Заняття 5. Формування SQL-запиту на вибірку. Використання агрегатних функцій. Підзапити

ЛОГОС
Центр комп'ютерного навчання

Визначення SQL-запиту на вибірку

- **Запит на вибірку даних** – це команда або інструкція, яка дається СУБД, щоб вона вивела певну інформацію з таблиць бази даних.

ЛОГОС
Центр комп'ютерного навчання

Загальний синтаксис інструкції SELECT

- SELECT [ALL | DISTINCT] {* |
  вираз-стовпець [AS псевдонім] [, ...]}
  FROM таблиця [, ...]
  ![WHERE умова пошуку]
  ![GROUP BY список стовпців групування]
  ![HAVING умова пошуку] вийняткові
  ![ORDER BY список стовпців сортування
  умова сортування];

ЛОГОС
Центр комп'ютерного навчання

Порядок виконання SQL-запиту на вибірку

- **FROM** – СУБД вибирає таблицю з бази даних.
- **WHERE** – з таблиці вибираються записи, які відповідають умові пошуку, і відкидаються решта (фільтр записів).
- **GROUP BY** – створюються групи записів, відібраних оператором WHERE (якщо він є в SQL-виразі), і кожна група відповідає якому-небудь значенню стовпця групування. Стовпець групування може бути будь-яким стовпцем таблиці, заданій в операторі FROM, а не лише тими, які вказані у виразі SELECT.
- **HAVING** – опрацьовує кожну із створених груп записів, залишаючи лише ті з них, які задовольняють умові. Цей оператор використовується лише разом з оператором GROUP BY.
- **SELECT** – вибирає з таблиці, віртуально створеної в результаті застосування наведених операторів, лише вказані стовпці.
- **ORDER BY** – сортує записи таблиці. При цьому в умову сортування можна вказувати лише ті стовпці, які вказані в операторі SELECT.

ЛОГОС
Центр комп'ютерного навчання

Усунення надлишковості вибраних даних

- Ключове слово ***DISTINCT*** (ВІДМІННІСТЬ) усуває повторювані значення з команди SELECT:

  SELECT DISTINCT стовп_1, …

  FROM таблиця;

- DISTINCT слідкує за тим, які значення стовп_1 були раніше, щоб вони не дублювались у результатній таблиці.

ЛОГОС
Центр комп'ютерного навчання

Усунення надлишковості вибраних даних в MySQL

- **ALL**
- **DISTINCT**
- **LIMIT** (вказується вкінці запиту)

ЛОГОС
Центр комп'ютерного навчання

Основні типи умов пошуку (предикатів)

- **порівняння** – порівнюються результати обчислення одного виразу з результатами обчислення іншого виразу;
- **діапазон** – перевіряється, чи попадає результат обчислення виразу у заданий діапазон значень;
- **належність до множини** – перевіряється, чи належить результат обчислення виразу до заданої множини значень;
- **відповідність шаблону** – перевіряється, чи відповідає деяке символьне значення заданому шаблону;
- **існування** – перевіряється чи існує хоча б один рядок, який задовольняє умові;
- **перевірка на невизначене значення** – перевіряється, чи містить заданий стовпець значення NULL.

ЛОГОС
Центр комп'ютерного навчання

Спеціальні SQL-предикати. Належність до множини

- **IN** та **NOT IN** визначає список значень, в який може або не може входити дане значення стовпця
- WHERE стовп IN (зн_1, …);
- Альтернативою є поєднання предикатів порівняння з логічною операцією OR.

ЛОГОС
Центр комп'ютерного навчання

Спеціальні SQL-предикати. Предикат діапазону

- **BETWEEN** визначає діапазон значень, в який має попадати задане значення стовпця. Включає граничні значення у діапазон.
- WHERE стовп BETWEEN зн_1 AND зн_2;
- На відміну від оператора IN, оператор BETWEEN є чутливим до порядку, тобто першим має бути менше значення (як символьне так і числове).
- Має особливості роботи з символьними значеннями!!!

ЛОГОС
Центр комп'ютерного навчання

Спеціальні SQL-предикати. Предикат шаблона

- **LIKE** (подібний) та **NOT LIKE** (не подібний) застосовуються тільки до полів типу CHAR або VARCHAR, в яких вони знаходить підстрічки. В якості умови вони використовують групові символи, або маски, яких є два типи:
  - символ підкреслення (_), який заміняє одиничний символ;
  - знак процента (%), який заміняє послідовність символів довільної довжини.
- WHERE стовп LIKE ' ';

ЛОГОС
Центр комп'ютерного навчання

Спеціальні SQL-предикати. Предикат існування

- **EXISTS** та **NOT EXISTS** – предикати, які повертають значення TRUE або FALSE, і які можна застосовувати окремо або разом з іншими булевими виразами.
- EXISTS не може використовувати агрегатні функції у своєму підзапиті.
- У зв'язаних підзапитах предикат EXISTS виконується для кожного рядка зовнішньої таблиці.
- Можна комбінувати предикат EXISTS із з'єднаннями таблиць.
- ```
  SELECT * FROM Customer
  WHERE EXISTS
  (SELECT * FROM Customer
  WHERE City = 'SanJose');
  ```

