

**К И Ї В С Ь К И Й Н А Ц І О Н А Л Ь Н И Й
У Н І В Е Р С И Т Е Т
І М Е Н І Т А Р А С А Ш Е В Ч Е Н К А**

В.В.Попов, Є.С.Вакал, О.В.Обвінцев, В.В.Личман

ОБЛІК ІНФОРМАЦІЇ З ВИКОРИСТАННЯМ MS ACCESS

МЕТОДИЧНИЙ ПОСІБНИК

для студентів механіко-математичного факультету

ОБЛІК ІНФОРМАЦІЇ З ВИКОРИСТАННЯМ MS ACCESS. Методичний посібник для студентів механіко-математичного факультету/ Упорядн. В.В.Попов, Є.С.Вакал, О.В.Обвінцев, В.В.Личман. 2001. – 36 с.

Рецензенти

Затверджено Радою

Зміст

1	Основні поняття баз даних та систем управління базами даних	5
2	Архітектура MS Access	7
3	Побудова бази даних	7
3.1	Таблиці	8
3.1.1	Проектування структури таблиць	8
3.1.2	Індекси та ключі	9
3.1.3	Побудова та редагування структури таблиць	10
3.1.4	Зв'язування таблиць	11
3.2	Форми	11
3.2.1	Види форм	12
3.2.2	Проектування та побудова форм	12
3.2.3	Редагування форм	16
3.2.4	Пошук та сортування даних	17
3.3	Звіти	17
3.3.1	Механізми створення звітів	17
3.3.2	Редагування звіту	19
3.3.3	Виконання обчислень та підведення підсумків	21
4	Робота з даними	22
4.1	Можливості та типи запитів	22
4.2	Створення запитів у середовищі MS Access	23
4.3	SQL – запити	23
5	Побудова програмного забезпечення	25
5.1	Макроси	25
5.2	Кнопкові форми	29
6	Теми учбових розробок для самостійної роботи студентів	31
7	Приклади таблиць та їх полів для учбової розробки баз даних	31
8	Список літератури	35

Вступ

Цей посібник присвячений MS Access - одній з сучасних реляційних систем управління базами даних (СУБД), яку можуть використовувати спеціалісти різних галузей промисловості, науки, причому її використання не потребує спеціальних знань з програмування. Саме з цих позицій висвітлюються можливості MS Access для забезпечення комп'ютеризації кадрової, виробничої діяльності, процесів обліку у фінансовій сфері тощо. Посібник знайомить читача з основними поняттями баз даних та систем управління базами даних, конструктивно “веде” його по всіх ділянках процесу створення та використання баз даних на основі MS Access. Будь-яка СУБД, в тому числі MS Access, містить велику інформацію довідкового характеру, повне викладання якої не має сенсу. Але і запропонована частка СУБД MS Access може бути опанована лише під час постійних практичних і (головне!) самостійних робіт.

1 Основні поняття баз даних та систем управління базами даних

Сукупність типових даних деякої предметної області можна структурно уявити у вигляді множини пов'язаних між собою окремих інформаційних частин. До будь-якої такої сукупності даних у загальному випадку можна використовувати термін *база даних* [2]. Наприклад, базою даних можна вважати сукупність інформації про кадровий стан підприємства, розклад занять у навчальному закладі, родинні зв'язки, бібліотечний фонд тощо. Якщо поняття бази даних наблизити до інформаційних систем, які розміщуються на комп'ютерних носіях даних, то під відповідним терміном можна розуміти набір записів та файлів, які організовані спеціальним чином. Спеціальне програмне забезпечення, необхідне для використання та модифікації баз даних користувачем, називається *системою управління базами даних*. Основні функції СУБД:

- забезпечення користувача інструментарієм, що дозволяє оперувати даними в абстрактних термінах, які не пов'язані із способом збереження даних в ЕОМ;
- забезпечення секретності і пріоритетності доступу до даних;
- захист цілісності даних;
- синхронізація доступу до даних;
- захист даних від аварій та відновлення даних.

Відповідні функції, що містяться в арсеналі СУБД, можна розподілити за типом на такі групи:

- Визначення даних. Функції СУБД цієї групи визначають структуру даних та їх тип, зв'язки між даними, елементи підстановок для значень тощо.
- Обробка даних. Ці функції СУБД зосереджені на обробці даних різними методами, причому залучаються такі дії, як фільтрація та сортування.
- Керування даними. Функції СУБД групи керування даними дозволяють розподілити дані за користувачами, надаючи їм право доступу до них, причому з наданням можливості або запереченням можливості коригування даних.

Кожну інформаційну область за своєю структурою можна віднести до деякого типу. Найпоширені типи – *ієрархічні*, *мережні*, *реляційні*. Так, якщо інформаційна структура даних пов'язана із ієрархією її частин, то таку структуру називають *ієрархічною*. До ієрархічних інформаційних областей можна віднести дані, що пов'язані з родинними зв'язками, структур міністерств тощо. Графічно дані з ієрархічною структурою уявляють у вигляді так званого *дерева* або сукупності дерев - *лісу*. Якщо інформаційна структура даних пов'язана із деякою мережою, то відповідну інформаційну область за своєю структурою називають *мережною*. Прикладом мережних інформаційних областей є дані про стан електричних мереж, транспортних шляхів тощо. Графічно дані з мережною структурою уявляють у вигляді так званого *графа*. Нарешті, якщо дані уявити у графічному вигляді як пов'язані між собою двовимірні таблиці, то одержану сукупність даних можна назвати *реляційною* (від англійського слова *relation* - *відношення*). Відповідно з типом даних називають і системи управління базами даних. Найбільш поширеними сьогодні є реляційні СУБД, прикладом яких є ORACLE, MS SQL Server, MS FoxPro [3], MS Access [1] тощо.

Переваги реляційних СУБД - це легкість використання та ефективність реалізації. У даному навчальному посібнику розглядаються СУБД реляційного типу, а саме MS Access. Треба відмітити, що будь-яку інформаційну структуру можна визначити через модель іншої структури. Але саме СУБД реляційного типу дозволяє це зробити з найменшими втратами.

У подальшому рядки двовимірних таблиць будемо називати *записами*, стовпчики – *полями*. У термінах реляційних СУБД подібні таблиці називаються *відношеннями*, їх записи – *кортежі* відношень, поля – *атрибути* відношень. Тепер зрозуміло, чому використовується термін *реляційна СУБД (relation - відношення)*. Записи відрізняються своїм номером, а поля – своїм ім'ям. Основні умови щодо змісту таблиць такі:

- однакові записи забороняються;
- всі записи повинні мати однакову кількість полів;
- значення полів атомарні, тобто таблиця не може мати своїми компонентами інші таблиці.

Елементами даних MS Access є *сталі*, *змінні пам'яті* та *поля таблиць*. Елементи даних та функції над ними складають *вираз*, які надалі будемо позначати словом Expr, додаючи, якщо це потрібно, ще один з символів C, M, N, D, L – початкові букви англійських назв типів даних (наприклад, ExprN). Типи даних розподіляються на базові і такі, що визначені користувачем. Деякі базові типи даних наведені у Табл. 1.

Табл. 1 - Базові типи даних MS Access

Тип даних	Позначення	Назва в MS Access
Рядок (<i>Character</i>)	C	текст
Текст (<i>Memo</i>)	M	текст Мемо
Числовий (<i>Numeric</i>)	N	числовий
Дата (<i>Date</i>)	D	дата/час
Логічний (<i>Logical</i>)	L	логічний

Приклади сталих: “010201” – рядок (ExprC), 010201 – число (ExprN), 01.02.01 – дата (ExprD). Типи полів таблиць визначаються під час визначення таблиці або модифікації її структури. Треба пам'ятати, що тип даних *рядок* – це множина послідовностей символів. Кількість символів такої послідовності називається довжиною рядка і не може перевищувати 255. Тип даних рядок використовується для позначення кодів, назв, прізвищ тощо. Існує операція конкатенації даних типу рядок. Позначається ця операція знаком +. Наприклад, значення змінних A і B є відповідно ‘Micro’ і ‘soft’, то значенням виразу A+B, буде рядок ‘Microsoft’. Тип даних *текст* (Мемо) використовується для даних, значення яких – символи, а їх кількість може сягати за 255. *Числовий* тип даних використовується для даних, що характеризують кількісну їх властивість, наприклад, вага, стаж, заробітна плата тощо. Тип даних *дата* очевидний, він використовується для обробки дат. Логічний тип даних використовується в виразах, що характеризують ситуацію, причому значення таких виразів або *істина* або *хибність*. Наприклад, вираз 10>100 логічний, а значення такого виразу – *хибність*. Для даних

логічного типу існують декілька операцій: OR - логічне додавання, AND – логічне множення, NOT – логічне заперечення. Наприклад, значення виразу NOT (10>100) – *істина*.

