

Лабораторна робота № 10

Тема. Створення і використання запитів на вибірку з групуванням записів та перехресних запитів. Використання параметрів поєднань таблиць та властивостей запитів для ефективного аналізу даних.

Мета. Формування вмінь та навичок створення запитів на вибірку з групуванням даних, перехресних запитів, корегування параметрів об'єднань таблиць та властивостей запитів. Закріплення вмінь і навичок створення та корегування запитів в режимі конструктора. Систематизація та застосування знань варіантів поєднань таблиць БД при конструюванні запитів. Усвідомлення ролі зв'язків між таблицями при формуванні джерела даних запиту.

Теоретичні відомості

Розглянемо детальніше механізм формування джерела даних запиту. Єдина динамічна таблиця джерела даних містить всі поля з базових таблиць чи запитів (назвемо їх базовими таблицями, оскільки обробка в запиті іншого запиту нічим не відрізняється від обробки таблиці). Зрозуміло, що якщо запит обробляє одну базову таблицю, то джерело даних буде співпадати з нею. Якщо ж джерело містить більше базових таблиць, то його записи формуються згідно встановлених **поєднань**. Поєднання між базовими таблицями джерела даних найчастіше реалізуються за допомогою зв'язків. **Кожен зв'язок між базовими таблицями фактично вказує на поєднання записів, в яких значення пов'язаних полів співпадають** (як і в схемі даних БД). Крім цього, на кількість записів джерела впливають встановлені **параметри поєднань**. Дослідимо роль зв'язків та параметрів поєднань при формуванні джерела на прикладі двох базових таблиць для обліку даних студентів та їх пропусків (рис. 49).

The image shows two screenshots of a database interface. The left screenshot shows a table named 'Студенти' with columns 'КодСтудента' and 'ПІБ'. The right screenshot shows a table named 'Пропуски' with columns 'КодПропу', 'КодСтудента', 'ДатаПропуска', and 'НомерПари'.

КодСтудента	ПІБ
1	Семешук
2	Бондарець
3	Собчук
4	Козачок
*	(№)

КодПропу	КодСтудента	ДатаПропуска	НомерПари
1	2	19.09.2016	1
2	2	19.09.2016	2
3	3	23.09.2016	1
*	(№)		

Рис. 49. Дані базових таблиць *Студенти* та *Пропуски* для формування джерела даних запиту

Дані пропусків студентів зберігаються в окремій таблиці, оскільки один студент може мати багато пропусків (саме тому між цими таблицями

встановлено структурний зв'язок по полях *КодСтудента*, які перебувають у співвідношенні 1:∞). Бачимо (див. рис. 49), що студенти з кодами 1 і 4 взагалі не мають пропусків, студент з кодом 2 має два пропуски, а студент з кодом 3 має один пропуск. Отож, в Access застосовується чотири типи параметрів поєднань:

➤ **поєднання тільки тих записів, в яких пов'язані поля обох базових таблиць співпадають.** Цей тип поєднання використовується на практиці найчастіше. Якщо поєднати базові таблиці, наведені на рис. 49, по співпадаючих значеннях полів *КодСтудента*, то отримаємо джерело, наведене на рис. 50. Зверніть увагу, що, по-перше, дані студентів з кодами 1 і 4 не ввійшли в джерело, оскільки для них немає відповідних записів у другій таблиці і тому ці дані не будуть виведені запитом; по-друге, дані студента з кодом 2 ввійшли в джерело двічі, оскільки за ним зафіксовано два пропуски; по-третє, для ідентифікації полів з однаковими іменами, які входять в різні базові таблиці, перед цими іменами вказується назва базової таблиці, яка відділяється крапкою.

Студенти.КодСтудє	ПІБ	КодПропуска	Пропуски.КодСту	ДатаПропус	НомерПари
2	Бондарець	1	2	19.09.2016	1
2	Бондарець	2	2	19.09.2016	2
3	Собчук	3	3	23.09.2016	1

Рис. 50. Джерело даних з таблиць *Студенти* та *Пропуски*, поєднаних по співпадаючих значеннях поля *КодСтудента*

Якщо ж поєднати ці самі базові таблиці по співпадаючих значеннях поля *КодСтудента* з першої таблиці і *НомерПари* з другої таблиці (а це можна зробити, наприклад, в режимі конструктора перетягуванням між полями, як в схемі даних), то отримаємо джерело, наведене на рис. 51. Тобто ті самі базові таблиці з тими самими параметрами поєднання, але іншими зв'язками можуть породити відмінне джерело даних.

Студенти.КодСтудє	ПІБ	КодПропуска	Пропуски.КодСту	ДатаПропус	НомерПари
1	Семешук	1	2	19.09.2016	1
2	Бондарець	2	2	19.09.2016	2
1	Семешук	3	3	23.09.2016	1

Рис. 51. Джерело даних з таблиць *Студенти* та *Пропуски*, поєднаних по співпадаючих значеннях поля *КодСтудента* з таблиці *Студенти* і *НомерПари* з таблиці *Пропуски*

В режимі конструктора цей параметр поєднання обирається для зв'язків по замовчуванню і виглядає як ламана, проведена між полями базових таблиць (див., наприклад, рис. 11). Для перетворення зв'язків інших типів до даного типу необхідно виділити зв'язок і в його контекстному меню обрати пункт *Параметри об'єднання*, після чого в однойменному вікні, що з'явиться на екрані (як на рис. 52), обрати перший перемикач. Між собою можуть бути пов'язані по декілька полі з кожної базової таблиці, але всі зв'язки між довільними парами таблиць мають бути одного типу.

