

Лабораторна робота № 12

Тема. Розробка та використання запитів на корегування даних.

Мета. Формування вмінь та навичок конструювання запитів на створення таблиць, доповнення, оновлення та знищення записів, використання будівничого виразів. Вдосконалення вмінь і навичок створення та корегування структури запитів в режимі конструктора. Систематизація знань можливостей використання запитів для пакетного корегування даних.

Теоретичні відомості

Дослідимо можливості пакетного корегування даних за допомогою відповідних запитів та їх синтаксис як в режимі конструктора, так і SQL. Нагадаємо, що для виконання дій, передбачених такими запитами, необхідно або завантажувати їх з вікна додатку Access або натискати кнопку стрічки меню *Конструктор* ► *Виконати*. Перед виконанням запиту на корегування рекомендується перейти в *Режим таблиць* щоб переглянути записи, над якими **буде виконуватися** корегування.

Для перетворення запиту на вибірку в запит на корегування, перебуваючи в режимі конструктора, необхідно у групі стрічки меню *Конструктор* ► *Тип запиту* змінити його тип (рис. 60), натиснувши відповідну кнопку. В режимі ж SQL для таких дій необхідно змінювати саму структуру запиту.

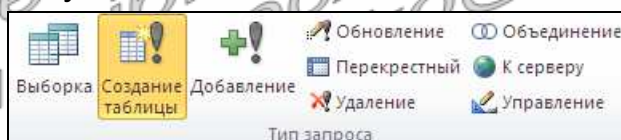


Рис. 60. Кнопки стрічки меню для зміни типу запиту в режимі конструктора

Серед запитів на корегування найменше відрізняються за структурою від запитів на вибірку **запити на створення таблиць** (📄!). Обираючи такий тип запиту в режимі конструктора (див. рис. 60), потрібно лише додатково вказати назву нової таблиці, в яку будуть записуватися результати вибірки даних цим запитом. В режимі ж SQL для перетворення запиту на вибірку у запит на створення таблиці необхідно після речення `SELECT` дописати тільки

```
INTO <ім'я нової таблиці> .
```

Наголосимо, що ім'я таблиці, яку створює запит, не має співпадати з назвою цього запиту, інакше при виконанні такого запиту буде створено нову таблицю, але сам запит буде знищено, навіть якщо при цьому він відкритий в режимі конструктора.

Запити на створення таблиць найчастіше застосовуються для автоматичного архівного зберігання поточного стану даних таблиць чи результатів їх аналізу. Наприклад, для зберігання сум замовлень кожного клієнта в кінці поточного місяця доцільно створити запит, який в режимі SQL запишеться так:

```
SELECT Клієнти.НазваКлієнта, Sum([Кількість]*[Ціна]) AS Сума
INTO СумиЗамовленьПоКлієнтахЗаПоточнийМісяць
FROM Клієнти INNER JOIN (ЗаголовкиЗамовлень INNER JOIN
ПунктиЗамовлень ON ЗаголовкиЗамовлень.КодЗамовлення =
ПунктиЗамовлень.КодЗамовлення) ON Клієнти.КодКлієнта =
ЗаголовкиЗамовлень.КодКлієнта
WHERE (((Month([ДатаЗамовлення])=Month(Date())) AND
((Year([ДатаЗамовлення])=Year(Date()))))
GROUP BY Клієнти.НазваКлієнта
ORDER BY Sum([Кількість]*[Ціна]) DESC;
```

Для того, щоб створювана таблиця не була автоматично знищена після наступного виконання цього запиту у наступному місяці, щоразу після виконання запиту її слід перейменовувати, додатково фіксуючи у назві поточний рік та місяць, а при потребі – і поточну дату.

Для запитів на оновлення (🔧) вказувати ім'я корегованої таблиці не потрібно, адже під час їх виконання змінюються записи джерела даних. В режимі конструктора після вибору цього типу запитів (див. рис. 60, кнопка *Обновление*) у бланку замість рядків *Групповая операция*, *Сортировка* та *Вывод на экран* виводиться рядок *Обновление*, де вказуються значення, що заносяться в обране поле з джерела. Для полів відбору значення в цьому рядку не вказується. Наприклад, для обнулення проценту знижки при поверненні товарів доцільно в режимі конструктора створити запит, аналогічний наведеному на рис. 61. Зверніть увагу: для прискорення виконання цього запиту обнулюється процент знижки не для всіх повернень (*Кількість < 0*), а лише для тих, де процент знижки не вказаний або додатний (*Is Null or >0*).