2 Архітектура MS Access

СУБД MS Access належить до так званих об'єктно-орієнтованих систем. Під об'єктами у MS Access розуміють все, що має ім'я. Множину об'єктів MS Access складають *таблиці, запити, форми, звіти, макроси та модулі*. Ці об'єкти визначаються користувачем, а їх сукупність складає *базу даних*. Всі об'єкти взаємопов'язані, причому таблиці знаходяться в основі всіх зв'язків. На цій основі, як правило, будуються всі інші об'єкти (але взагалі кажучи це не обов'язково). Це пояснюється перш за все тим, що саме в таблицях зберігаються дані, які вже надалі обробляються формами, запитами тощо. Таким чином *таблиці* призначені для збереження даних. Кожна таблиця містить інформацію певного типу, наприклад, кадровий склад працівників підприємства. Таблиці можуть бути пов'язані між собою, що дає змогу представити інформацію як багатовимірну. *Запит* призначений для виборки необхідних даних з таблиць. За допомогою запитів можна також модифікувати дані або створювати інші таблиці. *Форми* призначаються для перегляду, введення і редагування даних таблиць а також надають користувачу засоби керування програмами, які використовують базу даних. Форми, як правило, використовуються для того, щоб забезпечити необхідний зовнішній вигляд документа, дані якого вводяться у таблицю. *Звіти* призначаються для перегляду та друку результуючої інформації. Звіти можуть бути включені до документів інших програм. *Макроси та модулі* призначені для обробки *подій*. Терміном *подія* у MS Access називають зміну стану об'єкта. Так, подія – це відкриття того чи іншого об'єкта, його закриття, зміна стану самого об'єкта. Наприклад, для таблиць це введення запису, видалення запису тощо. Для визначення макросів та модулів необхідні знання з теорії програмування та вміння програмувати в системі програмування Visual Basic for Applications (VBA), яка вбудована у MS Access. Всі наведені вище об'єкти можуть бути визначені у MS Access як вручну, так і за допомогою так званих *майстрів*, які допомагають користувачеві за декілька кроків діалогу створити необхідне. Майстри – це один з найпотужніших засобів MS Access, якими слід користуватись на перших кроках опанування даної СУБД, або в інших “нештатних” ситуаціях.

Примітка: у зв'язку з наявністю російськомовної і відсутністю на сьогодні україномовної кирилізованої версії MS Access у даному посібнику використовуються російськомовні назви елементів інтерфесу (тобто, назви кнопок, дій меню тощо).

3 Побудова бази даних

Після запуску MS Access користувач має змогу:

- побудувати базу даних самостійно;
- побудувати базу даних за допомогою майстра;
- відкрити існуючу базу даних.

При самостійному способі побудови бази даних і першому відкритті вікна бази даних MS Access (більш докладно процес побудови таблиць, форм, запитів, звітів у середовищі MS Access наведено у [4]) активізує вкладку “Таблиця” з трьома кнопками: “Открыть”, “Конструктор”, “Создать”. Саме остання кнопка буде спочатку активною,

вона призначена для побудови нової таблиці. Таким чином, за угодою MS Access дає змогу розпочати процес побудови бази даних з визначення таблиць.

3.1 Таблиці

3.1.1 Проектування структури таблиць

Процес визначення таблиці починається на стадії її проектування. Саме на цій стадії розробник бази даних має зробити аналіз інформації, яку необхідно обробляти. Для визначеності розглянемо інформацію про кадри підприємства, яка складається з особового складу, відомостей про переміщення тощо. Цю інформацію у реляційних СУБД необхідно представити сукупністю двовимірних таблиць. Нехай, для прикладу, база даних, що створюється, називається *Кадри* та складається з двох таблиць: *Особи* та *Переміщення*. Визначимо таблицю *Особи*. Визначимо, що в цій таблиці розміщується деяка інформація з першої сторінки відомої картки особи, а саме:

- 1) табельний номер (номер картки особи)
- 2) прізвище
- 3) ім'я
- 4) по-батькові
- 5) дата народження
- 6) стать
- 7) освіта

Кожний описаний вище пункт картки особи – це майбутнє поле таблиці! Визначимо тепер назву, тип даних, розмір та необхідну множину значень цих полів та зведемо все у Табл. 2:

Табл. 2 - Поля таблиці осіб

Пункт картки особи	Назва поля	Тип даних	Розмір, байт	Множина значень
1	ТабНомер	C	4	
2	Прізвище	C	16	
3	Ім'я	C	12	
4	ПоБатькові	C	16	
5	ДатаНародження	D		
6	Стать	C	1	'Ч';'Ж'
7	Освіта	C	16	'середня';'вища'; 'середня-технічна'

Як бачимо, деякі клітинки цієї таблиці не заповнені. Так, розмір поля з типом D визначається автоматично, а можливі значення даних пунктів 1)-5) картки особи тут

очевидно не потрібні. На цьому процес проектування таблиці Особи призупиняється, щоб визначитись з так званими первинними ключами.

3.1.2 Індеси та ключі

Для зв'язування таблиць у реляційних СУБД використовують *ключі*. Ключем називають сукупність полів таблиці, значення яких ідентифікують її записи. *Первинним ключем* називають такий ключ, який однозначно ідентифікує записи таблиці. Іншими словами, значення первинного ключа є унікальним для даної таблиці. Часто первинний ключ складається з одного поля. Наприклад, первинним ключем таблиці Особи має бути поле *ТабНомер*, і аж ніяк не *Прізвище*, оскільки записів з однаковим табельним номером не повинно бути. Водночас можна очікувати на записи, де є однакові прізвища.

Впорядкування записів таблиці за значенням ключа називають *індексацією*, а отриманий образ таблиці – *індексом*. Взагалі фізично таблиці та індекси можуть зберігатися у різних файлах (як, наприклад, у СУБД MS FoxPro). У MS Access всі об'єкти бази даних (у тому числі і індекси) зберігаються в одному файлі.

Для уникнення непотрібних повторювань даних, яке може бути причиною помилок при введенні даних, і нераціонального використання дискового простору комп'ютера проектування таблиць завершується їх *нормалізацією* – процесом зменшення надлишкової інформації [3]. Так, згідно з принципами нормалізації таблицю Особи слід редагувати, замінивши в ній поле *Освіта* на *КодОсвіти* (тип даних С, розмір 1), а крім цього ввести ще таблицю Освіта з двома полями *КодОсвіти* та *Освіта*, які співпадають з відповідними полями у модифікованій та немодифікованій таблиці Особи. Таблицю Освіта будемо ще називати *таблиця-довідник Освіта*.

Взагалі таблиці створюються так, щоб задовольнити трьома *нормальними формами*:

Перша нормальна форма

1. Таблиця не має записів, що повторюються.
2. У таблиці відсутні групи полів, що повторюються.
3. Рядки повинні бути не впорядковані.
4. Стовпчики повинні бути не впорядковані.

Зрозуміло, щоб задовольнити умові 1, кожна таблиця повинна мати первинний ключ.

Друга нормальна форма

1. Таблиця задовольняє умовам першої нормальної форми.
2. Будь-яке неіндексне поле однозначно ідентифікується повним набором індексних полів.

Очевидно, що таблиці, які мають один первинний ключ автоматично задовольняють умові 2.

Третя нормальна форма

1. Таблиця задовольняє умовам другої нормальної форми.
2. Жодне з неіндексних полів таблиці не ідентифікується за допомогою іншого неіндексного поля.

Процес нормалізації таблиць, як правило, супроводжується створенням окремих додаткових таблиць, які пов'язані між собою з використанням введених первинних ключів. Повертаючись до таблиці *Особи*, відмітимо, що її первинним ключем буде поле *ТабНомер*. Очевидно первинним ключем таблиці *Освіта* буде поле *КодОсвіти*. Тепер настає черга побудови таблиць:

Табл. 3 - Таблиця *Особи*

Номер поля	Назва поля	Тип даних	Розмір, байт	Множина значень
↔ 1	ТабНомер	С	4	
2	Прізвище	С	16	
3	Ім'я	С	12	
4	ПоБатькові	С	16	
5	ДатаНародження	D		
6	Стать	С	1	'Ч';'Ж'
7	КодОсвіти	С	1	Значення поля КодОсвіти з таблиці <i>Освіта</i>

Табл. 4 - Таблиця *Освіта*

Номер поля	Назва поля	Тип даних	Розмір, байт	Множина значень
↔ 1	КодОсвіти	С	1	
2	Освіта	С	16	

Тут знак ↔ означає первинний ключ.

3.1.3 Побудова та редагування структури таблиць

Для визначення таблиці треба у вікні бази даних (активізація вікна бази даних - F11) натиснути кнопку “Создать” та вибрати той чи інший режим, наприклад, режим *конструктор*. В цьому режимі кожний рядок в верхній частині вікна визначає поле таблиці. Три стовпчики необхідні для визначення поточного поля: назва, тип даних, опис. Тут опис – для необов'язкових пояснень. Перед рядком є можливість поставити ознаку первинного ключа за допомогою значка з малюнком у вигляді золотого ключика, що розташований на панелі інструментів. Розмір поля необхідно поставити у частині вікна з назвою “Свойства поля”. Щоб забезпечити підстановку значень (в нашому прикладі значення полів *Стать* та *КодОсвіти* таблиці *Особи*) необхідно в стовпчику “Тип данных” знову звернутися до вибору можливих типів і вибрати “Мастер подстановок”. Після очевидного діалогу буде забезпечено необхідний вибір значень при формуванні записів таблиці. Слід відмітити, що характер двох підстановок в даному прикладі різний. Для поля *Стать* -це підстановка фіксованих значень “М”

або “Ж”, а для поля *КодОсвіти* - це підстановка даних поля *КодОсвіти* таблиці *Освіта*. Майстер підстановок забезпечує вибір даних серед елементів фіксованих значень або серед значень поля таблиці-довідника, причому для надійності вибору цей процес, як правило, посилюється візуалізацією інших полів таблиці-довідника (в даному випадку при виборі значень поля *КодОсвіти* візуалізується ще й поле *Освіта*). Визначення підстановок за рахунок значень полів таблиць-довідників автоматично веде до так званого зв'язування таблиць з відношенням “один-до-багатьох”.

Якщо первинний ключ користувач не визначив, то MS Access допоможе при необхідності це зробити введенням додаткового поля, значення якого – порядковий номер запису.

Для редагування структури таблиці необхідно натиснути кнопку “Конструктор” вікна база даних, а далі діяти за схемою, що наведена вище.

