Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут прикладної математики та фундаментальних наук

Кафедра прикладної математики

**Звіт**

про виконання лабораторної роботи №2

з курсу «Організація без даних та знань»

**Виконав :**   
студент групи ПМ-23   
Музика Денис

**Перевірив:**

кандидат наук   
 Пабирівський Віктор Володимирович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата) (підпис викладача)

**Лабораторна робота №2. «Написання SQL-запитів»**

Завдання лабораторної роботи виконувати в призначеній предметній області. Змістовне призначення запитів визначати самостійно виходячи з логіки предметної області.

Створити запити типу:

* SELECT на базі однієї таблиці з використанням сортування, накладенням умов зі зв’язками OR та AND.
* SELECT з виводом обчислюваних полів (виразів) в колонках результату.
* SELECT на базі кількох таблиць з використанням сортування, накладенням умов зі зв’язками OR та AND.
* SELECT на базі кількох таблиць з типом поєднання Outer Join.
* SELECT з використанням операторів Like, Between, In, Exists, All, Any.
* SELECT з використанням підсумовування та групування.
* SELECT з використанням під-запитів в частині Where.
* SELECT з використанням під-запитів в частині From.
* ієрархічний SELECT‑запит.
* SELECT‑запит типу CrossTab.
* UPDATE на базі однієї таблиці.
* UPDATE на базі кількох таблиць.
* Append (INSERT) для додавання записів з явно вказаними значеннями.
* Append (INSERT) для додавання записів з інших таблиць.
* DELETE для видалення всіх даних з таблиці.
* DELETE для видалення вибраних записів таблиці.

Запити можуть бути складними і поєднувати у собі одразу декілька типів (згідно завдання). У роботі має бути не менше двох таких складних запитів. При цьому обов’язково необхідно відобразити у звіті відповідні особливості запиту.

**Призначення та SQL представлення:**

--SELECT на базі однієї таблиці з використанням сортування, накладенням умов зі зв’язками OR та AND.

SELECT \* FROM Clients

WHERE taxCode = 8642 OR taxCode = 65432

ORDER BY name DESC;

-- SELECT з виводом обчислюваних полів (виразів) в колонках результату.

SELECT Clients.name, Clients.directorName, Rating.actualRating \* 10 AS 'Score'

FROM Clients

LEFT OUTER JOIN Rating ON Clients.id = Rating.clientID;

-- SELECT на базі кількох таблиць з використанням сортування, накладенням умов зі зв’язками OR та AND.

--Цей запит вибирає імена клієнтів, номери контрактів та дати початку контрактів з таблиць

--Clients, CreditContract та CreditContracts, відповідно. Умови накладаються на коди податків клієнтів та

--кількість коштів, які були видані на контракти. Результати сортуються за датою початку контракту у зворотному порядку.

SELECT c.name AS client\_name, cc.startDate

FROM Clients AS c

JOIN CreditContract AS cc ON c.id = cc.clientID

LEFT JOIN CreditContracts AS ccc ON c.id = ccc.clientID AND cc.id = ccc.contractID

WHERE c.taxCode = 5432 OR (ccc.contractID IS NULL AND cc.amount > 1000)

ORDER BY cc.startDate DESC;

--SELECT з використанням операторів Like, Between, In, Exists, All, Any.

SELECT \* FROM Clients WHERE name LIKE '%S%';

SELECT \* FROM CreditContract WHERE interestRate BETWEEN 5 AND 10;

SELECT \* FROM Clients WHERE id IN (1, 3, 5);

SELECT \* FROM Clients WHERE EXISTS (SELECT \* FROM CreditContract WHERE Clients.id = CreditContract.clientID);

--SELECT з використанням підсумовування та групування.

SELECT c.name, COUNT(cc.id) AS totalContracts, SUM(cc.amount) AS totalCreditAmount

FROM Clients c

LEFT JOIN CreditContract cc ON cc.clientID = c.id

GROUP BY c.name

ORDER BY totalContracts DESC;

--SELECT запит типу CrossTab.

--У цьому прикладі, перша колонка виводить ім'я клієнта, а наступні колонки

--відображають кількість кредитів відкритих відповідно до місяця.

SELECT \*

FROM

(

SELECT c.name, MONTH(p.paymentDate) AS month, COUNT(cc.id) AS count

FROM Clients c

JOIN CreditContract cc ON c.id = cc.clientID

JOIN PaymentSchedule ps ON cc.id = ps.contractID

JOIN PaymentAction p ON ps.paymentActionID = p.id

GROUP BY c.name, MONTH(p.paymentDate)

) AS SourceTable

PIVOT

(

SUM(count)

FOR month IN ([1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12])

) AS PivotTable;

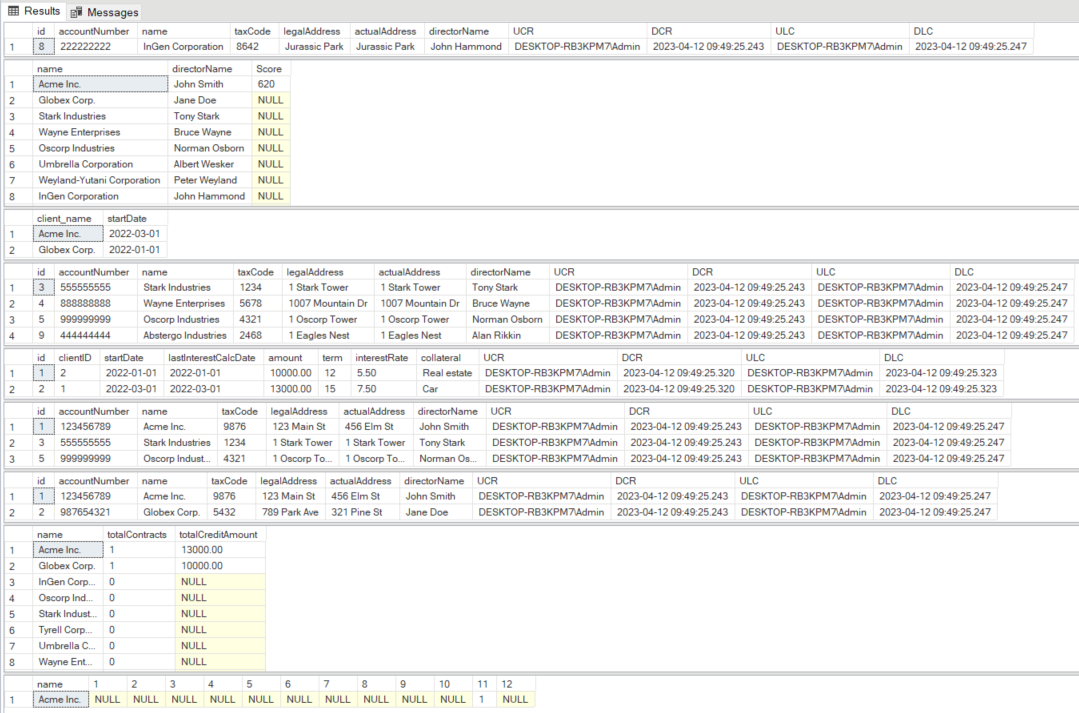
--UPDATE на базі однієї таблиці.

UPDATE Clients

SET legalAddress = '456 Main St', actualAddress = '123 Elm St'

WHERE name = 'Acme Inc.';

**Результат виконання:**

****