

Лабораторна робота № 4-5

ВИКОРИСТАННЯ SELECT. ЗАПИТИ НА ВИБІРКУ ІЗ ГРУПОВИМИ ОПЕРАЦІЯМИ. ПІДЗАПИТИ У MOBI SQL

Мета роботи: дослідження властивостей DML-інструкції SELECT для групування даних та відображення проміжних підсумків.

Хід роботи:

1. → Виконати запити наведені у ході роботи для БД «Торгівельна фірма».

2. → Створити самостійно наступні запити:

→ вивести загальну кількість товарів на підприємстві.

```
SELECT SUM(NaSklade) AS TotalProducts
FROM tovar;
```

	TotalProducts
1	330

Рис. 1 Результат виконання

→ вивести загальну кількість співробітників підприємства.

```
SELECT COUNT(*) AS TotalEmployees
FROM sotrudnik;
```

	TotalEmployees
1	3

Рис. 2 Результат виконання

→ вивести загальну кількість постачальників підприємства.

```
SELECT COUNT(*) AS TotalSuppliers
FROM postachalnik;
```

	TotalSuppliers
1	3

Рис. 3 Результат виконання

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.22.121.05.000 –Лр.4-5						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Васюта В.В.			Звіт з лабораторної роботи №4-5			Лім.		Арк.	Акрушіє
Перевір.		Коротун О.В.								1	12
Реценз.								ФІКТ, гр. ІПЗ-22-1			
Н. Контр.											
Зав.каф.		Єфіменко А.А.									

—>Вивести кількість за кожним товаром, що придбані у поточному місяці.¶

```
SELECT t.Nazva, SUM(zt.Kilkist) AS PurchasedQuantity
FROM zakaz z
INNER JOIN zakaz_tovar zt ON z.id_zakaz = zt.id_zakaz
INNER JOIN tovar t ON zt.id_tovar = t.id_tovar
WHERE CONVERT(DATE, z.date_rozm) BETWEEN '2017-06-01' AND '2017-06-30'
GROUP BY t.Nazva;
```

	Nazva	PurchasedQuantity
1	Кефир	44
2	Молоко	206
3	Сметана	98
4	Творог	36

Рис. 4 Результат виконання

—>Вивести суму, на яку були придбані товари у поточному місяці.¶

```
SELECT SUM(t.Price * zt.Kilkist) AS TotalSpent
FROM zakaz z
INNER JOIN zakaz_tovar zt ON z.id_zakaz = zt.id_zakaz
INNER JOIN tovar t ON zt.id_tovar = t.id_tovar
WHERE CONVERT(DATE, z.date_rozm) BETWEEN '2017-06-01' AND '2017-06-30';
```

	TotalSpent
1	6636.00

Рис. 5 Результат виконання

—>Вивести суму продажу товарів за кожним постачальником.¶

```
SELECT p.Nazva, SUM(t.Price * zt.Kilkist) AS TotalSales
FROM zakaz_tovar zt
INNER JOIN tovar t ON zt.id_tovar = t.id_tovar
INNER JOIN postachalnik p ON zt.id_postav = p.id_postach
GROUP BY p.Nazva;
```

	Nazva	TotalSales
1	Краз Н.Н	1491.00
2	ПП Фриз Н.Н.	4392.00
3	ТОВ "Арей"	4206.00

Рис. 6 Результат виконання

—>Вивести загальну кількість замовлень за кожним постачальником, що продає молоко.¶

```
SELECT p.Nazva, COUNT(z.id_zakaz) AS TotalOrders
FROM zakaz z
INNER JOIN zakaz_tovar zt ON z.id_zakaz = zt.id_zakaz
INNER JOIN tovar t ON zt.id_tovar = t.id_tovar
INNER JOIN postachalnik p ON zt.id_postav = p.id_postach
WHERE t.Nazva = 'Молоко'
GROUP BY p.Nazva;
```

