

Лекція № 6

Тема: Методика навчання роботі з електронними таблицями. Методика навчання, створення та використання баз даних, інформаційно-пошукових систем.

Питання

1. Методичні рекомендації щодо викладення теоретичного матеріалу розділу «Методика навчання роботи з електронними таблицями».
2. Рекомендації по організації практичної роботи з електронними таблицями.
3. Призначення інформаційних систем и баз даних (БД). Класифікація БД.
4. Рекомендації по організації практичної роботи з базами даних.

Методика навчання роботи з електронними таблицями.

Основні цілі. Познайомити учнів із призначенням і структурою електронної таблиці. Навчити основним прийомам роботи з табличним процесором. Навчити організації простих табличних розрахунків за допомогою електронних таблиць.

Досліджувані питання даного розділу.

- ☞ Призначення і структура електронної таблиці (ЕТ).
- ☞ Табличний процесор: середовище, режими роботи.
- ☞ Інформація в комірках таблиці.
- ☞ Правила запису формул.
- ☞ Принцип відносної адресації.
- ☞ Організація табличних обчислень.
- ☞ Сортування даних у ЕТ.
- ☞ Графічна обробка даних.
- ☞ Абсолютна адресація.
- ☞ Математичні, логічні, статистичні функції.

Згідно з шкільною програмою з інформатики в 9-12 класах відводиться годин на вивчення даної теми:

10 кл. – 8 год.;

11 кл. – 5 год.

Предметом вивчення даного розділу є *електронні таблиці. (ЕТ)*. Поява електронних таблиць історично збігається з початком поширення персональних комп'ютерів. Перша програма для роботи з електронними таблицями — табличний процесор, була створена в 1979 році, призначалася для комп'ютерів типу Apple II і називалася VisiCalc. У 1982 році з'являється знаменитий табличний процесор Lotus 1-2-3, призначений для IBM PC. Lotus поєднував у собі обчислювальні можливості електронних таблиць, ділову графіку і функції реляційної СУБД. Популярність табличних процесорів росла дуже швидко. З'являлися нові програмні продукти цього класу: Multiplan, Quattro Pro, SuperCalc і інші. Одним із самих популярних табличних процесорів сьогодні є MS Excel, що входить до складу пакета Microsoft Office.

Методичні поради:

1. Використання індуктивного методу.

Пояснення основних можливостей та принципів опрацювання ЕТ будуються від конкретних прикладів до узагальнень у вигляді правил-алгоритмів, які не залежать від конкретної програми та знань користувача.

2. Мотиваційне навчання.

Кожна вказівка спочатку демонструється на прикладах, пояснюється мета та необхідність введення.

3. Навчання через систему доцільних задач.

Для кожної вказівки добирається система завдань з практично значущим і зрозумілим змістом для відповідної вікової групи учнів. Учні знайомляться з умовою завдання та самостійно шукають потрібні вказівки для одержання результатів. Застосовується дидактичний принцип – навчання за зразком – орієнтовною основою дій.

4. *Наочні орієнтири* для контролю за результатами виконання завдання.
 5. *Використання завдань двох видів*:
 - введення вхідних даних, та опрацювання в середовищі табличного процесора;
 - опрацювання наперед введених табличних даних та аналіз одержаних результатів.
 6. *Використання навчальної допомоги* різного рівня при використанні завдання для самостійного виконання.
 7. *Проведення лабораторних робіт* комплексного характеру для закріплення знань та вмій учнів в нових умовах.
 8. *Використання методу проектів* для поглибленого вивчення можливостей електронних таблиць.
 9. Ефективний прийом – *аналогія та порівняння*, наприклад з середовищем текстового редактора, з структурою оперативної пам'яті.
- Завдання на самостійне знаходження спільних рис та відмінностей активізують розумову діяльність учнів.

Що ж таке електронна таблиця? Це засіб інформаційних технологій, що дозволяє вирішувати цілий комплекс задач:

- 1) *насамперед, виконання обчислень*. Здавна багато розрахунків виконуються в табличній формі, особливо в області діловодства: численні розрахункові відомості, кошториси витрат і т.п. Крім того, рішення чисельними методами цілого ряду математичних задач зручно виконувати в табличній формі. Електронні таблиці являють собою зручний інструмент для автоматизації таких обчислень. Рішення багатьох обчислювальних задач на ЕОМ, що раніш можна було здійснити тільки шляхом програмування, стало можливо реалізувати на електронних таблицях;
- 2) *математичне моделювання*. Використання математичних формул у ЕТ дозволяє представити взаємозв'язок між різними параметрами деякої реальної системи. Основна властивість ЕТ — миттєве перерахування формул при зміні значень вхідних в них операндів. Завдяки цій властивості, таблиця являє собою зручний інструмент для організації чисельного експерименту: підбор параметрів, прогноз поведінки системи, що модулюється, аналіз залежностей, планування. Додаткові зручності для моделювання дає можливість графічного представлення даних;
- 3) *використання електронної таблиці як бази даних*. Звичайно, у порівнянні із СУБД електронні таблиці мають менші можливості в цій області. Однак деякі операції маніпулювання даними, властиві реляційним СУБД, у них реалізовані. Це пошук даних за заданих умов і сортування інформації.

Пояснення нового матеріалу зручно будувати на конкретному прикладі. Наприклад, таблиця обліку продажу молочних продуктів у магазині. Розглядаючи таку таблицю як базу даних, її поля можна розділити на два типи: незалежні, тобто ті, що утримують деяку вхідну інформацію й обчислюються по визначених формулах. До незалежних полів відносяться ціна одиниці товару, кількість поставленого товару, кількість проданого товару. До полів, що обчислюються, відносяться залишок, рівний різниці між поставленим і проданим товаром, виторг від продажу, рівний добуткові кількості проданого товару на ціну. *Основною властивістю електронної таблиці є можливість миттєвого перерахування полів, що обчислюються, при зміні значень операндів, що входять у формули.*

Можливість розміщення в комірках таблиці формули — це перша базова ідея електронних таблиць. Друга базова ідея — це принцип відносної адресації.

На уроках учні мають освоїти конкретний табличний процесор. Як і в попередніх темах курсу, що розглядають інформаційні технології, рекомендується дотримуватися методичної схеми віртуального виконавця, елементами якої є вивчення середовища, режимів роботи, системи команд, даних.

Методичні рекомендації з викладу теоретичного матеріалу

При роботі з табличним процесором (ТП) створюється документ, що називається *електронною таблицею*. Електронна таблиця формується в оперативній пам'яті комп'ютера. Надалі її можна переглядати, змінювати, записувати на магнітний диск для збереження, друкувати на принтері.

Середовище ТП.

Всі відомості, що стосуються середовища табличного процесора, режимів роботи, системи команд обов'язково супроводжуються демонстрацією на простих прикладах.

При роботі з табличним процесором на екран виводиться робоче поле таблиці і панель діалогу. Електронна таблиця представляється у вигляді матриці, що складається з рядків і стовпців. Рядки нумеруються зверху вниз, починаючи від 1. Стовпці іменуються латинськими буквами (одно-і дволітерними іменами) за абеткою в напрямку з ліва на право. Число рядків і стовпців залежить від конкретного типу ТП.

На перетинанні рядків і стовпців утворюються комірки (клітки), кожна з яких має своє позначення (ім'я, адреса), що складається з імені стовпця і номера рядка: А1, С5, АВ356 і т.п. На екрані дисплея видна не вся електронна таблиця (документ), а тільки її частина. Документ у повному обсязі зберігається в оперативній пам'яті, а екран можна вважати «вікном», через яке користувач має можливість переглядати його. Крім основного розділу пам'яті, де зберігається електронна таблиця, використовуються ще додаткові розділи: буфер для збереження фрагментів таблиці, що були скопійованими; розділ пам'яті для розміщення довідкової інформації.

