

## Лабораторна робота № 9

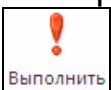
**Тема.** Розробка та використання запитів на вибірку без групування записів.

**Мета.** Формування вмінь та навичок створення запитів на вибірку без групування записів в режимі конструктора для комплексного аналізу даних таблиць. Усвідомлення послідовності етапів виконання запитів на вибірку даних. Вдосконалення вмінь створення та корегування об'єктів додатків СУБД MS Access.

### Теоретичні відомості

Запити дають змогу відбирати, групувати, сортувати, корегувати записи однієї чи одночасно декількох таблиць, виконувати обчислення як в межах запису, так і для кожної групи, а також ефективно обробляти дані в мережі. Запит може обробляти дані таблиць чи інших запитів. Але запити – це лише команди для обробки даних. В них не зберігаються дані записів. Створюються ці команди в режимах конструктора чи SQL. При перегляді ж запиту в режимі таблиці згідно його команди відображаються в табличному вигляді або результати виконання аналізу даних або записи, над якими **буде виконуватися** корегування. Ці тимчасові таблиці не зберігаються у файлах БД, а лише відображаються на екрані, тому й називаються **динамічними наборами даних**. Експлуатувати ці набори можна майже так само, як і таблиці, але при цьому потрібно чітко усвідомлювати, що внесені зміни зберігаються не в запиті, а в базовій таблиці, і тому стають доступними іншим користувачам в мережі.

Формувати більшість запитів в Access доцільно в режимі конструктора, де використовується стандарт **QBE** (Query by Example – запити по зразку). У цьому режимі, використовуючи інтерактивні засоби та регіональні національні стандарти, користувач формує бланк, де вказує, які поля необхідно опрацювати. Розробити ж будь-який запит можна в режимі SQL по однойменному стандарту. Тут команда кожного запиту записується за регіональним американським стандартом текстовими реченнями, тому запити в режимі SQL створюються і відлагоджуються довше, ніж в режимі конструктора, зате дозволяють задати оптимальніші умови відбору. Для виконання запиту необхідно завантажити його з вікна БД або в режимі конструктора у вкладці стрічки меню *Конструктор* натиснути кнопку



Наголосимо, що виконання запитів відбувається шляхом послідовної реалізації їх окремих речень згідно стандарту SQL, тому запити, створені по стандарту QBE в режимі конструктора перед виконанням всерівно перетворюються до стандарту SQL. І навпаки, більшість запитів, створених в режимі SQL, можна відкрити для

подальшого корегування в режимі конструктора. Всі запити виконуються **пакетно**, тобто СУБД, реалізуючи кожен запит, або повністю опрацює дані згідно його команди або, при неможливості виконання перерве обробку і всі задіяні таблиці поверне до початкового стану.

За призначенням запити Access поділяються на дві групи – запити **аналізу даних** та **запити на корегування**. В режимі таблиці для запитів аналізу даних фактично відображається результат їх виконання, а для запитів на корегування – записи, над якими буде здійснено дію. До першої групи запитів належать:

- **запити на вибірку** – найуживаніші запити, які використовуються для відбору, групування, сортування записів джерела та виконання обчислень над його даними (наприклад, підрахунок загальних сум оформлених замовлень по кожному співробітнику за минулий квартал);
- **перехресні запити** – виконують групування відібраних записів як по рядках, так і по стовпцях та проводять розрахунки над відповідними даними на їх перетині (наприклад, розрахунок сум замовлень (дані на перетині) по кожному товару (заголовки рядків) за кожен місяць (заголовки стовпців) в поточному році (умова відбору));
- **запити на поєднання** – здійснюють поєднання результатів відбору декількох запитів на вибірку. Такі запити можливо розробити лише в режимі SQL, оскільки вони обробляють дані декількох джерел;
- **запити до сервера** – виконують аналіз даних таблиць, розміщених на віддаленому сервері. Створюються лише в режимі SQL. Дані сервера мають бути попередньо записані в джерелах ODBC панелі керування операційної системи.

До групи запитів на корегування належать:

- **запити на оновлення** – змінюють дані, які містяться в записах таблиць з збереженням цілісності;
- **запити на додавання** – використовуються для пакетного створення нових записів в обраній таблиці. При цьому перевіряється цілісність не лише задіяних полів, а й даних записів в цілому;
- **запити на видалення** – вилучають записи, що відповідають критеріям відбору, з обраної таблиці. При встановленій властивості каскадного видалення для зв'язків з обраною таблицею на стороні 1 автоматично вилучаються також пов'язані записи в таблицях на стороні ∞;
- **запити на створення нових таблиць** – автоматично зберігають дані, відібрані командою кожного запиту, в новій таблиці. Якщо назва нової таблиці співпадає з назвою існуючої таблиці чи запиту, то такий об'єкт замінюється створюваною таблицею після отримання згоди від користувача;

➤ **запити керування** – використовуються для створення чи видалення таблиць або для зміни їх структури чи індексів. Створюються лише в режимі SQL.

У цій лабораторній ми розглянемо лише найпростіші запити на вибірку без групування записів, а в наступних – складніші запити.

Виконання запиту на вибірку відбувається в такій послідовності:

- формується джерело даних (динамічна таблиця з даних базових таблиць);
- відбираються записи з сформованого джерела;
- виконується групування відібраних записів і обчислення в групах;
- здійснюється відбір з сформованих груп;
- виконується сортування результатів відбору;
- формуються значення полів і виводяться результати дії запиту.

Обов'язковими з цих етапів є лише перший та останній. Наявність решти етапів залежить від команди запиту.