	Nazva	TotalOrders
1	ПП Фриз Н.Н.	8
2	ТОВ "Арей"	6

Рис. 7 Результат виконання

—→Вивести середню суму, на яку замовлявся товар.¶

```
SELECT AVG(TotalSpent) AS AverageOrderAmount
FROM (
    SELECT SUM(t.Price * zt.Kilkist) AS TotalSpent
    FROM zakaz_tovar zt
    INNER JOIN tovar t ON zt.id_tovar = t.id_tovar
    GROUP BY zt.id_zakaz
) AS OrderTotals;
```

	AverageOrderAmount
1	1121.000000

Рис. 8 Результат виконання

—→Вивести вартість замовлень усіх клієнтів, що мешкають у Житомирі.¶

```
SELECT SUM(t.Price * zt.Kilkist) AS TotalOrderValue
FROM zakaz z
INNER JOIN zakaz_tovar zt ON z.id_zakaz = zt.id_zakaz
INNER JOIN tovar t ON zt.id_tovar = t.id_tovar
INNER JOIN klient k ON z.id_klient = k.id_klient
WHERE k.City = 'Житомир';
```

	TotalOrderValue
1	8520.00

Рис. 9 Результат виконання

—→Вивести середню ціну на товари по кожному постачальнику.¶

```
SELECT p.Nazva, AVG(t.Price) AS AveragePrice
FROM tovar t
INNER JOIN postachalnik p ON t.id_postav = p.id_postach
GROUP BY p.Nazva;
```

	Nazva	AveragePrice
1	Краз Н.Н	21.000000
2	ПП Фриз Н.Н.	26.500000
3	ТОВ "Аррей"	13.000000

Рис. 10 Результат виконання

3.→Створити для БД «Рейтинг» наступні запити (база даних додається до лабораторної роботи):¶

—→Сумарний рейтинг студента з кожної дисципліни.¶

```
SELECT Kod_student, SUM(Reiting) AS Total_Rating
FROM Reiting
GROUP BY Kod_student;
```

	Kod_student	Total_Rating
1	46	43
2	47	66
3	48	54
4	49	89
5	50	59

Рис. 11 Результат виконання

→Розрахувати кількість студентів у кожній групі.¶

```
SELECT Kod_group, COUNT(Kod_stud) AS Students_Count
FROM dbo_student
GROUP BY Kod_group;
```

	Kod_group	Students_Count
1	?-53	19
2	?-54	19

Рис. 12 Результат виконання

→Розрахувати кількість дисциплін за групою.¶

```
SELECT Kod_group, COUNT(*) AS Discipline_Count
FROM dbo_groups
GROUP BY Kod_group;
```

	Kod_group	Discipline_Count
1	?-53	1
2	?-54	1
3	?-55	1

Рис. 13 Результат виконання

→Розрахувати кількість проведених занять у кожній групі.¶

```
SELECT Kod_group, COUNT(*) AS Classes_Count
FROM Rozklad_pids
GROUP BY Kod_group;
```

	Kod_group	Classes_Count
1	?-53	1
2	?-54	27

Рис. 14 Результат виконання

→Розрахувати середній бал за групою.¶

```
SELECT Kod_group, AVG(Reiting) AS Avg_Rating
FROM Reiting r
JOIN dbo_student s ON r.Kod_student = s.Kod_stud
GROUP BY Kod_group;
```

	Kod_group	Avg_Rating
1	?-53	60
2	?-54	68

Рис. 15 Результат виконання

→Розрахувати середній бал з дисципліни.¶

```
SELECT pp.K_predmet, AVG(r.Reiting) AS Avg_Rating
FROM Reiting r
JOIN Predmet_plan pp ON pp.K_predmet = pp.K_predmet
GROUP BY pp.K_predmet;
```

	K_predmet	Avg_Rating
1	343	68
2	345	68
3	347	68
4	348	68
5	349	68

Рис. 16 Результат виконання

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.22.121.05.000 –Лр.4-5	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