3.1.4 Зв'язування таблиць

Процес зв'язування таблиць може розпочинатись ще на етапі визначення підстановок (див. п. 3.1.3). Завершує цей процес робота з визначення схеми даних. Ця робота розпочинається з натискання на значок “Схема данных”, що розташований на панелі інструментів. В результаті відкривається вікно з такою ж назвою “Схема данных”, де вже існують графічні образи зв'язаних при підстановці таблиць. Решта таблиць може бути додана при необхідності. Зв'язки між таблицями можна редагувати, видаляти. Для редагування або видалення існуючого зв'язку необхідно правою кнопкою миші клацнути по лінії зв'язку (графічне зображення зв'язку), а далі дотримуватись очевидних дій меню та діалогу. Так, дія “Удалить” означає видалення зв'язку (це можна забезпечити і натисканням клавіатурної клавіші DELETE), а дія “Изменить связь” – редагування зв'язку. Причому в останній дії передбачено діалог, який містить питання щодо цілісності даних. Якщо забезпечити цілісність (вибір значком “галочка”), то отримаємо зв'язок “один-до-багатьох”. Якщо між таблицями попередньо ніякі зв'язки не встановлені, то встановити їх можна за допомогою лівої кнопки миші: необхідно не відпускаючи кнопки миші з'єднати нею відповідні поля. В результаті відкриється вікно “Связи”, де необхідно вибрати “Обеспечение целостности данных”.

3.2 Форми

Форми – осоновний засіб побудови інтерфейсу користувача, що забезпечує найбільш зручний спосіб перегляду та редагування даних а також контроль за ходом виконання прикладної програми. Таким чином, форми будуються для:

- *Виведення та редагування даних.* Це найбільш поширений спосіб використання форм. Вони забезпечують виведення на екран монітору даних у необхідному вигляді. Використання таких форм значно спрощує редагування даних, їх введення та видалення з бази даних. При цьому деякі дані можна зробити доступними лише для перегляду, а деякі – зовсім не демонструвати. Крім цього є можливість в таких формах забезпечити процес обчислення полів в залежності від параметрів, які задає користувач.
- *Керування ходом виконання прикладної програми.* Сучасні прикладні програми, як правило, мають оболонку для надання користувачеві можливості виконувати ті чи інші дії у певній послідовності. Таку оболонку

можна побудувати за допомогою форм, а саме – кнопочових форм. Такі форми містять кнопки, що викликають при їх виборі дію певного макроса або процедури VBA. Ці кнопки називаються командними. Командні кнопки можуть використовуватись і для виклику іншої форми, а також багатьох інших дій, що пов'язані з ходом виконання прикладної програми, а саме, виконувати запити, команди меню, фільтрувати дані, друкувати звіти тощо.

- *Введення даних.* Можна побудувати форми, які призначені лише для введення в базу даних нових значень, які автоматизують виконання програми за певним алгоритмом.
- *Виведення повідомлень.* Такі форми забезпечують інформацією події, що пов'язані з виконанням прикладної програми, наприклад, повідомлення про різноманітні помилки.
- *Друк інформації.* Як правило для друку призначені звіти, але надрукувати інформацію можна і за допомогою форм. Очевидно для введення та виведення даних діють різні параметри, тому в таких формах інструментально підтримується їх подвійна роль.

3.2.1 Види форм


За своїм видом форми поділяються на:

1. *Прості форми* - форми, вигляд яких практично не відрізняється від зображення при звичайному перегляді таблиць.
2. *Багатосторінкові форми* - форми, які використовуються у випадку, коли записи містять велику кількість інформації, що не може розміститись на одному екрані монітора.
3. *Стрічкові форми* - форми, які використовуються у випадку, коли записи містять невелику кількість інформації, наприклад у разі малої кількості полів. На відміну від простої форми, яка демонструє лише один запис, в стрічковій формі записи виводяться одні за одним, як в таблиці.
4. *Підлеглі форми* - форми, які використовуються у випадку, коли демонструються записи двох або більше таблиць, що пов'язані між собою відношенням "один до багатьох".
5. *Спливаючі форми* - форми, які після свого виведення на екран монітору, залишаються на передньому плані; лише з безпосереднім їх закриттям ці форми уступають передній план іншим формам.
6. *Монопольні форми* - форми, що не дають діяти жодним іншим об'єктам, поки самі активні.

3.2.2 Проектування та побудова форм

Проектування форм для таблиць оснований перш за все на первинних документах введення інформації. Розміщення інформації, стиль, фон, які використані у первинному документі, беруться за основу розробки форми. Інформація у формі міститься у так званих *елементах управління*. Наприклад, елементом управління може бути *надпис, поле таблиці, кнопка, перемикачі, прапорці, вимикачі, групи, список, поле із списком, об'єкти OLE*. Елемент управління *надпис* має такі основні властивості як шрифт, його

розмір та колір, колір фону. *Поле* - елемент управління, що використовується найчастіше. В цей елемент виводяться дані з базових таблиць, запитів, а також значення, що обчисленні безпосередньо у самій формі. *Кнопки* найчастіше використовуються для виконання окремих дій, наприклад, для виклику макросу, іншої форми, звіту тощо. *Перемикачі, прапорці, вимикачі* використовуються тоді, коли необхідно вибрати з деякої множини необхідний елемент. У разі, коли для декількох елементів управління відповідні дані розміщуються на значній площині екрану, використовують *групу* з цих елементів управління. При виборі елемента групи відповідна інформація виводиться на екран в підлеглий формі. Якщо необхідно вивести в формі перелік значень, який залишається на час роботи форми завжди відкритим, то використовують елемент управління *список*. Так, список можна використовувати для перегляду таблиць у вікні бази даних, що приводить до підстановки значень полів цих таблиць у поля поточних таблиць. *Поля із списком* схожі на список, за винятком того що для розміщення цього елемента управління на відміну від списку необхідно значно менше місця на екрані монітору, а саме один рядок. *Об'єкти OLE* використовуються для відображення фотографій, електронних таблиць, текстових документів тощо.

Як правило, побудова форми відбувається після її проектування, коли вже побудовані відповідні таблиці. Для цього необхідно у вікні бази даних в закладці *Форми* натиснути кнопку *Создать*, вибрати той чи інший режим побудови форми а в якості джерела даних таблицю (або запит). Якщо вибрати режим *Конструктор*, то MS Access відчиняє відповідне вікно форми. У тому випадку, коли форма нова, на користувача чекає лише порожня область даних з сірою сіткою для зручності подальшого розміщення елементів управління. Крім вікна форми відчиняється, як правило, і панель елементів для розміщення вибраних в цій панелі елементів управління в області даних форми. Якщо панель елементів не відчинена, то відчинити її можна через меню MS Access, або піктограму . Вигляд панелі наведений нижче. Кнопки позначені номерами. Розмір форми можна змінити безпосередньо мишею так, як це звичайно можна зробити у Windows (встановити вказівник миши на край та потягнути) або через властивості форми (клацнути правою кнопкою миши, коли її вказівник знаходиться в заголовку форми). За допомогою панелі елементів в формі розміщуються вказані користувачем елементи. Це відбувається так: спочатку необхідно клацнути мишею необхідну кнопку панелі, а потім вказати мишею місце на формі, де буде розташований відповідний елемент. Розмір елемента можна змінювати під час його розміщення (тягнути мишею), або пізніше звичайним способом зміни розмірів.

Нижче вказано зміст кнопок панелі елементів (Рис. 1):

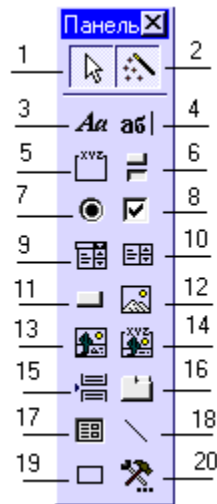


Рис. 1 - Панель інструментів

1. Кнопка вибору елементів в формі. Використовується для вибору одного чи декількох елементів форми та подальшого виконання тих чи інших дій над ними (зміна розміщення, розмірів, видалення, тощо).
2. Кнопка майстра елементів. Ця кнопка може перебувати у двох позиціях в залежності від її вмикання. Якщо кнопку увімкнути, то при розміщенні в формі елементів управління їх властивості допомагають встановити відповідні майстри – процедури, що вбудовані в MS Access.
3. Кнопка надпису. Використовується для побудови елементу управління *надпис* – текст заголовку, пояснення, тощо.
4. Кнопка поля. Використовується для побудови елементу управління *поле*, що відображає значення виразів відповідного типу, наприклад, поля таблиці (запису).
5. Кнопка групи. Використовується для побудови елементу управління *група*, який містить декілька вимикачів, перемикачів, прапорців.
6. Кнопка вимикач. Використовується для побудови елементу управління *вимикач*. Два значення цього елементу моделюють дані логічного типу: *істина* та *хибність* (або -1 та 0 , *Так* та *Ні*).
7. Кнопка перемикач. Використовується для побудови елементу управління *перемикач*. Часто цей елемент називають *радіокнопкою*. Значення радіокнопки, як і кнопки вимикача, набувають із множини значень логічного типу. Перемикачі відрізняються тим, що з групи перемикачів може бути вибраний тільки один.
8. Кнопка прапорець. Використовується для побудови елементу управління *прапорець*. Значення цього елементу аналогічні значенням елементів вимикач і перемикач.
9. Кнопка поле із списком. Використовується для побудови елементу управління *поле із списком*. Цей елемент дає можливість вибрати значення з наперед визначених, або з значення поля таблиці (запиту).

10. Кнопка список. Використовується для побудови елемента управління *список*. Як і поле із списком, цей елемент дає можливість вибрати значення з наперед визначених. Але на відміну від елемента управління поле із списком, він завжди відкритий.
11. Кнопка кнопка. Використовується для побудови елемента управління *кнопка*. За допомогою елемента управління *кнопка* можна здійснювати запуск макроса, процедури VBA.
12. Кнопка малюнок. За допомогою цієї кнопки на формі можна розмістити малюнок, зокрема, цей малюнок можна використовувати і як фон всієї форми.
13. Кнопка вільна рамка. Використовується для включення в форму об'єктів OLE. При цьому об'єкт OLE стає часткою форми, але не зберігається у БД (малюнок, звук, діаграма, слайд).
14. Кнопка приєднана рамка. Використовується для включення у форму об'єктів OLE, посилання на які містяться в таблиці БД.
15. Кнопка розриву сторінки. Використовується для побудови багатосторінкових форм, дозволяє вставляти розрив сторінки форми.
16. Кнопка підлеглої форми. Використовується для побудови такого елемента управління форми, який включає іншу форму, пов'язану з поточними даними.
17. Кнопка лінія. Використовується для включення в форму відрізків прямих ліній.
18. Кнопка прямокутник. Використовується для включення в форму прямокутників.

