

Лабораторна робота № 11

Тема. Створення запитів на вибірку та на поєднання в режимі SQL.

Мета. Формування вмінь та навичок створення запитів на вибірку даних в режимі SQL для аналізу даних таблиць. Закріплення вмінь і навичок використання запитів для аналізу даних. Усвідомлення послідовності етапів виконання запитів на вибірку та ролі зв'язків між таблицями при формуванні джерела даних. Вдосконалення вмінь створення та корегування об'єктів додатків СУБД MS Access.

Теоретичні відомості

У випадках, коли необхідно одночасно вивести результати виконання декількох запитів, застосовують **запити на поєднання**. Ці запити поєднують результати відбору з декількох джерел (а не з одного, як, наприклад, запити на вибірку), тому створювати і змінювати їх структуру можливо лише в режимі SQL. Синтаксис запитів на поєднання в цьому режимі такий:

```
SELECT <поля результатів відбору 1>  
FROM <джерело даних 1>  
[WHERE <умови відбору записів 1>]  
[GROUP BY <поля групування 1>]  
[HAVING <умови відбору груп 1>]  
UNION | UNION ALL  
SELECT <поля результатів відбору 2>  
FROM <джерело даних 2>  
[WHERE <умови відбору записів 2>]  
[GROUP BY <поля групування 2>]  
[HAVING <умови відбору груп 2>]  
...  
[ORDER BY <поля сортування>];
```

Бачимо, що запити на поєднання можуть об'єднувати результати багатьох запитів на вибірку. Необхідно лише забезпечити, щоб кількість полів виводу у кожному з цих запитів-частин на вибірку була однаковою і кожне поле у всіх частинах мало сумісний тип даних.

Оператор **UNION** виводить лише унікальні записи з усіх частин (тобто автоматично відкидає записи-дублікати). Для відображення ж всіх записів з усіх запитів-частин необхідно між ними в режимі SQL вказувати оператор **UNION ALL**. Наголосимо, що речення сортування **ORDER BY** вказується при потребі в кінці всього запиту на поєднання лише один раз і забезпечує сортування між собою всіх записів з усіх частин. В цьому реченні записуються або поля з першої частини, або (що використовується частіше) номери стовпців.

Література: [2, С. 372-436]

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 цій області до розділу *Запроси*.

Створення найпростіших запитів на вибірку з сортуванням записів в режимі SQL

4. Створіть запит *SQLАлфавітнийСписокПостачальників* в режимі SQL для формування алфавітного списку постачальників з зазначенням адреси та дати реєстрації. Для цього:
 - 4.1. Розпочніть створення запиту в режимі конструктора, натиснувши на вкладці стрічки меню *Создание* у групі *Запросы* кнопку *Конструктор запросов*;
 - 4.2. Закрийте вікно *Добавление таблицы*, оскільки текст запиту буде вводиться в режимі SQL;
 - 4.3. Для створення тексту запиту перейдіть в режим SQL одним з двох способів: за допомогою пункту контекстного меню *Режим SQL* вільного поля області джерела даних запиту чи з використанням аналогічного пункту кнопки *Главная* ► *Режим стрічки меню*;
 - 4.4. У вікні режиму SQL введіть текст запиту:

```
SELECT Постачальники.НазваПостачальника,  
Постачальники.Адреса, Постачальники.ДатаРеєстрації  
FROM Постачальники  
ORDER BY Постачальники.НазваПостачальника;
```
 - 4.5. Завантажте створений запит на виконання, обираючи в контекстному меню заголовка вікна запиту чи у списку кнопки *Режим* стрічки меню позицію *Режим таблиць*;
 - 4.6. Перегляньте структуру створеного запиту в режимі конструктора. Де в цьому режимі відображається кожна структурна частина запиту, створеного в режимі SQL?
 - 4.7. Закрийте вікно конструктора та збережіть запит під назвою *SQLАлфавітнийСписокПостачальників*.
5. Самостійно створіть аналогічний запит *SQLАлфавітний-СписокКлієнтів* для відображення в алфавітному порядку назв клієнтів з зазначення місця роботи (назви необхідних полів можна вписати чи безпосередньо скопіювати з вікна режиму конструктора таблиці *Клієнти*). Переконайтеся в дієздатності запиту та перегляньте і проаналізуйте його структуру в режимі конструктора.