→Розрахувати поточний рейтинг студента з кожної дисципліни.¶

```
SELECT r.Kod_student, pp.K_predmet, r.Reiting AS Current_Rating
FROM Reiting r
JOIN Predmet_plan pp ON pp.K_predmet = pp.K_predmet;
```

	Kod_student	K_predmet	Current_Rating
1	84	388	80
2	85	388	60
3	86	388	39
4	87	388	45
5	88	388	52

Рис. 17 Результат виконання

→Відобразити найменший рейтинг студентів з дисципліни.¶

```
SELECT pp.K_predmet, MIN(r.Reiting) AS Min_Rating
FROM Reiting r
JOIN Predmet_plan pp ON pp.K_predmet = pp.K_predmet
GROUP BY pp.K_predmet;
```

	K_predmet	Min_Rating
1	343	35
2	345	35
3	347	35
4	348	35
5	349	35

Рис. 18 Результат виконання

→Відобразити найбільший студентський рейтинг з дисципліни.¶

```
SELECT pp.K_predmet, MAX(r.Reiting) AS Min_Rating
FROM Reiting r
JOIN Predmet_plan pp ON pp.K_predmet = pp.K_predmet
GROUP BY pp.K_predmet;
```

	K_predmet	Min_Rating
1	343	100
2	345	100
3	347	100
4	348	100
5	349	100

Рис. 19 Результат виконання

→Розрахувати кількість проведених занять за видами для кожної дисципліни.¶

```
SELECT K_predm_pl, COUNT(Zdacha_type) AS Classes_Count_By_Type
FROM Rozklad_pids
GROUP BY K_predm_pl, Zdacha_type;
```

	K_predm_pl	Classes_Count_By_Type
1	93	1
2	86	1
3	87	3
4	89	2
5	90	8

Рис. 20 Результат виконання

—→Розрахувати кількість груп за кожною спеціальністю.¶

```
SELECT K_spets, COUNT(Kod_group) AS Groups_Count
FROM Navch_plan
JOIN dbo_groups ON Navch_plan.K_navch_plan = dbo_groups.K_navch_plan
GROUP BY K_spets;
```

	K_spets	Groups_Count
1	1	1
2	2	2

Рис. 21 Результат виконання

—→Запит на знищення даних з таблиці «Reiting» за визначеним кодом студента (в поле параметра вводиться прізвище студента).¶

```
DELETE FROM Reiting
WHERE Kod_student IN (SELECT Kod_stud FROM dbo_student WHERE Sname = 'Vasiuta');
```

	K_zapis	Kod_student	Reiting	Prisutn
7	29	92	91	0
8	31	92	81	0
9	32	92	100	0
10	33	92	77	0
11	34	92	80	0
12	34	109	86	0
13	35	92	65	0
14	36	92	50	0

Рис. 22 Результат до видалення

```
select * from Reiting
WHERE Kod_student = 109
```

K_zapis	Kod_student	Reiting	Prisutn
---------	-------------	---------	---------

Рис. 23 Результат після видалення

—→Запит на знищення даних з таблиці «Рага» за визначеним кодом дисципліни (у поле параметра вводиться назва дисципліни).¶