Будь-яка кнопка панелі інструментів, за винятком кнопки вибору, стає неактивною після її використання. Для неперервного використання однієї кнопки панелі інструментів в декількох місцях форми можна кожного разу активізувати цю кнопку, або зафіксувати її подвійним клацанням миши.

За допомогою панелі інструментів можна будувати форми в режимі *конструктор*.

Але часто, особливо при побудові стандартних простих форм для таблиць-довідників, можна успішно користуватися зручним способом – майстром побудови форм. Для цього необхідно в вікні "Новая форма" вибрати "Мастер форм", а в якості головного джерела даних вибрати необхідну таблицю (запит). Після використання кнопки "Ок" необхідно пройти декілька вікон діалогу майстра з користувачем:

- вікно вибору таблиці та її полів;
- вікно вибору зовнішнього вигляду;
- вікно вибору стилю;
- вікно збереження форми.

У вікні вибору таблиці та її полів крім таблиці (запиту), що визначена головною, можна вибрати інші з відповідними полями. У вікнах вибору зовнішнього вигляду і стилю форми майстр забезпечує на елементарному рівні її дизайн. Вікно збереження

форми призначено для введення заголовку форми та її назви. На кожному кроці діалогу майстер дає можливість перейти на крок назад або вперед, завершити процес побудови форми в будь-який момент.

3.2.3 Редагування форм

Після побудови форми за допомогою майстра або конструктора існує чимало причин її редагування: покращання дизайну, підвищення ефективності роботи з формою тощо. Так після діалогу з майстром побудови форми виявляється, що довжини елементів можна змінити і, як наслідок, за рахунок звільнення місця в площині форми можна раціонально розмістити всі елементи, розташувати їх у певній послідовності. Редагуванню форм значно допомагають вікна властивостей самої форми, кожного її розділу (заголовок, область даних, розділ приміток) і елементів управління. Множина властивостей залежить від вибраного елемента. Щоб відкрити вікно властивостей елемента, необхідно спочатку його виділити, а потім клацнути правою кнопкою миши і вибрати "Свойства" (або через меню *Вид->Свойства*). Форма має більш ніж 70 властивостей, елементи управління – більш ніж 30, і тому всі властивості для зручності групуються і зібрані на вкладках:

- *Макет*
- *Данные*
- *События*
- *Другие*

Існує і загальна вкладка *Все*. Властивості задаються з множини значень (наприклад, логічних *Да/Нет*), констант відповідного типу (наприклад, розмір фонового малюнку) а також складними виразами. Для побудови останнього зручно користуватись так званим побудовувачем, який викликається кнопкою з трьома крапками (такі кнопки з'являються, якщо властивість допускає використання побудовувача). Так, якщо на вкладці *Данные* у вікні властивостей форми вибрати властивість *Источник записей*, то в якості побудовувача після його виклику стане побудовувач запитів. Особливо зручно користуватись побудовувачем у випадку, коли необхідно побудувати та обчислити вираз. Наприклад, обчислення добутку ab , де a і b - поля таблиці, і подальше присвоєння значення добутку полю c , можна виконати обробкою подій по виходу з поля a або з поля b . Вказана обробка подій буде забезпечена, якщо скористатися побудовувачем двох однакових за змістом процедур VBA:

```
Private Sub a_Exit()  
    c=a*b  
End Sub
```

```
Private Sub b_Exit()  
    c=a*b  
End Sub
```

В результаті під час роботи з формою на стадії її відкриття значення поля c в цьому прикладі буде автоматично переобчислюватись, як тільки користувач переходить від поля a , або поля b до іншого елемента форми.

Для зміни розміру елемента форми необхідно виділити його, а потім користуватися маркерами, що з'являються по кутах та серединах сторін прямокутника виділення. Маркер в лівому верхньому куті – найбільший, їм не можна користуватись при зміні розмірів. Для переміщення елемента можна мишею стати між маркерами на будь-якій стороні прямокутника виділення (вказівник прийме вигляд долоні) та перетягти елемент у необхідне місце. Якщо до поточного елемента приєднаний надпис, то і він також буде при цьому перетягуватись. Для того, щоб приєднаний надпис не перетягувався, необхідно користуватись верхнім лівим маркером (вказівник прийме вигляд стиснутої долоні).

3.2.4 Пошук та сортування даних

Пошук та сортування даних у формі, що відкрилась, можна виконувати так, як у відкритій таблиці.

Для простого пошуку спочатку необхідно стати на відповідне поле, а далі командою *Правка->Найти* (або кнопкою *Найти*) відкрити вікно діалогу пошуку. В цьому вікні можна задати умови пошуку, причому, на відміну від пошуку у таблиці, пошук у формі можна здійснювати і по полях, що обчислюються.

Сортування по полю форми виконується ще простіше: необхідно після вибору поля форми натиснути одну із кнопок сортування на панелі інструментів (сортування по збільшенню значення поля, або - по зменшенню).

3.3 Звіти

Звіти використовуються для того, щоб вибрати інформацію з однієї або декількох таблиць, виконати необхідні обчислення, підвести підсумки. При цьому можна забезпечити групування даних та необхідний дизайн змісту звіту. Користувачу надається можливість переглянути або надрукувати зміст звіту.

Звіти створюються згідно з умовами відносно структури необхідного документу. Найпростішою є така структура:

- *верхній колонтитул*
- *заголовок групи даних*
- *область даних*
- *примітка групи даних*
- *нижній колонтитул*

Цією структурою передбачено групування даних, причому групування одного рівня. Якщо структура документу передбачає групування декількох рівней, то це відіб'ється у структурі звіту тим, що з'являться вкладені підзаголовки підгруп даних та відповідні примітки. Зрозуміло, що множина об'єктів звіту може і не вмещувати деякі з описаних. Наприклад, необов'язковим може бути нижній колонтитул.

3.3.1 Механізми створення звітів

У звітах, як правило, використовується інформація з різних таблиць. Тому простіше розпочати розробку звіту з попередньої роботи – побудови базового запиту. При побудові базового запиту необхідно визначитись з множиною таблиць та їх схемою відношень, множиною полів цих таблиць, що будуть використовуватись у звіті, умовою

вибору записів та їх сортування. З такої сукупності інформації і буде складений звіт. Звичайно, деякі поля можуть виконувати свою роль при побудові базового запиту, але, взагалі кажучи, саме їх значення у звіті можуть бути відсутніми. Після підготовчої роботи з створення базового запиту можна будувати звіт.

Звіти часто будуються за допомогою одного з майстрів або конструктора звітів. Для першого знайомства із звітом необхідно обрати основного майстра розробки звітів, який дозволяє вибрати поля для звіту, задати формати, умови групування та функції для підсумків. Вся робота майстра виконується через діалог з користувачем і складається з таких кроків:

- 1) *вибір головної таблиці;*
- 2) *вибір доступних полів головної таблиці а також полів з інших таблиць;*
- 3) *вибір умов групування;*
- 4) *задання умов сортування та підсумків;*
- 5) *вибір вигляду макету звіту та його орієнтації;*
- 6) *вибір стилю звіту;*
- 7) *задання імені звіту.*

Кожний з цих кроків здійснюється через відповідне вікно діалогу з користувачем, причому можна рухатись як вперед по вікнах, так і назад, а також здійснити відміну всього процесу побудови звіту.

Вікно вибору головної таблиці . Тут передбачена можливість вибрати необхідний елемент з фіксованого списку

- Конструктор
- Мастер отчетов
- Автоотчет: в столбец
- Автоотчет: ленточный
- Диаграмма
- Почтовые наклейки

(наприклад, *майстер звітів*), а з нефіксованого списку – таблицю (запит), яка буде надалі використовуватись як джерело даних.

Вікно вибору полів . Тут можна включити в звіт декілька або всі доступні поля в тій же послідовності, як і в базовому запиті (кнопки > та >>).

Вікно вибору умов групування даних. Групування даних може бути виконано за майстром декількома рівнями. На питання *Добавить уровни группирования?* необхідно вибрати відповідне поле групування або відмовитись від подальшого рівня групування. У випадку, коли групування вибране можна налаштувати інтервали групування кнопкою *Группировка...*

Вікно завдання умов сортування та підсумків даних. Сортування даних може бути виконано за майстром по чотирьох полях. Підсумки та умови їх виведення у звіті можна визначити кнопкою *Итоги....*

Вікно вибору вигляду макету звіту та його орієнтації. В арсеналі майстра існує множина макетів звіту, серед яких можна вибрати потрібний. В цьому ж вікні можна автоматично налаштувати ширину кожного стовпчика звіту у відповідності з шириною сторінки так, щоб весь звіт розмістився на одній сторінці.

Вікно вибору стилю звіту. Майстер пропонує багато стилей, серед яких можна вибрати потрібний.

Вікно завдання імені звіту. Тут задається ім'я звіту та надається можливість переглянути сам звіт або редагувати його.

3.3.2 Редагування звіту

Передусім необхідно визначитись з рівнями та критеріями групування. Бажано уникнути можливості відриву заголовку групи від її змісту, що трапляється під час друку, коли інформація продовжується на наступній сторінці. Щоб запобігти цьому, треба властивість "Не разрывать" встановити у значення "Первая область данных". Інші значення властивості "Не разрывать" залежать також від вибору того чи іншого розміщення інформації нв аркушах надрукованого звіту. Цікаво, що звіт може автоматично бути спадкоємцем властивостей базового запиту відносно фільтру, порядку сортування, тощо. Для передачі таких властивостей від базового запиту до звіту необхідно редагувати властивості звіту. Це можна зробити командою меню *Правка->Выделить отчет* і далі правою кнопкою миши клацнути в околі, що за правим краєм області даних, та перейти до вікна "Отчет", де і встановити необхідні властивості.