Для створення команди запиту на вибірку в режимі конструктора потрібно на стрічці меню натиснути кнопку **Создание** ► **Конструктор запросов**, після чого на екрані з'являється порожній бланк запиту і вікно для формування джерела (рис. 47). Верхня частина вікна запиту в режимі конструктора називається **областю джерела даних**, а нижня – **бланком запиту**.

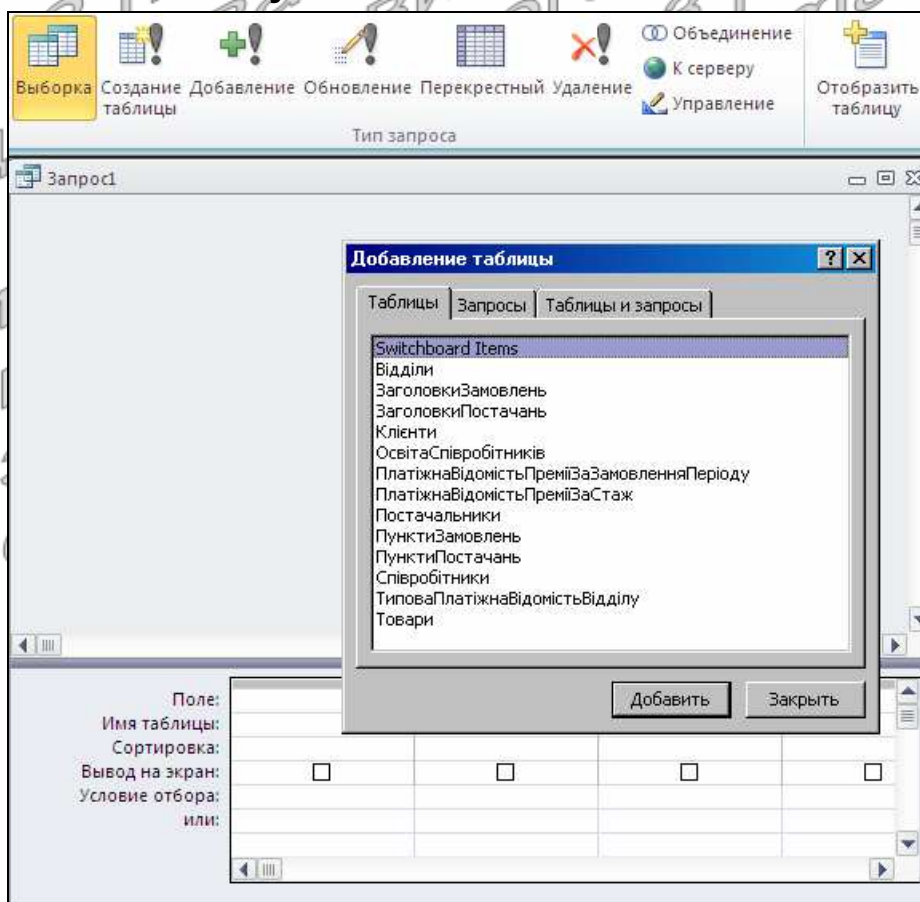


Рис. 47. Порожній бланк запиту на вибірку та вікно для доповнення джерела новими таблицями чи запитамі в режимі конструктора

По замовчуванню в режимі конструктора створюється саме запит на вибірку. Інший тип запиту обирається натисненням відповідної кнопки на вкладці стрічки меню *Конструктор* (верхня частина рис. 47).

Для формування джерела даних у вікні *Добавление таблицы* необхідно відмітити потрібні таблиці чи запити і натиснути кнопку *Добавить*. Це саме вікно можна надалі викликати з контекстного меню вільного місця області джерела даних або кнопкою стрічки меню *Отобразить таблицу* (див. рис. 47). Початкові зв'язки між таблицями джерела формуються з схеми даних, але надалі розробник в режимі конструктора запиту може встановити нові чи знищити існуючі зв'язки і ці зміни ніяк не впливають на схему даних (і це логічно, адже запитальні зв'язки не повинні впливати на структурні зв'язки інфологічної моделі). Наголосимо, що в джерелі даних не слід використовувати зайві таблиці чи запити, адже їх наявність може в рази збільшити кількість записів джерела даних і, як наслідок, спотворити результати виконання запиту. Знищення образу таблиці в області джерела даних запиту не призводить до її вилучення з схеми даних, чи, тим більше, до знищення самої таблиці. В області джерела даних режиму конструктора відображається структура обраних таблиць чи запитів, тобто їх поля, оскільки тут формується команда обробки даних (див. рис. 47).

Бланк запиту складається з таких рядків:

➤ *Поле* – містить назви полів з таблиць чи запитів джерела або вирази для обчислень значень, які використовуються запитом. Для вказування поля з джерела можна використати один з таких способів: перетягнути ім'я поля з області джерела даних в бланк; обрати чи набрати це ім'я безпосередньо в рядку *поле*; двічі натиснути ЛКМ або клавішу *Enter* на імені поля в джерелі (при цьому поле перенесеться у вільний стовбець бланку запиту). Крім цього, для вибору всіх полів таблиці чи запиту (наприклад, для формування джерела форми чи звіту) достатньо перетягнути в бланк \* з її образу. Тоді навіть перейменування чи створення нового поля у базовій таблиці не призведе до втрати дієздатності запиту.