Важливим елементом електронної таблиці є *табличний курсор* — прямокутник, виділений кольором або рамкою. Комірка таблиці, в якій у даний момент часу знаходиться курсор, називається *поточною коміркою*. При переміщенні курсору по таблиці відбувається переміщення «вікна» по документу, в результаті чого стають видимими різні його частини. Частина таблиці, заповнена інформацією, називається *активною таблицею*.

У різних табличних процесорів може розрізнятися розташування на екрані робочого поля і панелі діалогу. Наприклад, у середовища ТП SuperCalc, що працює в операційній системі MS DOS, на панелі діалогу меню команд постійно не відбивається і викликається натисканням клавіші «\» — слеш. Для табличних процесорів, що працюють в ОС Windows (у т.ч. Excel), властива наявність панелі інструментів, лінійок прокручування.

Рядок введення призначений для відображення даних, що вводяться в поточну комірку. У цьому рядку можна переглянути і відредагувати збережену в комірці формулу; у самій комірці користувач бачить результат обчислення за формулою.

Головне меню містить основні команди керування ЕТ; являє собою ієрархічну систему команд. Команди, що викликають відкриття підменю, можна назвати режимами. Команди, що виконуються, ініціюють визначені дії над електронною таблицею.

Рядок підказки використовується для виведення повідомлень, що підказують користувачеві можливі дії при даному стані таблиці.

Допоміжна область керування містить у собі панель інструментів, лінійки прокручування, рядок стану.

Основні режими роботи

Режим готовності. У цьому режимі відбувається вибір поточної комірки або виділення блоку комірок.

Режим введення даних. Відбувається посимвольне введення даних із клавіатури в поточну комірку.

Режим редагування. Використовується при необхідності відредагувати вміст комірки без повної її зміни.

Командний режим. Режим вибору і виконання команд з ієрархічної системи меню. Після виконання команди відбувається повернення до режиму готовності.

Крім перерахованих основних режимів роботи ТП можна говорити про режими відображення таблиці і режимах керування обчисленнями.

Режими відображення таблиці. В комірках, які зберігають формули, можуть відображатися результати обчислення за формулами або самі формули. Перший режим називається *режимом відображення значень*, другий — *режимом відображення формул*. Робочим станом таблиці є режим відображення значень. Режим відображення формул може використовуватися при формуванні і налагодженні таблиці.

Режим керування обчисленнями. Табличний процесор робить обчислення за формулами, скануючи таблицю у визначеному порядку. Таке сканування завжди починається з клітки А1.

Порядок обчислень може бути встановлений *по рядках* або *по стовпцях*. Деякі ТП дозволяють встановлювати цей порядок за бажанням користувача.

При кожному введенні нових даних в комірку вся таблиця автоматично перераховується заново (режим *автоматичного перерахування*). У деяких ТП існує можливість установки *режиму ручного перерахування*, тобто таблиця заново перераховується тільки після подачі спеціальної команди.

Система команд.

Команди ТП організовані в ієрархічну систему, верхнім рівнем якої є головне меню. Крім того, виконання команд може ініціюватися через панель інструментів, контекстне меню, «гарячі клавіші».

Команди редагування таблиці дозволяють маніпулювати з фрагментами таблиці: *видаляти, копіювати, переміщати, вставляти*. Вставки і видалення стовпців або рядків приводять до зрушення інших рядків або стовпців таблиці. При цьому діюча в таблиці відносна адресація автоматично модифікує формули відповідно до їх адрес, що змінилися. Прийом копіювання дозволяє швидко будувати великі таблиці, що містять однотипні елементи.

Команди форматування дозволяють змінювати зовнішній вигляд таблиці, її оформлення. До елементів формату відносяться:

- ☞ напрямки вирівнювання даних щодо границь комірки;
- ☞ висота рядка і ширина стовпця;
- ☞ тип, накреслення і розмір шрифту;
- ☞ формат представлення чисел (звичайний, експонентний, розрядність);
- ☞ вид розлінування таблиці; колір фону та ін.

В електронній таблиці діє деякий набір стандартних параметрів формату «за замовчуванням». Командами форматування його можна змінювати як стосовно всієї таблиці, так і в окремих її фрагментах.

Команди роботи з файлами містять у собі стандартний набір команд, що дозволяють відкривати і зберігати файли, організовувати виведення до друку отриманого документа.

Команди роботи з таблицею як з базою даних. Здатність ТП шукати і вибирати дані з таблиці дозволяє використовувати електронну таблицю як нескладну базу даних. При роботі з базами даних мають справу з *записами і полями*. В електронних таблицях базою даних є сама таблиця, записами — рядки таблиці, полями — клітки таблиці. У ТП реалізовані команди *пошуку і сортування*.

Щоб організувати *пошук і витяг даних*, необхідно задати:

- ☞ *вхідний блок*, тобто діапазон комірок, у яких зберігаються дані (записи і поля); важлива вимога: усі рядки в цьому блоці повинні бути однорідні;
- ☞ *блок критеріїв*, тобто діапазон кліток, що містять умову, відповідно до якої здійснюється пошук і вибірка даних із вхідного блоку;
- ☞ *вихідний блок*, тобто діапазон кліток, в який будуть витягнуті дані з вхідного блоку, відповідно до умови, яка міститься в блоці критеріїв. Завдання цих блоків здійснюється спеціальними командами.

Сортування рядків таблиці виконується за значеннями визначеного стовпця. У команді вказується порядок сортування: по зростанню або спаданню значень (у тім же змісті, що й у БД).

Команди графічної обробки даних дають можливість відображати числову інформацію в графічному вигляді, найчастіше — у виді діаграм. Команди графічного режиму можна розбити на дві групи:

- ☞ команди опису діаграм (задають дані, що будуть виведені в графічному вигляді, задають тип діаграм і т.д.);
- ☞ команди виведення діаграм.

Дані в комірках таблиці. Дані для табличних процесорів — це інформація, що утримується в комірках таблиці, представлена у визначеній символічній формі.

Тут ми знову зустрічаємося з такими властивостями даних, як тип і структура, з поняттями константи і змінної, арифметичного і логічного виразу, поняттям адресації.

Вмістом комірки електронної таблиці може бути формула або текст. Частковим випадком формули є числова константа або змінна. Більш загальним — арифметичний або логічний вираз.

Типи даних.

Табличний процесор повинен «знати», якого типу дані зберігаються в конкретній комірці таблиці для того, щоб правильно інтерпретувати її вміст. Тип даних визначається безліччю значень, які може приймати величина і сукупністю операцій, застосованих до величин цього типу. Звідси, наприклад, впливає, що не можна застосовувати арифметичні операції до комірок таблиці, в яких зберігається текстова інформація. Основний набір типів даних в електронних таблицях: числа, символічна, логічна, дата, грошова та ін.

Структури даних. Мінімальним структурним елементом даних, представлених в електронній таблиці, є комірка. Основну роботу виконують в комірках: їх заповнюють, редагують, очищають.

Комірки поєднуються в структури даних — стовпці і рядки. Табличні процесори дозволяють оперувати з рядками або стовпцями як єдиним цілим. Наприклад, можна видаляти або вставляти рядки (стовпці), міняти їх місцями.

Базовим структурним поняттям в електронних таблицях є поняття *діапазону комірок (блоку)*. Воно використовується в багатьох командах табличних процесорів і в деяких функціях. Діапазон — це безліч комірок, що утворюють у таблиці область прямокутної форми (матрицю). Мінімальний діапазон — це комірка, рядок і стовець також є блоком, максимальний діапазон — вся таблиця. Деякі табличні процесори дозволяють задавати ім'я для діапазону кліток, що дає можливість працювати з блоком як єдиним цілим.

Числові константи розділяються на цілі і дійсні. Дійсні константи можна записувати двома способами: у формі з фіксованою крапкою й в експонентній формі (у формі з крапкою, що плаває).