	K_predm_pl	K_predmet	K_navch_plan	Chas_Lek	Cahs_pr	Chas_all	Chas_Labor	Chas_sem	Kilk_modul	Cahs_sam	Semestr	Shifr_predmet	k_fk
170	356	418	13	NULL	NULL	135	NULL	NULL	NULL	NULL	5	.10	1
171	357	424	13	NULL	NULL	324	NULL	NULL	NULL	NULL	2	.11	2
172	358	431	13	NULL	NULL	126	NULL	NULL	NULL	NULL	5	.13	2
173	359	391	13	NULL	NULL	108	NULL	NULL	NULL	NULL	8	.14	1
174	360	407	13	NULL	NULL	54	NULL	NULL	NULL	NULL	8	.15	2
175	361	416	13	NULL	NULL	144	NULL	NULL	NULL	NULL	7	.16	1
176	362	474	13	NULL	NULL	126	NULL	NULL	NULL	NULL	7	.17	2
177	363	456	13	NULL	NULL	144	NULL	NULL	NULL	NULL	7	.18	1
178	364	352	13	NULL	NULL	162	NULL	NULL	NULL	NULL	6	NULL	6
179	365	429	13	NULL	NULL	216	NULL	NULL	NULL	NULL	8	NULL	6
180	366	351	13	NULL	NULL	324	NULL	NULL	NULL	NULL	8	NULL	5
181	368	482	17	5	40	1500	750	750	2	2	2	DB	5

Рис. 24 До видалення

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.22.121.05.000 –Лр.4-5	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

```
DELETE FROM Predmet_plan
WHERE K_predmet in (SELECT K_predmet FROM predmet WHERE Nazva = 'DataBase');
```

	K_predm_pl	K_predmet	K_navch_plan	Chas_Lek	Chas_pr	Chas_all	Chas_Labor	Chas_sem	Kilk_modul	Chas_sam	Semestr	Shifr_predmet	k_fk
169	355	345	13	NULL	NULL	324	NULL	NULL	NULL	NULL	1	.09	1
170	356	418	13	NULL	NULL	135	NULL	NULL	NULL	NULL	5	.10	1
171	357	424	13	NULL	NULL	324	NULL	NULL	NULL	NULL	2	.11	2
172	358	431	13	NULL	NULL	126	NULL	NULL	NULL	NULL	5	.13	2
173	359	391	13	NULL	NULL	108	NULL	NULL	NULL	NULL	8	.14	1
174	360	407	13	NULL	NULL	54	NULL	NULL	NULL	NULL	8	.15	2
175	361	416	13	NULL	NULL	144	NULL	NULL	NULL	NULL	7	.16	1
176	362	474	13	NULL	NULL	126	NULL	NULL	NULL	NULL	7	.17	2
177	363	456	13	NULL	NULL	144	NULL	NULL	NULL	NULL	7	.18	1
178	364	352	13	NULL	NULL	162	NULL	NULL	NULL	NULL	6	NULL	6
179	365	429	13	NULL	NULL	216	NULL	NULL	NULL	NULL	8	NULL	6
180	366	351	13	NULL	NULL	324	NULL	NULL	NULL	NULL	8	NULL	5

Рис. 25 Після видалення

→Запит на оновлення даних у таблиці «Reiting» передбачити збільшення балів за модульні контролю на 15%.¶

```
UPDATE Reiting
SET Reiting = Reiting * 1.15
```

	K_zapis	Kod_student	Reiting	Prisutn
1	10	92	58	0
2	16	92	74	0
3	17	92	44	0
4	18	62	88	0
5	19	92	58	0
6	28	92	96	0
7	29	92	104	0

Рис. 26 Результат виконання

→Запит на оновлення даних в таблиці «Reiting» передбачити зменшення балів за іспит на 15%.¶

```
UPDATE Reiting
SET Reiting = Reiting * 0.85
```

	K_zapis	Kod_student	Reiting	Prisutn
1	10	92	49	0
2	16	92	62	0
3	17	92	37	0
4	18	62	74	0
5	19	92	49	0
6	28	92	81	0

Рис. 27 Результат виконання

→Запит на вставку даних до таблиці «Reiting» передбачити вставку даних студентів визначеної групи (код пари та початковий бал задається динамічно).¶

```

DECLARE @K_group varchar(10) = 'c-53';
DECLARE @Reiting smallint = 76;

INSERT INTO Reiting (K_zapis, Kod_student, Reiting, Prisutn)
SELECT 36, s.Kod_stud, @Reiting, 2
FROM dbo_student s
JOIN dbo_groups g ON s.Kod_group = g.Kod_group
WHERE g.Kod_group = @K_group;