Для формування ж виразу не достатньо вказати лише формулу його обчислення за національним стандартом. Справа в тому, що кожен вираз у бланку повинен мати унікальне ім'я (псевдонім), яке в подальшому використовується для звертання до значень його поля. Ім'я записується перед виразом і відділяється від нього двокрапкою (наприклад, *Сума: Кількість\*Ціна*). Якщо розробник не вказує ім'я виразу, то Access автоматично генерує імена типу *Выраженный1*, *Выражение2* і т.д. У полі двокрапка двічі використовуватися не може, тобто кожен стовбець запиту може мати лише один заголовок;

➤ *Имя таблицы* – вказує, з якої таблиці береться *Поле*. Обов'язково зазначається лише у випадках, коли ім'я поля присутнє у декількох



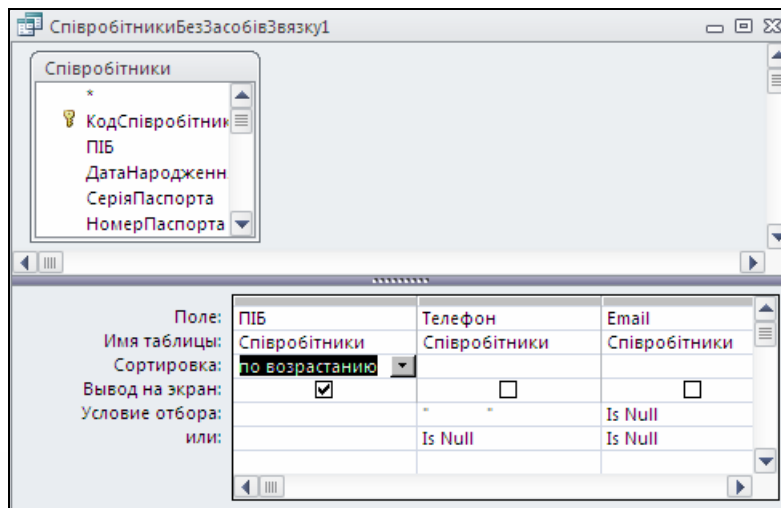
таблицях чи запитах, які формують джерело даних і не зазначається для виразів;

- *Сортировка* – забезпечує впорядкування записів (а не тільки значень поля), сформованих запитом. Це поле зі списком може набувати одного з трьох значень: *по возрастанию*, *по убыванию* та *отсутствует* (порожнє, застосовується по замовчуванню). Найважливішим є перше зліва поле сортування. Саме за ним впорядковуються записи динамічної таблиці результатів дії запиту. І лише ті записи, у яких значення першого поля сортування однакове, впорядковуються між собою за другим полем сортування і т.д.;
- *Групповая операция* – використовується для групування відібраних записів та виконання обчислень в групах. З'являється в бланку лише після натиснення в його контекстному меню чи на вкладці

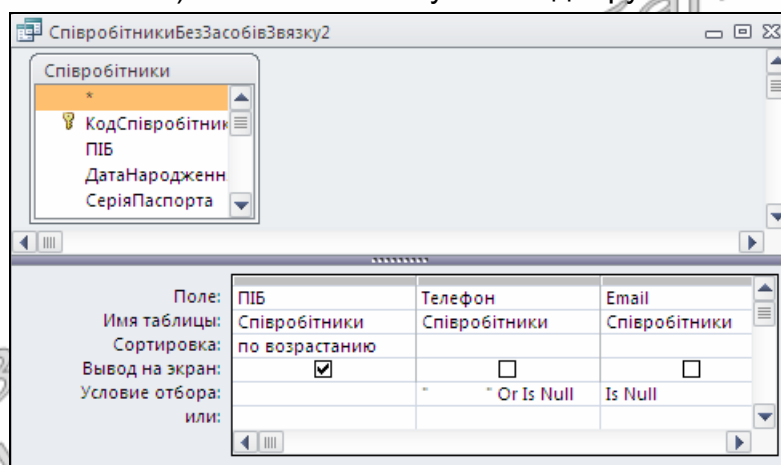


*Конструктор* стрічки меню кнопки

- *Вывод на экран* – прапорець, що встановлюється для полів, які мають відобразитися при виконанні запиту. Запит на вибірку має виводити хоча б одне поле;
- *Условие отбора / или* – містять умови для відбору записів з джерела чи з сформованих груп. Умови, записані в одному рядку, мають для відібраних записів чи груп виконуватися одночасно (поєднуються логічним сполучником **And**). У випадку ж вказування умов в різних рядках (застосовуючи рядки бланку *или*) для відбору запису джерела чи групи достатньо виконання умов хоча б з одного рядка. Тобто такі умови поєднуються логічним сполучником **Or**. Отже, запис чи група відбираються запитом тоді і лише тоді, коли для нього виконуються **всі умови хоча б одного** з рядків умов відбору. Наприклад, для відбору співробітників без зазначеного номера телефону та адреси електронної пошти необхідно врахувати, що телефон може бути не вказаний у двох випадках: або це поле ще не заповнювалося, тоді у ньому буде зберігатися значення по замовчуванню (12 пробілів, тобто " "), або телефон ввели, а потім видалили (тоді значення поля буде порожнім, тобто *Is Null*). Для кожного з цих випадків, згідно з умовою запиту, адреса електронної пошти має бути порожньою (*Is Null*, що відповідає як значенню по замовчуванню, так і значенню поля після видалення). Вигляд режиму конструктора цього варіанту запиту наведено на рис. 48а. Зверніть увагу, що поля *Телефон* та *Email* не виводяться на екран, оскільки при виконанні в них не відображаються дані (слід завжди дотримуватися цього підходу, щоб не відволікати увагу користувача на такі поля). Оптимізувати умови відбору цього запиту можливо, явно використавши логічний оператор *Or* для поля *Телефон* (рис. рис. 48б). Тоді при виконанні запиту для кожного запису джерела буде виконуватися максимум 3, а не 4 операції порівняння (Чому?).