ЛОГОС
Центр комп'ютерного навчання
  ```

Спеціальні SQL-предикати. Перевірка на значення NULL

- **IS NULL** застосовується для виявлення записів, в яких той чи інший стовпець має невідоме значення.
- **IS NOT NULL** застосовується, коли необхідно виключити з результатів запис з NULL-значеннями.
- WHERE стовп IS NULL;

ЛОГОС
Центр комп'ютерного навчання

Аргументи GROUP BY та HAVING

- **GROUP BY** служить для групування записів за значеннями одного або декількох стовпців.
- Якщо в SQL-виразі використовується оператор WHERE, який задає фільтр записів, то оператор GROUP BY знаходиться і виконується після нього.
- Для визначення, які записи повинні увійти в групи, служить оператор HAVING, який використовується разом з GROUP BY.
- Якщо оператор **HAVING** не використовується, то групуванню підлягають усі записи, відфільтровані оператором WHERE.
- Якщо WHERE не використовується, то групуються усі записи таблиці.

ЛОГОС
Центр комп'ютерного навчання

Аргумент ODER BY

- Застосовується для упорядкування (сортування) записів.
- Записується і виконується вкінці запиту.
- Сортує записи усієї таблиці або окремих її груп, у випадку застосування оператора GROUP BY.
- Після імені стовпця групування можна вказувати ключове слово, яке задає режим сортування: ***ASC*** – за зростанням (за замовчуванням) і ***DESC*** – за спаданням.
- Усі стовпці, які впорядковуються, повинні вказуватись у виразі SELECT.

ЛОГОС
Центр комп'ютерного навчання

Агрегатні (статистичні) функції в SQL

- **COUNT** (параметр) – обчислює кількість записів, вказаних у параметрі.
  - Якщо необхідно отримати кількість усіх записів, то в якості параметра вказується * (або унікальний ідентифікатор).
  - Якщо в якості параметра вказано ім'я стовпця, то функція поверне кількість записів, в яких цей стовпець має не NULL значення.
  - Щоб знати, скільки різних значень має стовпець, перед його іменем вказується ключове слово DISTINCT.

ЛОГОС
Центр комп'ютерного навчання

- **SUM** (параметр) – обчислює суму значень стовпця, вказаного як параметр.
- **AVG** (параметр) – обчислює середнє арифметичне значень стовпця, вказаного в параметрі.
- Параметр може представляти собою вираз, який містить ім'я стовпця. Тоді використання DISTINCT не дозволяється.
- З функціями SUM та AVG можуть використовуватись лише числові поля.

ЛОГОС
Центр комп'ютерного навчання

- **MAX** (параметр) – обчислює найбільше з усіх вибраних значень стовпця.
- **MIN** (параметр) – обчислює найменше з усіх вибраних значень стовпця.
- З функціями COUNT, MAX і MIN можуть використовуватись і числові, і символьні поля.

ЛОГОС
Центр комп'ютерного навчання

Використання агрегатних функцій з групуванням

- Фраза GROUP BY дозволяє визначати підмножину значень в деякому стовпці і застосовувати агрегатну функцію до цієї підмножини. Потрібно утворювати групи і лише тоді виконувати певні операції. Приклад – групувати продавця, і тоді рахувати певні операції з ним.
- Тоді необхідно оголосити стовпці та агрегатні функції у фразі SELECT.

ЛОГОС
Центр комп'ютерного навчання

- Приклад 1. Знайти найбільшу суму, отриману кожним продавцем:
- SELECT snum, MAX(amt)
  FROM Orders
  GROUP BY snum;
- Результат:

1001	9891.88
1002	5160.45
1003	1713.23
1004	1900.10
1007	1098.16

ЛОГОС
Центр комп'ютерного навчання

- Приклад 2. Знайти найбільший платіж, який проводив кожен продавець кожен день:
- SELECT snum, odate, MAX(amt)

  FROM Orders

  GROUP BY snum, odate;

ЛОГОС
Центр комп'ютерного навчання

- Результат:

1001	03-09-2011	767.19
1001	05-09-2011	4723.00
1001	06-09-2011	9891.88
1002	03-09-2011	5160.45
1002	04-09-2011	75.75
1002	06-09-2011	1309.95
1003	04-09-2011	1713.23
1004	03-09-2011	1900.10
1007	03-09-2011	1098.16

ЛОГОС
Центр комп'ютерного навчання

- Приклад 3. Знайти максимальну суму, отриману кожним продавцем кожного дня, значення якої більше 3 000.00.
  - **Не можна** використовувати агрегатну функцію у фразі WHERE, оскільки предикати оцінюються в термінах одиничного рядка, а агрегатні функції оцінюються в термінах груп рядків. Це означає, що не можна написати в команді SELECT з попереднього прикладу 2 таку фразу:

    WHERE MAX(amt) > 3000.00

ЛОГОС
Центр комп'ютерного навчання

- Правильною буде наступна інструкція:

```
SELECT snum, odate, MAX(amt)
FROM Orders
GROUP BY snum, odate
HAVING MAX(amt) > 3000.00;
```

- Результат:

1001	05-09-2011	4723.00
1001	06-09-2011	9891.88
1002	03-09-2011	5160.45

ЛОГОС
Центр комп'ютерного навчання

- Аргументи у фразі HAVING повинні мати одне значення на групу виводу.
- Неправильна інструкція:

```
SELECT snum, MAX(amt)
FROM Orders
GROUP BY snum
HAVING odate = '2009-10-03';
```

- Поле odate не може використовуватись фразою HAVING, тому що воно може мати (і дійсно має) більше, ніж одне значення на групу виводу.
- ***Щоб уникнути такої ситуації, оператор HAVING повинен використовувати лише агрегатні функції і поля, вибрані оператором GROUP BY.***

ЛОГОС
Центр комп'ютерного навчання

- Приклад 4. Правильний спосіб написання попереднього запиту :

```
SELECT snum, MAX(amt)
FROM Orders
WHERE odate =  '2009-10-03'
GROUP BY snum;
```

- Приклад 5. Вивести найбільші платежі, які провели продавці Serres (1002) і Rifkin (1007):

```
SELECT snum, MAX(amt)
FROM Orders
GROUP BY snum
HAVING snum IN (1002,1007);
```

ЛОГОС
Центр комп'ютерного навчання

Використання підзапитів

- При використанні підзапитів у предикатах, які використовують операції порівняння, необхідно, щоб результат підзапиту видавав лише один рядок. В іншому випадку команда не виконається.
- Якщо в результаті підзапиту не буде ніяких значень, то інструкція виконається, але не видасть ніяких результатів. Предикат, в якому розміщений такий підзапит, є невідомий і має такий ефект як невірний, тому команда не має результатів.

ЛОГОС
Центр комп'ютерного навчання

- Приклад 6. Вивести інформацію про усі операції купівлі-продажу, які обслуговуються продавцем Motika, припустивши, що його номер нам невідомий.

```
SELECT * FROM Orders
WHERE snum =
(SELECT snum  FROM Sellers
WHERE sname = 'Motika');
```

- Стандарт ANSI забороняє додавати для порівняння два підзапити: <підзапит> <оператор> <підзапит>.

ЛОГОС
Центр комп'ютерного навчання

Використання агрегатних функцій у підзапитах

- Приклад 7. Вивести усі операції за 3 жовтня з платежем вище середнього:

```
SELECT * FROM Orders
WHERE odate = '2009-10-03'
AND amt >
(SELECT AVG(amt)
FROM Orders);
```

ЛОГОС
Центр комп'ютерного навчання

- Згруповані агрегатні функції за допомогою оператора GROUP BY, видають декілька значень.
- Таким чином вони не використовуються у підзапитах, навіть, коли оператор GROUP BY або HAVING виводять одну групу.
- У підзапитах необхідно використовувати одиничну агрегатну функцію у виразі WHERE, щоб уникнути небажаних груп.

ЛОГОС
Центр комп'ютерного навчання

- Приклад 8. Запит, який повинен знайти середнє значення комісійних продавців у Лондоні, **не можна** використовувати у підзапиті.

```
SELECT AVG(comm)
FROM Sellers
GROUP BY city
HAVING city = 'London';
```

- Інший спосіб, який **можна** використовувати у підзапиті:

```
SELECT AVG(comm)
FROM Sellers
WHERE city = 'London';
```

ЛОГОС
Центр комп'ютерного навчання

Підзапити, в результаті яких виходить декілька значень

- IN – використовується з підзапитами.
- BETWEEN – не використовується з підзапитами.
- LIKE – не використовується з підзапитами.

ЛОГОС
Центр комп'ютерного навчання

- Приклад 9. Вивести всю інформацію про операції купівлі-продажу для продавців у Лондоні. (Тут використано з підзапитом оператор IN, оскільки інструкція не буде працювати з оператором порівняння):

```
SELECT *  FROM Orders
WHERE snum IN
(SELECT snum  FROM Sellers
WHERE city = 'London');
```

ЛОГОС
Центр комп'ютерного навчання

Підзапити у фразі HAVING

- Такі підзапити можуть використовувати свої агрегатні функції, якщо вони не виводять декількох значень.
- Приклад 10. Порахувати кількість замовників з рейтингом, вищим середнього, у місті San Jose:

```
SELECT rating, COUNT(cnum)
FROM Customers
GROUP BY rating
HAVING rating >
(SELECT AVG(rating)
FROM Customers
WHERE city = 'SanJose');
```

ЛОГОС
Центр комп'ютерного навчання

Завдання 1. Написати запити на вибірку для власної бази даних

- 1 запит, який відображає тільки декілька стовпців із певної таблиці;
- 3 запити з використанням мінімум будь-яких 2 спеціальних предикатів;
- 3 запити із використанням 3 різних агрегатних функцій;
- 3 запити із використанням підзапитів (мінімум один з них з агрегатною функцією).

ЛОГОС
Центр комп'ютерного навчання