

## Лабораторна робота №3

### Тема: Прості запити. Групові операції. Використання агрегатних функцій.

#### Теоретичні відомості:

Для формування запитів використовується команда SELECT.

```
SELECT [DISTINCT] перелік_полів  
FROM перелік_таблиць  
[WHERE умова]  
[GROUP BY ім'я_поля [HAVING умова_для_групи]]  
[ORDER BY перелік_полів];
```

де **DISTINCT** – для виключення появи однакових рядків;

**SELECT** *перелік\_полів* – перелік полів і допустимих виразів над полями, які включаються у результуючу таблицю. Елементи списку розділяються комами. Для заміни списку всіх полів використовують \*. Якщо використовується декілька таблиць, тоді перед назвою поля необхідно вказувати назву або псевдонім таблиці. Назва або псевдонім таблиці і назва поля розділяються крапкою. Назву поля у результуючій таблиці можна змінити за допомогою ключового слова AS

*ім'я\_таблиці.поле* **AS** *нове\_ім'я\_поля*

Після службового слова **FROM** через кому вказується перелік таблиць. Для будь-якої таблиці через пропуск можна задати її друге ім'я (псевдонім).

#### Приклад:

Вивести прізвища студентів з таблиці STUDENTS. Вивід прізвищ відбувається у стовпець з назвою SURNAME. В запиті для таблиці STUDENTS задано псевдонім ST.

```
SELECT ST.SFAM AS SURNAME  
FROM STUDENTS ST;
```

**ORDER BY** *перелік\_полів* – задає сортування записів в результуючій таблиці на основі одного чи декількох полів. Поле можна задавати числом, що вказує на положення цього поля у результуючій таблиці. Для кожного поля можна вказати ключ DESC для сортування записів у порядку спадання або ключ ASC для сортування у порядку зростання. По замовчуванню встановлюється сортування по зростанню.

**GROUP BY** *ім'я\_поля* – групує запити по вказаному полю. Якщо необхідно вивести запити у групі, які задовольняють певній умові, використовують оператор **HAVING**.

**WHERE** *умова* – задає умови для вибору записів.

В умовах розділу **WHERE** можуть використовуватись оператори

1) =, <>, >, <, >=, <=

2) **AND** (логічне і), **OR** (логічне або), **NOT** (логічне заперечення).

3) Оператор **IN**(множина) визначає набір допустимих значень, розділених комами.

#### Приклад:

Вивести прізвища студентів, які отримують стипендію 25 і 17.

```
SELECT SFAM  
FROM STUDENTS  
WHERE STIP IN (17, 25).
```

Символьні значення в наборі беруться в одиночні лапки.

4) Оператор **BETWEEN A AND B** визначає діапазон значень, в який має входити шукане значення.

**Приклад:**

Вивести інформацію про студентів, які склали іспити у період з 09.06.2005 по 12.06.2005.

```
SELECT SNOM
FROM USPISH
WHERE DATA BETWEEN '09.06.2005' AND '12.06.2005'
```

5) Оператор **LIKE** використовується для порівняння символічних даних відповідно до шаблону. Шаблон задається за допомогою одинарних лапок. Дозволяється використання наступних групових символів:

- знак підкреслення **\_** замінює один символ;
- знак відсотків **%** замінює будь-яку послідовність символів;
- **[a-m]** – означає, що в даній позиції має бути один символ з вказаного діапазону;
- **[!safd]** – означає, що в даній позиції має бути один символ, який не входить в даний діапазон.

**Приклад:**

Вивести список дисциплін, які у назві містять слово „програмування”.

```
SELECT PNAME
FROM PREDMET
WHERE PNAME LIKE '%ПРОГРАМУВАННЯ%';
```

6) Оператор **IS [NOT] NULL** дозволяє вивести поля, які мають або не мають (NOT) невизначені значення.

**Приклад:**

Вивести номери залікових книжок студентів, які по різних причинам не здавали іспит.

```
SELECT SNOM
FROM USPISH
WHERE OCINKA IS NULL;
```

### Функції

1) **COUNT(ім'я\_поля)** – підрахунок кількості рядків або не-NULL значень поля.

Допускає використання **DISTINCT**, **ALL** або **\***.