а) неоптимізовані умови відбору



б) оптимізовані умови відбору

Рис. 48. Два варіанти структури запиту *СпівробітникиБезЗасобівЗв'язку* в режимі конструктора

Для відбору записів, що відповідають різним умовам, без внесення змін в структуру запиту в режимі конструктора чи SQL в умовах запиту використовують **параметри** (наприклад, замість конкретного номера місяця доцільно ввести текст *[Введіть номер місяця для аналізу даних]*. Чому текст параметра береться в квадратні дужки?). При виконанні запиту Access не віднаходить кожен параметр в джерелі даних і виводить вікно для введення його значення від користувача (ось чому поява зайвих вікон для введення даних при виконанні запиту найчастіше свідчить про неправильне введення імені поля). Безпосередній відбір записів виконується лише після введення всіх параметрів. Для кожного параметра можна забезпечити контроль типу даних, як це описано в лабораторній роботі, щоб уникнути аварійного завершення виконання запиту.

Надалі, описуючи синтаксис запитів і елементів виразів, в < > будемо позначати місця для ідентифікаторів та виразів користувача, в [ ] – необов'язкові елементи, а через | будемо перераховувати елементи, серед яких користувач має обрати лише один.

В режимі SQL запити на вибірку записуються згідно синтаксису:

```
SELECT <поля результатів відбору>  
FROM <джерело даних>  
[WHERE <умови відбору записів>]  
[GROUP BY <поля групування>]  
[HAVING <умови відбору груп>]  
[ORDER BY <поля сортування>];
```

Бачимо, що обов'язковими речення в інструкції відбору є лише FROM та SELECT, які відповідають першому та останньому етапу виконання запиту на вибірку. Окремі речення в середині інструкції вибору, як правило, набираються з нового рядка, хоча й можуть вводитися через пробіл після попереднього речення. В режимі SQL ім'я таблиці вказується перед іменем її поля і відділяється від нього крапкою. Ідентифікатори та вирази користувача у кожному з речень перераховуються через кому. Завершується кожна інструкція в режимі SQL крапкою з комою.

Таблиці, запити і зв'язки між ними, які формують джерело даних і в режимі конструктора відображаються в його верхній частині, в режимі SQL описуються після FROM. Якщо джерело містить лише одну таблицю чи запит, то після FROM вказується лише його назва (наприклад, FROM Відділи), якщо ж більше – то крім назв базових об'єктів ще й описуються зв'язки між ними.

Після слова SELECT перераховуються лише ті поля, що виводяться при виконанні запиту (для них в режимі конструктора встановлюється прапорець *Вивод на екран*). Якщо формується вираз, то його ім'я вказується після формули виразу і відділяється від нього сполучником AS. Наприклад, SELECT ЗаголовкиЗамовлень . ДатаЗамовлення, Кількість\*Ціна AS Сума.

Окремі умови відбору записів, які в режимі конструктора записуються в рядку *Умовие отбора* та нижчих і можуть міститися у різних стовпцях, в режимі SQL послідовно вказуються після WHERE, але поєднуються між собою сполучниками AND чи OR (Коли і як саме?).

Як і в режимі конструктора (рядок *Сортировка*), найважливішим полем сортування в режимі SQL є перше поле, яке зазначається відразу після ORDER BY, адже саме за ним впорядковуються записи. Лише ті записи, в яких значення першого поля сортування однакові, між собою впорядковуються за другим полем сортування і т.д. По замовчуванню сортування виконується за зростанням значення поля. Для явного сортування за зростанням після виразу поля вказують ASC. При необхідності ж сортування за спаданням після виразу поля вказують DESC. Наприклад, ORDER BY Відділи.НазваВідділу DESC, Співробітники.ПІБ.

Особливості формування та відбору груп розглянемо в наступній лабораторній роботі. Тут же висвітлимо принципи формування виразів, які застосовуються у всіх частинах запиту для виконання обчислень.

## Конструювання виразів в Access

**Вирази** – це конструкції, що містить хоча б один оператор і хоча б одну константу, ідентифікатор чи функцію.

Вирази використовують при конструюванні умов відбору в запитах, під час формування обчислювальних полів, в мовах програмування для виконання обчислень і формування умов для операторів. Запис виразу залежить від того, в якому режимі він створюється, оскільки в режимі конструктора діють національні стандарти, а в режимі SQL – американський. При переході з режиму в режим зміна синтаксису виразу відбувається автоматично.

Основні правила запису **констант**:

- рядкові константи беруться в подвійні лапки, наприклад, "Факультет";
- дати записуються між символами #, причому в режимі конструктора – у форматі #dd.mm.yyyy#, а в режимі SQL – #mm.dd.yyyy#;
- ціла частина числа відмежовується від дробової в режимі конструктора комою, а в режимі SQL – крапкою. Вона може містити мантису, яка записується після E.

**Ідентифікатори** – це імена об'єктів, які повертають активне значення. Наприклад, для запитів ідентифікаторами є імена полів з джерела даних. Якщо назва ідентифікатора містить декілька слів, то вони беруться в квадратні дужки. Для уникнення помилок назви ідентифікаторів рекомендується перетягувати, вибирати зі списку або вводити за допомогою будівничого виразів.