Запис числової константи з фіксованою крапкою припускає, що число містить цілу і дробову частину, розділені десятковою крапкою. Наприклад, числова константа -3,1415 записується, як -3.1415 (у ТП Excel може вживатися кома). При записі числової константи в експонентній формі спочатку записується мантиса, потім — латинська буква E (прописна або рядкова), після неї — порядок. Мантиса може бути записана, як ціла константа або константа з фіксованою крапкою, а порядок — тільки як ціла двозначна константа. Числова константа в експонентній формі трактується, як мантиса, помножена на 10 у ступені, рівній порядку. Наприклад, числа 0,0001 і 1 000 000 можуть бути записані в такий спосіб: 1E-4 або 1E+6.

Змінні. Кожну комірку таблиці будемо інтерпретувати як комірку пам'яті табличного процесора. Кожна комірка має своє ім'я. У кожній комірці може зберігатися інформація того або іншого вигляду. Тут простежується пряма аналогія з поняттям змінної в мовах програмування. Змінна — це поійменоване місце в пам'яті (комірка), куди можна записати значення. Змінна приймає різні значення визначеного типу. Кожна змінна позначається символічним ім'ям (ідентифікатором). Комірку таблиці можна розглядати як змінну. Отже, A1, C5, G10 та ін. — імена змінних.

Вирази (формули). В електронних таблицях використовуються два види виразів: арифметичні і логічні. Вираз, що визначає спосіб обчислення деякого числового значення за математичною формулою, називають *арифметичним виразом*. Існують визначені правила запису арифметичних виразів. Ці правила аналогічні тим, що використовуються в мовах програмування.

Вирази складаються з констант, змінних, знаків операцій, функцій, круглих дужок. У ТП SuperCalc *SQRT* — функція обчислення квадратного кореня; *SUM* — функція підсумовування. У Excel функція квадратного кореня — *КОРІНЬ*, функція підсумовування — *СУМ*. Порядок обчислення виразів відбувається відповідно до пріоритетів виконання арифметичних операцій, з розміщенням дужок. Засвоєння учнями навичок запису формул вимагає практичних вправ.

Логічні вирази (логічні формули) будуються за допомогою операцій відношення (<, >, =, <=, >=,) і логічних операцій (логічне «І», логічне «АБО», логічне заперечення «НІ»). Результатом обчислення логічного виразу є логічні величини «істина» або «неправда». З логічними виразами учні вже зустрічалися при вивченні розділу «Програмування».

За допомогою логічних виразів, зокрема, задаються умови, перевірку яких здійснює умовна функція. Форма запису умовної функції значною мірою залежить від типу табличного процесора. Якщо в клітку заноситься умовна функція, то на екрані відображається результат її обчислення, тобто те або інше значення в залежності від умови, яка задається логічним виразом. Звичайно умовна функція має таку структуру:

ЯКЩО (умова; дія_1; дія_2).

Якщо умова істинна, то виконується дія_1, інакше — дія_2. Умовна функція може мати вкладену структуру. Використовуючи її, можна будувати найпростіші алгоритми розгалуження, наприклад, обчислення розривних функцій.

Адресація. Варто звернути увагу учнів на визначену спорідненість структури електронної таблиці й оперативної пам'яті ЕОМ. В обох випадках використовується принцип адресації для збереження і пошуку інформації. Різниця полягає в тому, що в ОЗУ найменшою одиницею, що адресується є байт, а в таблиці — клітка (комірка).

Символічні імена змінних ϵ , у той же час, їх адресами в таблиці. В таблиці може бути встановлений режим відносної адресації або режим абсолютної адресації. У режимі відносної адресації зміни в місці розташування формули шляхом копіювання блоку, переносу блоку, вставки або видалення рядків або стовпців приводять до автоматичної зміни адрес змінних у формулах, що знаходяться в зміщених комірках. Інакше кажучи, формули модифікуються у відповідності зі своїм новим положенням.

При скасуванні режиму відносної адресації встановлюється режим абсолютної адресації. У цьому випадку при зсуві кліток модифікації формул не відбувається.

Звичайно режим відносної адресації працює в таблиці за замовчуванням. Абсолютна адресація застосовується до окремих посилань на комірки у формулах. Для цього використовується символ «заморожування» адреси — «\$». За допомогою цього символу можна «заморозити» як всю адресу, наприклад $\$B\2 , так і окремі її частини, наприклад $\$B2$, $B\$2$.

Рекомендації з організації практичної роботи

Головна задача для учнів на мінімальному рівні вивчення даної теми: навчитися основних методів організації розрахунків за допомогою електронних таблиць. Для цього вони повинні освоїти наступні практичні прийоми роботи в середовищі електронної таблиці:

- ☞ здійснювати переміщення табличного курсору;
- ☞ встановлювати курсор у потрібну комірку;
- ☞ вводити дані: числа, тексти, формули;
- ☞ редагувати дані в комірках;
- ☞ копіювати інформацію в комірках;
- ☞ вставляти і видаляти рядки і стовпці.

Теоретичні питання, що на першому етапі викликають найбільші труднощі — це правила запису формул і розуміння принципу відносної адресації. Їх відпрацьовування варто проводити на задачах і вправах.

Основні *правила запису формул* зводяться до наступного:

- ☞ усі символи у формулі записуються в один рядок;
- ☞ проставляються всі знаки операцій (на відміну від алгебри, де знак множення часто пропускається);
- ☞ використовуються круглі дужки для впливу на послідовність виконання операцій;
- ☞ враховуються пріоритети операцій, що розташовані в такому порядку: зведення в ступінь; множення і ділення; додавання і віднімання;
- ☞ пріоритет стандартних функцій вище арифметичних операцій; аргумент записується в круглих дужках після імені функції;
- ☞ послідовно записані операції однакового старшинства виконуються в порядку запису, тобто зліва на право.

Усі ці правила збігаються з правилами запису виразів у мовах програмування.

Для вправ на дану тему варто давати задачі як прямі (задано математичний вираз, записати формулу для електронної таблиці) так і зворотні (дана формула, записати математичний вираз).

Тепер *про принцип відносної адресації*. Як уже відзначалося, це один з базових принципів функціонування електронної таблиці. *Адреси комірок, що використовуються у формулах, визначаються щодо місця розташування формули*. Зміст цього принципу варто пояснити на прикладах. У даній нижче таблиці формулу в комірці С1 табличний процесор сприймає так: *скласти*

значення, з комірки, що розташована на дві клітинки лівіше зі значенням з комірки, що розташована на одну клітинку лівіше даної формули.

	A	B	C
1	5	3	=A1 + B1

При переносі цієї формули в будь-яку іншу комірку будь-яким способом (копіюванням, внаслідок вставки або видалення фрагментів), збережеться сформульований вище зміст формули. Внаслідок цього зміняться посилання на комірки. Наприклад, при копіюванні формули з комірки C1 в комірку C2 формула прийме вигляд: A2 + B2. При копіюванні в комірку F6 прийме вид: F4 + F5.

Для закріплення розуміння принципу відносної адресації варто виконати кілька завдань. Умови пропонувані задач повинні бути наступного типу: даний фрагмент електронної таблиці (наприклад такий, як приведений вище). Які формули запишуться до комірок блоку D1:F1, якщо в них скопіювати формулу з комірки C2? Результат виконання цього завдання в режимі відображення формул і в режимі відображення значень буде наступним:

	A	B	C	D	E	F
1	5	3	=A1 + B1	=B1+C1	=C1+D1	=D1+E1

	A	B	C	D	E	F
1	5	3	8	11	16	30

Тут заливкою відзначені комірки, у які зроблене копіювання.

Найбільш ефективною буде така постановка завдання: спочатку вирішити задачу теоретично, а потім перевірити отримане рішення на комп'ютері. У цьому випадку відбувається як закріплення розуміння теоретичного питання, так і відпрацювання навичок копіювання даних в електронній таблиці.