У наступній таблиці (Табл. 5) зведені властивості звіту та їх зміст:



Табл. 5 - Властивості звіту

Властивість	Зміст
<i>Источник записей</i>	Ім'я базової таблиці(запиту), дані яких формують звіт.
<i>Фильтр</i>	Фільтр-спадкоємець джерела даних.
<i>Фильтр включен</i>	<i>Да</i> – автоматичне використання фільтру при кожному відкритті звіту.
<i>Порядок сортировки</i>	Умова сортування
<i>Сортировка включена</i>	<i>Да</i> – автоматичне сортування при кожному відкритті звіту.
<i>Подпись</i>	Введення тексту.
<i>Блокировка записей</i>	Блокування записів від можливої їх зміни іншими користувачами під час друку звіту
<i>Верхний колонтитул</i>	Умова друку верхнього колонтитулу на сторінках звіту
<i>Нижний колонтитул</i>	Умова друку нижнього колонтитулу на сторінках звіту
<i>Группировка по датам</i>	Вибір стандартів дат і часу: американський або за угодою (вікно <i>Язык и стандарты</i> панелі управління Windows)

Властивість	Зміст
<i>Неразрывная группа</i>	Забезпечення на кожній сторінці звіту нерозривності групованих даних.
<i>Рисунок</i>	Введення повної назви файлу з растровим малюнком
<i>Тип рисунка</i>	<i>Внедренный</i> – копіювання малюнка у звіт, <i>Связанный</i> – завантаження малюнка при кожному відкритті звіту на основі інформації властивості <i>Рисунок</i>
<i>Масштаб рисунка</i>	Визначення розмірів малюнка у звіті: <i>Фрагмент</i> – розміри малюнка зберігаються. Малюнок обрізається, якщо його розміри більші за відповідні розміри рамки. <i>По размеру рамки</i> – пропорційна зміна розмірів малюнка до відповідних розмірів рамки наскільки це можливо. <i>Вписать в рамку</i> – зміна розмірів малюнка до відповідних розмірів рамки.
<i>Выравнивание рисунка</i>	Позиціювання малюнка у кутах або в центрі рамки (за умови вибору значення масштабу малюнка <i>Фрагмент</i>)
<i>Мозаичное заполнение</i>	<i>Да</i> – заповнення копіями малюнка всього простору рамки (за умови вибору значення масштабу малюнку <i>Фрагмент</i>)
<i>Страницы с рисунком</i>	Виведення малюнка на першій або на кожній сторінці звіту, невиведення малюнка на кожній сторінці звіту.
<i>Строка меню</i>	<i>True</i> -виведення меню звіту під час його перегляду.
<i>Контекстное меню</i>	Виведення контекстного меню звіту під час його перегляду (за допомогою правої кнопки миші).
<i>Число делений по X</i>	Задання відстані між вертикальними лініями сітки.
<i>Число делений по Y</i>	Задання відстані між горизонтальними лініями сітки.
<i>Формат для печати</i>	<i>Да</i> – можливість вибору шрифтів True Type та принтерних шрифтів, <i>Нет</i> – тільки екранних шрифтів.
<i>Для лазерного принтера</i>	Друк ліній із збільшеною швидкістю (принтерами, які мають спеціальні команди для цього).
<i>Файл справки</i>	Можливість побудови файлу прикладних довідок.
<i>Индикатор справки</i>	Виведення на екран індикатору прикладної довідки.
<i>Источник палитры</i>	Зміна кольорової палітри під час друку на кольоровому принтері.
<i>Дополнительные сведения</i>	Зберігання інформації для наступної її передачі іншим процедурам.
<i>Открытие</i>	Запуск процедури або макроса під час друку (перегляду) звіту.

Властивість	Зміст
<i>Закрытие</i>	Запуск процедури або макроса під час закриття вікна друку (перегляду) звіту.
<i>Отсутствие данных</i>	Запуск процедури або макроса, якщо під час відкриття звіту джерело записів не містить даних.
<i>Ошибка</i>	Запуск процедури або макроса, якщо звіт містить помилки.

3.3.3 Виконання обчислень та підведення підсумків

Як і у випадку побудови форми, для виконання обчислень у звіті зручно використовувати побудовувачі. Найпростіший спосіб полягає у тому, щоб спочатку вибрати інструмент побудови елемента *поле*, розмістити цей елемент у необхідному розділі звіту у певному місці. При цьому треба визначити підпис та вираз, значення якого планується обчислити у звіті. Підпис визначається звичайним редагуванням тексту. Для формування виразу необхідно відкрити вікно властивостей цього елемента форми (двічі клацнути лівою кнопкою миши по цьому елементу або стати на цей елемент та клацнути правою кнопкою миши). У вікні властивостей вибрати закладку *Данные* і на одноіменному рівні клацнути кнопку  для вибору поля у випадку, коли вираз є полем, і кнопку  для виклику побудовувача виразу. Вікно побудовувача виразу (Рис. 2) має такий вигляд:

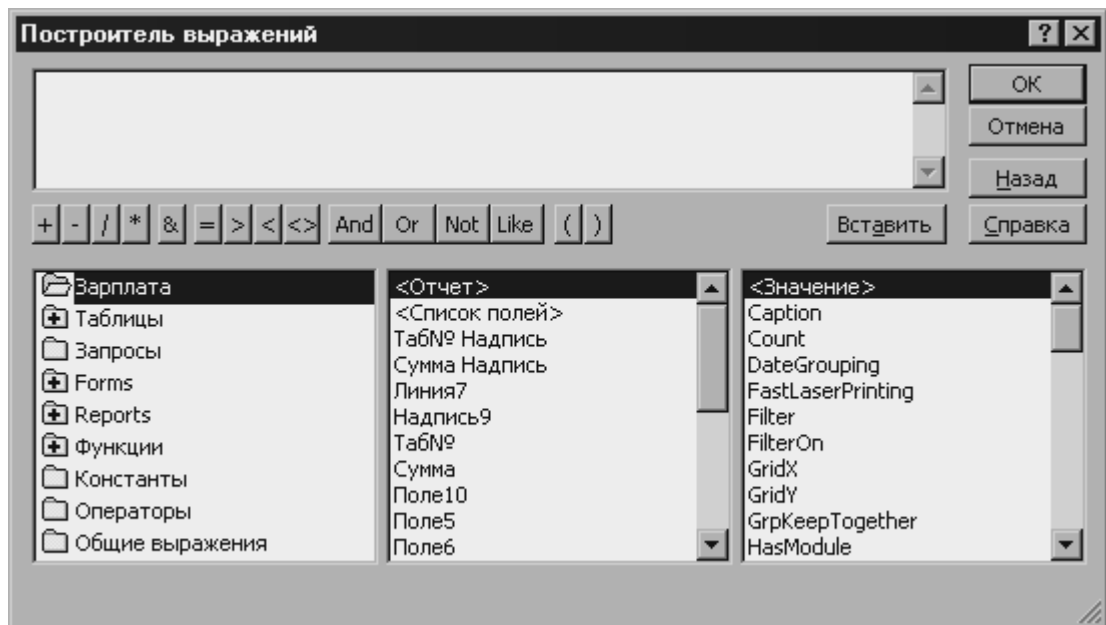


Рис. 2 - Побудовувач виразу

У цьому вікні можна скористатись необхідною функцією з переліку функцій, зібраних за своїм змістом у відповідні розділи. Аргументом цих функцій, як правило, є поля поточних таблиць звіту. Доступ до таких полів можливий, необхідно для цього використати середнім віконцем з трьох в нижній частині вікна побудовувача.

Для підведення підсумків у звіті існує простий спосіб: використання майстра побудови звітів, в арсеналі якого є відповідні інструменти. У випадку, коли звіт розробляється конструктором (без допомоги майстра), перш за все необхідно визначитись, до якої категорії належать ці підсумки: групи, звіту, забезпечити відповідні розділи звіту (піктограмою *Сортировка и группировка*), а потім скористатись побудовою виразу, де використати функцію отримання суми SUM().

Примітка: синтаксис виразу передбачає знак рівності "=" перед ним.

Головний принцип щодо оформлення звіту – правильна передача форми документу, яка затверджена міністерством, держадміністрацією, підприємством, тощо. Але деякі поради, якщо вони не протирічать затвердженій формі, можна все ж таки сформулювати.

- *Об'єднання текстових значень і приховування даних, що повторюються.* Якщо декілька рядків звіту мають співпадаючі значення, то такий звіт важко читається і виглядає непрофесійно. Властивість поля *Не виводить повторы* (вона доступна лише для звітів) дозволяє керувати виведенням на друк значень, що повторюються. Ця властивість діє для кожного поля виразу незалежно. У випадку, коли необхідно керувати сполученням значень декількох виразів (полів), необхідно об'єднати ці вирази (поля) у окремий елемент звіту, а потім застосувати вказану властивість до цього елемента.
- *Обчислення відсотків.* При формуванні звіту MS Access двічі проходить дані. На першому проході вона обчислює вирази у детальних рядках, підсумки у групах, загальні підсумки для всього звіту та визначає довжину звіту. На другому проході MS Access обчислює всі вирази, що базуються на підсумках першого проході. Така технологія обчислення звіту дає можливість в області даних та розділах обчислити відсотки від загальних значень. Обчислення відсотків досягається, якщо встановити властивість *Формат поля* у значення *Процентний*.