**Функції** – це послідовності команд, які повертають обчислення значення в місце виклику. В Access після назви функції вказують дужки, у яких при потребі перераховують її аргументи. Аргументи функцій відділяються між собою в режимі конструктора ;, а в SQL – ,. Різновиди функцій:

**а) функції обробки дат:**

- **Date()** – повертає активну дату в місце виклику;
- **Now()** – повертає активний момент часу;
- **Day(<дата>)** – повертає число дня в місяці від вказаної дати;
- **Month(<дата>)** – повертає місяць від значення аргумента;
- **Year(<дата>)** – повертає номер року від вказаної дати;
- **Weekday(<дата>)** – повертає номер дня в тижні від зазначеної дати;
- **Hour(<дата>)** – повертає годину дати в місце виклику;

**б) функції обробки рядків:**

- **Len(<рядок>)** – обчислює кількість символів у рядку;
- **Left(<рядок>; <кількість>)** – вирізає з рядка зліва вказану кількість символів;
- **Right(<рядок>; <кількість>)** – вирізає з рядка справа вказану кількість символів;
- **Mid(<рядок>; <позиція>; <кількість>)** – вирізає з рядка, починаючи з вказаної позиції, необхідну кількість символів;
- **Format(<змінна>; <шаблон>)** – форматує значення змінної по заданому шаблону;
- **InStr(<рядок>; <шаблон>)** – виконує пошук в рядку тексту згідно шаблону і повертає першу позицію входження (при наявності);

**в) Функції перетворення типів:**

- **CBool(<значення>)** – перетворює значення до логічного типу, якщо це можливо;
- **CStr(<значення>)** – перетворює значення до тексту;
- **Cdbl(<значення>)** – переводить значення до дійсного числа;
- **CLng(<значення>)** – перетворює значення до довгого цілого числа;
- **CCur(<значення>)** – перетворює значення до грошового типу;
- **CDate(<значення>)** – перетворює значення до дати;
- **Nz(<вираз>; <значення>)** – якщо вираз непорожній, то повертає його значення, інакше повертає значення другого аргумента;



#### г) математичні функції:

- **Int**(<значення>) – повертає цілу частину числа. Наприклад, вираз для визначення кількості років, які пройшли від дати влаштування співробітника на роботу:  $\text{Int}((\text{Date}() - [\text{Дата влаштування}]) / 365,25)$ ;
- **Round**(<значення>; <точність>) – заокруглює число до вказаної у точності кількості знаків. Наприклад,  $\text{Round}(\text{Оклад}/27,2;2)$ ;

#### д) функції змішаного типу:

- **IIf**(<умова>; <значення1>; <значення2>) – ця функція перевіряє істинність умови. Якщо умова істинна, то повертається *значення1*, інакше повертається *значення2*. Наприклад, для обчислення суми прибуткового податку, якщо до 340 грн. він не нараховується, а після 340 складає 19,5% від суми, що перевищує 340 грн. зазначають:  $\text{IIf}(\text{Оклад} \leq 340; 0; 0,195 * (\text{Оклад} - 340))$ ;
- **IsNull**(<значення>) – повертає *істинно*, якщо значення порожнє;
- **IsNumeric**(<значення>) – повертає *істинно*, якщо значення можна перетворити до числа;
- **IsDate**(<значення>) – повертає *істинно*, якщо значення можна перетворити до дати.

**е) Статистичні функції SQL** – це функції, які в режимі конструктора вибирають у рядку групової операції. Вони детально розглядаються в наступній лабораторній роботі;

**ж) Статистичні функції обробки підмножин** – це функції, що використовуються для обробки записів зовнішнього джерела. Всі функції цієї групи мають 3 аргументи: перший вказує, яке значення повертає функція; другий – описує джерело даних; третій – задає умови відбору записів джерела. Наприклад, функція **DLookup** повертає перше значення, яке відповідає умовам відбору. Назви інших функцій, обробки підмножин аналогічні функціям SQL, лише спереду додається буква D.

**Оператори**, що використовуються у виразах:

- а) **арифметичні**: +; -; \*; /; **mod** – остача від ділення, \ - ціла частина від ділення, ^ - степінь;
- б) **порівняння**: (<, ≤, >, ≥, =, <>);
- в) **логічні**: **NOT** – заперечення, **AND**, **OR**, **XOR** – оператор істинний, коли істинні обидва операнди;
- г) **злиття рядкових величин (конкатенації)**: &;
- д) **ідентифікації**: ! – використовують для відмежування класу об'єкта від самого об'єкта і об'єкта від вкладеного об'єкта); . – відмежовує стандартне поле запису від назви таблиці з джерела та стандартну властивість від самого об'єкта. В Access використовують такі типи об'єктів: Tables (таблиці); Queryes (запити); Forms (форми); Reports (звіти).

#### з) Інші оператори:

- **IS** – використовують для перевірки того, чи є поле порожнім чи ні (в поєднанні з *Null*, а саме *Is Null* або *Is Not Null*);
- **LIKE** <шаблон> – встановлює відповідність поля шаблону. В шаблоні можуть використовуватися символи \* (довільна кількість символів) та ? (один довільний символ);
- **IN** (<список>) – перевіряє належність поля до множини, елементи списку перераховують як аргументи функцій – в режимі конструктора через ;, а в SQL – через ,;
- **BETWEEN** <нижня межа> **AND** <верхня межа> – перевіряє на належність до діапазону. Межі в діапазон включаються, одна з меж може не вказуватися, але вживання **AND** обов'язкове.

Література: [4, С. 108-114, 120-121; 1, С. 157-167, 172-174]

### Підготовчий етап заняття. Актуалізація знань

1. Завантажте Access, відкрийте розроблену раніше БД *Sklad*.
2. В контекстному меню області переходів оберіть категорію подання об'єктів *Тип объекта*.
3. Перейдіть в області переходів до розділу *Запросы*. По можливості, для прискорення формування виразів, роздрукуйте дві попередні сторінки теоретичних відомостей.

