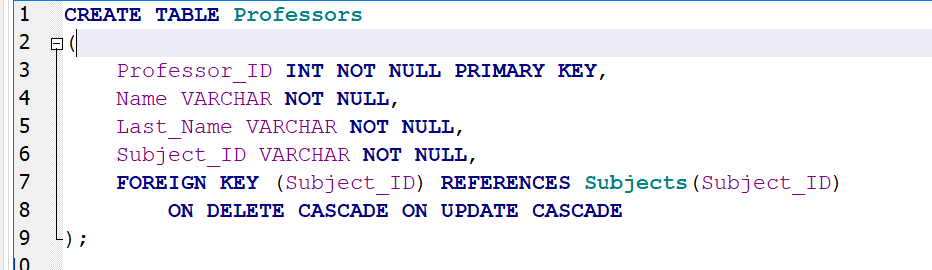
CREATE TABLE Subjects

(

Subject\_ID INT NOT NULL PRIMARY KEY,

Name VARCHAR NOT NULL

);



CREATE TABLE Professors

(

Professor\_ID INT NOT NULL PRIMARY KEY,

Name VARCHAR NOT NULL,

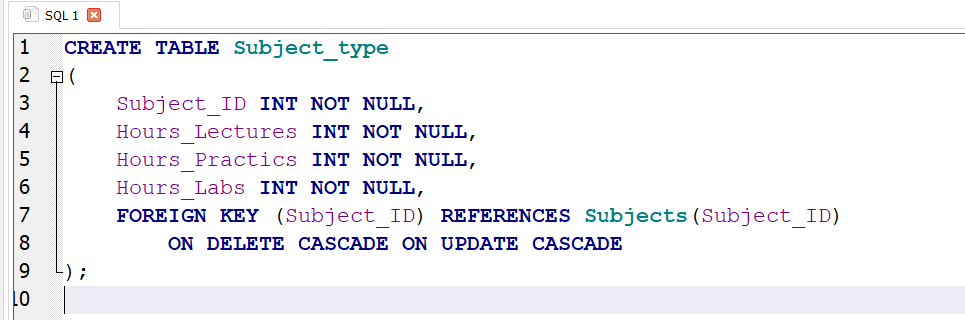
Last\_Name VARCHAR NOT NULL,

Subject\_ID VARCHAR NOT NULL,

FOREIGN KEY (Subject\_ID) REFERENCES Subjects(Subject\_ID)

ON DELETE CASCADE ON UPDATE CASCADE

);



CREATE TABLE Subject\_type

(

Subject\_ID INT NOT NULL,

Hours\_Lectures INT NOT NULL,

Hours\_Practics INT NOT NULL,

Hours\_Labs INT NOT NULL,

FOREIGN KEY (Subject\_ID) REFERENCES Subjects(Subject\_ID)

ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE Attendance\_lectures

(

Attendance\_ID INTEGER PRIMARY KEY AUTOINCREMENT,

Student\_ID INT NOT NULL,

Subject\_ID INT NOT NULL,

Day DATE NOT NULL,

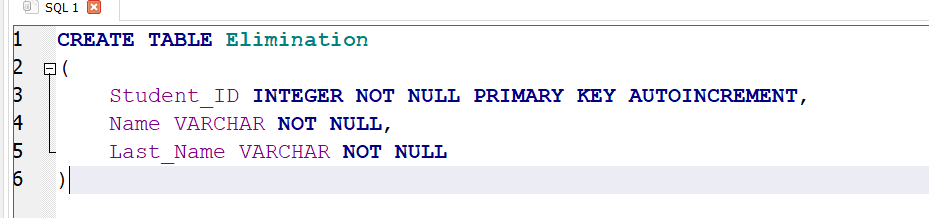
Type INT NOT NULL,

FOREIGN KEY("Student\_ID") REFERENCES "Students"("Student\_ID") ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY("Subject\_ID") REFERENCES "Subjects"("Subject\_ID") ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY("Type") REFERENCES "Reasons"("Reason\_ID")

);