```

	K_zapis	Kod_student	Reiting	Prisutn
16	36	49	76	1
17	36	50	76	1
18	36	51	76	1
19	36	52	76	1
20	36	53	76	1
21	36	54	76	1
22	36	55	76	1
23	36	56	76	1
24	36	57	76	1
25	36	58	76	1
26	36	59	76	1
27	36	60	76	1

Рис. 28 Результат виконання

→ Запит на вставку даних до таблиці «Рага» – передбачити вставку всіх дисциплін, назва яких починається з літери «М» (дата заняття та години задаються динамічно, однакові для всіх). ¶

```

INSERT INTO Predmet_plan(K_predmet, K_navch_plan, Chas_Lek, Chas_pr, Chas_all, Chas_Labor,
Chas_sem, Kilk_modul, Chas_sam, Semestr, Shifr_predmet, k_fk)
SELECT p.K_predmet, 17, @ALLHOURS / 2, @ALLHOURS / 2, @ALLHOURS, @ALLHOURS / 2, @ALLHOURS / 2,
5, null, 1, null, 4
FROM predmet p
WHERE Nazva LIKE 'M%';

```

Results		Messages											
176	362	474	13	NULL	NULL	144	NULL	NULL	NULL	NULL	7	.17	1
177	363	456	13	NULL	NULL	144	NULL	NULL	NULL	NULL	7	.18	1
178	364	352	13	NULL	NULL	162	NULL	NULL	NULL	NULL	6	NULL	6
179	365	429	13	NULL	NULL	216	NULL	NULL	NULL	NULL	8	NULL	6
180	366	351	13	NULL	NULL	324	NULL	NULL	NULL	NULL	8	NULL	5
181	369	483	17	750	750	1500	750	750	5	NULL	1	NULL	4
182	370	484	17	750	750	1500	750	750	5	NULL	1	NULL	4

	K_predmet	Nazva	Nazva_skor
135	477	?	?
136	478	?	
137	479	?	
138	480		
139	481		
140	482	DataBase	DB
141	483	Math	math
142	484	Management	mang

Рис. 29 Результат виконання

→Запит на оновлення даних → передбачити зміну порядкового номера змістовного модуля за певною дисципліною на нове значення.¶

```
UPDATE Predmet_plan
SET Kilk_modul = 5
WHERE K_predmet = 388
```

	K_predm_pl	K_predmet	K_navch_plan	Chas_Lek	Chas_pr	Chas_all	Chas_Labor	Chas_sem	Kilk_modul	Chas_sam	Semestr	Shifr_predmet	k_fk
1	86	388	17	NULL	NULL	108	NULL	NULL	5	NULL	1	.01	1

Рис. 30 Результат виконання

→Передбачити знищення студентів з таблиці «Students» за визначеним номером групи.¶

```
DELETE FROM dbo_student
WHERE Kod_group = 'c-54'
```

	Kod_stud	Name_ini	Sname	Name	Fname	N_ingroup	Kod_group
1	46	..				1	?-53
2	47	..			?	2	?-53
3	48	..				3	?-53
4	49	..				4	?-53
5	50	..				5	?-53
6	51	..				6	?-53
7	52	..				7	?-53
8	53	..				8	?-53
9	54	..				9	?-53
10	55	..				10	?-53
11	56	..				11	?-53
12	57	..				12	?-53
13	58	..				13	?-53
14	59	..				14	?-53
15	60	..				15	?-53
16	61	..				16	?-53
17	62	? ..	?			17	?-53
18	63	..			?	18	?-53
19	64	..?			?	19	?-53
20	109	Davyde...	Davy...	Artem		2	?-53