При знайомстві з прийомом «заморожування» адрес комірок у формулах, корисно виконати аналогічні вправи. Наприклад, у тій же таблиці в комірку C1 записана формула: A\$1 + B\$1. Який вигляд прийме формула, якщо її скопіювати в блок D1: F1 і в блок C2: F2 ? Результати рішення цієї задачі будуть наступними:

	A	B	C	D	E	F
1	5	3	=A\$1+B\$1	=B\$1+B\$1	=C\$1+B\$1	=D\$1+B\$1
2			=A\$1+B\$1	=B\$1+B\$1	=C\$1+B\$1	=D\$1+B\$1
	A	B	C	D	E	F
1	5	3	8	6	11	9
2			8	6	11	9

Основні типи розрахункових задач, що учні повинні навчитися вирішувати на електронних таблицях:

- 1) одержання нескладних розрахункових відомостей;
- 2) статистична обробка числових таблиць;
- 3) побудова діаграм по табличним даним;
- 4) сортування таблиці за значеннями параметра (стовпця);
- 5) табулювання функцій.

Прикладом рішення задачі першого типу є побудова таблиці обліку продажу молочних продуктів. Інша задача цього ж типу: побудувати таблицю розрахунку щомісячної оплати за витрату електроенергії за даними показаннями лічильника і вартості 1 квт.ч. Обговоримо хід рішення цієї задачі.

Почати рішення задачі варто з проектування таблиці. Це найбільш складний етап для учнів. По-перше, необхідно пояснити алгоритм підрахунку оплати за електрику (не всім дітям це очевидно).

Показання лічильника знімаються наприкінці кожного місяця. Витрата електроенергії за місяць визначається як різниця показання лічильника в даному місяці й у попередньому. Потім оплата підраховується як добуток витрати на ціну 1 квт.ч. Для розрахунку плати за січень необхідно знати показання лічильника в грудні минулого року. З усього сказаного потрібно зробити висновки: які величини є вхідними даними, які — обчислюються за формулами. Вхідні дані — показання лічильника кожен місяць і вартість 1 квт.ч, а що обчислюються — щомісячна витрата електроенергії і сума оплати.

Тепер можна переходити до побудови таблиці. Варто відразу ж сформулювати учням деякі правила оформлення таблиці. По-перше, у таблиці обов'язково повинен бути заголовок. Рядки і стовпці таблиці повинні бути поійменовані. Поступово вчитель повинен познайомити учнів з основними засобами форматування таблиці для одержання оформленого документа. Початкова частина таблиці буде виглядати так (передбачається, що ціна 1квт.ч дорівнює 20 коп., а плата підраховується в гривнях):

	A	B	C	D
1	Відомість оплати електроенергії			
2	<i>місяць</i>	<i>Лічильник</i>	<i>Витрата в квт.ч</i>	<i>Оплата в грн.</i>
3	Грудень	4000		
4	Січень	4200	=B4- B3	=C4 * 0,2
5	Лютий	4350	=B5 - B4	=C5 * 0,2
6	Березень	4475	=B6 - B5	=C6 * 0,2

Під час роботи з табличним процесором *обов'язково потрібно використовувати прийом копіювання формул*. Формули в комірках C4, D4 є вхідними. Усі формули, розташовані нижче них, отримані шляхом копіювання. З цього приклада учням стане зрозумілим практичний зміст принципу відносної адресації: він дозволяє швидко створювати великі таблиці без переписування формул.

Розбір цієї задачі приводить до ідеї використання абсолютної (замороженої) адреси. Зручно ціну 1 квт.ч зберігати в окремій комірці (наприклад, в B16), а в комірці D4 записати формулу C4 * \$B\$16. При копіюванні формули абсолютна адреса мінятися не буде. Тепер, якщо відбудеться зміна ціни 1 квт.ч, то буде достатньо внести зміни лише в одну комірку B16.

Дуже часто в числових таблицях підраховуються різні підсумкові дані: суми, середні значення, найбільші і найменші значення. Одержання таких даних називається *статистичною обробкою таблиці*. В усіх табличних процесорах є для цього відповідні функції. У задачах того ж типу, що розглянуто вище, варто додати завдання на подібну статистичну обробку даних. Наприклад, до таблиці оплати електроенергії додати обчислення загальної суми грошей, виплачених за рік, середньомісячної витрати електроенергії, найбільшої і найменшої місячної плати.

Представлення табличних даних у графічній формі часто використовується на практиці. Графічна обробка додає наочність, видимість результатам розрахунків. Табличні процесори надають користувачеві на вибір безліч типів діаграм (гістограм, графіків). Такі графічні засоби прийнято називати *діловою графікою*.

Для побудови діаграми користувач повинен вказати її тип і повідомити табличному процесорові, з яких блоків таблиці потрібно вибрати всю необхідну інформацію.

Великі можливості дає використання в електронній таблиці *умовної і логічної функцій*. Таблиця без використання умовної функції реалізує в собі лінійний обчислювальний алгоритм. Використання умовної функції вносить до таблиці структуру розгалуження. Потреба в розгалуженні з'являється при ускладненні умови задачі. Наприклад, якби правило оплати за електроенергію звучало так: за перші 100 квт.ч потрібно платити по 20 копійок; за кожен квт.ч, витрачений вище цього, потрібно платити по 30 копійок. Тоді розрахункова формула в комірці D4 виглядала б так:

ЯКЦО (C4<=100; C4*0.2; 20+(C4-100)*0.3).

Можливість *сортування даних* у таблиці існує в табличних процесорах завдяки наявності режиму бази даних. Звичайно сортування організується у виділеному блоці таблиці. Як і в базах даних, вказується стовпець, по якому виконується сортування, і порядок сортування: по зростанню або по спаданню значень у стовпці. Наприклад, у розглянутій таблиці можна відсортувати рядки, починаючи з четвертого, по спаданню розміру оплати. Тоді спочатку розташується самий «дорогий» місяць, далі — по спаданню.

Таблювання функції — одна з часто розв'язуваних прикладних задач математики. Таблювання означає побудову таблиці значень функції для значень аргументу, що змінюються у визначеному інтервалі з даним кроком. Таблювання дозволяє дослідити функцію: простежити характер зміни, виділити області коренів, визначити екстремальні значення. Застосовуючи прийом копіювання, в електронній таблиці можна швидко побудувати таблицю значень функції великого розміру. Для цього досить ввести два початкові рядки таблиці і потім скопіювати їх униз на потрібне число рядків. Наприклад, таблиця функції

$$F(x)=x^2+0,5*x,$$

обчисленої з кроком 0,2 починаючи від значення $x = 0$, будується в такий спосіб

	A	B
1	Крок =	0,2
2	X	F(X)
3	0	=A3*A3+0.5*A3
4	=A3+\$B\$1	=A4*A4+0.5*A4
5	=A4+\$B\$1	=A5*A5+0.5*A5

Якщо потрібно змінити початкове значення x у таблиці, то досить виправити комірку A3, а для зміни кроку потрібно змінити вміст комірки B1. Формули записуються за правилами, що прийняті у використовуваному табличному процесорі.

Застосування електронної таблиці як інструменту для математичного моделювання розглядається в поглибленому варіанті вивчення базового курсу.

Що ж таке *математична модель*? Це опис стану або поведінки деякої реальної системи (об'єкта, процесу) мовою математики, тобто за допомогою формул, рівнянь та ін. математичних співвідношень.

Реалізація математичної моделі — це застосування визначеного методу розрахунків значень вихідних параметрів за значеннями вхідних параметрів. Технологія електронних таблиць — один з можливих методів реалізації математичної моделі. Іншими методами реалізації математичної моделі може бути складання програм на мовах програмування, застосування математичних пакетів (MathCad, Математика й ін.), застосування спеціалізованих програмних систем для моделювання. Реалізовані такими засобами математичні моделі будемо називати *комп'ютерними математичними моделями*.