CREATE TABLE Elimination

(

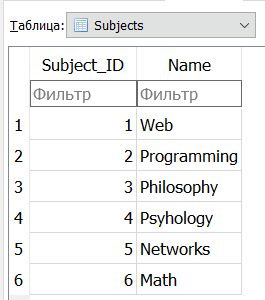
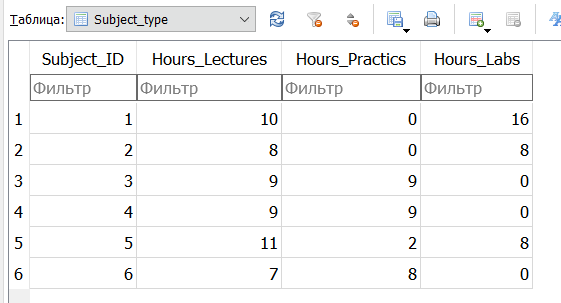
Student\_ID INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,

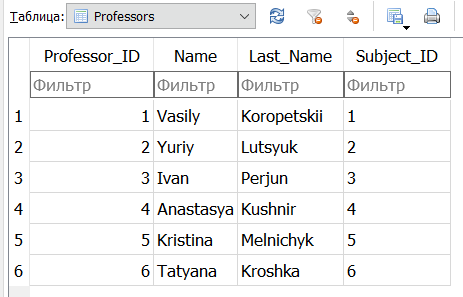
Name VARCHAR NOT NULL,

Last\_Name VARCHAR NOT NULL

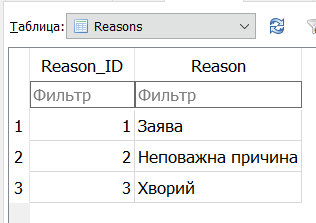
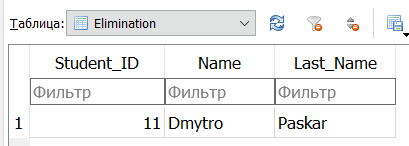
)

Таблиця предемет Скільки годин виділенні для кожного предмета

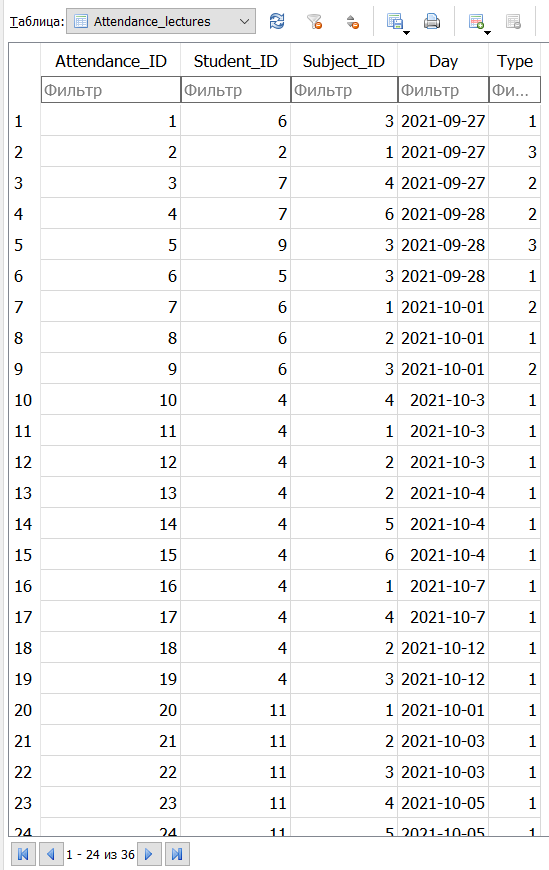


Таблиця студенти Таблиця викладачі

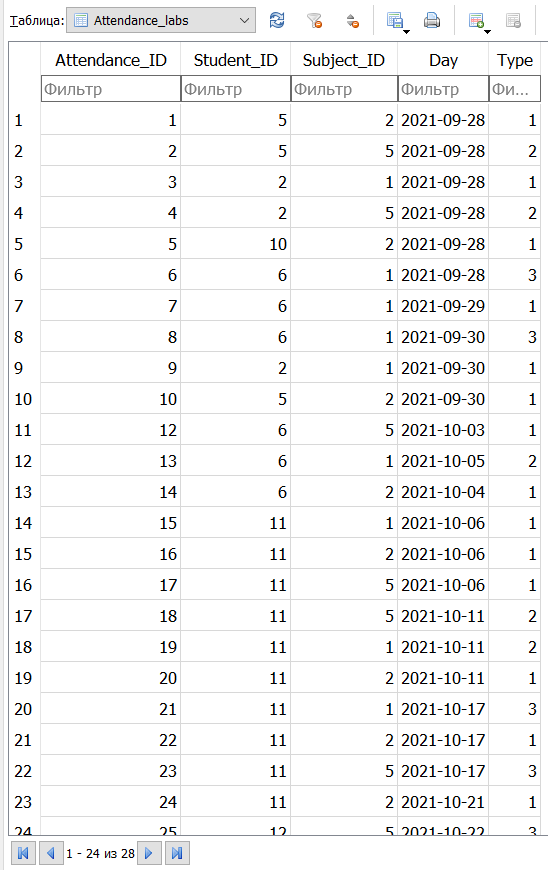
Таблиця причини Н Таблиця кандидатів на виключення