Рис. 31 Результат виконання

→Запит на вставку даних до таблиці «Reiting» → передбачити вставку даних студентів визначеної групи (код пари та присутність задаються динамічно).¶

```
DECLARE @Kod_group varchar(10) = 'c-54'
DECLARE @Prisutn int = 0

INSERT INTO Reiting(K_zapis, Kod_student, Reiting, Prisutn)
SELECT 36, s.Kod_stud, 66, @Prisutn
FROM dbo_student s
JOIN dbo_groups g ON s.Kod_group = g.Kod_group
WHERE g.Kod_group = @Kod_group;
```

21	36	119	66	0
22	36	120	66	0
23	36	121	66	0
24	36	122	66	0
25	36	123	66	0
26	36	124	66	0
27	36	125	66	0
28	36	126	66	0
29	36	127	66	0
30	36	128	66	0
31	36	129	66	0
32	36	130	66	0
33	36	131	66	0

Рис. 32 Результат виконання

→ Передбачити оновлення даних у таблиці «Reiting» – поле «Prisutnist», для студента із визначеним прізвищем автоматично встановлюється true.¶

```
UPDATE Reiting
SET Prisutn = 1
WHERE Kod_student IN (SELECT Kod_stud FROM dbo_student WHERE Sname = 'Vasiuta');
```

18	18	63	37	0
19	18	64	42	0
20	19	109	88	1
21	36	119	66	0

Рис. 33 Результат виконання

4. Створити запити до БД у відповідності до індивідуального завдання.¶

→ Запити, що наводяться маркованим переліком у варіанті¶

→ Запити на вставку та оновлення даних – для одного запису в кожній таблиці¶

```
INSERT INTO Departments (Department_ID, Department_Name) VALUES (4, 'Відділ кадрів');
```

```
INSERT INTO Employees (Employee_ID, Last_Name, First_Name, Middle_Name, Birth_Year,
Employment_Date, Position, Salary, Department_ID, Identification_Code)
VALUES (4, 'Ковальчук', 'Марія', 'Василівна', 1987, '2023-06-12', 'HR менеджер', 2800.00, 4,
'13579');
```

```
INSERT INTO Sick_Leaves (Sick_Leave_ID, Employee_ID, Start_Date, End_Date) VALUES (3, 1,
'2024-04-01', '2024-04-05');
```

```
INSERT INTO Business_Trips (Trip_ID, Employee_ID, Start_Date, End_Date) VALUES (3, 1, '2024-
05-10', '2024-05-15');
```

```
UPDATE Departments SET Department_Name = 'Відділ персоналу' WHERE Department_ID = 4;
```

```
UPDATE Employees SET Position = 'HR спеціаліст' WHERE Employee_ID = 4;
```

```
UPDATE Sick_Leaves SET End_Date = '2024-04-10' WHERE Sick_Leave_ID = 3;
```

```
UPDATE Business_Trips SET End_Date = '2024-05-20' WHERE Trip_ID = 3;
```

	Department_ID	Department_Name								
1	1	Відділ розробки								
2	2	Відділ маркетингу								
3	3	Бухгалтерія								
4	4	Відділ персоналу								

	Employee_ID	Last_Name	First_Name	Middle_Name	Birth_Year	Employment_Date	Position	Salary	Department_ID	Identification_Code
1	1	Іванов	Петро	Олександрович	1985	2020-01-15	Програміст	3000.00	1	12345
2	2	Петров	Олег	Іванович	1990	2018-03-20	Менеджер з продажу	2500.00	2	67890
3	3	Сидорова	Олена	Петрівна	1988	2019-07-10	Головний бухгалтер	3500.00	3	54321
4	4	Ковальчук	Марія	Василівна	1987	2023-06-12	HR спеціаліст	2800.00	4	13579

	Sick_Leave_ID	Employee_ID	Start_Date	End_Date
1	1	1	2024-03-10	2024-03-20
2	2	3	2024-03-05	2024-03-10
3	3	1	2024-04-01	2024-04-10

