**Лабораторна робота № 4**

**Теоретичні відомості**

При розв‘язуванні прикладних задач доводиться реалізовувати алгоритми оброблення масивів даних, які мають не однорідну структуру (можуть складатися з числових значень величин, індексів-вказівників, тексту та ін.). Найпростіша структура даних – це вектор елементів. У загальному вигляді це дво- або багатовимірні масиви.

***Дані*** – це факти та ідеї, подані у формалізованому вигляді для обробки за допомогою певного алгоритму або для передачі іншим користувачам.

***Структура даних*** – сукупність правил та обмежень, які відображають зв‘язки, що існують між окремими частинами даних. Вона визначається користувачем і залежить від конкретної задачі.

***База даних (БД)*** – сукупність взаємопов‘язаних даних (файлів), призначених для спільного застосування.

***Система управляіння базами даних*** ***(СУБД)*** – комплекс програм, які забезпечують взаємодію користувача з базою даних.

За допомогою СУБД вирішуються такі основні завдання:

* створення бази даних,
* занесення, редагування і вилучення даних,
* упорядковування даних,
* вибір сукупності даних, які відповідають заданим критеріям,
* оформлення вихідних даних та ін.

Сукупність СУБД і бази даних утворює ***банк даних***. Банк даних забезпечує:

* подання інформації у вигляді, адаптованому до різних за підготовкою користувачів,
* несуперечність інформації, що зберігається,
* організацію обмеженого доступу до даних,
* одноразове введення даних і багаторазове їх використання,
* можливість виключення надмірності даних і т.д.

Недоліком такого підходу є велика ємність необхідної пам‘яті ПК.

***Microsoft Access*** – повнофункціональна програма для роботи з реляційними базами даних. Вона дозволяє:

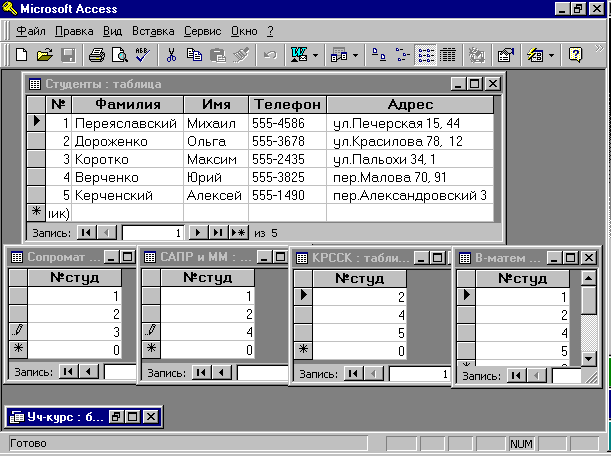
* створювати нові бази даних та працювати зі створеними;
* поповнювати таблиці новими даними, створювати і налагоджувати нові таблиці, форми, звіти;
* автоматизувати роботу з ними, щоб їх могли використовувати люди, які не знайомі з Access.

Якщо стоїть задача працювати з однією таблицею даних, наприклад списком телефонних номерів, то зручніше застосовувати *Microsoft Excel* чи створити текстовий файл, використовуючи *Microsoft Word*. Реляційні бази даних застосовують, якщо дві або більше таблиць з різноманітними характеристиками зв‘язані поміж собою. Можливо зберігати всю цю інформацію в одній таблиці, такій як *Excel*, але вона буде мати великі розміри і вміщувати багаторазово повторювану інформацію, що громіздко і нееффективно. В реляційних базах даних інформація зберігається у вигляді деякого числа взаємопов‘язаних таблиць, що дозволяє позбавитися чи мінімізувати повторювану інформацію. Наприклад, таблиця в *Excel* виглядає так:



*Рис 1. Таблиця в Excel.*

А якщо скористатися Access, то отримаємо декілька таблиць, що вміщують дані про студентів, які навчаються за різними предметами. При цьому обсяг інформації, що вводиться, набагато менший, ніж у попередньому випадку.