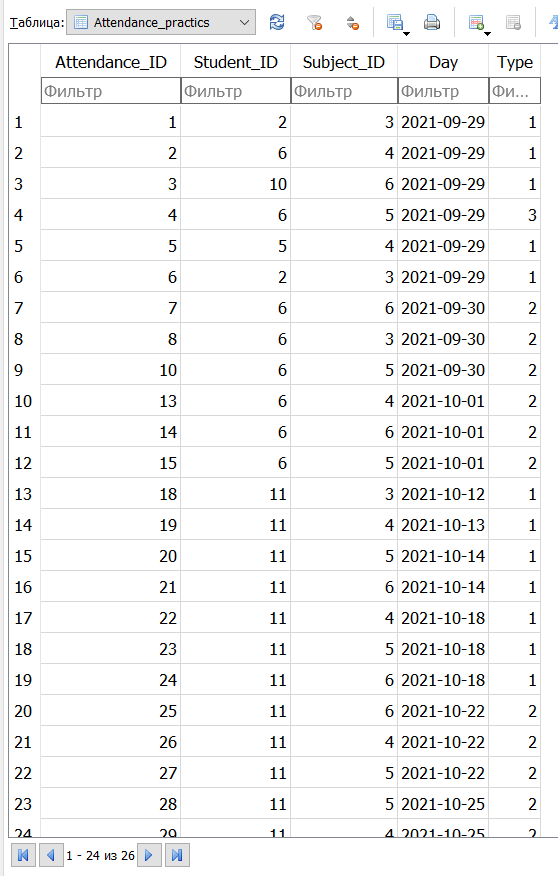
Таблиця відвідувань на лекційних заннятях



Таблиця відвідувань лабораторних заннять



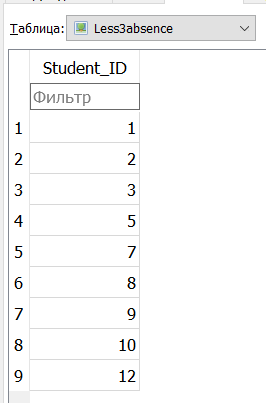
Таблиця відвідувань практичних заннять



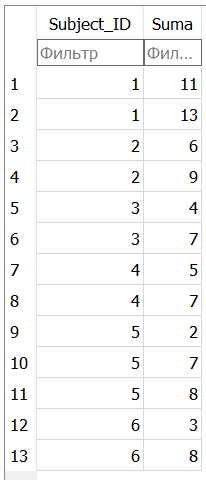
Подання. Список студентів з зазначенням скільки пар пропустив студент по кожному предмету та з яких причин.



Cписок студентів які пропустили менше 3 лекційних занять

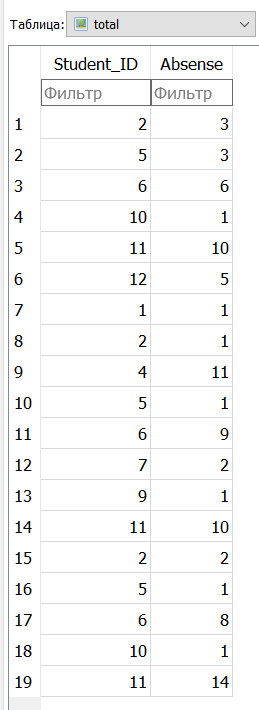


Подання sub\_att: предмет та кількість пропусків по ньому. Тип не зазначено так як в умові для подальшого оброблення інформації, він не потрібен.