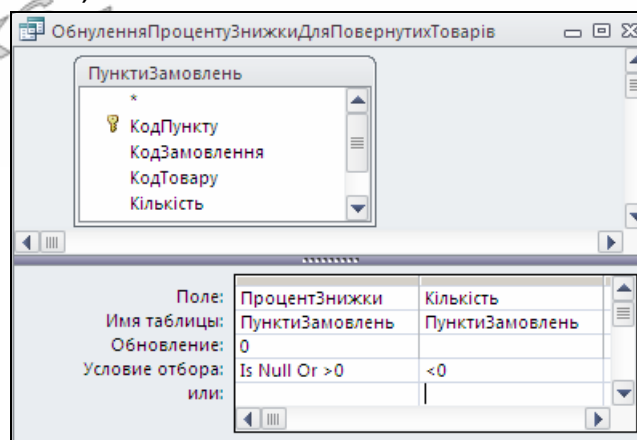


Рис. 61. Структура запиту *ОбнуленняПроцентуЗнижкиДляПовернутихТоварів* в режимі конструктора

В режимі SQL синтаксис запитів на оновлення такий:

```
UPDATE <джерело даних>
SET <ім'я поля> = <вираз> [, ...]
[WHERE <умови відбору записів>];
```

тобто поля і значення, які в них заносяться, перераховуються в реченні SET, а сам запит починається словом UPDATE. Наприклад, розглянутий вище запит в режимі SQL має вигляд:

```
UPDATE ПунктиЗамовлень SET ПунктиЗамовлень.ПроцентЗнижки = 0
WHERE (ПунктиЗамовлень.ПроцентЗнижки Is Null Or ПунктиЗамовлень.
ПроцентЗнижки > 0) AND ПунктиЗамовлень.Кількість < 0;
```

У запитах на оновлення ніколи не змінюються обчислювальні поля та поля-лічильники.

Кожен з **запитів на додавання** (👤) застосовується для доповнення новими записами однієї з існуючих таблиць. Тому в режимі конструктора після вибору даного типу запиту (див. рис. 60, кнопка *Добавление*) потрібно вказати назву цієї таблиці. Після цього у бланку запиту з'явиться додатковий рядок *Добавление*, де **обираються** поля таблиці для доповнення, в які записується значення з рядка *Поле*. В доповнювану таблицю записується стільки записів, скільки є у джерелі. Запит на додавання може й не мати джерела. Тоді він доповнює таблицю одним записом значеннями виразів з рядка *Поле*. У запитах на додавання, аналогічно запитам на оновлення, ніколи не записуються значення в обчислювальні поля та не рекомендується встановлювати поля-лічильники. Access перевіряє цілісність як окремих полів, так і цілих створюваних записів, тому всі обов'язкові поля доповнюваної таблиці повинні або мати значення по замовчуванню або заповнюватися таким запитом. Наприклад, запит на доповнення таблиці *Клієнти* записами з даних тих постачальників, які у цій таблиці відсутні, в режимі конструктора виглядає як на рис. 62. Зверніть увагу: у поле *КодКлієнта* значення не записуються. Воно лише використовується для відбору даних тих постачальників, які відсутні в таблиці *Клієнти*.

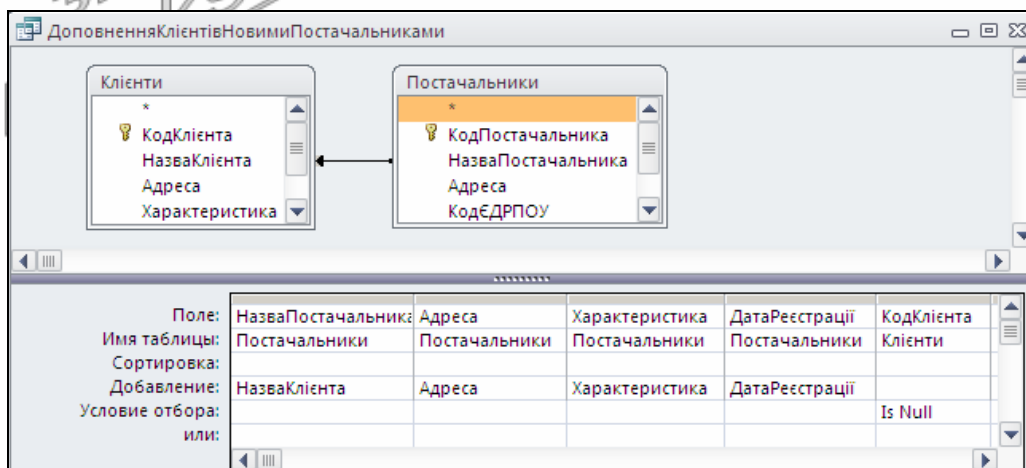


Рис. 62. Структура запиту *ДоповненняКлієнтівНовимиПостачальниками* в режимі конструктора

