**ЛАБОРАТОРНА РОБОТА №7**

**ПРЕДИКАТИ ТА ОПЕРАТОРИ ПОРІВНЯННЯ**

Мета: ознайомлення студентів з предикатами та операторами порівняння в SQL, навчання основам застосування цих інструментів для фільтрації та аналізу даних в реляційних базах даних. Розробка навичок використання предикатів і операторів порівняння для виконання ефективних запитів до бази даних, що дозволить студентам здійснювати точний вибір даних відповідно до заданих умов.

**ТЕОРЕТИЧНІ ОСНОВИ**

У базах даних, предикати та оператори порівняння є фундаментальними інструментами для визначення умов у запитах SQL, які дозволяють фільтрувати, оновлювати або видаляти дані згідно з заданими критеріями. Вони відіграють ключову роль у створенні ефективних і точних запитів до бази даних.

***Оператори порівняння***

Оператори порівняння дозволяють порівнювати два значення або вирази. Стандартні оператори порівняння включають:

- `=` (рівно)

- `>` (більше ніж)

- `<` (менше ніж)

- `>=` (більше ніж або дорівнює)

- `<=` (менше ніж або дорівнює)

- `<>` або `!=` (не дорівнює)

Ці оператори можна застосовувати для порівняння числових значень, рядків, дат та інших типів даних.

***Логічні предикати***

Логічні предикати використовуються для комбінування або інвертування умов порівняння, створюючи складніші умови фільтрації. Основні логічні предикати:

- `AND` - вибирає рядок, якщо обидві умови істинні.

- `OR` - вибирає рядок, якщо хоча б одна з умов істинна.

- `NOT` - інвертує результат умови порівняння.

***Спеціалізовані предикати***

- `BETWEEN` - перевіряє, чи значення лежить в межах вказаного діапазону.

- `IN` - перевіряє, чи значення входить до списку вказаних значень.

- `LIKE` - використовується для пошуку за шаблоном у рядкових даних.

- `IS NULL` - перевіряє, чи значення є NULL.

Ці інструменти можна застосовувати у вибірках (`SELECT`), оновленнях (`UPDATE`), видаленнях (`DELETE`) та інструкціях вставки (`INSERT` при використанні в `SELECT` для визначення значень), дозволяючи виконувати точний вибір та маніпуляцію даними згідно з бізнес-логікою і вимогами безпеки.

Правильне використання предикатів та операторів порівняння має вирішальне значення для оптимізації продуктивності запитів. Неоптимізовані умови можуть призвести до непотрібного сканування великих обсягів даних, що суттєво знижує швидкість виконання запитів і завантажує системні ресурси. Оптимізація запитів з використанням ефективних умов фільтрації та вибору індексованих колонок для порівнянь може значно підвищити продуктивність та забезпечити швидкий доступ до потрібних даних.

Оператори порівняння та логічні предикати є невід'ємною частиною мови SQL, що дозволяє розробникам та аналітикам ефективно маніпулювати даними. Розуміння та вміле використання цих інструментів сприяє створенню оптимізованих, продуктивних та безпечних запитів до баз даних. Водночас, знання принципів оптимізації запитів є критично важливим для підвищення ефективності роботи з реляційними базами даних та забезпечення швидкого доступу до потрібної інформації.

**ЗАВДАННЯ**

***Задача 1***

Напишіть запит, який поверне список офісів східного регіону з ціллю по продажах менше або рівної 350000.00.

- Використовується таблиця [dbo].[OFFICES]

- Результуючий набір даних містить: Ідентифікатор офісу, місто, ідентифікатор керівника офісу

***Задача 2.1***

Напишіть запит, що повертає список замовлень, які були проведені не в 2008 році.

Враховуйте тільки замовлення на товари з ідентифікатором товарів (PRODUCT), які містять «A» другим символом або «0» в будь-якому місці.