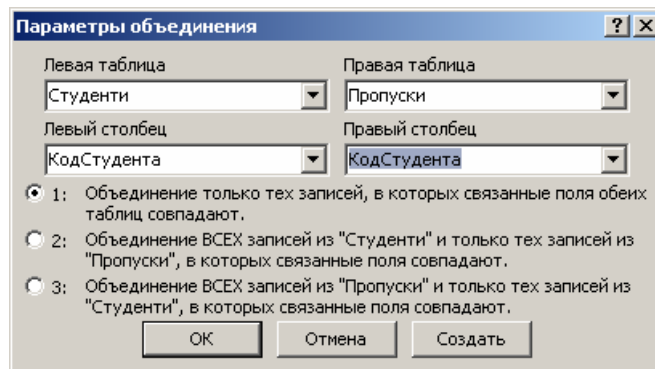


Рис. 52. Вікно вибору параметра поєднання в режимі конструктора

В режимі SQL джерело даних з двома базовими таблицями та зв'язком між ними записується згідно синтаксису

FROM <Таблиця 1> INNER JOIN | LEFT JOIN | RIGHT JOIN
 <Таблиця 2> ON <поле з Таблиці 1>=<поле з Таблиці 2> .

Зокрема, якщо застосовується описуване поєднання записів по співпадаючих значеннях полів обох базових таблиць, то між іменами цих таблиць використовується оператор INNER JOIN. В стандартах SQL такий тип поєднання називається **внутрішнім**. Наприклад, джерело рис. 50 в режимі SQL має такий вигляд:

FROM Студенти INNER JOIN Пропуски ON Студенти.КодСтудента =
 Пропуски.КодСтудента .

При необхідності опису джерела більш ніж з двох базових таблиць спочатку у круглих дужках записується поєднання перших двох таблиць, яке надалі сприймається як єдина таблиця, після чого дописується поєднання до третьої таблиці і т.д. Наприклад:

FROM (Студенти INNER JOIN Пропуски ON Студенти.
 КодСтудента = Пропуски.КодСтудента) INNER JOIN
 ПричиниПропусків ON Пропуски.КП=ПричиниПропусків.КП ;

- **поєднання ВСІХ записів з першої таблиці і тільки тих записів з другої таблиці, в яких пов'язані поля співпадають.** У випадку використання цього параметра поєднання в джерело даних запиту крім записів, утворених поєднанням записів базових таблиць, в яких значення пов'язаних полів співпадають, увійдуть ще й записи першої таблиці, для яких відсутні відповідні записи у другій таблиці. При цьому в джерелі даних для таких додаткових записів у всіх полях другої таблиці буде встановлено *Null*.

Студенти.КодСтудє	ПІБ	КодПропуска	Пропуски.КодСту	ДатаПропус	НомерПари
1	Семешук				
2	Бондарець	1	2	19.09.2016	1
2	Бондарець	2	2	19.09.2016	2
3	Собчук	3	3	23.09.2016	1
4	Козачок				

Рис. 53. Джерело даних з усіх записів таблиці *Студенти* і тільки тих записів таблиці *Пропуски*, в яких значення полів *КодСтудента* співпадають

другої до першої базової таблиці, а для задання такого параметра поєднання в режимі SQL необхідно між назвами таблиць використати оператор `RIGHT JOIN`;

- **поєднання кожного запису першої таблиці з кожним записом другої таблиці** (Декартів добуток записів). При використанні цього параметра поєднання в режимі конструктора зв'язки між базовими таблицями не встановлюються (саме тому додаткові непов'язані таблиці в джерелі призводять до створення додаткових записів), а в режимі SQL назви базових таблиць перераховуються через кому і поля з двох таблиць не порівнюються між собою (наприклад, `FROM Студенти, Пропуски`). Якщо в першій таблиці міститься N , а в другій – M записів, то в джерелі буде сформовано $M*N$ записів (як на рис. 55).

Студенти.КодСтудє	ПІБ	КодПропуска	Пропуски.КодСту	ДатаПропус	НомерПари
1	Семещук	1	2	19.09.2016	1
1	Семещук	2	2	19.09.2016	2
1	Семещук	3	3	23.09.2016	1
2	Бондарець	1	2	19.09.2016	1
2	Бондарець	2	2	19.09.2016	2
2	Бондарець	3	3	23.09.2016	1
3	Собчук	1	2	19.09.2016	1
3	Собчук	2	2	19.09.2016	2
3	Собчук	3	3	23.09.2016	1
4	Козачок	1	2	19.09.2016	1
4	Козачок	2	2	19.09.2016	2
4	Козачок	3	3	23.09.2016	1

Рис. 55. Джерело даних при поєднанні кожного запису таблиці *Студенти* з кожним записом таблиці *Пропуски*

В стандарті SQL використовується також оператор повного зовнішнього поєднання `FULL JOIN`, який повертає поєднання записів, в яких пов'язані поля співпадають, записи з першої базової таблиці без відповідних записів у другій і записи другої таблиці без пов'язаних записів у першій, але цей параметр поєднання в Access не реалізований.