4 Робота з даними

4.1 Можливості та типи запитів

Запит - один з найбільш потужних об'єктів MS Access, який дозволяє ефективно представити інформацію, що містять таблиці, з певними властивостями. В деякому розумінні запит подібний до фільтрів, коли з таблиць будується виборка за певною умовою. Але на відміну від фільтра запит дозволяє отримати більш змістовний результат. Перш за все, це пояснюється тим, що фільтр дає інформацію для перегляду (друку), але, на відміну від запиту автоматично не зберігається, як окремий об'єкт бази даних. Запити, маючи таку властивість, дозволяють динамічно поновлювати інформацію у своїх таблицях, якщо у таблицях бази даних виникла зміна інформації. Крім цього, запит має і зворотню дію: якщо змінювати інформацію у його таблицях, то таблиці бази даних, на базі яких побудований запит, будуть адекватно змінювати свою інформацію.

4.2 Створення запитів у середовищі MS Access

Будувати запит необхідно відповідно з постановкою задачі. Одним з найпростіших механізмів є *Конструктор*. Для того, щоб скористатись цим механізмом, перш за все у вікні побудови бази даних необхідно перейти на закладку *Запросы*. Через кнопку *Создать*, будується запит. На цьому кроці можна вибрати декілька способів побудови запиту, в тому числі за допомогою *Конструктора*. В результаті з'являється вікно, у верхній частині якого необхідно розмістити таблиці, інформація з яких нас цікавить (поступове додавання таблиць здійснюється у вікні *Добавление таблиць*). У нижній частині необхідно вказати поля запиту, в тому числі ті, що якісно впливають на запит, але значення яких не виводиться. Причому тут можна визначитись і відносно сортування по необхідних полях. Щоб внести інформацію щодо сортування, значення виборки тощо необхідно в нижній частині вікна (тут діє так звана розмітка QBE - це набір текстових вікон, які мають назву комірок, за допомогою яких здійснюється опис запиту) стати на перехрестя відповідного рядка і стовпчика. Після визначення назви запиту, його можна відкрити. В результаті одержується деякий результат у вигляді таблиці. Ця таблиця надалі може використовуватись у будь-яких операціях на рівні із звичайними таблицями.

4.3 SQL – запити

SQL (Structured Query Language – структурована мова запитів) – це стандартна мова програмування, яка має свої корені у 70-х роках минулого століття, коли фірма IBM створювала програмне забезпечення для роботи з реляційними базами даних. Сьогодні існує міжнародний стандарт мови SQL, якого дотримуються розробники програмного забезпечення з обробки реляційних баз даних. У MS Access вбудовано деякий діалект мови SQL, який використовується в таких діях, як побудова запитів на виборку, підсумовуючих запитів, перехрестних запитів, запитів на побудову таблиць, запитів на оновлення, додавання або видалення записів таблиці. Мову SQL MS Access використовує при роботі з базою даних, в тому числі неявно, коли користувач діє в рамках звичайного запиту – кожний запит, незалежно від того, як він створювався, зберігається у вигляді інструкції SQL. Необхідність використання мови SQL виникає в різних випадках. Наприклад, коли є необхідність побудувати так званий *підлеглий запит*, результати якого використовуються в якості умов в інших запитах. Синтаксис діалекту MS SQL необхідно знати, щоб безпосередньо на цій мові виконувати ті чи інші SQL-запити. Для початку цікаво познайомитись, як в термінах SQL зберігається будь-який запит. Для цього необхідно лише скористатись дією меню MS Access *Вид - Режим SQL*, коли стає активним той чи інший запит.

Ядром мови SQL є інструкція SELECT, яка використовується для відбору полів і записів таблиць бази даних. Синтаксис інструкції SELECT використовує п'ять основних речень і його можна представити в наступному вигляді:

```
SELECT <список полів>  
FROM <список таблиць>  
[WHERE <специфікація вибору записів>]  
[GROUP BY <специфікація групування>]  
[HAVING <специфікація вибору груп>]  
[ORDER BY <специфікація сортування>]
```

Тут речення FROM визначає таблиці або запити, які є джерелом даних для запиту, що створюється. Речення WHERE та HAVING визначають умови відбору записів і груп записів відповідно. Речення GROUP BY визначає поле (поля) для подальшого формування запиту відносно груп даних з однаковими значеннями цього поля (полів). Речення ORDER BY визначає сортування записів. Крім цих речень в кожному діалекті мови SQL можуть зустрітись інші речення стандарту SQL. У MS Access реалізовані чотири з них:

TRANSFORM – для побудови перехресного запиту;

IN – для зв'язку з віддаленою базою даних;

DISTINCTROW – для відбору унікальних записів;

WITH OWNERACCESS OPTION – для побудови запиту користувачами, що не мають прав доступу до відповідних таблиць цього запиту.

Приклад запиту з пов'язаних між собою таблиць *Особи*, *Освіта* щодо освіти жінок з сортуванням за прізвищем записів, які містять поля Прізвище, Освіта:

```
SELECT Особи.Прізвище, Освіта.Освіта
FROM Особи INNER JOIN Освіта
ON Особи.КодОсвіти = Освіта.КодОсвіти
WHERE (((Особи.Стать)="Ж"))
ORDER BY Особи.Прізвище;
```

Для формування списків та специфікацій інструкції SELECT використовуються вирази, що визначають деяке значення. Синтаксис виразів передбачає структуру, що базується на константах того чи іншого типу, функціях та виразах. Вирази, функції і константи з'єднуються між собою операторами: арифметичними – для побудови виразу з операндів числового типу (+, -, *, /, \, ^, MOD), логічними – для побудови виразу з операндів логічного типу (AND, OR, NOT) і операндів числового типу (=, >, >=, <=, <, <>), текстовими – для побудови виразу з операндів текстового типу (&). Приклади виразів:

- "beta"&"01" – конкатенація рядків *beta* та *01*
- price*50 - добуток змінної (або поля) числового типу *price* та числа 50
- "Дата народження: "+CDATE(date) – конкатенація рядка *Дата народження:* та текстового виразу після перетворення змінної (поля) *date* типу Дата/Время у значення текстового типу.

До запиту SQL часто включаються параметри, значення яких можуть бути введені користувачем або з елементів управління активної форми. Для використання таких параметрів необхідно їх попередньо визначити за допомогою оператора PARAMETERS, синтаксис якого передбачає присутність імені параметру (ідентифікатору) та його типу:

```
PARAMETERS < ідентифікатор> <тип даних> ...
```


Серед типів даних є логічний (Bit), грошовий (Currency), дата/час (DateTime), цілий (Integer), текстовий (Text) та інші.

Інструкції SQL використовуються також і для видалення, вставки або оновлення даних, а також для побудови нової таблиці з даними що існують. Запити на зміну даних – це достатньо потужний засіб, оскільки така дія впливає одразу на множину записів. Наприклад, інструкції UPDATE або DELETE діють на всі записи базових таблиць, для яких виконується умова відбору. Разом з інструкцією видалення існує інструкція додавання записів INSERT. Для побудови нової таблиці із значеннями, що вибрані з одної або декількох таблиць, використовується інструкція SELECT INTO

5 Побудова програмного забезпечення

5.1 Макроси

Використання макросів є одним із способів побудови програмного забезпечення. У MS Access можна визначити макрос, який виконує майже всі дії, які можна реалізувати на клавіатурі або мишею. Головна властивість макросів та, що вони можуть виконуватись як відповідь на ту чи іншу подію. Саму подію можна трактувати як твердження, яке, зрозуміло, має два значення: істина або хибність. Макроси використовуються для виконання таких дій:

- відкриття (закриття) об'єктів MS Access – таблиць, запитів, форм, звітів, макросів;
- відмова події, яка здійснила запуск макроса, і навіть вихід з програмного забезпечення;
- обчислення та встановлення значень елементів управління форм, звітів;
- емуляція роботи з клавіатурою;
- фільтрація даних;
- визначення спеціального рядка меню;
- відкриття (закриття) панелі інструментів;
- виконання будь-якої команди будь-якого меню MS Access;
- переходи між вікнами та стандартні дії з ними (переміщення, зміна розмірів і таке інше) ;
- виведення інформаційних повідомлень;
- вимикання деяких попереджаючих повідомлень з боку MS Access;
- копіювання, перейменування, видалення об'єктів MS Access;
- імпорт (експорт) таблиць баз даних, електронних таблиць, текстових файлів;
- запуск програмного забезпечення та передача йому параметрів

Для побудови того чи іншого макроса необхідно визначитись з послідовністю його макрокоманд. Ці макрокоманди можна вибрати під час створення макроса за допомогою кнопки *Создать* у закладці *Макросы*. У Табл. 6 наведені макрокоманди та їх короткий зміст.