*Рис 2. Таблиці в Access.*

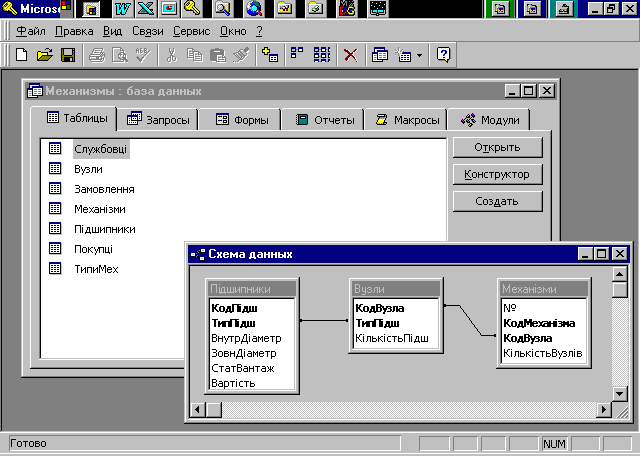
В реляційній базі даних кожний об‘єкт реального світу – *сутність* – записується у вигляді рядка, що називається *кортежем* чи *записом*. Кожний запис складається з одного чи декількох *атрибутів*, що характеризують об‘єкт. Множина записів з однаковим переліком характеристик утворюють *таблицю*. Кожна таблиця має ім‘я, унікальне в межах БД. Вона відображає тип об‘єкта реального світу – сутності. Кожен стовпчик таблиці – це сукупність значень конкретного атрибута об‘єкта, які вибирають з множини всіх можливих значень атрибута об‘єкта, що називається *доменом*.

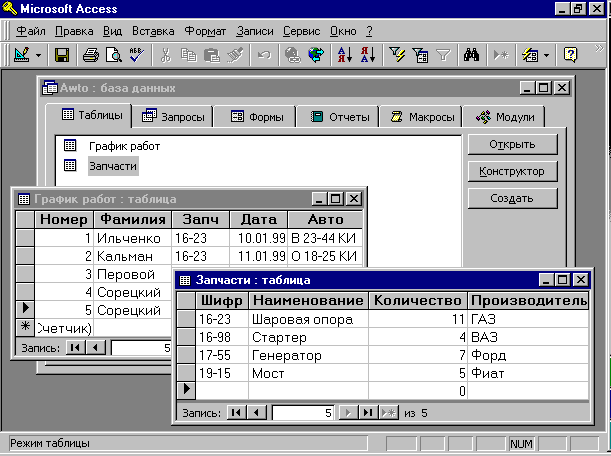
Різні таблиці в одній базі даних можуть мати стовпці з однаковими іменами. Будь-яка таблиця повинна мати, у крайньому разі, один стовпчик.

На відміну від стовпчиків, рядки не мають імен, порядок їх розташування у таблиці не визначений, а кількість не обмежена. Рядок можливо вибрати у таблиці, якщо в ній є хоч би один стовпчик, значення у котрому однозначно ідентифікують кожен рядок, тобто не повторюються. Такий стовпчик (чи декілька стовпчиків) називають *первинним ключем*. У цьому стовпці значення не дублюються, а таблицю, що вміщує такий стовпчик, називають від*ношенням*.

Взаємозв‘язок таблиць є найважливішим елементом реляційної моделі даних. Вона підтримується зо*внішніми ключами*, через які забезпечується взаємозв‘язок між об‘єктами головної та допоміжної таблиць.

У вікні схема даних ми бачимо таблиці, що пов‘язані через *ключові поля*:



**

*Рис 3. Пов‘язані таблиці.*

*Рис 4. Ключові поля в таблицях.*

Будь-яка БД вміщує не тільки файли бази даних, файли з розширенням *.dbf* (dBASE, FoxPro), та *.mdb* (Access), але і ряд інших файлів – керування і журнальних. Для зберігання інформації застосовується без*посереднє керування даними*. Воно здійснюється у зовнішній пам‘яті (на різних фізичних носіях) і може використовуватися не тільки для зберігання інформації, а і для службових цілей. В деяких СУБД для керування даними активно використовуються можливості існуючих файлових систем, в інших – робота проводиться самою СУБД, включаючи рівень пристроїв зовнішньої пам‘яті.

Обробка даних здійснюється в оперативній пам‘яті комп‘ютера. Оскільки часто обсягу оперативної пам‘яті буває не достатньо для зберігання всієї оброблюваної бази даних, то реальним способом покращення роботи з БД є *буферизація даних* в оперативній пам‘яті.