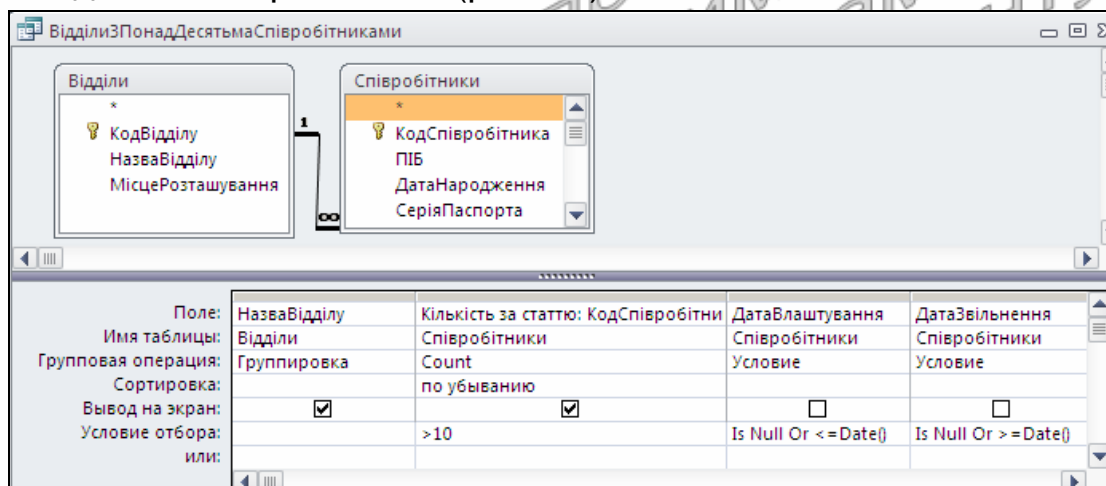
Розглянемо тепер особливості створення і використання запитів на вибірку з групуванням записів. Як вже зазначалося в попередній лабораторній роботі, для забезпечення групування відібраних записів і виконання обчислень у групах в режимі конструктора необхідно натиснути кнопку *Итоги*. Після цього в бланку запиту з'явиться додатковий рядок *Групповая операция*, в якому для кожного поля можна обрати одне з значень:

- **Группировка** – використовується для групування відібраних записів. Якщо це значення встановлюється для декількох полів, то групи створюються згідно унікальних комбінацій їх значень. Коли ж це значення не встановлено для жодного поля, то всі записи джерела буде віднесено до однієї групи. Поля, для яких в режимі

конструктора обрана *Групування*, в режимі SQL послідовно перераховуються в реченні GROUP BY. Решта з перерахованих нижче значень списку, крім останнього, - це статистичні функції SQL, що застосовуються до значення поля;

- **Sum** – обчислює суму значень поля в кожній групі;
- **Count** – обчислює кількість непорожніх значень поля в кожній групі;
- **Min** – обчислює мінімальне значення в кожній групі;
- **Max** – обчислює максимальне значення поля в кожній групі;
- **Avg** – обчислює середнє арифметичне в кожній групі;
- **Var** – обчислює дисперсію в кожній групі;
- **First** – повертає перше значення з джерела в кожній групі;
- **Last** - повертає останнє значення з джерела в кожній групі;
- **умовие** – містить умову відбору окремих записів з джерела і тому не виводиться на екран.

Створимо, наприклад, запит для відбору відділів, в яких працює більше десяти співробітників (рис. 56).



```
SELECT Відділи.НазваВідділу,
Count(Співробітники.КодСпівробітника) AS [Кількість за статтю]
FROM Відділи INNER JOIN Співробітники ON Відділи.КодВідділу =
Співробітники.КодВідділу
WHERE (Співробітники.ДатаВлаштування Is Null Or
Співробітники.ДатаВлаштування<=Date()) AND (Співробітники.
ДатаЗвільнення Is Null Or Співробітники.ДатаЗвільнення>=Date())
GROUP BY Відділи.НазваВідділу
HAVING (((Count(Співробітники.КодСпівробітника))>10))
ORDER BY Count(Співробітники.КодСпівробітника) DESC;
```

Рис. 56. Запит *ВідділиЗПонадДесятьмаСпівробітниками* в режимах конструктора та SQL

Зрозуміло, що джерелом даних цього запиту будуть таблиці *Відділи* та *Співробітники*, поєднані по співпадаючих значеннях полів *КодВідділу* (перший етап виконання). У такому джерелі назва кожного відділу буде повторюватися стільки разів, скільки є записів, пов'язаних з цим відділом. На другому етапі виконання запиту необхідно серед всіх співробітників відібрати працюючих зараз. На третьому – згрупувати відібраних співробітників по групах і в кожній групі

підрахувати кількість співробітників (групова операція *Count*). На четвертому етапі виконання потрібно відібрати групи (відділи), в які входить більше десяти записів (співробітників), на п'ятому – відсортувати сформовані групи за спаданням кількості працюючих співробітників, а на останньому – вивести результати відбору на екран.

Бачимо, що умови відбору груп в режимі конструктора задаються у тих самих рядках, що й умови відбору записів, хоча другі діють до, а перші – після групування записів. В режимі ж SQL умови відбору записів перераховуються в реченні *WHERE*, а груп – в реченні *HAVING*.

Ще одним типом запитів, які виконують групування даних, є перехресні запити, які групують записи джерела не тільки по рядках, а й по стовпцях. Майстер створення перехресних запитів для прискорення цього процесу дає змогу послідовно обрати джерело, поля групування по рядках, поле (може бути лише одне) та функцію групування по стовпцях, поле та групову функцію для обчислень в середині перехресної таблиці. Якщо у перехресному запиті необхідно проаналізувати дані декількох таблиць, то з цих таблиць необхідно попередньо створити проміжний запит-джерело, в якому вивести не менше трьох полів (Чому?). Міжнародний стандарт SQL на такі запити не затверджений, тому обмежимося розглядом їх структури в режимі конструктора (як на рис. 57).