Мета створення комп'ютерної математичної моделі — проведення *чисельного експерименту*, що дозволяє досліджувати систему, що моделюється, спрогнозувати її поведінку, підібрати оптимальні параметри та ін.

Отже, характерні ознаки комп'ютерної математичної моделі наступні:

- ☞ наявність реального об'єкта моделювання;
- ☞ наявність кількісних характеристик: вхідних і вихідних параметрів;
- ☞ наявність математичного зв'язку між вхідними і вихідними параметрами;
- ☞ реалізація моделі за допомогою визначених комп'ютерних засобів.

Методика вивчення розділу «Бази даних. Системи управління базами даних. Експертні системи»

Основні цілі.

Дати представлення про призначення інформаційних систем і баз даних. Познайомити з основами реляційних баз даних. Навчити основним прийомам роботи з однієї з реляційних СУБД.

Навчити організації пошуку, сортування, редагування даних. У заглибленому варіанті: дати представлення про проблеми проектування реляційних баз даних.

Досліджувані питання.

1. Призначення інформаційних систем і баз даних (БД).
2. Класифікація БД.
3. Структура реляційної бази даних (РБД).
4. Елементи РБД: головний ключ; ім'я, значення і тип поля.
5. Призначення СУБД; режими роботи СУБД.
6. Пошук інформації в базі даних.
7. Логічні вираження в умовах пошуку і видалення записів.
8. Сортування; ключі сортування.
9. Елементи проектування РБД; нормалізація даних.

Методичні рекомендації з викладу теоретичного матеріалу

Згідно з шкільною програмою з інформатики в 9-12 класах відводиться годин на вивчення даної теми:

11 кл. – 11 год.

1. Призначення інформаційних систем і баз даних (БД).

Вивчення баз даних варто почати з обґрунтування *актуальності* даного додатка комп'ютерної техніки. Задачу можна сформулювати в такий спосіб: мається великий обсяг даних про якусь реальну систему об'єктів або подій. Наприклад, про книги в бібліотеці, про працівників підприємства, про товари на складі, про дорожньо-транспортні випадки за тривалий період часу.

Необхідно організувати збереження цієї інформації таким чином, щоб її було зручно переглядати, поповнювати, змінювати, шукати потрібні зведення, робити будь-які вибірки здійснювати сортування в будь-якому порядку. Такою роботою працівники були змушені займатися і задовго до появи комп'ютерів. Основним засобом збереження даних був папір, дані зберігалися у вигляді списків у товстих журналах, папках, на картонних картках. Останній спосіб використовується, наприклад, у бібліотечних каталогах. Подібні систематизовані картотеки використовуються у відділах кадрів підприємств. Вони зручні тим, що легко можна витягти потрібну картку, замінити, додати нові картки, зберігаючи встановлений порядок. Проте, якщо така картотека містить тисячі карток, то, як би досконала ні була її організація, обробка даних у ній — справа тривала і трудомістка.

Інший приклад — архіви різних документів. Наприклад, існують історичні архіви, архіви судових справ, архіви патентів на винаходи. Часом такі архіви займають цілі будинки. Пошук у них потрібних документів вимагає значних зусиль. Крім того, існують кіноархіви, фотоархіви, архіви звукових записів.

У наш час у рішенні таких проблем допомагають комп'ютери. Комп'ютерні інформаційні системи дозволяють зберігати великі обсяги даних, здійснювати в них швидкий пошук, вносити зміни, виконувати всілякі маніпуляції з даними.

Варто привести приклади таких інформаційних систем. Наприклад, система продажу залізничних і авіаційних квитків, інший знайомий учням приклад: під час телерепортажів з великих міжнародних змагань, олімпійських ігор на екран миттєво виводиться досьє будь-якого спортсмена, про яке говорить коментатор — це працює комп'ютерна інформаційна система.

Основою всякої інформаційної системи є **база даних** — **організована сукупність даних на магнітних дисках**. Учні вже добре знають, що інформація на дисках зберігається у виді файлів. Тому перший висновок, який можна зробити щодо організації великих баз даних це те, що вони вимагають великих обсягів дискової пам'яті.

2. Класифікація баз даних.

Бази даних класифікують за різними ознаками:

- За характером збереженої інформації БД поділяються на фактографічні і документальні. Якщо проводити аналогію з описаними вище прикладами інформаційних сховищ, то фактографічні БД — це картотеки, а документальні — це архіви. У фактографічних БД зберігається коротка інформація в строго визначеному форматі. У документальних БД — усілякі документи. Причому це можуть бути не тільки текстові документи, але і графіка, відео і звук (мультимедіа).
- Класифікація за способом збереження даних поділяє БД на централізовані і розподілені. Вся інформація в централізованій БД зберігається на одному комп'ютері. Це може бути автономний ПК або сервер мережі, до якої мають доступ користувачі-клієнти. Розподілені БД використовуються в локальній і глобальній комп'ютерних мережах. У такому випадку різні частини бази зберігаються на різних комп'ютерах.
- Третя ознака класифікації баз даних — за структурою організації даних. Якщо до цього учні не мали справу з поняттям структури організації даних, учитель повинен зупинитися на цьому питанні, або повторити знайомий учням матеріал. Вчитель повинен розповісти про три способи організації даних: табличний, ієрархічний і мережний. Бази даних, що використовують відповідний спосіб організації інформації, називаються реляційними (табличні БД), ієрархічними і мережними БД.

3. Структура реляційної бази даних .

У базовому курсі інформатики розглядаються лише фактографічні реляційні бази даних. Це зв'язано не тільки з обмеженістю шкільного курсу, але і з тим фактом, що реляційний вид використовується сьогодні найбільш часто і є універсальним. Теоретично доведено, що будь-яка система даних може бути відбита за допомогою таблиць. Найпростіша реляційна БД містить одну таблицю, більш складна може складатися з безлічі взаємозалежних таблиць.

Структура таблиці обговорювалася в попередній темі (Табличні процесори) і тому учням уже знайома: у різних рядках утримується інформація про різні об'єкти описуваної системи, а стовпці відповідають різним атрибутам цих об'єктів. У термінології реляційних баз дані рядки таблиці називаються записами, стовпці — полями. Сама назва реляційна БД походить від англійського слова „relation" що переводиться як «відношення». Тут термін «відношення» розуміється як взаємозв'язок між полями таблиці. У реляційному підході таблиця називається відношенням.

Пояснення даного матеріалу варто проводити на конкретних прикладах, учитель повинен заздалегідь підготувати будь-яку БД. Наприклад, домашня бібліотека, погода, успішність, факультативи.

Варто підкреслити, що в базах даних кожна таблиця повинна мати своє ім'я. Вчителю буде зручно працювати, якщо ці таблиці вивести на демонстраційний монітор або, у крайньому випадку, винести на плакати.

Основні представлення, що повинні бути закріплені учнями:

- всяка таблиця містить у собі інформацію про деяку реальну систему (процес) і, отже, є її інформаційною моделлю;
- всякий запис у таблиці — інформація про конкретний об'єкт (події) даної системи;
- значення поля і кожного запису — це визначена характеристика (властивість, атрибут) об'єкта.

4. Елементи РБД: головний ключ; ім'я, значення і тип поля.

Основні поняття, зв'язані з записами і полями: головний ключ запису, ім'я поля, значення поля, тип поля.

Головний ключ — це поле або сукупність полів, що однозначно визначає запис у таблиці. Можна ще сказати так: головний ключ — це ідентифікатор запису. У базах даних слово ключ має кілька вживань: ключ пошуку — поле, за значенням якого шукається запис у БД, ключ сортування

— поле, за значенням якого відбувається упорядкування записів. Тому ідентифікатор записів треба називати головним ключем.