### Сортування та відбір даних за допомогою запитів

4. Створіть запит *АлфавітнийСписокСпівробітників* в режимі конструктора для формування алфавітного списку співробітників з зазначенням дати народження. Для цього:
  - 4.1. Перейдіть в режим конструктора для створення запиту, натиснувши на вкладці стрічки меню *Создание* у групі *Запросы* кнопку *Конструктор запросов*;
  - 4.2. Для формування джерела даних запиту у вікні *Добавление таблицы* виділіть таблицю *Співробітники* та натисніть кнопку *Добавить*. Закрийте вікно *Добавление таблицы*;
  - 4.3. З метою формування списку співробітників в перший стовпець бланку запиту внесіть поле *ПІБ* одним з двох способів: перетягніть поле при натиснутій лівій кнопці мишки з образу таблиці *Співробітники* у верхній частині вікна в бланк запиту у нижній його частині або виберіть назву поля зі списку в рядку *Поле* першого стовпця бланку запиту;
  - 4.4. В другий стовпець бланку запиту внесіть одним з двох описаних вище способів поле *ДатаНародження*;
  - 4.5. В рядку *Сортировка* для поля *ПІБ* оберіть зі списку значення *по возрастанию* для відповідного впорядкування списку співробітників при виконанні запиту;
  - 4.6. Закрийте вікно конструктора та збережіть запит під назвою *АлфавітнийСписокСпівробітників*.
5. Виділіть створений запит та завантажте його на виконання різними способами:
  - 5.1. Подвійним натисненням лівої кнопки мишки по назві запиту;
  - 5.2. Натисненням клавіші *Enter* після виділення назви запиту;
  - 5.3. За допомогою пункту *Открыть* контекстного меню запиту;
  - 5.4. Перейдіть в режим конструктора, обравши в контекстному меню запиту пункт *Конструктор*, та оберіть, перебуваючи у ньому, у списку кнопки *Режимы* пункт *Режим таблицы* (цей спосіб рекомендується використовувати надалі);
  - 5.5. Перейдіть в режим конструктора та натисніть на вкладці інструментів стрічки меню кнопку *Выполнить*.
6. Перегляньте текст створеного запиту в режимі SQL. Для цього перейдіть в режим конструктора та виберіть у списку кнопки *Режимы* пункт *режим SQL*. Обгрунтуйте структуру тексту запиту.

7. Завантажте запит в режимі конструктора та з'ясуйте призначення всіх пунктів списку кнопки *Режими*.
8. Самостійно створіть аналогічні запити:
  - 8.1. *АлфавітнийСписокВідділів* – для відображення в алфавітному порядку назв відділів з зазначенням місць розташування;
  - 8.2. *АлфавітнийСписокКлієнтів* – для відображення в алфавітному порядку назв клієнтів з зазначенням дати реєстрації та місця роботи;
  - 8.3. *АлфавітнийСписокПостачальників* – для відображення в алфавітному порядку назв постачальників з зазначенням коду ЄДРПОУ та дати реєстрації;
  - 8.4. *АлфавітнийСписокТоварів* – для відображення назв товарів в алфавітному порядку з зазначенням марки, одиниці виміру та виробника.

#### **Створення обчислювальних полів в запитах**

9. Доповніть бланк запиту *АлфавітнийСписокСпівробітників* полем для зазначення його віку. Для цього:
  - 9.1. Відкрийте обраний запит в режимі конструктора;
  - 9.2. В рядку *Поле* третього стовпця бланку запиту введіть формулу *Date()-ДатаНародження* для підрахунку кількості прожитих днів;
  - 9.3. Для перевірки дієздатності запиту завантажте його в режимі таблиці. Знову поверніться в режим конструктора. Чому перед формулою з'явився підпис *Выражение 1*? Яку роль він відіграє?
  - 9.4. Відредагуйте текст третього стовпця до вигляду *Вік: (Date()-ДатаНародження)/365,25* (оскільки середня тривалість року складає 365,25 дня). Перевірте функціональність запиту;
  - 9.5. Для відкидання дробової частини відредагуйте текст третього стовпця до вигляду *Вік: Int((Date()-ДатаНародження)/365,25)*. Перевірте функціональність запиту;
  - 9.6. Закрийте запит та збережіть внесені зміни.
10. Створіть запит *ДніНародженняСпівробітників* з переліком прізвищ співробітників, місяців і днів їх народження та кількості років, що виповнилися (виповняться) співробітнику в цьому році. Для цього:
  - 10.1. Розпочніть створення запиту в режимі конструктора;
  - 10.2. Сформуєте джерело даних запиту з записів таблиці *Співробітники*;
  - 10.3. Перетягніть в перший стовпець бланку запиту поле *ПІБ*;
  - 10.4. В другий стовпець запиту введіть вираз *Місяць народження: Month(ДатаНародження)*;
  - 10.5. Самостійно сформуєте вираз третього стовпця для виводу на екран числа дня народження співробітника;
  - 10.6. В четвертий стовпець запиту введіть вираз *Виповниться (виповнилося): Year(Date())-Year(ДатаНародження)*;

- 10.7. Самостійно встановіть ознаки сортування за сукупним зростанням значень полів *Місяць народження* та *День народження*;
- 10.8. Збережіть запит та переконайтеся в його дієздатності.
11. Самостійно доповніть запит *АлфавітнийСписокТоварів* полем *Мінімальна дата виготовлення* для відображення дати виготовлення товару, починаючи з якої він ще є придатним до вживання (значення цього поля дорівнює різниці поточної дати та терміну придатності товару в днях).
12. Самостійно створіть запит *ОкладиСпівробітників* з переліком прізвищ співробітників, окладу в гривнях, доларах (рівний окладу в гривнях, поділеному на поточний курс долара) та євро. Відсортуйте його записи за спаданням значення окладу в гривнях.