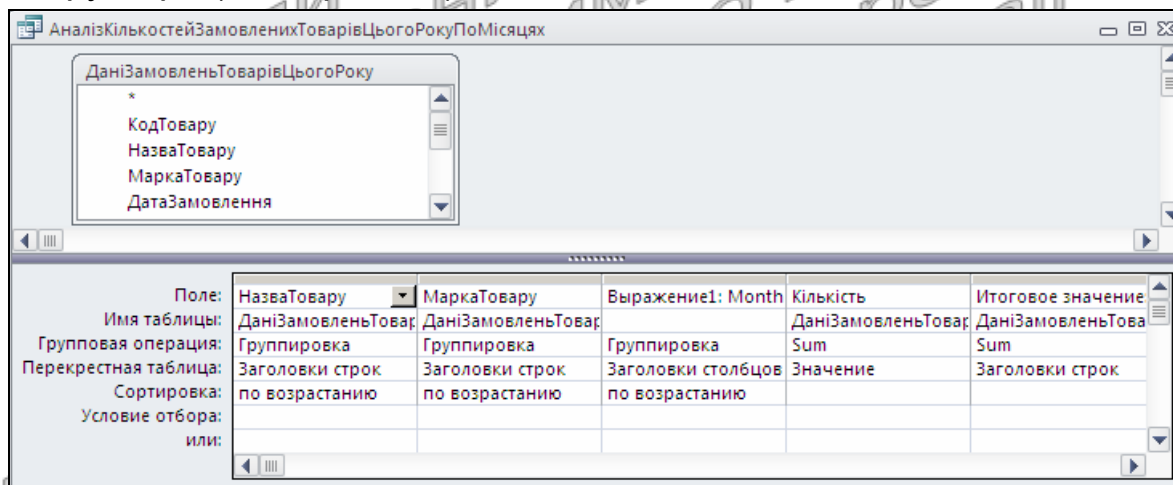


Рис. 57. Перехресний запит

АналізКількостейЗамовленихТоварівЦьогоРокуПоМісяцях в режимі конструктора Бачимо, що в цьому режимі у бланку перехресного запиту відображається додатковий рядок *Перекрестная таблица*. У ньому розробник для кожного поля може обрати одне з чотирьох значень: *Заголовки строк*, *Заголовки столбцов* (лише для одного поля), *Значение* (вираз для обчислення на перетині рядків і стовпців, може бути лише один), *не отображается* (містить умови для відбору записів з джерела). Наприклад, для аналізу загальних кількостей замовлень кожного товару у цьому році по місяцях доцільно дані замовлень цього року згрупувати у рядках по окремих товарах, а в стовпцях – по місяцях, підрахувавши на їх перетині відповідні загальні замовлені кількості (див. рис. 57). Звернемо увагу, що в режимі

конструктора по кожному рядку перехресного запиту можна виконати обчислення декількох підсумків, але для виразу такого підсумку у рядку *Перекрестная таблица* необхідно встановити значення *Заголовки строк* (див., наприклад, останній стовбець бланку на рис. 57).

При виконанні перехресного запиту відображаються лише ті рядки і стовпці, для яких є хоча б одне обчислене значення (як на рис. 58, де відображені дані лише двох місяців).

Назва товару	Марка това	Итоговое зна	6	7
вгущене молоко 'Бахмач'	Бахмач	13	8	5
кава 'Галка'	Львівська каве	6	5	1
майонез 'Делікатесний'	Торчин проду	2		2
морозиво 'Пломбір'	Київське морс	11	6	5
сік 'Карапуз'	Геснерія	20	10	10
цукерки 'Білочка'	Світоч	6	1	5
цукерки 'Зоряне сяйво'	ЕКО продукт	3	1	2
цукерки 'Ромашка'	Світоч	2	1	1
цукерки 'Столичні'	ЕКО продукт	1	1	
цукерки 'Червоний мак'	Світоч	12	7	5

Рис. 58. Перехресний запит *АналізКількостейЗамовленихТоварівЦьогоРоку ПоМісяцях* в режимі таблиці

Але для остаточного оформлення даних, виведених перехресним запитом, недостатньо створити лише цей запит, а на його основі потрібно розробити форму чи звіт. В цих же об'єктах неможливо відобразити змінну кількість полів (фактично, стовпців перехресного запиту). Тому перехресні запити розробляють так, щоб заздалегідь передбачити всі можливі стовпці, які можуть з'явитися при його виконанні і щоб назви цих стовпців не залежали від національних стандартів. Наприклад, для розглянутого вище перехресного запиту – це номери місяців від першого до дванадцятого (а не їх скорочені чи повні назви). Заголовки цих стовпців перераховують в режимі конструктора у полі *Заголовки столбцов* вікна властивостей перехресного запиту (як на рис. 59). Саме ці заголовки надалі будуть стовпцями перехресного запиту і полями для створення на його основі форм чи звітів.

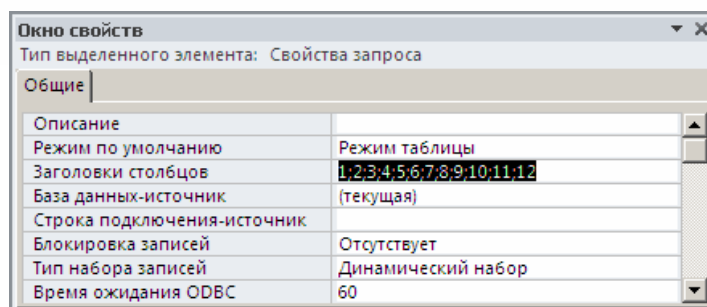


Рис. 59. Задання переліку змінних стовпців перехресного запиту *АналізКількостейЗамовленихТоварівЦьогоРокуПоМісяцях* у вікні його властивостей

Література: [4, С. 115-116; 1, С. 167-171]

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 об'єкта*.
3. Перейдіть в області переходів до розділу *Запроси*.
4. Доповніть запит *ДаніЗамовленьТоварівЗаПеріод* полем *ДеньЗамовлення: Weekday([ДатаЗамовлення])* для відображення номера дня в тижні, у якому було оформлено замовлення. Створіть аналогічне поле *ДеньПостачання* у запиті *ДаніПостачаньТоварівЗаПеріод*.
5. Самостійно створіть запити:
 - 5.1. *СпівробітникуВідділу* – для відображення в алфавітному порядку прізвищ працюючих співробітників вказаного відділу з зазначенням посади, окладу та стажу роботи з дня зарахування в роках. Для введення назви потрібного відділу у бланку запиту для поля *НазваВідділу* з таблиці *Відділи* вкажіть умову відбору *[Введіть назву відділу]*. Переконайтеся у дієздатності створеного запити;
 - 5.2. *СпівробітникуВказаноїНаціональності* – для відображення в алфавітному порядку прізвищ працюючих співробітників вказаної національності з зазначенням статі, відділу та посади;
 - 5.3. *ПостачанняВказаногоТовару* – для відображення дат постачання, кількості, ціни та суми обраного товару (параметром запити має бути не код, а назва товару);
 - 5.4. *ПоколінняСпівробітників* – для відображення в алфавітному порядку прізвищ співробітників, що народилися за вказаний проміжок часу з зазначенням статі, відділу та посади. Для параметрів запити забезпечте при введенні перевірку відповідності типу даних;
 - 5.5. *ЗвільненіСпівробітникиЗаПеріод* та *ВлаштованіСпівробітникиЗаПеріод* – для відображення в алфавітному порядку прізвищ співробітників, що звільнилися чи, відповідно, влаштувалися за вказаний проміжок часу з зазначенням статі, відділу та посади. Для параметрів запитів забезпечте при введенні перевірку відповідності типу даних.