Створення обчислювальних полів та формування умов відбору записів джерела даних запитів в режимі SQL

6. Доповніть запит *SQLАлфавітнийСписокПостачальників* полем *День реєстрації* для зазначення дня тижня, в якому було зареєстровано постачальника. Для цього:

6.1. Відкрийте вказаний запит в режимі SQL;

6.2. Відредагуйте текст запиту до вигляду:

```
SELECT Постачальники.НазваПостачальника,  
Постачальники.Адреса, Постачальники.ДатаРеєстрації,  
Weekday([ДатаРеєстрації]) AS [День реєстрації]  
FROM Постачальники  
ORDER BY Постачальники.НазваПостачальника;
```

6.3. Завантажте відредагований запит на виконання, переглядаючи його в режимі таблиці;

6.4. Перегляньте запит в режимі конструктора. Як в цьому режимі відображається обчислювальне поле?

6.5. Закрийте запит та збережіть внесені зміни.

7. Створіть в режимі SQL запит *SQLНекоректноЗареєстрованіПостачальники* для відображення даних постачальників, зареєстрованих в неділю. Для цього:

7.1. Скопіюйте запит *SQLАлфавітнийСписокПостачальників* в буфер обміну та вставте його під назвою *SQLНекоректноЗареєстрованіПостачальники*, оскільки ці запити мають однакові джерела даних, поля для відображення та сортування;

7.2. Відкрийте запит *SQLНекоректноЗареєстрованіПостачальники* в режимі SQL та відредагуйте його текст до вигляду:

```
SELECT Постачальники.НазваПостачальника, Постачальники.  
Адреса, Постачальники.ДатаРеєстрації,  
Weekday([ДатаРеєстрації]) AS [День реєстрації]  
FROM Постачальники  
WHERE Weekday([ДатаРеєстрації])=1  
ORDER BY Постачальники.НазваПостачальника;
```

7.3. Завантажте відредагований запит на виконання;

7.4. Перегляньте запит в режимі конструктора. Як в цьому режимі відображається умова відбору записів з джерела даних?

7.5. Закрийте запит та збережіть внесені зміни.

8. Самостійно доповніть в режимі SQL запит *SQLАлфавітнийСписокКлієнтів* полями *Число реєстрації*, *Місяць реєстрації* та *День реєстрації* (для відображення дня тижня, в якому було зареєстровано клієнта).

9. Самостійно створіть запит *SQLНекоректноЗареєстрованіКлієнти* для відображення даних клієнтів, зареєстрованих в неділю.

10. Перегляньте запит *ПрацюючіПенсіонера* в режимах конструктора та SQL. Як поєднуються в режимі SQL умови відбору, записані в режимі конструктора в одному рядку? В різних рядках?

Конструювання параметрів поєднання таблиць джерела даних в режимі SQL

11. Створіть запит *SQLСпівробітникиБезВідділів* для відображення прізвищ співробітників, у яких не вказаний відділ, де вони працюють. Для цього:

11.1. Розпочніть створення запиту та перейдіть в режим SQL;

11.2. У вікні режиму SQL введіть текст запиту:

```
SELECT Співробітники.ПІБ  
FROM Відділи RIGHT JOIN Співробітники ON  
Відділи.КодВідділу = Співробітники.КодВідділу  
WHERE Відділи.КодВідділу Is Null  
ORDER BY Співробітники.ПІБ;
```

11.3. Завантажте введений запит на виконання;

11.4. Перегляньте запит в режимі конструктора. Як в цьому режимі відображається обраний параметр поєднання між таблицями джерела даних? Встановіть в режимі конструктора інші параметри поєднання таблиць джерела даних та з'ясуйте, як ці зміни відображаються в режимі SQL;

11.5. Закрийте та збережіть створений запит.

12. Самостійно створіть в режимі SQL такі запити:

12.1. *SQLСпівробітникиБезПостачань*;

12.2. *SQLСпівробітникиБезПостачаньЗаПеріод*;

12.3. *SQLСпівробітникиБезЗамовлень*;

12.4. *SQLСпівробітникиБезЗамовленьЗаПеріод*.