При роботі з базами даних говорять про змінення їх стану, – тобто управління *транзакціями*. Під *транзакцією* розуміють послідовність операцій над БД, що розглядаються СУБД як єдине ціле. Поняття транзакції необхідно для підтримки логічної цілісності БД (під логічною цілісністю БД ми розуміємо непротирічність даних про об‘єкт в різних складових частинах БД). Транзакція або успішно виконується, і СУБД фіксує змінення БД, або жодна з цих змін ніяк не відображається на стані БД. Важлива властивість, що кожна транзакція починається при цілісному стані БД і залишає цей стан цілісним після свого завершення, робить дуже зручним використання транзакцій як одиниці активності користувача по відношенню до БД.

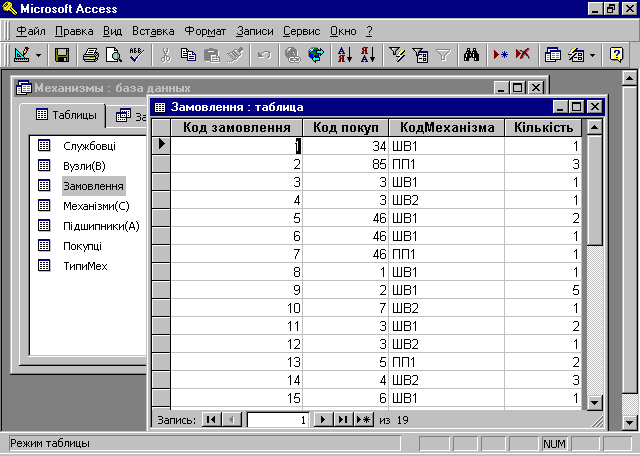
Однією з основних вимог до СУБД є *надійність зберігання даних* у зовнішній пам‘яті, під якою розуміють здатність відновлювати останній стан БД після будь-якого апаратного чи програмного збою. Зазвичай розглядають два види апаратних збоїв: так звані *м’які збої*, котрі можливо трактувати як раптове припинення роботи комп’ютера (наприклад, аварійне виключення живлення), і *жорсткі збої*, які характеризуються втратою інформації на носіях зовнішньої пам’яті.

Для захисту інформації від збоїв використовується *журнал*. Журнал – це особлива частина БД, що недоступна користувачам СУБД і підтирмується з особливою ретельністю ядром СУБД (іноді підтримуються дві копії журналу, що розташовуються на різних фізичних дисках), в яку поступають записи про всі зміни БД. При цьому реалізується стратегія попереднього запису в журнал (так званого протокола Write Ahead Log – WAL), який полягає в тому, що запис про змінення будь-якого об’єкту БД повинен попасти у зовнішню пам’ять журналу раніше, чим змінений об’єкт попадає у зовнішню пам’ять основної частини СУБД.

**Структура *Microsoft Access*.**

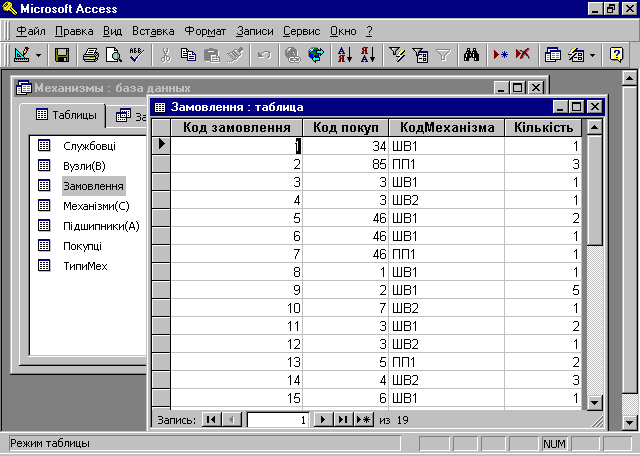
При запуску *Microsoft Access 97* користувач у початковому вікні може створити нову базу даних чи відкрити вже існуючу (для цього вибирають ім’я БД в нижній частині вікна чи використовують в меню *Файл* команду *Открыть*). Після цього на екрані з’являється вікно БД (див. рис. 3). В ньому розташовані всі частини бази даних. Це *таблицы*, *запросы*, *формы*, *отчеты*, *макросы* и *модули*. Частини бази даних працюють не по одинці, а всі разом: дані зберігаються в *таблицях* і можуть вводитися в таблиці безпосередньо чи з використанням *форм*; для вибору необхідних даних з однієї чи декількох пов’язаних таблиць застосовуються *запити*; форми і звіти грунтуються на запитах, які дозволяють об’єднувати як у формах, так і у звітах дані з декількох таблиць.