Вчителеві важливо розуміти, що тема Бази даних містить у собі ряд вузлових питань, що мають фундаментальне значення для курсу інформатики в цілому. В цій темі учні зустрічаються з поняттям величини. У розділі «Основи алгоритмізації і програмування» і «Табличні процесори» це поняття теж зустрічається. **Величина — це окремий інформаційний об'єкт, що має власне ім'я і займає місце в пам'яті комп'ютера.** З цього погляду поля є величинами. Кожне поле в таблиці має ім'я, для кожного поля визначений тип. Поняття типу величини зв'язано з її властивостями:

- безліччю значень, що може приймати величина;
- безліччю операцій, які можна виконувати з цією величиною;
- безліччю функцій, визначених над величинами даного типу;
- безліччю відносин над даними;
- формою внутрішнього представлення в пам'яті ЕОМ.

У найпростіших базах даних використовуються 4 основних типи величини: текстова, числова, дата і логічна. Поле текстового типу може зберігати послідовність символів (до 255); числові поля можуть містити цілі або дробові десяткові числа; дата — день/місяць/рік; логічні поля — значення логічних величин (так - ні, істина - лож).

Необхідно звернути увагу учнів на наступну обставину: текстове поле може складатися з цифр. З цієї причини іноді виникає плутанина з текстовим і числовим типом (потрібно на це звернути увагу учнів).

Значення числових полів можуть бути використані в обчисленнях. Іноді в умовах пошуку інформації присутні арифметичні вираження, операндами в яких можуть бути тільки значення числових полів. До текстових полів можна застосовувати операції відношення (менше, більше, дорівнює й ін.). Оскільки цифри в кодовій таблиці розташовані в порядку зростання, то відношення між рядками збережуть математичний зміст.

Необхідно такі поняття як «головний ключ», «тип поля», «ім'я поля» закріплювати за допомогою серії практичних завдань.

Наприклад,

Завдання 1 : дано ім'я таблиці і перелік полів, потрібно вказати головний ключ і визначити типи всіх полів.

Завдання 2 : визначена предметна область бази даних; потрібно озаглавити таблицю, визначити імена полів і їхні типи, призначити головний ключ. Звичайно, ця задача має відношення до області проектування БД і вже з цієї причини складна. Тому вимагати від учнів повноти її рішення не обов'язково.

5. Призначення СУБД; режими роботи СУБД.

Після знайомства з основними поняттями, що відносяться до організації інформації в реляційних БД, варто перейти до вивчення програмного забезпечення, призначеного для роботи з базами даних. Таке програмне забезпечення називається СУБД — система керування базами даних.

Для персональних комп'ютерів існує цілий ряд СУБД реляційного типу. Історично однією з перших таких систем була dBASE. Використані в цій системі формати представлення даних, мова обробки даних стали стандартом для ряду наступних СУБД.

Існують два способи роботи користувача з базою даних: робота за допомогою прикладних програм, заздалегідь складених програмістом у середовищі СУБД, і робота шляхом безпосередньої взаємодії із СУБД.

У початковий період появи і поширення баз даних використовувався тільки перший спосіб. Пізніше стали з'являтися СУБД, орієнтовані на роботу з користувачем. Для них були створені зручні діалогові оболонки, що дозволяють користувачеві легко реалізовувати свої потреби в роботі з базою даних:

- поповнювати і змінювати базу,
- здійснювати пошук даних за будь-якими умовами,
- створювати звітні документи.

Найбільш ідеальною СУБД такого типу є MS Access. У різних школах використовується різна техніка і програмне забезпечення. Варіанти доступних СУБД можуть бути найрізноманітнішими: від навчальної РБД для УКНЦ до професійної MS Access для IBM PC. У різних системах розрізняється спосіб взаємодії користувача із СУБД — інтерфейс користувача.

Є два принципово різних режими такої взаємодії: посимвольне введення команд і робота з діалоговою оболонкою. Ця ситуація подібна взаємодії користувача з операційною системою MS-DOS: робота в режимі командного рядка або робота через оболонку Norton Commander.

У будь-якій СУБД існує своя **мова опису даних і мова маніпулювання даними**. Якщо користувач змушений працювати в режимі посимвольного введення команд, то він повинен знати всі подробиці синтаксису командної мови. Діалогова оболонка — більш високорівневий засіб інтерфейсу. Однак варто розуміти, що, працюючи з оболонкою, користувач формує ті ж самі команди, залучаючи допоміжні засоби: меню, панелі інструментів, підказки й ін.

Будь-яка команда — це інформація, що керує виконанням визначеного виду роботи. Вона повинна містити всі необхідні дані для цього. Звичайно це ім'я команди і набір параметрів. З методичної точки зору важливо, щоб учні, виконуючи будь-які дії з БД, розуміли: яка віддається команда, які параметри для її виконання повинні бути повідомлені системі.

Вивчення конкретної СУБД варто проводити по стандартній методичній схемі: «середовище - режими роботи - система команд - дані».

Наприклад, для СУБД Access розрізняються наступні основні режими роботи:

- режим роботи з таблицею;
- режим роботи з запитом;
- режим роботи зі звітами
- режим роботи з формами.

(Режими роботи з макросами і модулями в базовому курсі не розглядаються.) У свою чергу, у кожному з відзначених режимів є підрежими:

- Перегляд,
- Конструктор,
- Створити.

Наприклад, працюючи в режимі «Таблиця-Перегляд», користувач може переглянути зміст таблиці, а також відредагувати деякі її поля. У режимі «Таблиця-Конструктор» можна переглянути опис структури таблиці і внести в неї зміни. У режимі «Таблиця-Створити» описується і створюється структура нової таблиці.

6. Пошук інформації в базі даних.

Якщо інформатика викладається в школах з поглибленим вивченням даного предмета, то після знайомства з навчальним матеріалом на 1-му рівні вивчення теми, можна перейти до додаткових задач другого рівня. Задачею 1-го рівня є: дати загальні представлення про бази даних, навчити роботи з готової БД: здійснювати пошук інформації; сортування, видалення і додавання записів. Задача 2-го рівня - познайомити з основами проектування БД; навчити створювати структуру і заповнювати базу даних.

Робота із СУБД починається з запуску відповідної програми, тому учням необхідно показати, де зберігається програма і як її завантажувати на виконання.

Перше поняття, що повинні засвоїти учні: база даних зберігається у файлі; щоб почати з нею працювати, необхідно відкрити файл із БД. Потім учитель повинен показати, як можна переглянути на екрані запису таблиці. Для цього повинна бути заздалегідь підготовлена демонстраційна база даних.

Представляючи учням демонстраційну базу даних, необхідно звернути увагу на те, що поряд із самою таблицею в пам'яті комп'ютера зберігається опис її структури, звідки користувач може

довідатися параметри полів: ім'я, тип, формат і ін. У СУБД Access це робиться в режимі «Таблиця-Конструктор».

Основна задача будь-якої інформаційної системи — пошук інформації в базі даних. Пошук відбувається по запиту користувача. У результаті виконання запиту виходить таблиця, що складається з полів, зазначених у команді. У цю таблицю включається інформація з тих записів, що задовольняють умові пошуку. Умова пошуку являє собою логічне вираження. Тут учні зустрічаються ще з однією фундаментальною складовою даної теми курсу — з основами математичної логіки. Крім прикладного застосування в інформатиці ця тема має велике загальноосвітнє значення.

7. Логічні вирази в умовах пошуку і видалення записів.

Визначення логічного виразу (це деяке висловлення, з приводу якого можна сказати істинно воно або ложно), стосовно до баз даних, визначення можна перефразувати так: логічний вираз — це деяке висловлення з приводу значень полів бази даних; це висловлення стосовно різних записів може бути істинно або помилково.

Логічні вирази поділяються на прості і складні. У простих виразах завжди використовується лише одне поле таблиці, і не застосовуються логічні операції. У складних логічних виразах використовуються логічні операції. Простий логічний вираз являє собою або ім'я поля логічного типу або відношення (у математиці говорять нерівність). Відношення для числових величин зберігають зміст математичних нерівностей; при обчисленні відношення для символічних величин враховують лексикографічний порядок; дати порівнюються в порядку їхньої календарної послідовності.

