# ПРОСТІ ЗАПИТИ

## ІНСТРУКЦІЯ SELECT

Інструкція SELECT отримує інформацію з бази даних та повертає її у виді таблиці результатів запиту.

Синтаксична діаграма інструкції SELECT представлена НИЖЧЕ

SELECT

рядки

, що отримуються

,

\*

FROM

ALL

DISTINCT

специфікатор таблиці

,

WHERE

умова відбору

GROUP BY

ім'я стовпця

,

HAVING

умова відбору

ORDER BY

сп

ецифікатор

сортування

,

*Рис.2. Синтаксична діаграма інструкції SELECT*

В частині SELECT вказується список стовбців, які мають бути отримані у результаті виконання запиту. Стовпці можуть містити значення, отримані з стовпців таблиць бази даних, або можуть обчислюватися під час виконання запиту.

В частині FROM вказується список таблиць, які місять елементи даних, до яких звертається запит.

В частині WHERЕ міститься умова для відбору рядків, які будуть включені у результат запиту.

Блок GROUP BY дозволяє створити підсумковий запит. Звичайний запит включає в результати запиту по одному запису для кожного рядка із таблиці. Підсумковий запит, в свою чергу, спочатку групує рядки бази даних за визначеною ознакою, а потім включає в результат запиту один підсумковий рядок для кожної групи.

Блок HAVING показує, що в результаті запиту, необхідно включити тільки деякі групи, створені за допомогою GROUP BY. В цій частині для відбору груп використовується умова відбору.

Блок ORDER BY впорядковує результати запиту на основі даних, що містяться в одному чи декількох стовпцях.

### Блок SELECT

В частині SELECT необхідно вказати елементи даних, які будуть отримані в результаті виконання запиту. Ці елементи задаються у вигляді списку стовпців, розділених комами. Для кожного елемента із цього списку в таблиці результатів буде створений стовпець. Стовпець результуючої таблиці може являти собою:

* ім'я стовпця, яке відповідає стовпцю однієї з таблиць, які перераховані в частині FROM;
* константу, яка показує, що в кожному рядку результату запиту має бути одне і те ж значення;
* вираз, який показує, що СУБД має обчислити значення за формулою, визначеною у виразі.

Вирази для обчислення значень певних стовпців можуть містити операції додавання, віднімання, множення та ділення. Тут також можна використовувати дужки.

Для того, щоб отримати всі стовпці таблиці, замість списку стовпців можна використовувати символ зірочки (\*).

Якщо із таблиці-результату запиту необхідно прибрати рядки, які містять однакові значення, то в частині SELECT перед списком стовпців необхідно вказати предикат DISTINCT, що забезпечить уникнення повторів при виводі результату.

### Блок FROM

Блок FROM містить список специфікаторів таблиць, розділених комами. Кожен специфіктор таблиці ідентифікує таблицю, що містить дані, які отримує запит.

### Блок WHERE

Для того, щоб вказати які саме рядки необхідно відібрати при виконанні запиту, використоувється Блок WHERE. У ній записують умову відбору рядків. Дял кожного з рядяків умова відбору може мати одне з трьох значень:

* якщо умова має значення TRUE, то рядок включається в результат відбору;
* якщо умова приймає значення FALSE, то рядок виключається з результатів запиту;
* якщо умова має значення NULL, то рядок виключається із результатів відбору.

Існує багато умов відбору, які дозволяють ефективно створювати різні типи запитів. Основними умовами відбору є:

1. Порівняння. Значення одного виразу порівнюється із значенням іншого виразу для кожного рядка даних. Існує шість різних способів порівняння виразів:

= , < > , < , <= , > , >=.

Результататом виконання СУБД порівняння двох виразів може бути:

* якщо порівняння істинне, то результат перевірки має значення TRUE;
* якщо порівняння хибне, то результат перевірки має значення FALSE;
* яякщ хоча б один з двох виразів має значення NULL, то результатом перевірки буде NULL.

2. Перевірка на належність діапазону значень. Перевіряється чи потрапляє вказане значення в визначений діапазон. Схематично таку форму умови відбору можна зобразити так:

*вираз, що перевіряється* BETWEEN *нижня межа* AND *верхня межа* або

*вираз, що перевіряється* NOT BETWEEN *нижня межа* AND *верхня межа*

При такій перевірці верхня та нижня межі вважаються частиною діапазону.