В режимі SQL запити на додавання записуються згідно синтаксису:

```
INSERT INTO <ім'я таблиці> [(<поле1>, [<поле2> ...])]  
VALUES ((<значення1>, [<значення2> ...]) |  
SELECT <поле1>, [<поле2> ...]  
[FROM <джерело даних>]  
[WHERE <умови відбору записів>]  
[GROUP BY <поля групування>]  
[HAVING <умови відбору груп>]  
[ORDER BY <поля сортування>]; .
```

Бачимо, що поля доповнюваної таблиці і значення, які в них заносяться, записуються в різних реченнях команди SQL. Поля для зберігання значень перераховуються в реченні INSERT INTO в круглих дужках після назви таблиці (коли послідовно встановлюються значення всіх полів, то їх можна не перераховувати). Значення ж для полів обчислюються з виразів після VALUES (якщо вставляється один запис) або послідовно беруться з полів результатів відбору команди SELECT. Наприклад, запит, наведений на рис. 62, в режимі SQL виглядає так:

```
INSERT INTO Клієнти (НазваКлієнта, Адреса, Характеристика,  
ДатаРеєстрації)  
SELECT Постачальники.НазваПостачальника, Постачальники.Адреса,  
Постачальники.Характеристика, Постачальники.ДатаРеєстрації  
FROM Постачальники LEFT JOIN Клієнти ON  
Постачальники.НазваПостачальника = Клієнти.НазваКлієнта  
WHERE Клієнти.КодКлієнта Is Null; .
```

І нарешті, для пакетного видалення записів з окремих таблиць, що формують відібрані записи джерела, створюють **запити на видалення** (✖). Кожен з таких запитів може видаляти записи з однієї чи декількох таблиць джерела. Тому при виборі цього типу запиту в режимі конструктора у бланку з'являється додатковий рядок *Удаление*, в якому для кожного стовпця можна обрати одне з двох значень: *Из* – для позначення кожної таблиці, з якої виконується вилучення (при цьому з даної таблиці в рядок *Поле* такого стовпця необхідно перетягнути *) чи *Условие* – для вказування полів чи виразів, згідно яких виконується відбір записів джерела (для таких полів потрібно задавати умови відбору в розміщених нижче рядках). Наприклад, запит з рис. 63 використовується для знищення тих пунктів постачань (перший стовбець), в яких кількість не вказана або відсутня (другий стовбець).

Перед завантаженням запитів на видалення необхідно у схемі даних проаналізувати встановлення прапорця *каскадного видалення пов'язаних записів* тих зв'язків, відносно яких таблиці, з яких безпосередньо чи опосередковано вилучаються дані, знаходяться на стороні 1. Адже видалення записів з таблиці при встановленому цьому

прапорці автоматично призведе до видалення пов'язаних записів з таблиць на стороні ∞ , а при зняттю – не вилучить ті записи на стороні 1, для яких є пов'язані записи на стороні ∞ .

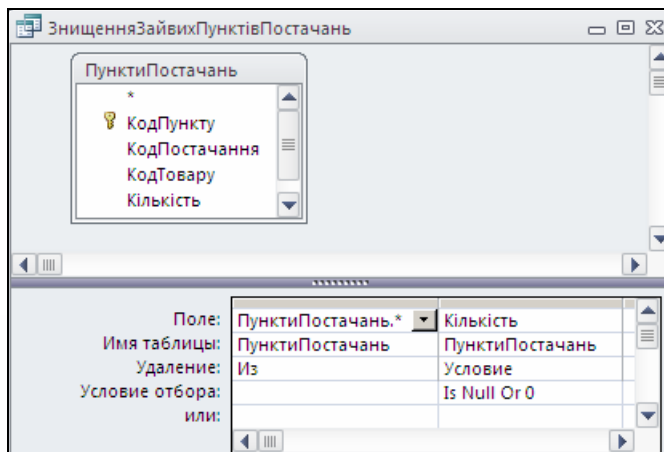


Рис. 63. Структура запити ЗнищенняЗайвихПунктівПостачань в режимі конструктора

Синтаксис запитів на видалення в режимі SQL такий (причому * без назви таблиці можна вживати лише у випадку, коли джерело містить єдину таблицю):

```
DELETE * | <ім'я таблиці1>.* [, <ім'я таблиці2>.* ...]
FROM <джерело даних>
[WHERE <умови відбору записів>];
```