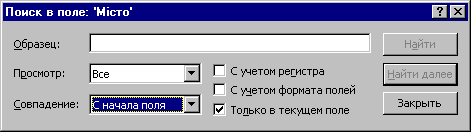
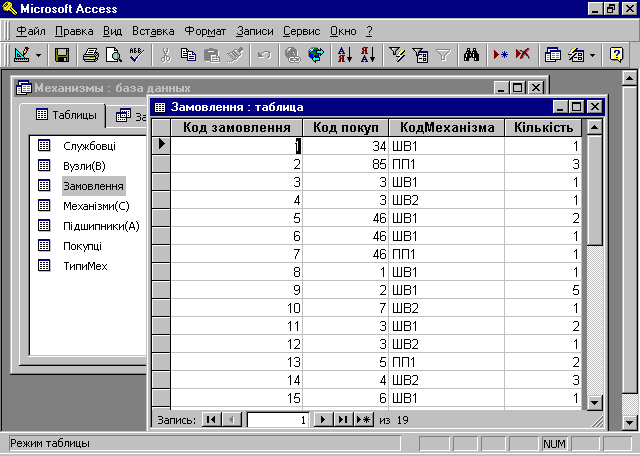
***Робота з таблицями.***

Таблиця є основною структурною одиницею накопичення і зберігання даних. Вона складається з рядків (записів) та стовпців (полів). Кожна таблиця вміщує інформацію на певну тему. Якщо відкрити вікно достатньо великої таблиці, побачимо, що вона не повністю вміщується в ньому, для переміщення по таблиці до потрібних комірок використовуються лінійки прокрутки. У нижній частині вікна таблиці розміщуються кнопки , що дозволяють знаходити записи по їх порядковому номеру (тобто по номеру внесення запису в таблицю). Кнопка | - позначає перехід до першого запису; кнопка - перехід до поперед-нього запису; кнопка - перехід до наступного запису; кнопка | - перехід до останнього запису; кнопка \* - створити новий запис.

Сірий квадратик перед кожним записом дозволяє виділити його. Виділений запис можливо скопіювати, перемістити чи видалити. В ньому при внесенні нового запису в таблицю виникає зображення олівця, – це означає, що запис ще не збережений в БД. По закінченні внесення запису відбувається перехід до наступного рядка, і тільки що внесений запис запам’ятовується.

При внесенні записів в таблицю для переміщення між стовпчиками використовують клавішу ***Tab***. Якщо потрібно скопіювати вміст сусідньої зверху комірки використовують сполучення клавіш **CTRL**+**’**(апостроф). При внесенні даних в таблицю можуть виникати помилки, для відміни щойно внесених змін використовують в меню (верхній рядок вікна Access) *Правка* команду *Восстановить запись*.

Пошук і організація даних в таблицях та запитах полегшується командами сортування по стовпцям, що вміщують потрібні характеристики. Сортування може проводитися по зростанню (А-Я) та у зворотному напрямку (Я-А). Для того, щоб відсортувати дані у стовпці, потрібно встановити в ньому курсор й натиснути одну з кнопок сортування. 

Пошук даних можливо проводити, знаючи лише початок поля, тобто декілька перших літер, для цього використовують кнопку із зображенням бинокля на панелі інструментів.  У вікні “Поиск в поле” в комірці *Совпадение* вибирають з розгорнутого списку “*С начала поля”*, і вказують границі пошуку, встановивши прапорець біля напису “*Только в текущем поле”*.

Для пошуку групи записів, що задовільняють загальній умові відбору, може використовуватися фільтр. Фільтрувати можна таблиці, запити і форми. Найшвидшим способом вибору є “*Фильтр по выделенному”* на панелі інструментів. Виділяють ключове слово для відбору і використовують фільтр. Для відміни фильтрації натискають кнопку *Удалить фильтр*. Для реалізації складних умов фільтрації використовують кнопку “*Изменить фильтр”* , де на вкладці “*Найти”* вписують в потрібному стовпчику умову відбору і на вкладці “*Или”* наступну умову відбору. Для очистки попередніх результатів відбору використовують кнопку “*Очистить бла*нк”. Таким чином можна відфільтрувати дані по умові «і» та «чи», по початковим літерам запису, по кінцевим літерам запису, по будь-якій частині текстової інформації.