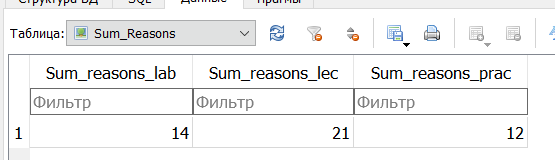


подання total

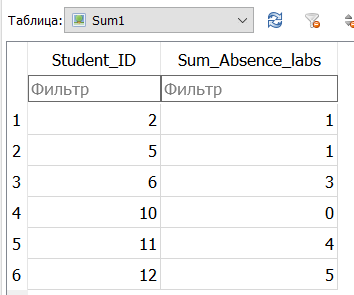
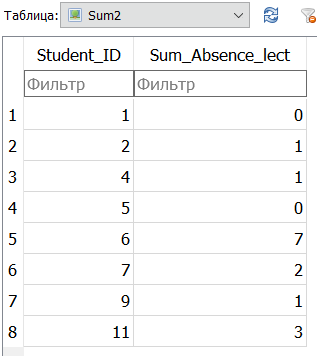
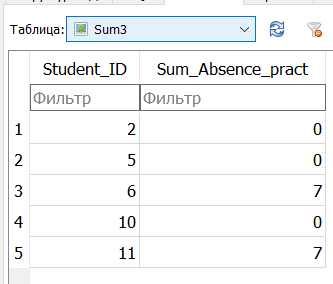
Інформація про те скільки студент має пропусків, з трьох таблиць, лекції, лабораторні, практичні.



Подання Sum\_Reasons

 Кількість пропусків по поважній причині.

Подання Sum1,Sum2,Sum3 кількість пропусків з неповажних причин



Запит N1

Знайти студентів, які не пропустили жодного заняття з початку семестру.

Version 1

SELECT

Student\_ID

FROM Students

WHERE Student\_ID NOT IN

(

SELECT

Student\_ID

FROM Attendance\_labs

)

AND Student\_ID NOT IN

(

SELECT

Student\_ID

FROM Attendance\_lectures

)

AND Student\_ID NOT IN

(

SELECT

Student\_ID

FROM Attendance\_practics

)

Version 2

SELECT

Student\_ID

FROM Students

WHERE Student\_ID NOT IN

(

SELECT

Student\_ID

FROM Attendance\_lectures

UNION

SELECT

Student\_ID

FROM Attendance\_labs

UNION

SELECT

Student\_ID

FROM Attendance\_practics

);

ЗАПИТ 2

Знайти студентів, які пропустили більше 15% загальної кількості лабораторних з заданого предмету.

SELECT

COUNT(Student\_ID) AS MoreThanFifteenpers,

Student\_ID

FROM Attendance\_labs

WHERE Subject\_ID = 1

GROUP BY Student\_ID

HAVING COUNT(Student\_ID) >

(

SELECT

Subject\_type.Hours\_Labs \* 0.15 AS Fifteenpers

FROM Subject\_type

WHERE Subject\_ID = 1

);

ЗАПИТ 3

Додати запис про нового студента групи.

INSERT INTO Elimination

SELECT

total.Student\_ID,

Students.Name,

Students.Last\_Name

FROM total INNER JOIN Students ON Students.Student\_ID = total.Student\_ID

GROUP BY total.Student\_ID

HAVING SUM(Absense) > 30

ЗАПИТ 4

Підрахувати студентів, що пропустили 10 годин лекцій з поважних причин.

SELECT

Student\_ID,

name,

Last\_Name

FROM Students

WHERE Student\_ID IN

(

SELECT

Student\_ID

From Attendance\_lectures

WHERE Type = 1

GROUP BY Student\_ID

HAVING COUNT(Student\_ID) = 10

);

ЗАПИТ 5

Видалити дані про студента, якщо кількість пропущених ним занять перевищує 30 годин.