Створення запитів з групуванням даних в режимі SQL

13. Створіть запит *SQLКількостіЗареєстрованихПостачальників-ПодняхТижня* в режимі SQL для відображення кількостей зареєстрованих постачальників за кожен день тижня. Для цього:

13.1. Розпочніть створення запиту та перейдіть в режим SQL;

13.2. У вікні режиму SQL введіть текст запиту:

```
SELECT SQLАлфавітнийСписокПостачальників.[День реєстрації],  
Count(SQLАлфавітнийСписокПостачальників.НазваПостачальника)  
AS [Кількість постачальників]  
FROM SQLАлфавітнийСписокПостачальників  
GROUP BY SQLАлфавітнийСписокПостачальників.[День реєстрації]  
ORDER BY SQLАлфавітнийСписокПостачальників.[День реєстрації];
```

13.3. Завантажте введений запит на виконання;

13.4. Перегляньте даний запит в режимі конструктора. Як в цьому режимі відображаються поля групування і поля з груповими операціями?

13.5. Закрийте та збережіть створений запит.

14. Самостійно створіть в режимі SQL запити *SQLКількостіЗареєстрованихКлієнтівПодняхТижня*, та *SQLКількостіЗареєстрованихКлієнтівПоМісяцях*.

Створення запитів на поєднання записів в режимі SQL

15. Створіть запит *SQLПостачальникиТаКлієнти* в режимі SQL для відображення алфавітного списку постачальників та клієнтів з зазначенням дати реєстрації. Для цього:
 - 15.1. Внесіть у форми *Постачальники* та *Клієнти* по запису з однаковими прізвищами та ініціалами і датами реєстрації;
 - 15.2. Розпочніть створення запиту та перейдіть в режим SQL;
 - 15.3. Не закриваючи вікна запиту, відкрийте в режимі SQL додатковий запит *АлфавітнийСписокПостачальників* та скопіюйте через буфер обміну його текст у вікно створюваного запиту. Закрийте додатковий запит;
 - 15.4. Не закриваючи вікна запиту, відкрийте в режимі SQL додатковий запит *АлфавітнийСписокКлієнтів* та скопіюйте через буфер обміну його текст у вікно створюваного запиту після скопійованого раніше тексту. Закрийте додатковий запит;
 - 15.5. Відредагуйте текст нового запиту до вигляду:

```
SELECT Постачальники.НазваПостачальника,  
Постачальники.ДатаРеєстрації  
FROM Постачальники  
UNION  
SELECT Клієнти.НазваКлієнта, Клієнти.ДатаРеєстрації  
FROM Клієнти  
ORDER BY 1;
```
 - 15.6. Завантажте введений запит на виконання та запам'ятайте кількість записів, що відображається при цьому на екрані;
 - 15.7. Спробуйте переглянути його структуру в режимі конструктора. Чому це не вдалося?
 - 15.8. Поверніться в режим SQL і після оператора UNION допишіть слово ALL. Знову завантажте введений запит на виконання і порівняйте кількість записів, що відображається на екрані, з кількістю записів, отриманою раніше. Віднайдіть ті записи, що призвели до розбіжностей результатів;
 - 15.9. Закрийте та збережіть створений запит.
16. Самостійно створіть запит *SQLФізичніОсоби* для відображення алфавітного списку унікальних прізвищ та ініціалів співробітників, постачальників і клієнтів.

Завершальний етап заняття. Повторення вивченого матеріалу

17. Стисніть створену БД.
18. Завершіть роботу Access.
19. Віднайдіть створену БД та заархівуйте її вміст.
20. Збережіть створений архів на власному флеш-носії в двох екземплярах.
21. Доповніть власну БД запитам на поєднання, аналогічними до створених на лабораторній роботі.

Контрольні запитання

1. Чи відрізняється послідовність виконання запитів, створених в режимі SQL, від послідовності виконання запитів, створених в режимі конструктора?
2. Яка загальна структура запитів на вибірку в режимі SQL?
3. Чим відрізняється синтаксис опису обчислювальних полів в режимах конструктора та SQL?
4. Як в режимі SQL описуються зв'язки та параметри поєднання між таблицями?
5. Де в режимі SQL описуються поля групування і де – поля з груповими операціями?
6. Де в режимі SQL описуються умови відбору записів джерела даних і де – умови відбору груп?
7. Як в Access слід створювати нові, але подібні до існуючих запити?
8. Яка загальна структура запитів на поєднання в режимі SQL?
9. Які основні вимоги висуваються до кожного складового запита на вибірку у загальному запиті на поєднання?

З питань продажів
друкованих примірників
за оптовими цінами
чи електронного варіанту
цього посібника без реклами
звертайтеся на e-mail
books_shoptko@ukr.net