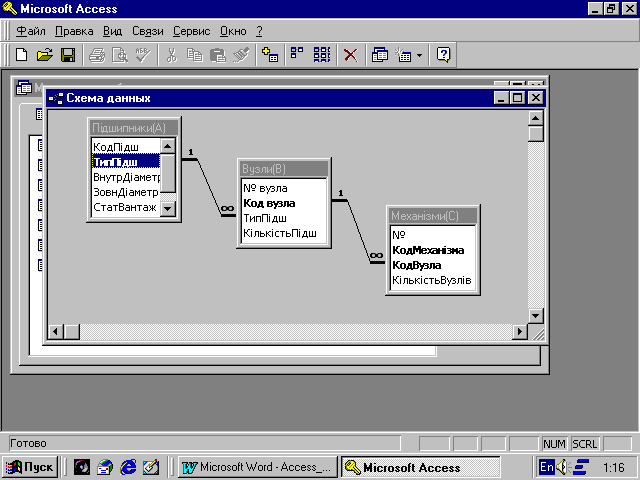
При необхідності таблиці однакової структури можна об’єднати. Для цього створюється третя таблиця, в неї копіюється перша, використовуючи буфер обміну, (Копировать-Вставить), при цьому в “*Параметрах вставки”* встановлюють перемикач “*Структура и данные”*, і друга таблиця – також через буфер обміну, але в *“Параметрах вставки”* встановлюють перемикач *“Добавление данных в таблицу”*.

Для того, щоб використати дані з іншого файла без їх копіювання, створюють *зв’язок* з даною таблицею. Переваги такого підходу полягають в тому, що ви завжди маєте доступ до інформації в її поточному стані; копія таблиці не може вміщувати поточної інформації. Наприклад, потрібно створити зв’язок з таблицею Excel. В меню Ф*айл* вибирають команду *“Внешние данны*е*”* - “*Связь с таблицами”*. У вікні діалогу ***Зв’язок*** встановлюємо *“Тип файлов”* Microsoft Excel; знаходимо папку, що вміщує файл з яким потрібно створити зв’язок; двічі натискаємо на цьому файлі – майстер зв’язку створить такий зв’язок. Ми отримаємо зв’язану таблицю, дані в якій будуть оперативно змінюватись по мірі змінення даних в таблиці Excel.

Для полегшення введення даних в таблиці можна створювати поля підстановки. Для створення поля підстановки таблицю відкривають у режимі конструктора. Далі по крокам виконують наступні дії:

1. Вводять ім’я поля у вільній комірці стовпця “*Имя поля”*. Нажимають клавішу ***Tab*** для переміщення у наступну комірку рядка.
2. У стовпчику “*Тип данных”* натискають на стрілці вниз поряд з виділеним написом *Текстовый*, вибирають у відкритому списку *Мастер подстановок*.
3. На екрані з’являється вікно першого кроку роботи *Мастера подстановок*. Вибираємо у ньому *«столбец подстановки исполь-зует значения из таблицы или запроса»*, нажимаемо кнопку *Далее*.
4. На другому кроці роботи в списку таблиць вибираємо ім’я таблиці, з якої буде проводитися підстановка, нажимаємо кнопку *Далее*.
5. На третьому кроці роботи зі списку *Доступных полей* вибираємо поле для підстановки, нажимаємо кнопку *Далее*.
6. У вікні четвертого кроку роботи встановлюємо ширину стовпця, натиснувши двічі на правій границі заголовку стовпця, нажимаємо кнопку *Далее*.
7. У наступному вікні вводимо нову назву стовпця підстановки і нажимаємо кнопку *Готово*.
8. У вікні діалогу, що пропонує зберегти таблицю, нажимаємо *Да*.

Поле підстановки створено. Тепер у стовпчику *Описание* режима конструктора таблиці можна ввести текст, що визначає, яку інформацію потрібно ввести в поле підстановки. Він буде відображатися в таблиці в режимі таблиці в нижньому рядку вікна.



