**Бази данних та інформаційні системи**

Лабораторна робота № 2 ( 7 балів)

Акостакіоає Флоріан 414 КН

Створити create view ( 3 штуки), текст запитів довільний, але використайте команди In, any, all, exists.

Студенти, які прагнуть отримати max реалізовують  view  with check option

Cтворити 3 вкладених підзапити ( Select, FROM, WHERE). Створити 3 підзапити з прозиціями UPDATE DELETE Insert into

Дивись презентацію [Select 3](https://moodle.chnu.edu.ua/mod/resource/view.php?id=131428),  VIEW. Студенти, які прагнуть отримати max ( роблять запит з пропозицією WITH).

Лабораторна робота виконується на основі попередньої, текс запитів розробник БД формує на власний розсуд відповідно до поставлених завдань.

ЗАПИТ 1

Видалити студентів які пропустили 3 лекційні заннятя з вишмату.

SAVEPOINT BeforeDeleteVika

DELETE

FROM Students

WHERE Student\_ID IN

(

SELECT

Student\_ID

FROM Attendance\_lectures

GROUP BY Subject\_ID, Student\_ID

HAVING Subject\_ID = 6 AND COUNT(Student\_ID) > 3

ORDER BY COUNT(Student\_ID) DESC

)

ROLLBACK TO BeforeDeleteVika

ЗАПИТ 2

UPDATE ПО РЕЗУЛЬТАТУ ЗАПИТУ.

Студент який прогуляв більше всього пар з конкретного предмету – міняємо прізвище на Лох. (без create view)

SELECT IN FROM, SELECT IN WHERE

UPDATE Students

SET Last\_Name = "Loh"

WHERE Student\_ID IN

(

SELECT

Student\_ID

FROM

(

SELECT

Student\_ID,

COUNT(Subject\_ID) AS Att

FROM Attendance\_labs

WHERE Subject\_ID = 1

GROUP BY Student\_ID

UNION

SELECT

Student\_ID,

COUNT(Subject\_ID) AS Att

FROM Attendance\_practics

WHERE Subject\_ID = 1

GROUP BY Student\_ID

UNION

SELECT

Student\_ID,

COUNT(Subject\_ID) AS Att

FROM Attendance\_lectures

WHERE Subject\_ID = 1

GROUP BY Student\_ID

)

GROUP BY Student\_ID

ORDER BY SUM(Att) DESC

LIMIT 1

)

Обновляємо прізвище студенту у якого більше всього пропусків по видам занять.(лекц, практ, лаб)

UPDATE Students

SET Last\_Name = "Loh"

WHERE Student\_ID IN

(SELECT

Student\_ID

FROM Attendance\_labs

GROUP BY Student\_ID

ORDER BY count(Student\_ID) DESC

LIMIT 1

)

OR Student\_ID IN

(SELECT

Student\_ID

FROM Attendance\_practics

GROUP BY Student\_ID

ORDER BY count(Student\_ID) DESC

LIMIT 1

)

OR Student\_ID IN

(SELECT

Student\_ID

FROM Attendance\_lectures

GROUP BY Student\_ID

ORDER BY count(Student\_ID) DESC

LIMIT 1

)

ЗАПИТ 3

Вставка або оновлення інформації.

SAVEPOINT BeforeInsert

INSERT OR REPLACE INTO Elimination

(Student\_ID, name, Last\_Name)

VALUES (11, 'Beniamin', 'Suleiman')

ROLLBACK TO BeforeInsert

ЗАПИТ 4

SELECT IN SELECT

SELECT

(

SELECT

count(Type)

FROM Attendance\_labs

WHERE Type = 1

)

+

(

SELECT

count(Type)

FROM Attendance\_practics

WHERE Type = 1

)

+

(

SELECT

count(Type)

FROM Attendance\_lectures

WHERE Type = 1

) AS TotalAbsenceByReason

ЗАПИТ 5

Sqlite не підтримує ANY, тому рішення яке я знайшов:

CREATE VIEW Sickness\_ANYparody AS

SELECT

\*

FROM Attendance\_labs

WHERE Day IN

(

SELECT

Day

FROM Attendance\_labs

WHERE Day BETWEEN '2021-10-10' AND '2021-10-20'

)

AND Type IN

(

SELECT

Type

FROM Attendance\_labs

WHERE Type = 3

)

ЗАПИТ 6

Тест оператора ALL в SELECT, але це не то.

SELECT ALL

\*

FROM Elimination

WHERE Student\_ID = 11

ЗАПИТ 7

CREATE VIEW OpExists AS

SELECT

\*

FROM Attendance\_lectures

WHERE EXISTS

(

SELECT

MAX(Day)

FROM Attendance\_lectures

WHERE Day = '2021-10-26'

)

ЗАПИТ 8

Оператор WITH, Звичайний запит.

WITH CommonTableExpression (Professor\_ID, Professor\_Name, Professor\_Last\_Name, Subject\_ID) AS

(

SELECT

Professor\_ID,

Name,

Last\_Name,

Subject\_ID

FROM Professors

)

SELECT

Professor\_ID,

Professor\_Name,

Professor\_Last\_Name,

CommonTableExpression.Subject\_ID,

Subject\_type.Hours\_Labs + Subject\_type.Hours\_Lectures + Subject\_type.Hours\_Practics AS TotalHours

FROM CommonTableExpression INNER JOIN Subject\_type

ON CommonTableExpression.Subject\_ID = Subject\_type.Subject\_ID

ЗАПИТ 9

WITH Рекурсія

\*заповнення таблиці Student\_rank\*

INSERT INTO Students\_manager

(Student\_ID,Manager)

SELECT Student\_ID, NULL FROM Students

CREATE VIEW ForRecursion AS

SELECT Students.Student\_ID, Name, Last\_Name, Students\_manager.Manager

FROM Students INNER JOIN Students\_manager

ON Students.Student\_ID=Students\_manager.Student\_ID

WITH Recursion (Student\_ID, Name, Last\_Name, Manager, Raank) AS

(

SELECT

Student\_ID,

Name,

Last\_Name,

Manager,

1 AS Raank

FROM ForRecursion

WHERE Manager IS NULL

UNION ALL

SELECT

t1.Student\_ID,

t1.Name,

t1.Last\_Name,

t1.Manager,

t2.Raank + 1

FROM ForRecursion t1

JOIN Recursion t2

ON t1.Manager = t2.Student\_ID

)

SELECT \* FROM Recursion ORDER BY Raank

ЗАПИТ 10

Знайти у яких викладачів немає пар з лабораторних робіт

WITH CTE (Professor\_ID, name, Last\_Name, Subject\_ID) AS

(

SELECT

Professor\_ID, name, Last\_Name, Subject\_ID

FROM Professors

)

, CTE\_subjtype (Subject\_ID, Hours\_Labs, Hours\_Lectures, Hours\_Practics) AS

(

SELECT

Subject\_ID, Hours\_Labs, Hours\_Lectures, Hours\_Practics

FROM Subject\_type

)

SELECT

\*

FROM CTE INNER JOIN CTE\_subjtype ON CTE.Subject\_ID = CTE\_subjtype.Subject\_ID

WHERE Hours\_Labs = 0