

Лабораторна робота № 7

Додаткові можливості обробки даних. Запити на модифікацію даних

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

Хід роботи:

Звіт містить:

1. Тему та мету лабораторної роботи
2. Скрипти самостійно створених процедур та тригерів до БД «Рейтинг» та результати їх виконання
3. Скрипти самостійно створених процедур та тригерів до БД за індивідуальним варіантом та результати їх виконання.

Побудувати запити до БД «Рейтинг» (див. рис. 1.9):

1. 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. 1 Результат до видалення

```
select * from Reiting
```

```
WHERE Kod_student = 109
```

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

Рис. 2 Результат після видалення

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.22.121.05.000 –Лр.7		
Змн.	Арк.	№ докум.	Підпис	Дата	Звіт з лабораторної роботи №7		
Розроб.		Васюта В.В.					
Перевір.		Коротун О.В.					
Реценз.							
Н. Контр.							
Зав.каф.		Єфіменко А.А.					
					Лім.	Арк.	Акрушів
						1	8
					ФІКТ, гр. ІПЗ-22-1		

2. Запит на знищення даних з таблиці «Para» 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. 3 До видалення

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	Cahs_pr	Chas_all	Chas_Labor	Chas_sem	Kilk_modul	Cahs_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. 4 Після видалення

3. Запит на оновлення даних у таблиці «Reiting» – передбачити збільшення балів за грудень на 15%.

4. Запит на оновлення даних в таблиці «Reiting»– передбачити зменшення балів за іспит 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. 5 Результат виконання

```
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. 6 Результат виконання

5. 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. 7 Результат виконання

6. 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).

```
DECLARE @ALLHOURS int = 1500;
INSERT INTO Predmet_plan(K_predmet, K_navch_plan, Chas_Lek, Cahs_pr, Chas_all, Chas_Labor,
Chas_sem, Kilk_modul, Cahs_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. 8 Результат виконання

7. Запит на оновлення даних – передбачити зміну порядкового номера відомості 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	Cahs_pr	Chas_all	Chas_Labor	Chas_sem	Kilk_modul	Cahs_sam	Semestr	Shifr_predmet	k_fk
1	86	388	17	NULL	NULL	108	NULL	NULL	5	NULL	1	.01	1

Рис. 9 Результат виконання

8. 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. 10 Результат виконання

9. 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. 11 Результат виконання

10. 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. 12 Результат виконання

Для БД за індивідуальним завданням реалізувати:

1. Запити на вставку та оновлення даних – для одного запису

```
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. 13 Результат виконання

2. 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. 14 Результат виконання

3. 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. 15 Результат виконання

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