В деяких СУБД визначені такі правила обробки значення NULL в перевірці BETWEEN:

* якщо вираз, що перевіряється має значення NULL або якщо обидва виразів, які визначають діапазон, рівні NULL, то і перевірка BETWEEN повертає NULL;
* якщо вираз, що визначає нижню межу діапазону, має значення NULL, то перевірка BETWEEN повертає FALSE, коли значення, що перевіряється більше, ніж верхня межа діапазону, і NULL в протилежному випадку;
* якщо вираз, що визначає верхню межу діапазону, має значення NULL, то перевірка BETWEEN повертає FALSE, коли значення, що перевіряється менше, ніж нижня межа діапазону, і NULL в протилежному випадку.

1. Перевірка на входження до множини. Перевіряється, чи співпадає значення виразу з одним із значень заданої множини. Схематично таку форму умови відбору можна зобразити так:

*вираз, що перевіряється* IN (*список констант через кому*) або

*вираз, що перевіряється* NOT IN (*список констант через кому*)

1. Перевірка на відповідність шаблону. Перевіряється чи відповідає рядкове значення, яке міститься в стовпці певному шаблону. Схематично таку форму умови відбору можна зобразити так: *ім'я стовпця* LIKE *шаблон* або

*ім'я стовпця* NOT LIKE *шаблон*

Шаблон являє собою рядок, в який може входити один або більше підстановочних знаків. В SQL використовуються такі підстановочні знаки:

1. %  співпадає з будь-якою послідовністю з нуля чи більше символів;
2. \_ (символ підкреслення)  співпадає з будь-яким окремим символом.

У випадку, коли підстановочний знак може виявитися елементом рядка, для побудови шаблону використовуються символи пропуску. Коли, в шаблоні зустрічається такий символ, символ, який слідує за ним, вважається не підстановочним. Структура умови в такому випадку є наступною:

*ім'я стовпця* LIKE *шаблон* ESCAPE *символ пропуску* або

*ім'я стовпця* NOT LIKE *шаблон* ESCAPE *символ пропуску*

Наприклад, якщо шаблон містить символ %, то умова буде такою:

WHERE name LIKE 'A$%BC' ESCAPE '$'

У цьому випадку '$' є символом пропуску і символ %, який слідує після нього є простим елементом рядка.

5. Перевірка на рівність значенню NULL. Значення NULL дозволяє застосовувати тризначну логіку в умовах відбору. У випадку, коли необхідно явно перевірити значення стовпців на рівність NULL використовується така структура умови:

*ім'я стовпця* IS NULL

або

*ім'я стовпця* IS NOT NULL

Дана перевірка завжди повертає значення TRUE або FALSE.

Перераховані прості умови відбору, після застосування до деякого рядка повертають значення TRUE, FALSE або NULL. За допомогою правил логіки ці прості умови можна об’єднувати в більш складні, використовуючи при цьому логічні операції AND, OR, NOT. Їх таблиці істинності наведені нижче:

*Таблиця 5*

###### Таблиця істинності оператора AND

|  |  |  |  |
| --- | --- | --- | --- |
| **AND** | **TRUE** | **FALSE** | **NULL** |
| **TRUE** | TRUE | FALSE | NULL |
| **FALSE** | FALSE | FALSE | FALSE |
| **NULL** | NULL | FALSE | NULL |

*Таблиця 6* **Таблиця істинності оператора OR**

|  |  |  |  |
| --- | --- | --- | --- |
| **OR** | **TRUE** | **FALSE** | **NULL** |
| **TRUE** | TRUE | TRUE | TRUE |
| **FALSE** | TRUE | FALSE | NULL |
| **NULL** | TRUE | NULL | NULL |

*Таблиця 7*

###### Таблиця істинності оператора NOT

|  |  |  |  |
| --- | --- | --- | --- |
| **NOT** | **TRUE** | **FALSE** | **NULL** |
|  | FALSE | TRUE | NULL |

Оператор NOT володіє найвищим пріоритетом, наступний пріоритет має оператор AND, найнижчий  OR.

### Блок ORDER BY

Для впорядкування результатів запиту використовується блок ORDER BY. Структура блоку є такою:

ORDER BY *ім'я чи порядковий номер стовпця* ASC/DESC

При впорядкуванні можна обирати зростаючий чи спадний порядок. За замовчуванням дані сортуються по зростанню. Щоб сортування відбувалося по спаданню, необхідно в речення включити ключове слово DESC.

Відповідно, ключове слово ASC задає впорядкування по зростанню.