Наприклад, запит з рис. 63 в режимі SQL виглядає так:

```
DELETE ПунктиПостачань.*, ПунктиПостачань.Кількість
FROM ПунктиПостачань
WHERE ПунктиПостачань.Кількість Is Null Or
ПунктиПостачань.Кількість=0;
```

На завершення теоретичних відомостей нагадаємо, що для прискорення виконання найуживаніших запитів на корегування (як і всіх інших запитів) доцільно створювати індекси в базових таблицях за основними полями умов відбору.

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 объекта*.
3. Перейдіть в цій області до розділу *Запросы*.

Розробка запитів на створення таблиць.

Використання будівничого виразів при конструюванні запитів

4. Створіть запит *СтворенняТиповоїПлатіжноїВідомостіВідділу* для генерації таблиці *ТиповаПлатіжнаВідомістьВідділу*, що має містити алфавітний список працюючих співробітників вказаного відділу з зазначенням відповідних окладів, утримань (прибуткового

податку, профспілкового внеску, додаткового податку) та сум до видачі. Для цього:

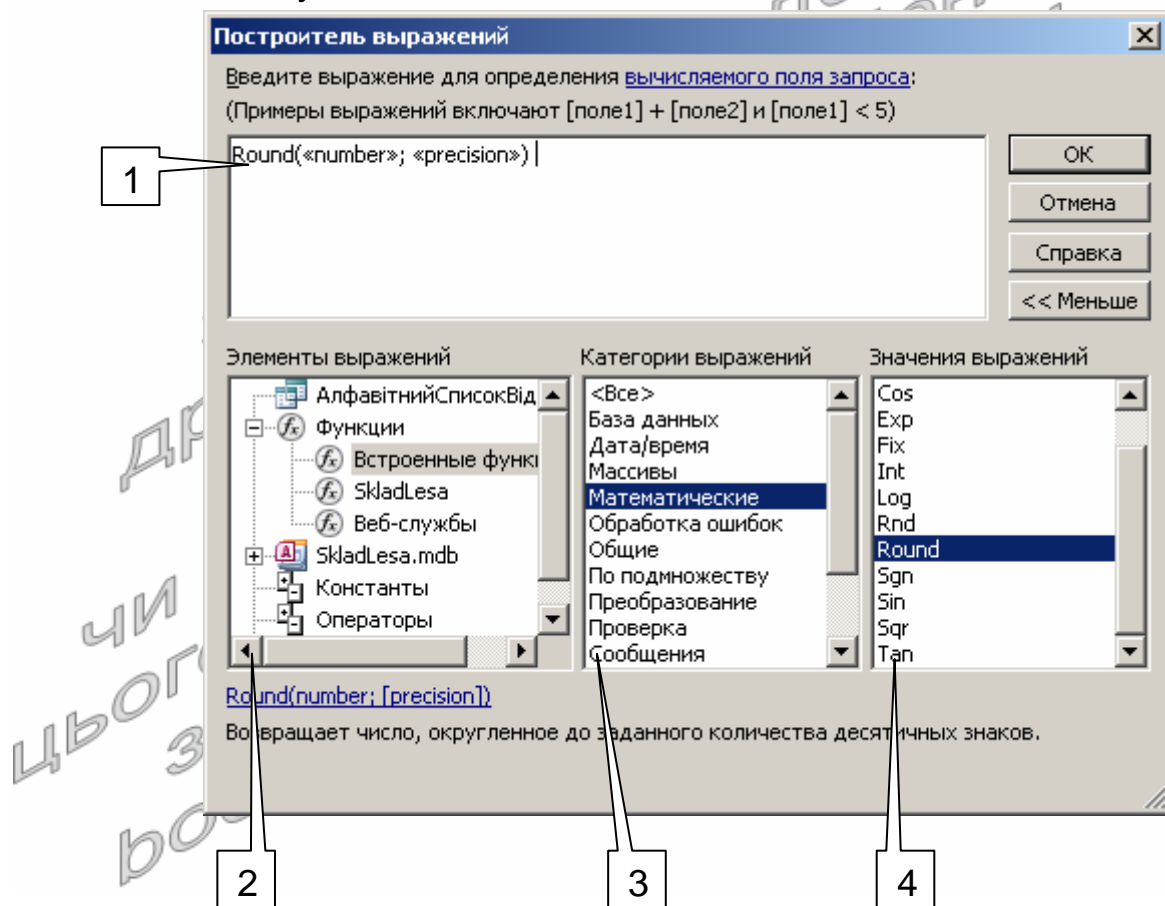
- 4.1. Розпочніть створення запиту в режимі конструктора та сформуєте його джерело даних з таблиць *Відділи* та *Співробітники*;
- 4.2. Перетворіть створюваний запит на вибірку в запит на створення таблиці, натискаючи на вкладці *Конструктор* стрічки меню у групі *Тип запроса* кнопку *Создание таблицы* або обираючи в контекстному меню вільної області джерела даних у підменю *Тип запроса* пункт *Создание таблицы*. У вікні *Создание таблицы*, що з'явиться на екрані, вкажіть назву створюваної таблиці – *ТиповаПлатіжнаВідомістьВідділу* та натисніть кнопку *ОК*;
- 4.3. В перший та другий стовпці бланку запиту перетягніть відповідно поля *ПІБ* та *Оклад* з таблиці *Співробітники*;
- 4.4. Для першого стовпця встановіть ознаку сортування за зростанням;
- 4.5. Припустимо, що прибутковий податок вираховується за формулою