Табл. 6 - Макрокоманди MS Access

<i>Макрокоманда</i>	<i>Зміст</i>
Восстановить	Поновлення попередніх розмірів вікна.
ВывестиВФормате	Виведення даних, що містяться у вказаному об'єкті бази даних MS Access (таблиці, форми, звіти або модулі) у файл в форматі MS Excel 97 (*.xls), в текстовому форматі MS-DOS (*.txt) або в форматі RTF (*.rtf). Також можливе виведення у файли формату HTML (*.html), Microsoft Internet Information Server (*.htx, *.idc) або у файл в форматі сторінок Microsoft ActiveX Server (*.asp).
ВыводНаЭкран	Визначає режим виведення на екран результатів виконання поточних операцій. Наприклад, ця макрокоманда дозволяє вивести на екран або сховати проміжні результати виконання макроса.
ВыделитьОбъект	Виділення вказаного об'єкта бази даних.
ВыполнитьКоманду	Виконання команди MS Access.
Выход	Вихід з MS Access. При цьому передбачений вибір параметрів зберігання об'єктів бази даних.
ДобавитьМеню	<p>Виконання наступних дій:</p> <ul style="list-style-type: none"> · побудова спеціального рядка меню для форми або звіту. Спеціальний рядок меню замінює вбудований рядок меню форми або звіту. · побудова спеціального контекстного меню для форми, елемента управління або звіту. Спеціальне контекстне меню замінює вбудоване контекстне меню форми, елемента управління або звіту. · побудова загального рядка меню. Загальний рядок меню замінює вбудоване головне меню у всіх вікнах MS Access, за винятком тих, що містять спеціальні рядки меню форми або звіту. · побудова загального контекстного меню. Загальне контекстне меню замінює вбудоване контекстне меню для всіх полів таблиць або запитів в режимі <i>Таблицы</i>, форм в режимі <i>Формы</i>, режимі <i>Таблицы</i> і режимі попереднього перегляду, а також звітів в режимі попереднього перегляду, за виключенням тих, де користувач додав спеціальне контекстне меню форми, звіту або елемента управління. <p><u>Примітка.</u> Для побудови нових меню рекомендується всеж таки використовувати діалогове вікно <i>Настройка панелей инструментов</i>, з меню <i>Вид команды - Панели инструментов</i> і кнопки <i>Настройка</i>.</p>
ЗадатьЗначение	Задання значень поля, елемента управління або властивості в формі

<i>Макрокоманда</i>	<i>Зміст</i>
ЗадатьКомандуМеню	Задання стану команд спеціального рядка меню або загального рядка меню для активного вікна.
Закричь	Зачинення вказаного вікна окно MS Access або поточного вікна (за угодою).
ЗапускЗапросаSQL	Запуск запиту на зміни MS Access за допомогою відповідної інструкції SQL. Крім того, ця макрокоманда дозволяє запустити керуючий запит.
ЗапускМакроса	Запуск макросу. Можна вказувати макрос з групи макросів.
ЗапускПриложения	Запуск із MS Access програмного забезпечення Windows або MS-DOS (наприклад, Microsoft Excel, Microsoft Word для Windows або Microsoft PowerPoint). Наприклад, ця макрокоманда дозволяє виконати вставку електронної таблиці MS Excel в базу даних MS Access.
ЗапускПрограммы	Виклик функції Visual Basic.
КомандыКлавиатуры	Передача натискань клавіш безпосередньо у MS Access або в активне програмне забезпечення MS Windows.
КопироватьОбъект	Копіювання вказаного об'єкта бази даних в іншу базу даних MS Access або в ту ж саму базу даних під новим ім'ям. Наприклад, цю макрокоманду використовують для копіювання або зберігання існуючого об'єкта в іншій базі даних або при використанні існуючого об'єкта в якості прототипу нового об'єкта.
КЭлементуУправления	Переведення фокусу на вказане поле або елемент управління у поточному записі форми, таблиці, запиту. Крім цього, ця макрокоманда використовується для автоматичного переміщення по формі у відповідності з визначеними умовами. Наприклад, якщо оператор введе «Так» в поле <i>Холост</i> , то поле <i>Супруг</i> буде пропущено автоматично, а фокус переданий наступному елементу управління.
НаЗапись	Зробити вказаний запис поточним для таблиці, форми або записів запиту.
НайтиЗапись	Пошук даних, що зодовільняють умовам пошуку цієї макрокоманди.
НаСтраницу	Передача фокусу в активній формі першому елементу управління. Ця макрокоманда використовується при побудові багатосторінкових форм з групуванням даних на різних сторінках.
ОбновитьОбъект	Завершення всіх відкладених операцій оновлення вказаного об'єкта бази даних або активного об'єкта бази даних (за угодою). При необхідності, вконується перерозрахунок значень елементів управління в цьому об'єкті.

<i>Макрокоманда</i>	<i>Зміст</i>
Обновление	Оновлення даних у вказаному елементі управління в активному об'єкті шляхом повторного перегляду даних.
ОстановитьВсеМакросы	Зупинка виконання всіх макросів, що виконуються
ОстановитьМакрос	Зупинка виконання поточного макросу, що виконується
ОткрытьЗапрос	Відкриття запиту на виборку або перехресного запиту в режимі <i>Таблицы</i> , в режимі <i>Конструктор</i> або в режимі попереднього перегляду. Крім цього, ця макрокоманда може вказати для запиту режим введення даних.
ОткрытьМодуль	Відкриття вказаної процедури в модулі Visual Basic. Ця процедура може бути процедурою Sub, процедурою Function або процедурою обробки подій.
ОткрытьОтчет	Відкриття звіту в режимі <i>Конструктор</i> , в режимі попереднього перегляду або виведення звіту на друк. При цьому допускається виборка записів, що включаються в звіт.
ОткрытьТаблицу	Відкриття таблиці в режимі <i>Таблица</i> , в режимі <i>Конструктор</i> або в режимі попереднього перегляду. При цьому допускається вибір режиму введення даних в таблицю.
ОткрытьФорму	Відкриття форми в режимі <i>Форма</i> , в режимі <i>Конструктор</i> форми, в режимі попереднього перегляду або в режимі <i>Таблица</i> . При цьому допускається вибір режиму введення даних та режиму вікна, а також відбір записів, які необхідно показати у формі.
ОтменитьСобытие	Відміна події, яка викликала запуск макросу.
ОтправитьОбъект	Включення вказаного об'єкта в режимі <i>Таблица</i> , <i>Форма</i> , <i>Отчет</i> або <i>Модуль</i> MS Access у повідомлення електронної пошти, перегляд та відправлення. Допускається включення об'єктів у форматі MS Excel (*.xls), текстовому форматі MS-DOS (*.txt), форматі RTF (*.rtf) або HTML у повідомлення електронної пошти Microsoft Exchange, Microsoft Mail, Microsoft Windows для робочих груп або іншого програмного забезпечення електронної пошти, яке використовує інтерфейс MAPI (Microsoft Mail Applications Programming Interface). Є можливість пересилки об'єктів MS Access у повідомлення електронної пошти VIM- інтерфейсу.
ПанельИнструментов	Виведення на екран або видалення з екрану вбудованої або спеціальної панелі інструментів. Користувач має можливість вказати виведення вбудованої панелі інструментів у всіх вікнах MS Access або тільки в одному вікні, для якого ця панель інструментів призначена (наприклад, вивести панель інструментів <i>Режим форми</i> тільки у вікні форми в режимі <i>Форма</i>).
Переименовать	Перейменування вказаного об'єкта бази даних.

<i>Макрокоманда</i>	<i>Зміст</i>
ПесочныеЧасы	Надати вказівнику форму пісового годинника (або іншого вказаного значка) на час виконання макроса. Ця макрокоманда дозволяє побудувати вказівник, що свідчить про виконання дій поточним макросом. Це особливо зручно, коли виконання макрокоманди або самого макроса займає достатньо великий час.
Печать	Друк активного об'єкта відкритої таблиці бази даних.
ПоказатьВсеЗаписи	Відміна будь-якого поточного фільтру.
Преобразовать БазуДанных	Перетворення бази даних при імпорті або експорті даних між поточною базою даних MS Access та іншою базою даних
ПреобразоватьТекст	Перетворення тексту при імпорті або експорті даних між поточною базою даних MS Access та текстовим файлом (а також файлом HTML (*.html)).
Преобразовать ЭлектроннуюТаблицу	Перетворення електронної таблиці при імпорті або експорті даних між поточною базою даних MS Access та файлом електронної таблиці.
ПрименитьФильтр	Застосування фільтру, запиту або інструкції WHERE (SQL) до таблиці, форми або звіту для виборки і сортування записів.
Развернуть	Збільшення розмірів активного вікна до розмірів вікна MS Access.
Свернуть	Згорання активного вікна.
СдвигРазмер	Зміна місця та розмірів активного вікна.
Сигнал	Подання звукового сигналу через динамик комп'ютера.
СледующаяЗапись	Знаходження наступного запису, який задовільняє умовам, що вказані у попередній макрокоманді <i>НайтиЗапись</i> або у діалоговому вікні <i>Поиск в поле</i> , яке відчиняється командою <i>Найти</i> з меню <i>Правка</i> .
Сообщение	Виведення на екран вікна, яке містить попередження або інформаційне повідомлення.
Сохранить	Збереження вказаного об'єкта MS Access, або активного об'єкта, якщо об'єкт не вказаний.
УдалитьОбъект	Видалення вказаного об'єкта бази даних.
УстановитьСообщения	Ввімкнення або вимкнення виведення системних повідомлень.

5.2 Кнопкові форми

Побудова кнопкових форм – це фінальна робота, яка в деякому розумінні логічно і структурно завершує побудову бази даних. Здійснюється вона у закладці *Формы* за допомогою кнопки *Создать* та *Конструктора*. На відміну від побудови

форм для введення даних, тут немає потреби у визначенні таблиць та їх полів. Головний технологічний прийом для виготовлення кнопкової форми – це розміщення елементів управління *Кнопка* у площині форми та надання цим елементам однієї властивості (функції): завантаження того чи іншого об'єкта. Крім цього, звичайно необхідно потурбуватись про дизайн форми: вибрати малюнок для фону, вдало розмістити кнопки а також вибрати їх розмір, картинку або підпис на них. Часто необхідно побудувати кнопкову форму, з якої можна відкривати форми, звіти, запити. Розробляються також і такі кнопкові форми, які у свою чергу можуть відкривати інші кнопкові форми.

Встановлення розмірів форми та визначення малюнка здійснюється за допомогою вікна властивостей форми та закладки *Макет* в цьому вікні. Для розміщення кнопок спочатку вмикається панель інструментів (якщо її немає на екрані монітору). Потім, попередньо вмикаючи на цій панелі кнопку майстра побудови елементів управління, вибирається елемент *Кнопка*. Далі вказується мишею місце його розміщення на кнопковій формі. Після цього майстер побудови елемента управління пропонує діалог:

перший крок – вибір категорії та дії. Тут в залежності від категорії відкривається відповідна множина дій (наприклад, можна вибрати категорію *Работа с формой* та дію *Открытие формы*).