Розробка запитів на вибірку даних з груповими операціями

6. Створіть запит *КількостіСпівробітниківПоВідділах* для відображення алфавітного списку назв відділів та кількостей співробітників, що в них працюють, виконуючи наступні дії:
 - 6.1. Розпочніть створення запити в режимі конструктора та сформуєте його джерело даних з таблиць *Співробітники* та *Відділи*;
 - 6.2. У перший стовпець бланку запити перетягніть поле *НазваВідділу* з таблиці *Відділи*;

- 6.3. В другий стовпець бланку запиту перетягніть поле *КодСпівробітника* з таблиці *Співробітники*. Дайте йому назву *Кількість співробітників*;
- 6.4. Для першого стовпця встановіть ознаку сортування за зростанням;
- 6.5. З метою визначення кількості співробітників кожного відділу згрупуйте записи джерела даних за назвою відділу і підрахуйте для кожної групи кількість відповідних кодів співробітників. Для цього:
 - 6.5.1. Натисніть кнопку Σ (*Итоги*) в контекстному меню бланку запиту чи на вкладці стрічки меню *Конструктор* у групі *Показать или скрыть*;
 - 6.5.2. В рядку *Групповая операция*, що з'явиться на екрані, оберіть зі списку для поля *Назва відділу* значення *Группировка* (встановлюється по замовчуванню), а для поля *Кількість співробітників* – значення *Count*;
- 6.6. Вкажіть повну умову відбору для пакетної обробки даних лише працюючих співробітників (працюючий співробітник – це особа, дата влаштування якої не порожня або не перевищує сьогоднішню, а дата звільнення – не вказана, або не менша сьогоднішньої). Для цього:
 - 6.6.1. Перетягніть в третій та четвертий стовпці бланку запиту з таблиці *Співробітники* поля *ДатаВлаштування* та *ДатаЗвільнення* відповідно;
 - 6.6.2. Задайте для третього стовпця умову відбору записів *Is Null Or <=Date()*, а для четвертого зазначте *Is Null Or >=Date()*;
 - 6.6.3. Оскільки третій та четвертий стовпці задають лише умову для відбору записів, то в рядку *Групповая операция* вкажіть для них значення *Условие* та забезпечте відсутність відображення цих полів при виконанні запиту. Чому ці поля неможливо вивести на екран?;
- 6.7. Перегляньте текст сформованого запиту в режимі SQL. Обгрунтуйте його структуру;
- 6.8. Збережіть створений запит та переконайтеся в його дієздатності.
7. Для всіх запитів, пов'язаних з аналізом даних працюючих співробітників, вкажіть повну умову відбору записів аналогічно попередньому запиту.
8. Самостійно створіть запити:
 - 8.1. *КількостіПрацюючихСпівробітниківПоНаціональностям* – для відображення кількостей працюючих співробітників кожної національності;
 - 8.2. *КількостіПрацюючихСпівробітниківПоСтатях* – для відображення кількостей працюючих співробітників кожної статі.
9. Створіть запит *СумиОформленихЗамовленьПоСпівробітниках ЗаПеріод* для відображення впорядкованих за спаданням загальних сум оформлених замовлень кожним співробітником за введений період з зазначенням їх прізвищ. Для цього:

- 9.1. Розпочніть створення запиту в режимі конструктора та сформуєте його джерело даних з таблиць *Співробітники*, *ЗаголовкиЗамовлень* та *ПунктиЗамовлень*;
- 9.2. В перший стовпець бланку запиту перетягніть поле *ПІБ* з таблиці *Співробітники*;
- 9.3. Для другого стовпця в рядку *Поле* введіть формулу для підрахунку суми пункту замовлення *Сума: Кількість*Ціна* та встановіть для нього ознаку сортування за спаданням;
- 9.4. Перегляньте результати відбору даних запитом в режимі таблиці. Чому прізвища окремих співробітників виводяться декілька разів? Від чого залежить кількість таких записів?
- 9.5. Для підрахунку загальних сум оформлених замовлень кожним співробітником знову перейдіть в режим конструктора та виконайте наступні дії:
 - 9.5.1. Вкажіть, що створюється підсумковий запит. Для цього натисніть кнопку *Итоги* одним з описаних вище способів;
 - 9.5.2. Оскільки запит повинен групувати дані за прізвищами співробітника та для кожного з них підраховувати загальну суму оформлених замовлень, то в рядку *Групповая операция* для поля *ПІБ* оберіть зі списку значення *Группировка* (встановлюється по замовчуванню), а для поля *Сума* – значення *Sum*;
- 9.6. Так як запит повинен аналізувати замовлення лише за введеним періодом, то перетягніть в третій стовпець його бланку поле *ДатаЗамовлення* з таблиці *ЗаголовкиЗамовлень* та вкажіть для нього в рядку *Групповая операция* значення *Условие*, а в рядку *Условие отбора* введіть вираз *between [Введіть початкову дату] and [Введіть кінцеву дату]*;
- 9.7. Для параметрів запиту забезпечте при введенні перевірку відповідності типу даних. Для цього натисніть кнопку *Параметры* на вкладці стрічки меню *Конструктор* та вкажіть у вікні *Параметры запроса*, що з'явиться на екрані, для параметрів *[Введіть початкову дату]* та *[Введіть кінцеву дату]* тип даних *Дата/время* (див. табл. 18);
- 9.8. Перегляньте текст сформованого запиту в режимі SQL. Обгрунтуйте його структуру;
- 9.9. Збережіть створений запит та переконайтеся в його дієздатності.
10. Самостійно створіть запити, впорядкувавши їх записи за спаданням значень основних показників (призначення кожного запиту відповідає його назві):
 - 10.1. *КількостіОформленихЗамовленьПоСпівробітникахЗаПеріод*;
 - 10.2. *СумиОформленихПостачаньПоСпівробітникахЗаПеріод*;
 - 10.3. *КількостіОформленихПостачаньПоСпівробітникахЗаПеріод*;
 - 10.4. *СумиПостачаньПоПостачальникахЗаПеріод*;
 - 10.5. *КількостіПостачаньПоПостачальникахЗаПеріод*;
 - 10.6. *СумиЗамовленьПоКлієнтахЗаПеріод*;
 - 10.7. *КількостіЗамовленьПоКлієнтахЗаПеріод*;
 - 10.8. *КількостіНародженихСпівробітниківПоДняхТижня*.