$$P = \begin{cases} 0, & H \leq 17 \\ 0,1 \times (H - 17), & 17 < H \leq 85 \\ 6,80 + 0,15 \times (H - 85), & 85 < H \leq 170 \\ 19,55 + 0,2 \times (H - 170), & 170 < H \leq 1020 \\ 189,55 + 0,3 \times (H - 1020), & H > 1020 \end{cases}$$

де P – прибутковий податок, H – нарахована сума. Тоді для автоматичного підрахунку прибуткового податку відносно окладу введіть в рядок *Поле* третього стовпця бланку запиту вираз *Прибутковий податок*: $Round(If(Оклад \leq 17; 0; If(Оклад \leq 85; 0,1 * (Оклад - 17); If(Оклад \leq 170; 6,8 + 0,15 * (Оклад - 85); If(Оклад \leq 1020; 19,55 + 0,2 * (Оклад - 170); 189,55 + 0,3 * (Оклад - 1020))))); 2)$ за допомогою будівничого виразів, виконуючи наступні дії:

- 4.5.1. В контекстному меню рядка *Поле* третього стовпця бланку запиту оберіть пункт *Построить* для виклику вікна будівничого виразів (рис. 64. Надалі в цьому підпункті номер поля вказує на номер виноска цього рисунка);
- 4.5.2. В полі для редагування виразу введіть заголовок стовпця *Прибутковий податок*.;
- 4.5.3. Для введення у вираз функції *Round* в полі 2 відкрийте папку *Функции*, а в ній – папку *Встроенные функции*; в полі 3 оберіть категорію функцій *Математические* та серед математичних функцій двічі натисніть лівою кнопкою мишки у полі 4 на функції *Round*. Обрана функція вставиться у полі 1 в положення курсору (див. рис. 64);

- 4.5.4. Для введення вкладеної функції виділіть у полі 1 перший аргумент «*Выражение*» вставленої функції *Round* та вставте замість нього функцію *If* з категорії *Управление*. Знищіть елемент «*Выражение*» після вставленої функції;
- 4.5.5. Назву поля *Оклад* введіть безпосередньо з клавіатури або вставте з полів нижньої частини будівничого виразів, відмітивши в полі 2 таблицю *Склад* ► *Таблицы* ► *Співробітники* та обравши з неї в 3 поле *Оклад*;
- 4.5.6. Самостійно завершіть формування виразу, вводячи оператори безпосередньо з клавіатури чи за допомогою полів 2-4. Не забувайте при цьому знищувати зайві елементи «*Выражение*» в 1;
- 4.5.7. Для запису сформованого виразу у бланк запити натисніть кнопку *OK*;



- 1 – поле для редагування тексту виразу;
- 2 – структура об'єктів, констант, функцій, операторів та загальних виразів;
- 3 – категорії обраного елемента з (2);
- 4 – значення обраного елемента з двох попередніх полів.