Основна проблема — навчити учнів формальному представленню умов пошуку у вигляді логічних виражень. Наприклад, від фрази «знайти всі книги, що лежать вище п'ятої полиці, потрібно перейти до логічного вираз «ПОЛИЦЯ» 5», або умова «вибрати всіх невстигаючих з фізики, представити у виді: ФІЗИКА <3.

Особливу увагу треба звернути на використання полів логічного типу в умовах пошуку. Звичайно до них не застосовуються відношення. Логічне поле саме несе логічне значення: істина або неправда. Наприклад, умова «вибрати всіх учнів, що відвідують танці» представиться одним ім'ям логічного поля: ТАНЦІ.

Складні логічні вирази містять у собі логічні операції. Розглядаються три основні операції математичної логіки: кон'юнкція (І), диз'юнкція (АБО), заперечення (НІ).

Звичайно при поясненні цього питання учитель відштовхується від семантичного змісту висловлення українською мовою, що містить союзи І, АБО, частку НІ.

Наприклад, висловлення: «Сьогодні буде контрольна з алгебри І з фізики» справедливо, якщо відбудуться обидві контрольні і ложно, якщо хоча б одна не відбудеться. Інше висловлення: «Сьогодні буде контрольна з алгебри АБО з фізики» буде істинно, якщо відбудеться хоча б одна контрольна робота. І, нарешті, висловлення: «Сьогодні НЕ буде контрольної» істинно, якщо контрольна не відбудеться, тобто якщо висловлення про те, що сьогодні буде контрольна, виявляється помилковим. З подібних прикладів учитель робить висновки про **правила виконання логічних операцій:**

якщо А і В логічні величини, то

- А й В істинно тільки в тому випадку, якщо істинні обидва операнда;
- А або В ложно тільки в тому випадку, якщо помилкові обидва операнда;
- не А змінює значення логічної величини на протилежне: не істина — неправда; не неправда — істина.

Ці правила зводяться в таблицю істинності.

При переході до складання умов пошуку в базах даних учні нерідко попадають у пастки природного змісту. Наприклад, задача формулюється так: із БД «Бібліотека» одержати відомості про всі книги Толстого і Тургенєва. Абсолютна більшість учнів записують цю умову в такий спосіб:

АВТОР=«Толстой» і АВТОР = «Тургенєв»

Союз, що пролунав у завданні, «і» машинально переноситься в логічний вираз. Після цього вчитель мусить пояснювати, що автором книги не може бути одночасно Толстой і Тургенєв. Тому в бібліотеці немає жодної книжки, що задовольняє такій умові. Тут варто застосувати логічну операцію АБО:

АВТОР = «Толстой» або АВТОР = «Тургенєв»

Тоді буде отримана шукана вибірка книг обох авторів.

Операція АБО поєднує в одну вибірку записи, що задовольняють кожній з умов. Операція І працює інакше: спочатку вибираються всі записи, що задовольняють першій умові, потім з відібраних записів вибираються ті, котрі задовольняють другій умові.

Корисно виконати з учнями кілька формальних завдань на обробку складних умов пошуку. Наприклад, намалюйте на дошці наступну таблицю:

Записи	Поля		
	A	B	C
R1	1	2	3
R2	1	3	1
R3	2	2	2
R4	3	3	3
R5	3	2	3

Запропонуйте серію завдань такого змісту: задано умову пошуку у формі логічного виразу; визначити, які записи йому задовольняють

Умова

Відповідь:

1) A=1 і B=2

R1

2) A=1 **або** A=3

R1,R2,R4,R5

3) A=1 **або** B=2

R1,R2,R3,R5

4) A=1 **або** B=2 **або** 3=3

R1,R2,R3,R4,R5

5) A=1 і B=2 і 3=3

R1

6) **ні** A=1

R3,R4,R5

На прикладі цієї ж таблиці відпрацьовується питання про старшинство операцій і порядок їхнього виконання. Повідомивши, що логічні операції по убуванню старшинства розташовані так ІІ, І, АБО, приведіть приклади логічних виражень, що містять різні операції.

7) A=1 і B=2 **або** 3=3

: R1,R4,R5

8) A=1 **або** B=2 і 3=3

: R1,R2,R5

9) **ні** A=1 **або** B=2 і 3=3

: R1,R3,R4,R5

10) (A=1 **або** B=2) і 3=3

: R1,R5

Після рішення таких формальних задач варто знову повернутися до змістовних задач. Тепер учні будуть набагато успішніше формалізувати складні умови пошуку в логічні вирази. Наприклад, потрібно вибрати всі книги Беляєва і Толстого, розташовані від 2-й до 5-й полки. Логічний вираз запишеться так:

(АВТОР = «Толстої Л.Н.» або АВТОР = «Беляєв А.Р.») і ПОЛКУ>=2 і ПОЛКУ<=5

Переходячи до роботи з конкретною СУБД, учитель знайомить учнів із правилами формування в ній команд запити. Якщо немає діалогової оболонки і команди задаються

посимвольним введенням, то потрібно описувати синтаксис команд, звертаючи увагу на всі деталі. При використанні діалогової оболонки команда формується в діалозі. Система підказує користувачеві кожен крок і надає меню для вибору чергового елемента команди. При цьому користувач бачить на екрані зформовану команду.

У СУБД Access для створення запитів використовується конструктор запитів. Формована команда виявляється схованою від користувача. Робота з конструктором вимагає певних навичок, які варто відпрацьовувати на вправах.

8. Сортування. Ключі сортування.

Ще один важливий вид маніпулювання інформацією в базі даних — сортування записів. Тут основними поняттями, що повинні засвоїти учні, є «ключ сортування», і «порядок сортування». Ключ сортування — те поле, за значенням якого відбувається упорядкування записів у таблиці. Порядок сортування має два варіанти: по зростанню значень ключа і по убутанню значень. Якщо ключів декілька, то серед них встановлюється ієрархія: первинний ключ, вторинний ключ і т.д. У першу чергу запису сортується за значенням первинного ключа; усередині групи записів з однаковими значеннями первинного ключа відбувається сортування по вторинному ключеві і т.д. СУБД Access дозволяє сортувати записи як у всій вихідній таблиці, так і в таблицях, одержаних у результаті виконання запиту на вибірку.

9. Елементи проектування РБД; нормалізація даних.

При поглибленому вивченні розділу «Бази даних» учні знайомляться з елементами проектування БД. Проектування БД — складна задача. Лише на перший погляд вона може показатися простою. Для невеликих навчальних БД помилки при проектуванні не настільки істотні. Але якщо створюється велика база, у якій будуть зберігатися багато тисяч записів, то помилки при проектуванні можуть коштувати дуже дорого. Основні наслідки неправильного проектування — надмірність інформації, її суперечливість, втрата цілісності, тобто взаємозв'язку між даними. В результаті БД може виявитися непрацездатною і вимагати дорогої переробки.