другий крок – вибір вже підготовленого об'єкта (наприклад, деякої конкретної форми).

третій крок – вибір множини даних об'єкта (наприклад, множини записів, а саме можна вибрати *Открыть форму и показать все записи*).


четвертий крок – вибір об'єкта розміщення на кнопці: текст або малюнок. (наприклад, текст і змінити текст *Открытие формы* на необхідний за змістом даної форми).

п'ятий(останній) крок – визначення ідентифікатора (назви) елемента управління. Тут можна погодитись із запропонованим з боку майстра ідентифікатором.

Нарешті, необхідно змінити значення деяких властивостей форми, які пропонуються конструктором форм за угодою у випадку конструювання форм для таблиць. Зведемо ці зміни у таблицю (Табл. 7)

Табл. 7 - Зміни властивостей форми

Властивість	Значення за угодою	Встановлене значення
<i>Полосы прокрутки</i>	<i>Все</i>	<i>Отсутствуют</i>
<i>Область выделения</i>	<i>Да</i>	<i>Нет</i>
<i>Поле номера записи</i>	<i>Да</i>	<i>Нет</i>
<i>Разделительные линии</i>	<i>Да</i>	<i>Нет</i>

Заголовок форми розміщуємо у верхній частині форми за допомогою інструменту , вибираючи шрифт, розмір шрифту та колір.

6 Теми учбових розробок для самостійної роботи студентів

Розробити базу даних за учбовою темою, передбачивши в ній декілька таблиць, що зв'язані між собою відношенням “один до багатьох”, форми для введення даних, звіти для виведення даних. Відкриття форм та звітів оформити через кнопкові форми. Розрахункові поля забезпечити макросами або процедурами. Звіт побудувати групуванням за одним з полів та сортуванням за іншим полем.

Табл. 8 - Варіанти завдань для самостійної роботи

<i>№ n/n</i>	<i>Тема</i>	<i>№ n/n</i>	<i>Тема</i>
1	Автоперевезення	11	Обмін валют
2	Гуртожиток	12	Основні фонди
3	Енергопостачання	13	Постачальники та клієнти
4	Екзаменгаційна сесія	14	Прейскурант цін на продукцію
5	Кадри	15	Продаж білетів (цирк, театр тощо)
6	Картинна галерея	16	Розподіл педнавантаження
7	Керування трудовими ресурсами	17	Склад
8	Купівля та продаж товару	18	Стипендія
9	Маркетинг	19	Табель врахування робочого часу
10	Матеріальні цінності	20	Туристичне агенство

Для створення баз даних за темами, зокрема з Табл. 8, необхідно розробити проект структур таблиць бази даних, забезпечити зв'язок таблиць і таким чином розробити схему даних. В кожній базі даних необхідно передбачити дві довідкові таблиці та одну облікову, в якій повинно бути поле для обчислення.

7 Приклади таблиць та їх полів для учбової розробки баз даних

У наступних прикладах наведено можливі таблиці та відповідні поля для деяких з учбових тем (Табл. 8). Крім того, пропонуються поля, що обчислюються, та можливі запити. Зовсім не обговорюється можлива схема даних, цілісність даних, які логічно впливають з наведеної інформації. Студент повинен сам вибрати типи даних для полів та їх розмір, продумати підстановку можливими фіксованими значеннями а також підстановку значень полів з таблиць-довідників.

Театральні білети

<i>СПЕКТАКЛІ</i>	<i>ТИПИ БІЛЕТІВ</i>	<i>ОБЛІК ПРОДАЖУ</i>
Код спектаклю	Тип білету	Код спектаклю

<i>СПЕКТАКЛІ</i>	<i>ТИПИ БІЛЕТІВ</i>	<i>ОБЛІК ПРОДАЖУ</i>
Назва	Назва	Дата
	Кількість	Тип білету
	Ціна	Кількість
		Сума

Поле, що обчислюється *ОБЛІК ПРОДАЖУ.Сума = ОБЛІК ПРОДАЖУ.Кількість * ТИПИ БІЛЕТІВ.Ціна*

Запит щодо вільних місць на даний спектакль.

Автоперевезення

<i>ВОДІЇ</i>	<i>ТИП ПЕРЕВЕЗЕНЬ</i>	<i>ШЛЯХОВІ ЛИСТИ</i>
Табельний №	Тип перевезень	Табельний №
Прізвище	Назва	Дата
Стать	Тариф заробітної плати	Тип перевезень
		Години в наряді
		Сума заробітної плати

Поле, що обчислюється *ШЛЯХОВІ ЛИСТИ.Сума заробітної плати = ШЛЯХОВІ ЛИСТИ.Години в наряді * ТИП ПЕРЕВЕЗЕНЬ.Тариф заробітної плати*

Запит щодо заробітної плати водіїв у заданий період.

Табель врахування робочого часу

<i>ПРАЦІВНИКИ</i>	<i>ПОСАДОВІ ТАРИФИ</i>	<i>ОБЛІК РОБОЧОГО ЧАСУ</i>
Табельний №	Посада	Табельний №
Прізвище	Погодинний тариф	Дата
Посада		Робочі години
		Сума заробітної плати

Обчислювальне поле *ОБЛІК РОБОЧОГО ЧАСУ.Сума заробітної плати = ОБЛІК РОБОЧОГО ЧАСУ. Робочі години * ПОСАДОВІ ТАРИФИ. Погодинний тариф*

Запит щодо заробітної плати працівників в заданий період.

Розподіл педнавантаження

<i>ПРАЦІВНИКИ</i>	<i>ДИСЦИПЛІНА</i>	<i>ПЕДНАВАНТАЖЕННЯ</i>
-------------------	-------------------	------------------------

<i>ПРАЦІВНИКИ</i>	<i>ДИСЦИПЛІНА</i>	<i>ПЕДНАВАНТАЖЕННЯ</i>
Табельний №	Код дисципліни	Табельний №
Прізвище	Назва	Код дисципліни
Стать	Годин на тиждень	Кількість тижнів
		Кількість годин

Поле, що обчислюється *ПЕДНАВАНТАЖЕННЯ*.Кількість годин = *ПЕДНАВАНТАЖЕННЯ*.Кількість тижнів * *ДИСЦИПЛІНА*.Годин на тиждень

Запит щодо недозавантажених працівників.

Купівля та продаж товарів

<i>ТОВАР</i>	<i>ОБЛІК КУПІВЛІ</i>	<i>ОБЛІК ПРОДАЖУ</i>
Номенклатурний №	Номенклатурний №	Номенклатурний №
Назва	Дата	Дата
Одиниця виміру	Ціна	Ціна
	Кількість	Кількість
	Сума	Сума

Поля, що обчислюються Сума=Ціна*Кількість

Запит щодо товарів, які були в обліку за певний період.

Кадри

<i>СПЕЦІАЛІСТИ</i>	<i>ПІДРОЗДІЛИ</i>	<i>ШТАТНИЙ РОЗКЛАД</i>
Код спеціаліста	Код підрозділу	Код спеціаліста
Назва	Назва	Код підрозділу
Тарифна ставка		Кількість спеціалістів
		Фонд заробітної плати

Поле, що обчислюється ШТАТНИЙ РОЗКЛАД.Фонд заробітної плати = ШТАТНИЙ РОЗКЛАД.Кількість спеціалістів * СПЕЦІАЛІСТИ.Тарифна ставка

Запит щодо фонду заробітної плати певного підрозділу.

Гуртожиток

<i>КІМНАТИ</i>	<i>МЕШКАНЦІ</i>	<i>ОБЛІК ПЛАТЕЖІВ</i>
----------------	-----------------	-----------------------

<i>КІМНАТИ</i>	<i>МЕШКАНЦІ</i>	<i>ОБЛІК ПЛАТЕЖІВ</i>
№ кімнати	Код	Код
Поверх	Прізвище	Місяць
Кількість місць	Стать	Рік
Вартість проживання	№ кімнати	Сума платежів
		Заборгованість

Поле, що обчислюється ОБЛІК ПЛАТЕЖІВ.Заборгованість = КІМНАТИ.Вартість проживання - ОБЛІК ПЛАТЕЖІВ.Сума платежів

Запит щодо мешканців, що мають заборгованість в заданому періоді.

Керування трудовими ресурсами

<i>СПЕЦІАЛЬНОСТІ</i>	<i>РЕГІОН</i>	<i>ОБЛІК ПРОПОЗИЦІЙ</i>
Код спеціальності	Код регіону	Дата
Назва	Назва	Код спеціальності
		Код регіону
		Кількість пропозицій спеціалістів
		Кількість пропозицій підприємств
		Дефіцит робочих місць

Поле, що обчислюється Дефіцит робочих місць = Кількість пропозицій спеціалістів - Кількість пропозицій підприємств

Запит щодо спеціальностей, що мають дефіцит робочих місць в заданому періоді.

8 Список літератури

1. Вейскас Д. Эффективная работа с Microsoft Access 7.0 для Windows 95/ Перев. с англ. – СПб.: Питер, 1997. – 848 с.
2. Крамм Р. Системы управления базами данных dBASEII для персональных компьютеров. – М.: Финансы и статистика, 1988. – 283 с.
3. Каратыгин С.А., Тихонов А.Ф., Тихонова Л.Н. Работа в Visual FoxPro на примерах: М: БИНОМ, 1995. – 512 с.
4. .В.Попов, Є.С.Вакал, О.В.Обвінцев, В.В.Личман. Розробка учбових баз даних у MS Access. – Київ, 2001. – 33 с.

Навчальне видання

ОБЛІК ІНФОРМАЦІЇ З ВИКОРИСТАННЯМ MS ACCESS

Методичний посібник для студентів механіко-математичного факультету

Упорядники

Попов Валерій Вікторович

Вакал Євген Сергійович

Обвінцев Олександр Вальдемарович

Личман Валерій Васильович