Рис. 64. Структура будівничого виразів

- 4.6. В рядок *Поле* четвертого стовпця введіть вираз для обчислення профспілкового внеску (1 % з нарахувань співробітника);

4.7. В рядок *Поле* п'ятого стовпця бланку запиту введіть вираз для обчислення додаткового податку, що розраховується за формулою $V = \begin{cases} 0.01 \times H, H \leq 150 \\ 0.02 \times H, H > 150 \end{cases}$, де V – внесок до пенсійного фонду,

H – нарахована сума;

4.8. В рядок *Поле* шостого стовпця самостійно введіть вираз для обчислення сум до видачі, як різниці між відповідними нарахованими (поле *Оклад*) і утриманими сумами;

4.9. Для формування типової платіжної відомості лише для співробітників окремого відділу в сьомий стовпець бланку запиту перетягніть поле *НазваВідділу* з таблиці *Відділи*, приховайте для нього вивід на екран в режимі таблиці та вкажіть умову відбору [*Введіть назву відділу*];

4.10. Перетягніть у восьмий та дев'ятий стовпці відповідно поля *ДатаВлаштування* та *ДатаЗвільнення* з таблиці *Співробітники*. Приховайте їх вивід на екран в режимі таблиці та вкажіть коректні умови відбору для аналізу даних лише працюючих співробітників;

4.11. У вікні властивостей числових полів запиту, що виводяться на екран, вкажіть потрібний формат поля та число десяткових знаків;

4.12. Збережіть запит під назвою *СтворенняТипової-ПлатіжноїВідомостіВідділу* та завантажте його на виконання. Після виконання запиту перегляньте вміст створеної таблиці *ТиповаПлатіжнаВідомістьВідділу*. Перегляньте структуру створеної таблиці в режимі конструктора.

5. Самостійно створіть запити:

5.1. *СтворенняПлатіжноїВідомостіПреміїЗаСтаж* – для генерації таблиці *ПлатіжнаВідомістьПреміїЗаСтаж* з алфавітним списком працюючих співробітників та відповідними сумами нарахованої премії за стаж (за кожен рік стажу – п'ять гривень);

5.2. *СтворенняПлатіжноїВідомостіПреміїЗаЗамовленняПеріоду* – для генерації таблиці *ПлатіжнаВідомістьПремії-ЗаЗамовленняПеріоду* з алфавітним списком працюючих співробітників та відповідними сумами нарахованої премії за оформлені замовлення протягом вказаного періоду (1% від загальної суми оформлених замовлень).

Створення запитів на оновлення даних таблиць

6. Доповніть таблицю *ПунктиЗамовлень* цілочисельним полем *ПроцентЗнижки*, встановивши для нього значення по замовчуванню рівним 0. Для сформованих раніше записів цієї таблиці автоматично внесіть в створене поле значення 0 при відсутності в ньому будь-яких значень. Для цього створіть запит

Ініціалізація *ПроцентуЗнижкиВПунктахЗамовлень*, виконуючи наступні дії:

- 6.1. Розпочніть створення запиту в режимі конструктора та сформуєте його джерело даних з таблиці *ПунктиЗамовлень*;
 - 6.2. Перетворіть запит на вибірку в запит на оновлення даних, натискаючи на вкладці *Конструктор* стрічки меню у групі *Тип запроса* кнопку *Обновление* або обираючи в контекстному меню вільної області джерела даних у підменю *Тип запроса* пункт *Обновление*;
 - 6.3. В перший стовпець бланку запиту перетягніть поле *ПроцентЗнижки* з таблиці *ПунктиЗамовлень*;
 - 6.4. Для редагування під час виконання запиту лише записів з відсутнім значенням в полі *ПроцентЗнижки* вкажіть для створеного першого стовпця бланку запиту умову відбору *Is Null*;
 - 6.5. З метою автоматичного внесення в поля *ПроцентЗнижки* відібраних записів значення 0 введіть в рядку *Обновление* першого стовпця запиту значення 0;
 - 6.6. Збережіть створений запит та переконайтеся в його дієздатності. Перегляньте та обгрунтуйте структуру запиту в режимі SQL.
7. Створіть запит *НарахуванняПроцентуЗнижкиВПунктахЗамовлень-ЗаПеріод* для автоматичного заповнення поля *ПроцентЗнижки* в записах таблиці *ПунктиЗамовлень*, що відповідають оформленим замовленням протягом вказаного періоду, згідно наступної шкали:
- $$P = \begin{cases} 0, H \leq 10 \\ 3, 10 < H \leq 100 \\ 5, H > 100 \end{cases}$$
- де P – процент знижки, H – сума пункту замовлення. Для цього:
- 7.1. Розпочніть створення запиту в режимі конструктора, сформуєте його джерело даних з таблиць *ЗаголовкиЗамовлень* та *ПунктиЗамовлень* і перетворіть його в запит на оновлення даних;
 - 7.2. В перший стовпець бланку запиту перетягніть поле *ПроцентЗнижки* з таблиці *ПунктиЗамовлень*;
 - 7.3. В рядку *Обновление* першого стовпця, використовуючи будівничого, введіть вираз для оновлення проценту знижки згідно шкали: *IIf([Кількість]*[Ціна]<=10; 0; IIf((([Кількість]*[Ціна]<= 100; 3; 5))*);
 - 7.4. Для аналізу пунктів, що стосуються лише оформлених замовлень протягом вказаного періоду, перетягніть в другий стовпець бланку запиту поле *ДатаЗамовлення* з таблиці *ЗаголовкиЗамовлень* та вкажіть для нього умову відбору

Between [Введіть початкову дату] and [Введіть кінцеву дату];

7.5. Збережіть створений запит та переконайтеся в його дієздатності.

8. Самостійно створіть запити:

8.1. *НарахуванняСвятковогоПроцентуЗнижкиВПунктахЗамовлень* – для автоматичного встановлення п'ятипроцентної знижки в пунктах замовлень, що оформлені 1 січня, 7 січня, 8 березня, 9 травня, 24 серпня будь-якого року;

8.2. *ПідвищенняОкладівНаДесятьПроцентів* – для автоматично підвищення окладів всіх працюючих співробітників на 10%.

Створення запитів на додавання до таблиць нових записів

9. Створіть запит *СтворенняЗаголовокЗамовленняЗаЗразком* для створення в таблиці *ЗаголовкиЗамовлень* нового запису з даними обраного запису та активною датою замовлення (тобто для створення аналогічного заголовка замовлення, але від поточної дати). Для цього:

9.1. Розпочніть створення запиту в режимі конструктора та сформуєте його джерело даних з таблиці *ЗаголовкиЗамовлень*;

9.2. Перетворіть створюваний запит на вибірку в запит на додавання записів до таблиці, натискаючи на вкладці *Конструктор* стрічки меню у групі *Тип запроса* кнопку *Добавление* або обираючи в контекстному меню вільної області джерела даних у підменю *Тип запроса* пункт *Добавление*. Після цього у вікні, що з'явиться на екрані, вкажіть назву таблиці для доповнення записами – *ЗаголовкиЗамовлень* та натисніть кнопку *ОК*;

9.3. Для копіювання даних співробітника та клієнта з обраного замовлення в новий запис перетягніть в перший та другий стовпці бланку запиту відповідно поля *КодСпівробітника* та *КодКлієнта* з таблиці *ЗаголовкиЗамовлень*;

9.4. З метою автоматичного встановлення в полі *ДатаЗамовлення* активної дати при створенні нового запису введіть для третього стовпця в рядку *Поле* вираз *Date()*, а в рядку *Добавление* оберіть зі списку поле *ДатаЗамовлення*;

9.5. Для доповнення таблиці даними лише обраного замовлення перетягніть в четвертий стовпець бланку запиту поле *КодЗамовлення*, **очистіть для нього значення в рядку *Добавление*** та вкажіть умову відбору *[Введіть код замовлення]*;

9.6. Збережіть створений запит та переконайтеся в його дієздатності. Перегляньте та обґрунтуйте структуру запиту в режимі SQL.

10. Самостійно створіть запит *СтворенняЗаголовкаПостачання-ЗаЗразком* для створення в таблиці *ЗаголовкиПостачань* нового запису з даними обраного запису та активною датою оформлення постачання.