Корегування параметрів поєднань між таблицями джерела даних

11. Створіть запит *НеосвідченіСпівробітники* для відображення алфавітного списку співробітників, яким в таблиці *ОсвітаСпівробітників* не відповідає жоден запис, з зазначенням відповідних назв відділів та посад. Для цього:
 - 11.1. Розпочніть створення запиту в режимі конструктора та сформуєте його джерело даних з таблиць *Відділи*, *Співробітники* та *ОсвітаСпівробітників*;
 - 11.2. Для аналізу даних всіх співробітників, а не лише тих, що мають відповідні записи в таблиці *ОсвітаСпівробітників*, змініть параметри поєднання таблиць *Співробітники* та *ОсвітаСпівробітників*, виконуючи наступні дії:
 - 11.2.1. Виділіть натисненням лівої кнопки мишки зв'язок між цими таблицями у верхній частині вікна запиту;
 - 11.2.2. В контекстному меню зв'язка оберіть пункт *Параметри об'єднання*;
 - 11.2.3. У вікні *Параметри об'єднання*, що з'явиться на екрані, встановіть перемикач поєднання в положення *Об'єднання ВСЕХ записей из 'Співробітники' и только тех из 'ОсвітаСпівробітників', в которых связанные поля совпадают* та натисніть кнопку ОК;
 - 11.3. Перетягніть в перший стовпець бланку запиту поле *НазваВідділу* з таблиці *Відділи*, а в другий та третій – відповідно поля *ПІБ* та *Посада* з таблиці *Співробітники*;
 - 11.4. Встановіть для першого та другого стовпця ознаку сортування за зростанням;
 - 11.5. Для відбору з базового джерела даних інформації про співробітників без відповідних записів в таблиці *ОсвітаСпівробітників* (тобто рядків джерела даних, в яких поля цієї таблиці не визначені) виконайте наступні дії:
 - 11.5.1. Перетягніть в четвертий стовпець бланку запиту поле *КодОсвіти* з таблиці *ОсвітаСпівробітників*;
 - 11.5.2. В рядку *Условие отбора* для цього стовпця введіть вираз *Is Null*;
 - 11.6. Самостійно забезпечте відображення на екрані даних лише перших трьох стовпців при виконанні запиту;
 - 11.7. Перегляньте текст сформованого запиту в режимі SQL. Обгрунтуйте структуру всіх частин запиту;
 - 11.8. Збережіть створений запит та переконайтеся в його дієздатності.
12. Самостійно створіть запити, впорядкувавши їх записи за зростанням значень основного поля (призначення кожного запиту відповідає його назві):
 - 12.1. *ТовариБезПостачань*;
 - 12.2. *ТовариБезПродажу*;
 - 12.3. *ПостачальникиБезПостачань*;
 - 12.4. *КлієнтиБезЗамовлень*;
 - 12.5. *ПостачанняБезПунктів*;
 - 12.6. *ЗамовленняБезПунктів*.