	Trip_ID	Employee_ID	Start_Date	End_Date
1	1	2	2024-02-15	2024-02-20
2	2	3	2024-03-01	2024-03-03
3	3	1	2024-05-10	2024-05-20

Рис. 34 Результат виконання

—→Запит на вставку та оновлення даних - для множини записів (можна створювати додаткові таблиці)¶

```
INSERT INTO Employees (Employee_ID, Last_Name, First_Name, Middle_Name, Birth_Year,
Employment_Date, Position, Salary, Department_ID, Identification_Code)
VALUES
(5, 'Семенов', 'Олексій', 'Павлович', 1982, '2024-04-01', 'Програміст', 3200.00, 1, '24680'),
(6, 'Ковальов', 'Іван', 'Олегович', 1995, '2024-03-15', 'Маркетолог', 2700.00, 2, '35791');
```

```
UPDATE Sick_Leaves
SET End_Date = '2024-04-15'
WHERE Employee_ID IN (1, 2);
```

```
UPDATE Business_Trips
SET End_Date = '2024-05-25'
WHERE Employee_ID = 2;
```

	Sick_Leave_ID	Employee_ID	Start_Date	End_Date
1	1	1	2024-03-10	2024-04-15
2	2	3	2024-03-05	2024-04-15
3	3	1	2024-04-01	2024-04-10

	Trip_ID	Employee_ID	Start_Date	End_Date
1	1	2	2024-02-15	2024-05-25
2	2	3	2024-03-01	2024-03-03

Рис. 35 Результат виконання

—→Запити на пошук побудовані в лабораторній роботі №3·
модифікувати на запити із параметрами що змінюються динамічно.¶

```
DECLARE @DepartmentID INT = 3;
DECLARE @BirthYearThreshold INT = 1990;

SELECT *
FROM Employees
WHERE Department_ID = @DepartmentID AND Birth_Year < @BirthYearThreshold;

DECLARE @SalaryThreshold DECIMAL(10, 2) = 2500.00;
DECLARE @PositionToSearch VARCHAR(100) = 'Програміст';

SELECT *
FROM Employees
WHERE Salary > @SalaryThreshold AND Position = @PositionToSearch;

DECLARE @StartDateFrom DATE = '2024-03-01';
DECLARE @StartDateTo DATE = '2024-03-31';

SELECT *
FROM Sick_Leaves
WHERE Start_Date BETWEEN @StartDateFrom AND @StartDateTo;

DECLARE @EmployeeCountThreshold INT = 2;

SELECT Departments.Department_Name, COUNT(*) AS Employee_Count
FROM Departments
JOIN Employees ON Departments.Department_ID = Employees.Department_ID
GROUP BY Departments.Department_Name
HAVING COUNT(*) < @EmployeeCountThreshold;
```

	Employee_ID	Last_Name	First_Name	Middle_Name	Birth_Year	Employment_Date	Position	Salary	Department_ID	Identification_Code
1	3	Сидорова	Олена	Петрівна	1988	2019-07-10	Головний бухгалтер	3500.00	3	54321

	Employee_ID	Last_Name	First_Name	Middle_Name	Birth_Year	Employment_Date	Position	Salary	Department_ID	Identification_Code
1	1	Іванов	Петро	Олександрович	1985	2020-01-15	Програміст	3000.00	1	12345
2	5	Семенов	Олексій	Павлович	1982	2024-04-01	Програміст	3200.00	1	24680

	Sick_Leave_ID	Employee_ID	Start_Date	End_Date
1	1	1	2024-03-10	2024-04-15
2	2	3	2024-03-05	2024-04-15

	Department_Name	Employee_Count
1	Бухгалтерія	1
2	Відділ персоналу	1

Рис. 36 Результат виконання

Висновок: дослідив властивості DML-інструкції SELECT для групування даних та відображення проміжних підсумків.