Теорія реляційних баз даних була розроблена в 70-х роках Е. Коддом. Е. Кодд запропонував технологію проектування баз даних, у результаті застосування якої, в отриманій БД не виникає відзначених вище недоліків (див. наприклад, книгу Дж. Мартін «Організація баз даних в обчислювальних системах» Мир, 1980). Сутність цієї технології зводиться до приведення таблиць, що складають базу даних, до третьої нормальної форми. Цей процес називається нормалізацією даних: спочатку всі дані, що планується включити в БД, представляються в першій нормальної формі, потім перетворюються до другого і на останньому кроці — до третьої нормальної форми. Проілюструємо процес нормалізації даних на прикладі. Ставиться задача: створити базу даних, що містить відомості про відвідування пацієнтами поліклініки свого дільничного лікаря. Спочатку будується одна таблиця, у яку заносяться прізвище пацієнта, його дата народження, номер ділянки, до якого приписаний пацієнт, прізвище дільничного лікаря, дата відвідування лікаря і встановлений діагноз хвороби. Нижче приведений екземпляр такої таблиці:

БД «Поліклініка»

Прізвище пацієнта	Дата народження	Номер ділянки	Прізвище лікаря	дата відвідування	Діагноз
Орлова Е.Ю.	25.01.47	1	Андрєєва І. В.	26.07.98	ОРЗ
Лосєв О.І.	20.04.65	2	Петрова О.І.	14.03.98	бронхіт
Жукова Л.Г.	05.03.30	2	Петрова О.І.	11.04.98	бронхіт
Орлова Е	30.01.70	1	Андрєєва І. В.	11.07.98	стенокардія
Бикова А.А.	25.01.47	1	Андрєєва І. В.	15.06.98	ангіна
Дуров М.Т	01.04.75	2	Петрова О.І.	26.07.98	гастрит

Неважко зрозуміти недоліки такої організації даних. По-перше, очевидна надмірність інформації повторення дати народження тієї самої людини, повторення прізвища лікаря тієї самої ділянки. У такий БД велика імовірність мати недостовірні, суперечливі дані. Наприклад, якщо на другій ділянці переміниться лікар, то доведеться переглядати всю базу і вносити зміни у всі записи, що відносяться до цієї ділянки. При цьому велика імовірність щось пропустити. Після кожного нового відвідування пацієнтом лікарні буде потрібно знову вводити його дату народження, номер ділянки, прізвище лікаря, тобто інформацію, що вже існує в БД.

Отримана таблиця відповідає першій нормальній формі для усунення відзначених недоліків, потрібно її подальша нормалізація. Структура такої таблиці (відношення) описується в такий спосіб:

ПОЛІКЛІНІКА (ПРИЗВИЩЕ, ДАТА НАРОДЖЕННЯ, ДІЛЯНКА, ЛІКАР, ДАТА ВІДВІДУВАННЯ, ДІАГНОЗ)

Необхідно установити ключ записів. Тут ключ складений, котрий містить у собі два поля ПРИЗВИЩЕ і ДАТА_ВІДВІДУВАННЯ. Кожен запис — це інформація про конкретне відвідування пацієнтом лікарні. Якщо допустити, що протягом одного дня даний пацієнт може зробити тільки один візит до дільничного лікаря, то в різних записах не буде повторюватися комбінація двох полів: прізвища пацієнта і дати відвідування лікаря.

Відповідно до визначення другої нормальної форми, усі неключові поля повинні функціонально залежати від повного ключа. У даній таблиці лише ДІАГНОЗ визначається одночасно прізвищем пацієнта і датою відвідування. Інші поля зв'язані лише з прізвищем, тобто від дати відвідування вони не залежать. Для перетворення до другої нормальної форми таблицю потрібно розбити на дві наступні:

ВІДВІДУВАННЯ (ПРИЗВИЩЕ, ДАТА ВІДВІДУВАННЯ, ДІАГНОЗ)
ПАЦІЄНТИ (ПРИЗВИЩЕ, ДАТА НАРОДЖЕННЯ, ДІЛЯНКА).

У таблиці ВІДВІДУВАННЯ як і раніше діє складений ключ із двох полів, а в таблиці ПАЦІЄНТИ — одне ключове поле ПРИЗВИЩЕ.

В другій таблиці мається так названа транзитивна залежність. Вона відображається в такий спосіб:



Значення поля ЛІКАР зв'язане із прізвищем пацієнта **транзитивно** через поле ДІЛЯНКА. Справді, усякий дільничний лікар приписаний до своєї ділянки й обслуговує хворих, що відносяться до даної ділянки.

Відповідно до визначення третьої нормальної форми, у таблиці не може бути транзитивних залежностей. Виходить, потрібна ще одна розбивка таблиці ПАЦІЄНТИ на 2 таблиці..

У підсумку одержуємо базу даних, що складає з трьох таблиць:

ВІДВІДУВАННЯ (ПРИЗВИЩЕ, ДАТА ВІДВІДУВАННЯ, ДІАГНОЗ)

ПАЦІЄНТИ (ПРИЗВИЩЕ, ДАТА_НАРОДЖЕННЯ, ДІЛЯНКА, ЛІКАР)

ЛІКАРІ (ДІЛЯНКА, ЛІКАР)

У третій таблиці ключем є номер ділянки, оскільки він повторюватися не може. У той же час можлива ситуація, коли один лікар обслуговує більше однієї ділянки. Отримана структура БД задовольняє вимогам *третьої нормальної форми*: у таблицях усі неключові поля цілком функціонально залежать від своїх ключів, і відсутні транзитивні залежності.

Ще одною важливою властивістю отриманої БД є те, що між трьома таблицями існує взаємозв'язок через загальні поля. Таблиця ВІДВІДУВАННЯ і ПАЦІЄНТИ зв'язані загальним полем ПРИЗВИЩЕ. Таблиця ПАЦІЄНТИ і ЛІКАРІ зв'язані через поле ДІЛЯНКА. Для зв'язаних таблиць існує ще одне поняття: тип зв'язку. Можливі три варіанти типу зв'язків: «один - до одного», «один -

до багатьох», «багато - до багатьох». У прикладі між зв'язаними таблицями існують зв'язки типу «до багатьох», і схематично вони відображаються так:

ЛІКАРІ <->>> ПАЦІЄНТИ <->>> ВІДВІДУВАННЯ

Зміст наступний: у кожного лікаря (на кожній ділянці) багато пацієнтів; кожен пацієнт відвідує лікаря безліч разів.

У приведеному прикладі показана процедура нормалізації в строгій відповідності з теорією реляційних баз даних. Розуміння змісту цієї процедури дуже корисно для вчителя.

Учитель не дає докладного опису процедури нормалізації, не приводиться строгого визначення трьох нормальних форм. Використовується нетрадиційний термін добре нормалізована база даних. У цьому понятті фактично закладені властивості третьої нормальної форми. Ці властивості сформульовані так: усі поля таблиці повинні відбивати безпосередні характеристики (атрибути) об'єкта, до якого ВІДНОСИТЬСЯ запис. Учням пропонується наступна, у деякому змісті, інтуїтивна методика одержання добре нормалізованої БД. Всю безліч даних потрібно розділити між різними об'єктами, до яких вони відносяться. На прикладі приведеної вище таблиці ПОЛІКЛІНІКА потрібно побачити три різних типи об'єктів, до яких відноситься дана інформація: це пацієнти поліклініки, лікарі і відвідування пацієнтами лікарів. Відповідно будуються три таблиці, що містять атрибути, що відносяться до цих трьох типів об'єктів.

Рекомендації з організації практичної роботи

Практичні завдання поділяються на три типи

- 1) задачі: теоретичні завдання для закріплення основних понять;
- 2) вправи: практичні завдання для роботи в середовищі СУБД із метою відпрацювання окремих навичок;
- 3) індивідуальні роботи: залікові завдання, що вимагають від учнів комплексного володіння теоретичними знаннями і практичними навичками.

Вправи виконуються на комп'ютері. Вчитель може використовувати матеріал вправ у ході пояснень для демонстрації прийомів роботи з базою даних засобами досліджуваної СУБД. Після такого пояснення варто запропонувати учням колективно виконати іншої вправи того ж типу. Робочий матеріал для вправ (файли з таблицями) учитель повинний підготувати заздалегідь.

Варіанти індивідуальних робіт учитель розподіляє серед учнів за своїм розсудом. Можливий варіант, коли усі роблять те саме завдання. У такому випадку вчителю легше здійснювати контроль, робити допомогати учням. За результатами виконання роботи кожен учень одержує оцінку.

Підбор практичного матеріалу дозволяє організувати вивчення теми «Бази даних на різних рівнях, використовувати різні СУБД».