### **Конструювання умов відбору записів в запитах**

13. Створіть запит *ЖінкиВідділів* для відображення прізвищ жінок, що працюють у кожному відділі. Для цього:
  - 13.1. Розпочніть створення запиту в режимі конструктора;
  - 13.2. Сформууйте джерело даних запиту з записів таблиць *Відділи* та *Співробітники*;
  - 13.3. Перетягніть в перший стовпець бланку запиту поле *НазваВідділу* з таблиці *Відділи*;
  - 13.4. Перетягніть в другий стовпець бланку запиту поле *ПІБ* з таблиці *Співробітники*;
  - 13.5. Для виводу на екран лише співробітників-жінок перетягніть в третій стовпець бланку запиту поле *Стать* з таблиці *Співробітники*, зніміть для нього прапорець *Вывод на экран* та вкажіть в рядку *Условие отбора* значення *0*;
  - 13.6. Оскільки працюючі співробітники – це, насамперед, ті, для яких не встановлено дату звільнення, то для виводу на екран при виконанні запиту лише працюючих співробітників перетягніть в четвертий стовпець бланку запиту поле *ДатаЗвільнення* з таблиці *Співробітники*, зніміть для нього прапорець *Вывод на экран* та введіть в рядок *Условие отбора* вираз *Is Null*;
  - 13.7. Для першого і другого стовпця встановіть ознаку сортування за зростанням;
  - 13.8. Збережіть запит та переконайтеся в його дієздатності.
14. Аналогічно забезпечте відображення в запитах *АлфавітнийСписокСпівробітників*, *ДніНародженняСпівробітників*, *ОкладиСпівробітників* даних лише працюючих співробітників.
15. Самостійно створіть наступні запити:
  - 15.1. *СпівробітникиБезТелефонів* – для відображення алфавітного списку співробітників без вказаних номерів телефонів з зазначенням відділу (не введений номер телефону

позначається дванадцятьма пробілами (чому?), а видалений – *Is Null*, тому загальна умова відбору для цього поля має вигляд " " *Or Is Null*);

15.2. *СпівробітникиЗEmail* – для відображення алфавітного списку співробітників з зазначенням відділу, що мають непорожні адреси електронної пошти.

16. Створіть запит *ПрацюючіПенсіонери* для відображення алфавітного списку працюючих пенсіонерів кожного відділу (пенсіонери – це жінки, старші 60 років та чоловіки, старші 62 років). Для цього:

16.1. Щоб уникнути повторного введення формули для обчислення віку співробітника, скопіюйте в буфер обміну запит *АлфавітнийСписокСпівробітників* та вставте його під назвою *ПрацюючіПенсіонери*;

16.2. Відкрийте скопійований запит в режимі конструктора та перетягніть в четвертий стовпець бланку поле *Стать*, а в п'ятий – поле *ДатаЗвільнення*;

16.3. Для відбору пенсіонерів–жінок внесіть в рядок *Условие отбора* для стовпця *Вік* вираз  $\geq 60$ , а для стовпця *Стать* – значення 0;

16.4. Для відбору пенсіонерів–чоловіків внесіть в рядок *или* для стовпця *Вік* вираз  $\geq 62$ , а для стовпця *Стать* – значення -1;

16.5. Для відбору лише працюючих пенсіонерів внесіть в рядки *Условие отбора* та *или* для стовпця *ДатаЗвільнення* вираз *Is Null*. Самостійно забезпечте відсутність відображення на екрані даного стовпця при виконання запиту;

16.6. Збережіть створений запит та перевірте його дієздатність.

17. Створіть запит *ОсвітченіСпівробітники* для відображення алфавітного списку співробітників з зазначенням відділу, посади та окладу, що мають середню спеціальну, незакінчену вищу або вищу освіту. Для цього:

17.1. Розпочніть створення запиту в режимі конструктора;

17.2. Сформууйте джерело даних запиту з записів таблиць *Відділи*, *Співробітники* та *ОсвітаСпівробітників*;

17.3. Перетягніть в перший стовпець бланку запиту поле *ПІБ* з таблиці *Співробітники*;

17.4. Перетягніть в другий стовпець бланку запиту поле *НазваВідділу* з таблиці *Відділи*;

17.5. Перетягніть в третій та четвертий стовпці бланку запиту відповідно поля *Посада* та *Оклад* з таблиці *Співробітники*;

17.6. Для виводу на екран лише співробітників, що мають середню спеціальну, незакінчену вищу або вищу освіту виконайте наступні дії:

17.6.1. Перетягніть в п'ятий стовпець бланку запиту поле *ТипОсвіти* з таблиці *ОсвітаСпівробітників*;



- 17.6.2. Оскільки для поля *ТипОсвіти* задано підстановку значень з фіксованого набору так, що відображуваному тексту *середня спеціальна* відповідає значення поля 3, тексту *незакінчена вища* – значення 4, тексту *вища* – значення 5, то для відбору записів джерела даних запиту лише з вказаними типами освіти задайте для даного поля в рядку *Условие отбора* значення 3, в рядку *или* – значення 4, а в наступному рядку *или* – значення 5;
- 17.7. Для виводу на екран при виконанні запиту лише працюючих співробітників перетягніть в шостий стовпець бланку запиту поле *ДатаЗвільнення* з таблиці *Співробітники*, зніміть для нього прапорець *Вывод на экран* та введіть в рядок *Условие отбора* вираз *Is Null*;
- 17.8. Для першого і другого стовпця встановіть ознаку сортування за зростанням;
- 17.9. Збережіть запит та переконайтеся в його дієздатності. Чому окремі прізвища співробітників виведено декілька разів?
- 17.10. Перегляньте текст створеного запиту в режимі SQL. Як в цьому режимі записано умову відбору записів джерела даних запиту?
- 17.11. Поверніться в режим конструктора. Для стовпця *ТипОсвіти* в рядок *Условие отбора* введіть еквівалентну до попередньої умову відбору записів  $\geq 3 \text{ and } \leq 5$  та знищіть умови відбору в нижчих рядках;
- 17.12. Завантажте відредагований запит в режимі таблиці. Чому на екран при зміненому варіанті умови відбору записів виведено аналогічні дані?
- 17.13. Перегляньте текст відредагованого запиту в режимі SQL. Як змінився текст умови відбору записів джерела даних запиту?
- 17.14. Поверніться в режим конструктора. Для стовпця *ТипОсвіти* в рядок *Условие отбора* введіть еквівалентну до попередньої умову відбору записів *between 3 and 5*;
- 17.15. Завантажте відредагований запит в режимі таблиці та проаналізуйте отримані дані;
- 17.16. Перегляньте текст відредагованого запиту в режимі SQL. Як змінився текст умови відбору записів джерела даних запиту?
- 17.17. Теоретично проаналізуйте та обґрунтуйте у звіті швидкість виконання трьох варіантів запиту.
18. Самостійно створіть наступні запити:
- 18.1. *СпівробітникиЗВищоюОсвітою* – для відображення алфавітного списку працюючих співробітників з зазначенням відділу, що здобули вищу освіту;