Створення запитів на основі запитів

13. Створіть запит *ТовариБезПостачаньЗаПеріод* для формування алфавітного списку товарів, що не надходили протягом введеного періоду часу. Для формування такого списку слід спочатку визначити товари, що **надходили** за введений період, а потім з переліку всіх товарів виключити знайдені товари та вивести результати виключення на екран. Створений раніше запит *ДаніПостачаньТоварівЗаПеріод* відбирає дані постачань товарів за введений період, тому модифікуйте його структуру та використайте при створенні нашого запиту:
 - 13.1. Відкрийте запит *ДаніПостачаньТоварівЗаПеріод* в режимі конструктора, перетягніть на місце першого поля бланку запиту (а не замість нього) поле *КодТовару* з таблиці *ПунктиПостачань*, закрийте запит та погодьтеся з збереженням внесених змін;
 - 13.2. Розпочніть створення запиту в режимі конструктора та сформуєте його джерело даних з таблиці *Товари* (містить дані всіх товарів) та запиту *ДаніПостачаньТоварівЗаПеріод* (містить дані товарів, які постачалися за період);
 - 13.3. Переконайтеся, що зв'язок між таблицями джерела даних встановлено поєднанням полів *КодТовару*. При відсутності такого зв'язку встановіть його перетягування відповідних полів між таблицями (аналогічно способу встановлення зв'язків між таблицями у схемі даних);
 - 13.4. Самостійно встановіть між таблицями джерела даних запиту параметр поєднання *Объединение ВСЕХ записей из 'Товары' и только тех из 'ДаніПостачаньТоварівЗаПеріод', в которых связанные поля совпадают*;
 - 13.5. Перетягніть в перший та другий стовпці бланку запиту поля *НазваТовару* та *МаркаТовару* з таблиці *Товари*. Встановіть для цих стовпців ознаку сортування за зростанням;
 - 13.6. Для відбору лише товарів, що не постачалися протягом вказаного періоду, перетягніть в третій стовпець бланку поле *КодТовару* з запиту *ДаніПостачаньТоварівЗаПеріод*, вкажіть для нього умову відбору *Is Null* та приховайте його відображення на екрані при виконанні запиту;
 - 13.7. Перегляньте текст сформованого запиту в режимі SQL. Обгрунтуйте структуру всіх частин запиту;
 - 13.8. Збережіть створений запит під назвою *ТовариБезПостачаньЗаПеріод* та переконайтеся в його дієздатності.
14. Самостійно створіть запити, впорядкувавши їх записи за зростанням значень основного поля (призначення кожного запиту відповідає його назві):
 - 14.1. *ТовариБезПродажуЗаПеріод*;
 - 14.2. *ПостачальникиБезПостачаньЗаПеріод*;
 - 14.3. *КлієнтиБезЗамовленьЗаПеріод*.
 - 14.4. *МаксимальніСумиПостачаньПоПостачальниках* (створивши перед цим запит *СумиПостачань*);

- 14.5. *МінімальніСумиПостачаньПоПостачальниках*;
- 14.6. *МаксимальніСумиЗамовленьПоКлієнтах* (створивши перед цим запит *СумиЗамовлень*);
- 14.7. *МінімальніСумиЗамовленьПоКлієнтах*.

Створення складних запитів на основі запитів

15. Створіть запит *ЗалишкиТоварівНаСкладі* для відображення алфавітного списку назв, марок і загальних отриманих, замовлених та наявних кількостей для кожного товару. Для цього:
 - 15.1. Створіть запит *КількостіОтриманихТоварів* з зазначенням коду та загальної отриманої кількості кожного товару (це поле повинно мати заголовок *ВсьогоОтримано*);
 - 15.2. Створіть запит *КількостіЗамовленихТоварів* з зазначенням коду та загальної замовленої кількості кожного товару (це поле повинно мати заголовок *ВсьогоЗамовлено*);
 - 15.3. Розпочніть створення запиту *ЗалишкиТоварівНаСкладі* в режимі конструктора та сформуєте його джерело даних з таблиці *Товари* та запитів *КількостіОтриманихТоварів* та *КількостіЗамовленихТоварів*;
 - 15.4. Оскільки окремі товари можуть взагалі не постачатися чи не реалізовуватися, то встановіть параметри поєднання запитів з таблицею *Товари* по полях *КодТовару* для забезпечення аналізу всіх товарів;
 - 15.5. Перетягніть в перший та другий стовпці бланку запиту відповідно поля *НазваТовару* та *МаркаТовару* з таблиці *Товари* (Чому?). Встановіть для цих стовпців ознаку сортування за зростанням;
 - 15.6. У рядок *Поле* третього стовпця бланку запиту введіть вираз *Отримано: IIF(IsNull(ВсьогоОтримано);0;ВсьогоОтримано)* для відображення кількості отриманого товару;
 - 15.7. У рядок *Поле* четвертого стовпця бланку запиту введіть вираз *Замовлено: IIF(IsNull(ВсьогоЗамовлено);0;ВсьогоЗамовлено)* для відображення кількості замовленого товару;
 - 15.8. У п'ятому стовпці самостійно введіть вираз для підрахунку залишку товарів, як різниці між отриманою та замовленою кількостями;
 - 15.9. Забезпечте відображення при виконанні запиту записів лише про ті товари, що надходили **або** замовлялися на складі;
 - 15.10. Перегляньте текст сформованого запиту в режимі SQL. Обгрунтуйте структуру всіх частин запиту;
 - 15.11. Збережіть створений запит та переконайтеся в його дієздатності.
16. Самостійно створіть запит *ЗалишкиТоварівПоВідділах* для відображення залишків товарів у кожному відділі. При цьому врахуйте, що кожен товар може знаходитися у кожному відділі, тому спочатку створіть запити *ТовариВідділів* (джерелом даних цього запиту будуть непоєднані таблиці *Відділи* та *Товари*), *КількостіОтриманихТоварівПоВідділах* (джерело даних цього

запиту має містити поєднані таблиці *Співробітники*, *ЗаголовкиПостачань* і *ПунктиПостачань*) та *КількостіЗамовленихТоварівПоВідділах*. У джерелі даних запиту *ЗалишкиТоварівПоВідділах* запит *ТовариВідділів* має поєднуватися з двома іншими запитами як по полю *КодТовару*, так і по полю *КодВідділу*, оскільки залишки мають обчислюватися для **кожного** товару у **кожному** з тих відділі, де він надходив **або** замовлявся.

Використання властивостей запитів для ефективного аналізу даних

17. Створіть запит *ПостачанняНаМаксимальнуСуму* для відображення даних постачання на максимальну суму. Для цього:
 - 17.1. Розпочніть створення запиту в режимі конструктора та сформуєте його джерело даних із запиту *СумиПостачань*;
 - 17.2. Перетягніть з джерела даних в бланк запиту поля *КодПостачальника*, *ДатаПостачання* та *Сума*;
 - 17.3. Для стовпця *Сума* встановіть ознаку сортування даних за спаданням;
 - 17.4. Для відображення даних лише першого постачання у відсортованому списку відкрийте вікно властивостей запиту, обравши відповідний пункт контекстного меню заголовка вікна чи вільного поля джерела даних, у виведеному вікні задайте для поля *Набор значень* значення *1* та закрийте вікно властивостей запиту;
 - 17.5. Перегляньте текст сформованого запиту в режимі SQL. Як відобразилася введена властивість запиту в цьому режимі;
 - 17.6. Збережіть створений запит та переконайтеся в його дієздатності (Цей самий запит можна було б створити з використанням групових операцій, але тоді довелося б створювати два запити: перший – для визначення максимальної суми постачання, а другий – для відбору даних постачання з цією сумою (в джерелі даних треба було б поєднати перший запит з запитом *СумиПостачань* по полю *Сума*)).
18. Перегляньте інші властивості запитів, полів та таблиць джерела даних, встановіть їх призначення та опишіть у письмовому звіті в таблиці згідно структури табл. 19.