**COUNT(DISTINCT ім'я\_поля)** – підрахунок не-NULL значень поля, виключаючи дублікати.

**COUNT([ALL] ім'я\_поля)** – підрахунок не-NULL значень поля, включаючи дублікати.

**COUNT(\*)** – підрахунок кількості рядків, включаючи NULL значення полів.

**Приклад:**

Підрахувати кількість записів у таблиці **STUDENTS**:

```
SELECT COUNT(*) FROM STUDENTS;
```

**Приклад:**

Підрахувати кількість студентів у таблиці **USPISH** без повторень.

```
SELECT COUNT(DISTINCT SNOM) FROM USPISH;
```

- 2) **SUM**(ім'я\_поля) – розраховує суму усіх значень вибраного поля;
- 3) **AVG**(ім'я\_поля) – знаходить середнє арифметичне значень вибраного поля;
- 4) **MAX**(ім'я\_поля) – знаходить максимальне значення серед значень вибраного поля;
- 5) **MIN**(ім'я\_поля) – знаходить мінімальне значення серед значень вибраного поля;
- 6) **UPPER**(ім'я\_поля) – перетворює символну стрічку до верхнього регістра;
- 7) **CAST**(ім'я\_поля **AS** тип\_даних) – перетворює дані поля до певного типу даних. Використовується в умовах пошуку для порівняння даних.
- 8) **EXTRACT**(частина **FROM** ім'я\_поля) – вибирає інформацію про дату і час з даних типу **DATE**, **TIME**, **TIMESTAMP**.

*Частина* – може бути **YEAR** (значення – 0-5400), **MONTH** (1-12), **DAY** (1-31), **HOURL** (1-23), **MINUTE** (1-59), **SECOND** (0-59.9999), **WEEKDAY** (0-6, 0 – неділя, 1 – понеділок і т.д.), **YEARDAY** (1-366).

Тип даних, які є результатом функції – **SMALLINT**, для секунд – **DECIMAL**(6,4).

#### Приклад:

Вивести інформацію про всіх студентів, які здали іспити у місяці червні.

```
SELECT SNOM FROM USPISH
WHERE EXTRACT(MONTH FROM DATA)=6;
```

#### **Завдання до виконання:**

- 1) Завантажте програму *IBExpert*.
- 2) Відкрийте базу даних **Univer**.
- 3) Використовуючи команду **SELECT** у командному вікні *SQL Editor*, виконайте наступні запити:
  1. Вивести на екран прізвища, імена та розмір стипендії всіх студентів. Дані відсортувати в алфавітному порядку по прізвищах.
  2. Вивести розрахунок стипендії після підвищення її у два рази для кожного студента. Задати назву для нового стовпця **NSTIP**.
  3. Вивести на екран прізвища та імена студентів, стипендія яких дорівнює 75.
  4. Вивести на екран номери залікових книжок студентів, які не мають трійок по певному предмету.
  5. Вивести інформацію про студентів, які мають 4 і 5 по певному предмету.
  6. За допомогою оператора **IN** вивести прізвища студентів, імена яких *Анатолій* або *Володимир*.
  7. Вивести інформацію про студентів, прізвища яких починаються з літер із діапазону „А” – „М”.
  8. Вивести номери залікових книжок студентів, які складала іспити від 1.06.2005 до 15.06.2005.
  9. Вивести номери залікових книжок студентів, які складала іспити в червні.
  10. Вивести номери залікових книжок студентів, які не принесли фото.
  11. Вивести номери залікових книжок студентів, які не здали іспити (отримали незадовільні оцінки). Дані не повинні повторюватись.

12. Вивести список викладачів, прізвища яких починаються на літеру „B”.
  13. Вивести список дисциплін, в назві яких є слово „бази”.
  14. Підрахувати кількість записів у таблиці **STUDENTS**.
  15. Підрахувати суму стипендій, яка виплачується на групу
  16. Підрахувати кількість студентів, які отримують стипендію.
  17. Вивести середній бал успішності по таблиці **USPISH**.
  18. Вивести середній бал кожного студента.
  19. Вивести мінімальну і максимальну оцінки для кожного студента по результатам сесії.
  20. Вивести мінімальну позитивну оцінку для кожного студента.
  21. Підрахувати кількість студентів, які склали іспит з кожного предмета (на будь-яку оцінку).
  22. Підрахувати кількість студентів, які склали іспит з кожного предмета позитивно.
  23. Підрахувати кількість студентів, які навчаються на бюджеті і платній формі.
- 4) Завершіть роботу.

#### **Контрольні запитання:**

- 1) Яка команда використовується для формування запитів?
- 2) Як задати сортування записів в запитах?
- 3) Як задаються умови відбору для запитів?
- 4) Для чого використовується опція **GROUP BY**?
- 5) Які агрегатні функції Ви знаєте?
- 6) Яка функція використовується для роботи з даними типу **DATE**?