- Використовується таблиця [dbo].[ORDERS]

- Задійте предикат LIKE

- Задійте предикат NOT BETWEEN

- Результуючий набір даних містить усі стовпці

***Задача 2.2***

Напишіть запит, що повертає загальну суму проведених замовлень по ідент. виробника товарів (MFR),

які були проведені не в 2008 році. Враховуйте тільки замовлення на товари з ідентифікатором товарів (PRODUCT), які містять

A другим символом або 0 в будь-якому місці.

- Використовується таблиця [dbo].[ORDERS]

- Задійте предикат LIKE

- Задійте предикат NOT BETWEEN

- Задійте агрегатну функцію SUM

- Результуючий набір даних містить: Ідент. виробника товарів, кіль-ть унікальних замовлень

- Відсортуйте результат за загальною сумою (за зростанням)

***Задача 2.3***

Напишіть запит, що повертає ідент. виробника з найбільшою загальною сумою проведених замовлень.

Враховуйте ймовірність того, що одразу кілька виробників можуть мати одну і ту ж загальну суму.

Враховуйте тільки замовлення, які були проведені не в 2008 році.

Враховуйте тільки замовлення на товари з ідентифікатором товарів (PRODUCT), які містять

A другим символом або 0 в будь-якому місці.

- Використовується таблиця [dbo].[ORDERS]

- Задійте предикат LIKE

- Задійте предикат NOT BETWEEN

- Задійте агрегатну функцію SUM

- Задійте фільтр TOP і оператор WITH TIES

- Результуючий набір даних містить: Ідент. виробника товарів, загальна сума

***Задача 3***

Напишіть запит, який поверне ідент. офісу з найбільшою кількістю працівників.

Враховуйте ймовірність того, що відразу кілька офісів можуть мати одну й ту саму кількість працівників.

Враховуйте лише працівників на посаді Sales Rep та у віці 29, 45, 48.

- Використовується таблиця [dbo].[SALESREPS]

- Задіяти предикат IN

- Задіяйте агрегатну функцію COUNT

- Задійте фільтр TOP та оператор WITH TIES

- Результуючий набір даних містить: Ідент. офісу, кількість працівників

**КОНТРОЛЬНІ ЗАПИТАННЯ**

1. Що таке оператор порівняння в SQL? Наведіть приклади операторів порівняння.

2. Як використовуються логічні оператори `AND`, `OR`, та `NOT` у SQL-запитах? Дайте приклади їх застосування.

3. Опишіть, як використовується предикат `LIKE` для пошуку за шаблоном. Як можна використати символи `%` та `\_` у шаблонах?\*\*

4. Що таке предикат `IN` і як він використовується в SQL? Наведіть приклад запиту з використанням `IN`.

5. Як використовується предикат `BETWEEN`? Чи включаються крайні значення діапазону, заданого в `BETWEEN`?

6. Опишіть, як застосовується оператор `IS NULL` у SQL-запитах. Чому не можна використовувати оператори `=` або `<>` для перевірки на `NULL`?

7. Яка різниця між `=` та `IS` при порівнянні значень з `NULL` у SQL?

8. Як можна використати предикат `NOT` для інвертування умови? Наведіть приклад.

9. Поясніть, як використовується `NOT BETWEEN` та які дані він вибирає.

10. Які стратегії можна використати для оптимізації запитів із складними умовами фільтрації?

11. Опишіть, як `TOP` і `WITH TIES` можуть бути використані разом. Які типи сценаріїв це дозволяє обробляти?

12. Як можна використовувати оператори порівняння та предикати для роботи з датами в SQL? Наведіть приклад.

**ДОДАТКОВІ МАТЕРІАЛИ:**

<https://www.w3schools.com/sql/sql_where.asp>

<https://learn.microsoft.com/en-us/sql/t-sql/language-elements/comparison-operators-transact-sql?view=sql-server-ver16>

<https://learn.microsoft.com/en-us/sql/t-sql/queries/predicates?view=sql-server-ver16>