- 18.2. *СпівробітникиБезЗасобівЗв'язку* – для відображення алфавітного списку працюючих співробітників з зазначенням відділу без введених номерів телефонів та адрес електронної пошти;
- 18.3. *СтажРоботиВТоваристві* – для відображення списку працюючих співробітників з зазначенням відділу та кількості відпрацьованих років в товаристві відносно активної дати. Відсортуйте виведені записи за спаданням кількості відпрацьованих років.

### **Створення запитів з параметрами відбору записів**

19. Створіть запит *ДаніПостачаньТоварівЗаПеріод* для відображення алфавітного списку назв та марок отриманих товарів за введений період з зазначенням відповідної назви постачальника, дати постачання, кількості, ціни та суми. Для цього:
  - 19.1. Розпочніть створення запиту в режимі конструктора;
  - 19.2. Сформууйте джерело даних запиту з записів таблиць *Постачальники*, *ЗаголовкиПостачань*, *ПунктиПостачань* та *Товари*;
  - 19.3. Перетягніть в перший та другий стовпці бланку запиту відповідно поля *НазваТовару* та *МаркаТовару* з таблиці *Товари*;
  - 19.4. Перетягніть в третій стовпець бланку запиту поле *ДатаПостачання* з таблиці *ЗаголовкиПостачань*;
  - 19.5. Перетягніть в четвертий стовпець бланку запиту поле *ПІБ* з таблиці *Постачальники*;
  - 19.6. Для першого, другого та третього стовпця встановіть ознаку сортування за зростанням;
  - 19.7. Для відбору даних постачань за 2016 рік введіть в рядку *Условие отбора* для поля *ДатаПостачання* вираз *between #01.01.2016# and #31.12.2016#*;
  - 19.8. Збережіть запит та переконайтеся в його дієздатності;
  - 19.9. Для відбору даних постачань за довільний проміжок часу поверніться в режим конструктора та змініть в рядку *Условие отбора* для поля *ДатаПостачання* умову відбору на *between [Введіть початкову дату] and [Введіть кінцеву дату]*;
  - 19.10. Завантажте запит на виконання, ввівши у першому вікні початкову, а в другому – кінцеву дати.
  - 19.11. Для забезпечення коректності введених параметрів задайте для них тип даних. Для цього відкрийте вікно *Параметры запроса* за допомогою кнопки *Параметры* вкладки інструментів стрічки меню або за допомогою аналогічного пункту контекстного меню вільного поля області джерела даних запиту, заповніть виведену таблицю згідно табл. 18 та натисніть *OK*.

**Типи даних для параметрів запиту *ДаніПостачаньТоварівЗаПеріод***

Параметр	Тип даних
[Введіть початкову дату]	Дата/время
[Введіть кінцеву дату]	Дата/время

- 19.12. Завантажте запит на виконання та переконайтеся у функціонуванні перевірки введених параметрів на відповідність вказаним типам даних.
20. Самостійно доповніть створений запит полями для виводу на екран кількості, ціни та суми поставленого товару.
21. Самостійно створіть запит *ДаніЗамовленьТоварівЗаПеріод* для відображення алфавітного списку назв та марок замовлених товарів з зазначенням відповідної дати замовлення, назви клієнта, кількості, ціни та суми.

**Завершальний етап заняття. Повторення вивченого матеріалу**

22. Стисніть створену БД.
23. Завершіть роботу Access.
24. Віднайдіть створену БД та заархівуйте її вміст.
25. Збережіть створений архів на власному носії в двох екземплярах.
26. Розробіть аналогічні запити до запитів цієї лабораторної роботи у власній БД. Продемонструйте розроблені БД викладачу.

**Контрольні запитання**

1. Що спільного між стандартами мов запитів QBE та SQL? Чим вони відрізняються? Які регіональні стандарти у них застосовуються?
2. Яка різниця між дією запиту та операцією пошуку?
3. Як перейти в режим конструктора для створення нового запиту?
4. Які поля відображаються в результаті виконання запиту? Де записуються назви цих полів в режимі SQL?
5. Як здійснити сортування полів при виконанні запиту? Як сортуються дані при вказуванні ознаки сортування для багатьох полів? Де записуються назви цих полів в режимі SQL?
6. Як здійснити відбір записів джерела даних при виконанні запиту? Де описуються критерії відбору записів в режимі SQL?
7. Як оформляються константи різних типів даних при конструюванні умов відбору?
8. Як створити обчислювальне поле в запиті? Значення яких полів може використовуватися при формуванні обчислювального поля? Коли слід створювати обчислювальне поле, а коли – задавати параметри відбору записів джерела даних?
9. Для чого використовуються параметри при виконанні запиту? Чому текст параметра з декількох слів необхідно брати в квадратні дужки?
10. Як забезпечити перевірку типу даних для параметра запиту? Навіщо це робити? Коли виконується така перевірка?