Створення запитів на видалення записів таблиць

11. Створіть запит *ЗнищенняЗайвихТоварів* для автоматичного знищення у таблиці *Товари* записів з даними товарів, що не постачалися і не замовлялися. Для цього:

11.1. В схемі даних відкоригуйте зв'язки між таблицею *Товари* та таблицями *ПунктиПостачань* і *ПунктиЗамовлень* так, щоб **заборонити** каскадне знищення даних для уникнення вилучення товарів, дані яких використовуються в постачаннях чи замовленнях. Для цього в контекстному меню відповідних зв'язків по чергово оберіть пункт *Изменить связь* і у вікні, що з'явиться на екрані, зніміть чи переконайтеся, що вже знятий прапорець *Каскадное удаление связанных записей*;

11.2. Розпочніть створення запиту в режимі конструктора та сформуєте його джерело даних з таблиці *Товари*;

11.3. Перетворіть створюваний запит на вибірку в запит на знищення записів, натискаючи на вкладці *Конструктор* стрічки меню у групі *Тип запроса* кнопку *Удаление* або обираючи в контекстному меню вільної області джерела даних у підменю *Тип запроса* пункт *Удаление*;

11.4. Для знищення записів саме з таблиці *Товари* перетягніть в перший стовпець бланку запиту символ * (позначення всіх полів) з її праобразу у верхній частині вікна;

11.5. Збережіть створений запит та переконайтеся в його дієздатності. Перегляньте та обгрунтуйте структуру запиту в режимі SQL.

12. Самостійно створіть запити на знищення записів *ЗнищенняЗайвихПунктівЗамовлень* (для знищення тих записів з таблиці *ПунктиЗамовлень*, в яких кількість не вказана або рівна нулю), *ЗнищенняЗайвихПунктівПостачань* (для знищення тих записів з таблиці *ПунктиПостачань*, в яких кількість не вказана або рівна нулю), *ЗнищенняЗайвихПостачальників*, *ЗнищенняЗайвихКлієнтів* (заборонивши перед цим відповідні каскадні знищення в схемі даних).

Керування запитом на корегування з головної кнопки форми

13. Для впорядкування всіх запитів на корегування даних забезпечте їх виклик безпосередньо з головної кнопки форми. Для цього:

13.1. В режимі конструктора доповніть набір вкладок форми *Головна* новою вкладкою *Корегування даних*. За допомогою пункту контекстного меню набору вкладок

Последовательность вкладок розташуйте її перед вкладкою *Довідники*;

- 13.2. Для завантаження кожного запиту на корегування створіть на цій вкладці відповідні кнопки, обираючи у вікні *Создание кнопок* майстра створення кнопок категорію *Разное* та дію *Выполнить запрос*. Для кожного з запитів, назви яких не виводяться у вікні майстра (наприклад, для запитів на вилучення), створіть кнопку, яка викликає інший запит на корегування, а потім змініть назву завантажуваного запиту на потрібну в макросі *События* ► *Нажатие кнопки* вікна властивостей створеної кнопки;
- 13.3. Згрупуйте кнопки за типом операцій корегування, кожну групу кнопок помістіть в окремий прямокутник;
- 13.4. Збережіть зміни структури головної кнопкової форми та переконайтеся в її дієздатності.

Завершальний етап заняття. Повторення вивченого матеріалу

14. Стисніть створену БД;
15. Завершіть роботу Access;
16. Віднайдіть створену БД та заархівуйте її вміст;
17. Збережіть створений архів на власному флеш-носії в двох екземплярах.
18. Доповніть власну БД запитамі на корегування, аналогічними до створених на лабораторній роботі.

Контрольні запитання

1. У чому полягає основна відмінність між запитамі на вибірку і запитамі на корегування?
2. У чому полягає відмінність між дією запитів на корегування та самостійним корегуванням даних?
3. Як двома способами перетворити запит на вибірку в запит на корегування?
4. Які різновиди запитів на вибірку і запитів на корегування використовуються в Access?
5. Який синтаксис запису мають різні варіанти запитів на корегування в режимі SQL?
6. Які поля містить вікно будівничого виразів? Яке їх функціональне призначення?
7. Яке обмеження накладається на назви таблиць, що створюються запитамі?
8. Поля якого типу не змінюються запитамі на оновлення? Чому?
9. Які поля мають обов'язково заповнюватися запитамі на додавання записів? Як ще можливо реалізувати таке заповнення?
10. Як використовуються параметри зв'язків між таблицями при виконання запитів на вилучення?
11. Як здійснюється керування запитамі на корегування у розроблених додатках?