Таблиця 19

Властивості запиту

Назва властивості	Призначення властивості	Відображення властивості в режимі SQL
...

19. Самостійно створіть запити:
 - 19.1. *ЗамовленняНаМаксимальнуСуму* – для відображення даних замовлення на максимальну суму;
 - 19.2. *ДесяткаОсновнихПостачальників* – для відображення десяти постачальників з максимальними загальними сумами постачань (створивши перед цим запит-джерело даних *РейтингПостачальників*);

- 19.3. *ДесяткаОсновнихКлієнтів* – для відображення десяти клієнтів з максимальними загальними сумами замовлень (створивши перед цим запит-джерело даних *РейтингКлієнтів*);
- 19.4. *ДесяткаОсновнихТоварів* – для відображення десяти товарів з максимальними загальними сумами замовлень (створивши перед цим запит-джерело даних *РейтингТоварів*).

Створення перехресних запитів

20. Створіть перехресний (з групуванням по рядках і стовпцях) запит *АналізСумЗамовленихТоварівЦьогоРокуПоМісяцях* для відображення щомісячних сум замовлень кожного товару (в рядках запиту мають вказуватися дані замовлень окремих товарів, а в стовпцях – дані замовлень по кожному місяцю). Для цього:
- 20.1. Самостійно створіть запит-джерело даних *ДаніЗамовленьТоварівЦьогоРоку*, що відбирає дані замовлень поточного року, з аналогічною структурою до запиту *ДаніЗамовленьТоварівЗаПеріод*. З метою відбору даних створіть **додаткове** поле для обчислення року від дати замовлення та вкажіть для нього умову відбору *Year(Date())* і знищіть попередні умови та параметри відбору стосовно відбору дат за період;
- 20.2. Створіть заданий перехресний запит за допомогою майстра, виконавши наступні дії:
- 20.2.1. Завантажте майстер створення перехресних запитів, натиснувши на вкладці *Создание* стрічки меню кнопку *Мастер запросов* та обравши у виведеному вікні *Новый запрос* пункт *Перекрестный запрос*;
- 20.2.2. На першому кроці майстра оберіть запит *ДаніЗамовленьТоварівЦьогоРоку* для формування джерела даних перехресного запиту;
- 20.2.3. На другому кроці майстра перемістіть з списку доступних в список обраних поля *НазваТовару* та *МаркаТовару* для формування заголовків рядків;
- 20.2.4. На третьому кроці майстра оберіть поле *ДатаЗамовлення* для формування заголовків стовпців;
- 20.2.5. На четвертому кроці майстра оберіть зі списку інтервал групування *Місяць* для поля-заголовка стовпця;
- 20.2.6. На п'ятому кроці майстра оберіть поле *Сума* та групову функцію *Сумма* для здійснення підрахунків на перетині рядків і стовпців, оскільки запит повинен знаходити загальну щомісячну суму замовлень по кожному товару;
- 20.2.7. На останньому кроці майстра задайте назву запиту та завершіть його створення;
- 20.3. Переконайтеся у дієздатності створеного запиту;

- 20.4. Перегляньте текст сформованого запиту в режимах конструктора та SQL. Визначте призначення та обгрунтуйте структуру всіх частин запиту в цих режимах.
21. Самостійно створіть перехресні запити (призначення кожного запиту відповідає його назві):
- 21.1. Аналіз Сум Заамовлених Товарів Цього Року По Кварталах;
 - 21.2. Аналіз Сум Заамовлених Товарів Цього Року По Днях Тижня;
 - 21.3. Аналіз Кількостей Заамовлених Товарів Цього Року По Місяцях;
 - 21.4. Аналіз Кількостей Заамовлених Товарів Цього Року По Кварталах;
 - 21.5. Аналіз Кількостей Заамовлених Товарів Цього Року По Днях Тижня;
 - 21.6. Аналіз Кількостей Заамовлень Цього Року По Клієнтах Та Місяцях (створивши попередньо запит Заамовлення Цього Року);
 - 21.7. Аналіз Кількостей Заамовлень Цього Року По Клієнтах Та Кварталах;
 - 21.8. Аналіз Кількостей Заамовлень Цього Року По Клієнтах Та Днях Тижня.

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. Створіть у власній БД аналогічні до створених на лабораторній роботі запити з груповими операціями та перехресні запити.

Контрольні запитання

1. Що позначає символ * в першому рядку образів таблиць джерела даних запиту?
2. Як створити, перемістити чи видалити поля бланку запиту?
3. Які групові операції можна виконати над полями при конструюванні підсумкових запитів? Як кожна з цих операцій описується в режимі SQL?
4. Які параметри поєднань можна встановлювати між таблицями джерела даних? Який з цих параметрів встановлюється автоматично? Де описуються параметри поєднань в режимі SQL?
5. Для яких запитів результати виконання відрізняються від результатів перегляду в режимі таблиці?
6. Які з властивостей запиту, полів та таблиць джерела даних відображаються в режимі SQL, а які – ні? Чому?
7. Чим відрізняється перехресний запит від підсумкового? Які обмеження накладаються на джерело даних перехресного запиту? Які обмеження накладаються на структуру перехресного запиту?