CREATE VIEW total AS

SELECT

Attendance\_labs.Student\_ID,

COUNT(Attendance\_labs.Student\_ID) AS Absense

FROM Attendance\_labs

GROUP BY Student\_ID

UNION ALL

SELECT

Attendance\_lectures.Student\_ID,

COUNT(Attendance\_lectures.Student\_ID)

FROM Attendance\_lectures

GROUP BY Student\_ID

UNION ALL

SELECT

Attendance\_practics.Student\_ID,

COUNT(Attendance\_practics.Student\_ID)

FROM Attendance\_practics

GROUP BY Student\_ID

DELETE FROM Students

WHERE Student\_ID IN

(

SELECT

Student\_ID

FROM total

GROUP BY Student\_ID

HAVING SUM(Absense) > 30

)

ЗАПИТ 6

Знайти студентів, які пропустили більше 20 годин занять від початку семестру.

SELECT

Student\_ID,

SUM(Absense) AS total\_absense

FROM total

GROUP BY Student\_ID

HAVING total\_absense > 20

Запит 7

Знайти дисципліни, практичні заняття з яких були пропущені студентами.

SELECT

Subject\_ID,

name

FROM Subjects

WHERE Subject\_ID IN

(

SELECT

Subject\_ID

FROM Attendance\_practics

)

Version 2 ( практика з дисципліни яку не пропустив ні один студент)

я неправильно прочитав запитання і зробив випадково інший запит (hard diff)

SELECT

Subject\_ID,

name

FROM Subjects

WHERE Subject\_ID IN

(

SELECT DISTINCT

Subject\_ID

FROM Subject\_type

WHERE Hours\_Practics <> 0 AND Subject\_ID NOT IN

(

SELECT DISTINCT

Subject\_ID

FROM Attendance\_practics

)

)

Запит 8

Підрахувати загальну кількість пропусків, які підкріплені довідками.

CREATE VIEW Sum\_Reasons AS

SELECT

(SELECT SUM(CASE WHEN Type = 1 THEN 1 ELSE 0 END)

FROM Students INNER JOIN Attendance\_labs ON Attendance\_labs.Student\_ID = Students.Student\_ID) AS Sum\_reasons\_lab,

(SELECT SUM(CASE WHEN Type = 1 THEN 1 ELSE 0 END)

FROM Students INNER JOIN Attendance\_lectures ON Attendance\_lectures.Student\_ID = Students.Student\_ID) AS Sum\_reasons\_lec,

(SELECT SUM(CASE WHEN Type = 1 THEN 1 ELSE 0 END)

FROM Students INNER JOIN Attendance\_practics ON Attendance\_practics.Student\_ID = Students.Student\_ID) AS Sum\_reasons\_prac

SELECT

Sum\_reasons\_lab + Sum\_reasons\_lec + Sum\_reasons\_prac AS Result

FROM Sum\_Reasons

Запит 9

Знайти інформацію про студентів, які відвідують лекції з певних предметів, проте не відвідують лабораторні заняття з них.

CREATE VIEW Less3absence AS

SELECT

Student\_ID

FROM Students

WHERE Student\_ID NOT IN

(

SELECT

Attendance\_lectures.Student\_ID

FROM Attendance\_lectures

GROUP BY Student\_ID

HAVING COUNT(Attendance\_lectures.Student\_ID) > 3

)

SELECT

Student\_ID

FROM Less3absence

WHERE Student\_ID IN

(

SELECT

Student\_ID

FROM Attendance\_labs

GROUP BY Student\_ID

HAVING COUNT(Attendance\_labs.Student\_ID) > 4

)

Запит 10

Знайти детальну інформацію по предмету з зазначенням скільки кожен студент пропустив лекційних, практичних та лабораторних занять.

CREATE VIEW Attendance\_by\_Student AS

SELECT

Students.name,

Students.Last\_Name,

Attendance\_lectures.Subject\_ID,

Attendance\_lectures.Type,

COUNT(Attendance\_lectures.Type) AS Attendance

FROM Students,Attendance\_lectures

WHERE Students.Student\_ID = Attendance\_lectures.Student\_ID

GROUP BY Attendance\_lectures.Type,Attendance\_lectures.Student\_ID

UNION ALL

SELECT

Students.name,

Students.Last\_Name,

Attendance\_practics.Subject\_ID,

Attendance\_practics.Type,

COUNT(Attendance\_practics.Type) AS Attendance

FROM Students,Attendance\_practics

WHERE Students.Student\_ID = Attendance\_practics.Student\_ID

GROUP BY Attendance\_practics.Type,Attendance\_practics.Student\_ID

UNION ALL

SELECT

Students.name,

Students.Last\_Name,

Attendance\_labs.Subject\_ID,

Attendance\_labs.Type,

COUNT(Attendance\_labs.Type) AS Attendance

FROM Students,Attendance\_labs

WHERE Students.Student\_ID = Attendance\_labs.Student\_ID

GROUP BY Attendance\_labs.Type,Attendance\_labs.Student\_ID

ORDER BY Subject\_ID

Запит 11

Знайти предмет, заняття з якого не пропускав з неповажних причин жоден студент.