***Зв’язки між таблицями***. Зв’язки між таблицями створюються у вікні “*Схема даних*”. Воно відкривається ідноіменною кнопкою на панелі інструментів. На кінцях створених зв’язків виникає 1 чи ∞, що означає відношення до одного чи до багатьох. Поля, по яким встановлюються зв’язки, повинні мати однаковий тип даних, перше з них (де відношення “до одного”) повинно бути ключовим. Іноді виникає потреба розбити одну таблицю на декілька, тому що деякі відомості використовуються не надто часто, або певна частина даних не призначення для загального доступу. У цьому випадку ми створюємо окрему таблицю, яка має той же первинний ключ, що і основна таблиця. У такому випадку між записами в таблиці встановлюється зв’язок “*один-до-одного*”, тобто один запис однієї таблиці пов’язаний лише з одним записом іншої таблиці. У випадку, коли утворюють зв’язок між одним записом таблиці 1 і багатьма записами таблиці 2, виникає зв’язок типу “*один-до-багатьох* ”. Наприклад, коли треба пов’язати запис у таблиці 1, що вказує назву і характеристики велосипеда, з переліком деталей, з яких він зібраний, у таблиці 2. На малюнку зображені зв’язки між таблицями “*один-до-багатьох*” (1-∞). Коли виникає потреба створити зв’язок “*багато-до-багатьох*”, розбивають таблиці так, щоб створити два зв’язки “багато-до-одного ” та “один-до-багатьох”. Тільки так в MS Access можна реалізувати цей зв’язок.

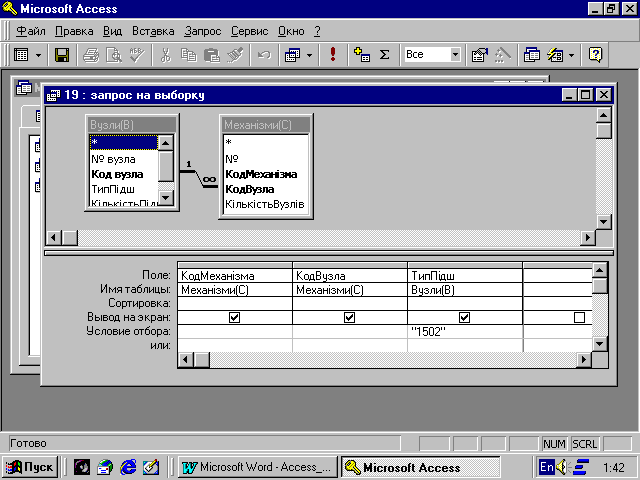
**Помічник.** Якщо вам знадобиться допомога, скористайтеся підказкою Microsoft Access, вибравши в меню **?** команду *Вызов справки*, чи *Помощником* (). В переліку оберіть розділ *Полный список разделов*, перейдіть на вкладку *Предметный указатель* і у вікні *Введите первые буквы нужного слова* введіть слово *таблица*. В нижньому вікні ви побачите список тем, що вміщують слово *Таблица*. Якщо вибрати тему *Таблицы, общие сведения* і натиснути кнопку *Ввести*, то з’явиться вікно довідки, в якому виберемо *Общие сведения о таблицах*. Далі побачимо текст довідки. Файл довідки складається з трьох частин, на це вказують цифри у лівому верхньому куті вікна.

*Таблиця 1. Властивості полів.*

|  |  |
| --- | --- |
| Властивість | Функція поля |
| Ключове поле | Поле, що однозначно ідентифікує запис. Таблиця може вміщувати декілька ключових полів, які працюють разом. Таблиця не може вміщувати два однакові значення у ключовому полі. Ключові поля позначаються значком ключа на початку поля у сірому квадраті (в режимі конструктора). |
| Індексоване поле | Індекс – це засіб MS Access, що підтримує зберігання відсортованих даних по даному полю записів. Він прискорює пошук і сортування, але уповільнює роботу БД. Зазвичай індекс необхідний тільки для ключових полів. |
| Тип даних | Визначає, які дані зберігаються у полі (текст, число, так/ні, дата, час і т.і.) |
| Формат поля | Визначає, як будуть відображатися значення поля (у грошовому форматі, як проценти, у науковому форматі чи ін.) |
| Маска вводу | Зразок для даних, що вводяться у поле (підходить для використання в полях, що вміщують телефонні номери, поштові індекси, дати, номери документів) |
| Підпис | Визначає, як буде називатися поле у формах, таблицях звітах. |

***Створення запитів.*** При роботі з таблицями можливо виконувати різні операції з даними – перегляд, сортування, фільтрація, оновлення і друк даних. Але досить часто виникає потреба проводити обчислення, передивлятися дані з декількох таблиць – ці операції реалізуються за допомогою *запитів*. Запит – це вираз, у якому вказуються таблиці з необхідними стовпцями та умовами відбору даних в них.

При виконанні запиту на вибірку даних Microsoft Access створює набір записів, що відповідають умові відбору. В більшості випадків з цим набором записів можна працювати, як зі звичайною таблицею. Але, на відміну від реальної таблиці, набір записів фізично не існує, Access створює його тільки під час виконання запиту.

Будь-який запит в Microsoft Access реалізується за допомогою майстра запитів чи мови Visual Basic та мови запитів SQL. Зокрема створення підпорядкованих запитів, результати яких використовуються в якості умов порівняння в інших запитах, можливо в Access тільки за допомогою SQL.

При створенні запитів інструкція SELECT є ядром мови SQL. Вона використовується для відбору рядків і стовпців і вміщує п‘ять основних речень (дужками [ ] обмежені необов‘язкові речення):

SELECT <список\_полів>

FROM <список\_таблиць>

[WHERE <специфікація\_відбору\_рядків>]

[GROUP BY <специфікація\_групування>]

[HAVING <специфікація\_відбору\_груп>]

[ORDER BY <специфікація\_сортування>]

Перед створенням запиту всі таблиці, що використовуються в запиті повинні бути зв’язані між собою. При створенні запитів слід чітко виписати правила відбору даних в кожній з таблиць.

*Таблиця 2. Умови відбору.*

|  |  |  |
| --- | --- | --- |
| Оператор | Функція | Приклади |
| And | Записи, що задовольняють першу та другу умови одночасно | like \*сч\* *and* like \*де\* -- умова, що вміщує текст *сч* та *де* |
| Or | Записи, що задовільняють хоча б одній з двох умов | лічильник *оr* дешифратор – умова, яка виконується при наявності у записі слова *лічильник* чи *дешиф-ратор* |
| Not | Записи, що не задовільняють дану умову | *not* счетчик – будь-яке значення, що відмінне від значення *счётчик* |
| & | Злиття двох рядкових виразів | [Місто]&»,»&[Країна]&»\_«&[Ін-декс] – об’єднує поля *Місто*, *Країна*, *Індекс*, розділяючи перші два атрибути комою, і другі – пробілом |
| Like | Записи, що вміщують даний вираз | *like* сч\* – значення, що починаються з *сч*, *like* \*сон – значення, що закінчуються на *сон*, *like* [т-ф]\* – значення, що починаються на одну з літер “т”, ”у”, ”ф”, *like* \*шиф\* —значення, що вміщують *шиф* |
| Between… And | Записи, що мають значення поміж двома вказаними | *Between* 01.01.99 *аnd* 10.01.99 – дати між 1 та 10 лютого 1999 г. |
| In | Записи, що мають в якості значення один з елементів даного списку | in (прямолінійні, обертальні) – значення, що вміщують *прямолінійні* чи *обертальні* |
| Is Null | Записи, що не мають значення в даному полі | *is nul* – в деяких полях значення не вводяться, (наприклад в полі *Телефон*) |
| Is Not Null | Записи, що мають значення в даному полі | *Is not nul* – в деяких полях значення вводяться, (например в поле *Телефон*) |
| =, <>, >, < | Позначення відношень дорівнює, не дорівнює, більше, менше | = 42 (значення дорівнює 42), <>3 (значення не дорівнює 3), >01.01.99 (дата після 1 счня 1999 г.), <3,7 (значення, менше, чим 3,7) |
| \*, /, +, – | Множення, ділення, додавання, віднімання | [Ціна]\*[Кількість] (добуток значен-ня поля *Ціна* на значення поля *Кількість*), [Вага]/12 – результат від ділення значення поля *Вага* на 12 |

**Створення звітів.** Для представлення результатів роботи БД найкращим засобом є звіти. Вони формуються на основі багатотабличних запитів чи таблиць. Звіти в порівнянні з іншими засобами виведення інформації мають певні переваги:

* у звітах передбачені можливості для групування та обчислення проміжних і загальних підсумків для великих наборів даних,
* звіти можуть створюватися у формі бланків рахунків, замовлень, поштових наклейок, матеріалів для презентацій та інших документів для зручного ведення справ,
* звіти дозволяють керувати виведенням повторюваних значень.

Звіти можуть бути простими і складними. У складних звітах до основного звіту є можливість додавати підпорядковані звіти чи підпорядковані форми. Наприклад підпорядковані звіти можуть бути використані при підрахунку поквартальних підсумків і потім впроваджені у звіт за рік.