SELECT

Subject\_ID

FROM Subjects

WHERE Subject\_ID NOT IN

(

SELECT

Subjects.Subject\_ID

FROM Subjects INNER JOIN Attendance\_labs ON Subjects.Subject\_ID = Attendance\_labs.Subject\_ID

WHERE Type = 2 OR Type = 3

)

AND Subject\_ID NOT IN

(

SELECT

Subjects.Subject\_ID

FROM Subjects INNER JOIN Attendance\_lectures ON Subjects.Subject\_ID = Attendance\_lectures.Subject\_ID

WHERE Type = 2 OR Type = 3

)

AND Subject\_ID NOT IN

(

SELECT

Subjects.Subject\_ID

FROM Subjects INNER JOIN Attendance\_practics ON Subjects.Subject\_ID = Attendance\_practics.Subject\_ID

WHERE Type = 2 OR Type = 3

)

Запит 12

Знайти кількість пропущених практичних занять з певного предмету студентами у жовтні.

SELECT

COUNT(Attendance\_practics.Student\_ID) AS absence\_practics

FROM Students INNER JOIN Attendance\_practics ON

Attendance\_practics.Student\_ID = Students.Student\_ID

WHERE Attendance\_practics.Day > '2021-09-30' AND Attendance\_practics.Day < '2021-11-01' AND

Attendance\_practics.Subject\_ID =

(SELECT

Subject\_ID

FROM Subjects

WHERE name = 'Psyhology')

Запит 13

Створити звіт з зазначенням загальної кількості пропущених занять (вказати види) студентами групи.

SELECT

Name,

Last\_Name,

Type,

SUM(Attendance) AS Total\_absence

FROM Attendance\_by\_Student

GROUP BY Type,Name

ЗАПИТ 14

Визначити студента, що пропустив максимальну кількість занять з неповажних причин.

3 таблиці подання, Sum1, Sum2, Sum3 Приклад: Sum3. Sum1 та Sum2 аналогічно.

CREATE VIEW Sum3 AS

SELECT

Student\_ID,

SUM(CASE WHEN Attendance\_practics.Type <> 1 THEN 1 ELSE 0 END) AS Sum\_Absence\_pract

FROM Attendance\_practics

GROUP BY Attendance\_practics.Student\_ID

SELECT

Students.Student\_ID,

COALESCE(Sum1.Sum\_Absence\_labs,0) +

COALESCE(Sum2.Sum\_Absence\_lect,0) +

COALESCE(Sum3.Sum\_Absence\_pract,0) AS Maximum

FROM Students LEFT JOIN Sum1 ON Students.Student\_ID = Sum1.Student\_ID

LEFT JOIN Sum2 ON Students.Student\_ID = Sum2.Student\_ID

LEFT JOIN Sum3 ON Students.Student\_ID = Sum3.Student\_ID

ORDER BY Maximum DESC

LIMIT 1

Запит 15

Скласти рейтинг предметів за рівнем їх відвідування студентами.

CREATE VIEW Sub\_att AS

SELECT

Subject\_ID,

COUNT(Subject\_ID) AS Suma

FROM Attendance\_labs

GROUP BY Subject\_ID

UNION

SELECT

Subject\_ID,

COUNT(Subject\_ID) AS Suma

FROM Attendance\_lectures

GROUP BY Subject\_ID

UNION

SELECT

Subject\_ID,

COUNT(Subject\_ID) AS Suma

FROM Attendance\_practics

GROUP BY Subject\_ID

SELECT

Sub\_att.Subject\_ID,

Subjects.Name,

SUM(Sub\_att.Suma) AS Attendance

FROM Sub\_att INNER JOIN Subjects ON

Subjects.Subject\_ID = Sub\_att.Subject\_ID

GROUP BY Subjects.Subject\_ID

ORDER BY SUM(Sub\_att.Suma) ASC