

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Запорізький національний технічний університет

Факультет післядипломної освіти

“Проектування баз даних”

**Програма, методичні вказівки до
лабораторних робіт і самостійної роботи**

слухачів спеціальності 7.080403

“Програмне забезпечення автоматизованих систем”

Частина I

2002

Проектування баз даних. Програма, методичні вказівки до лабораторних робіт і самостійної роботи слухачів спеціальності 7.080403 “Програмне забезпечення автоматизованих систем” /Укл.: С.К.Корнієнко, С.Н.Сердюк. – Запоріжжя: ЗНТУ, 2002. - 66 с.

Укладачі:

С.К.Корнієнко, доцент, к.т.н.,
С.Н.Сердюк, доцент, к.т.н.

Рецензент: Пінчук В.П., доцент, к.т.н

Відповідальний за випуск: С.К.Корнієнко

Затверджено
на засіданні кафедри ПЗ

Протокол № 2 від 13.10.2002

Методичні вказівки, призначені для використання на лабораторних заняттях із дисципліни "Проектування баз даних", а також в процесі самостійної роботи слухачів факультету післядипломної освіти.

Методичні вказівки містять програму дисципліни, практичні рекомендації з проектування баз даних, які можуть бути корисними під час курсового та дипломного проектування, а також указівки із виконання лабораторних робіт.

ЗМІСТ

1 ПРОГРАМА ДИСЦИПЛІНИ ТА МЕТОДИЧНІ ВКАЗІВКИ З ЇЇ ВИВЧЕННЯ.....	5
1.1 ОСНОВИ ПОБУДОВИ БАНКІВ ІНФОРМАЦІЇ	5
1.2 КОНЦЕПТУАЛЬНЕ ПРОЕКТУВАННЯ БАЗ ДАНИХ	13
1.3 РЕЛЯЦІЙНА МОДЕЛЬ ДАНИХ	16
1.3.2 <i>Типи функціональних залежностей. Нормалізація баз даних. Проектування реляційних баз даних.....</i>	<i>19</i>
1.4 ЗАХИСТ ДАНИХ.....	20
2 ПРОЕКТУВАННЯ БАЗ ДАНИХ.....	23
2.1 ЖИТТЄВИЙ ЦИКЛ ІНФОРМАЦІЙНОЇ СИСТЕМИ	23
2.2 КОНЦЕПТУАЛЬНЕ ПРОЕКТУВАННЯ	24
2.2.1. <i>Технологія аналізу предметної області.....</i>	<i>24</i>
2.2.2 <i>Виявлення інформаційних об'єктів та зв'язків між ними..</i>	<i>25</i>
2.2.3 <i>Побудова інформаційної структури</i>	<i>26</i>
2.3. ЛОГІЧНЕ ПРОЕКТУВАННЯ	28
2.3.1. <i>Вибір СКБД.....</i>	<i>28</i>
2.3.2 <i>Відображення концептуальної схеми на логічну схему БД.</i>	<i>30</i>
2.4 ОПТИМІЗАЦІЯ ЗАПИТІВ	33
3 РОЗРОБКА ІНТЕРФЕЙСА КОРИСТУВАЧА.....	35
3.1 ЗАГАЛЬНІ ПРИНЦИПИ РОЗМІЩЕННЯ ІНФОРМАЦІЇ НА ЕКРАНІ.....	35
3.2 ВИДІЛЕННЯ ІНФОРМАЦІЇ НА ЕКРАНІ.....	38
4 ЛАБОРАТОРНІ РОБОТИ.....	40
ЛАБОРАТОРНА РОБОТА № 1 "Створення та редагування електронної таблиці"	40
ЛАБОРАТОРНА РОБОТА №2 "РОБОТА З ФОРМУЛАМИ ТА ФУНКЦІЯМИ"	45
ЛАБОРАТОРНА РОБОТА № 3 "Макроси, звідні таблиці, консолідація даних"	53
ЛІТЕРАТУРА.....	64
Додаток А Словник основних термінів.....	65

ВСТУП

Сучасна економіка неможлива без ефективного управління, яке базується на використанні систем обробки даних, або інформаційних систем.

Проектування інформаційної системи уявляє собою трудомісткий, тривалий і в багатьох випадках неформалізований процес. Це комплексна проблема, яка торкається в кінцевому рахунку не тільки питань обробки даних, але й усієї організації в цілому.

Основним компонентом будь якої системи обробки даних є база даних або знань. Таким чином, якість проектування бази даних впливає на якість інформаційної системи в цілому, що, в свою чергу, впливає на якість прийняття управлінських рішень.

1 ПРОГРАМА ДИСЦИПЛІНИ ТА МЕТОДИЧНІ ВКАЗІВКИ З ЇЇ ВИВЧЕННЯ

1.1 Основи побудови банків інформації

1.1.1 Засоби організації інформаційних масивів. Поняття бази даних.

Відомі два підходи щодо організації інформаційних масивів: файлова організація та організація у вигляді бази даних.

Файлова організація передбачає, спеціалізацію та зберігання інформації, орієнтованої, як правило, на одну прикладну задачу. Така організація дозволяє досягнути високої швидкості обробки інформації, але має певні недоліки.

Базою даних (БД) називається поєднана структурована сукупність взаємозв'язаних даних, що відносяться до конкретної предметної області та призначаються для задоволення інформаційних потреб багатьох користувачів.

Централізований підхід щодо керування даними дозволяє значно скоротити надлишок інформації. Це, в свою чергу, дозволяє усунути протиріччя та забезпечити цілісність даних. Крім того, дякуючи повному контролю над базою даних, з'являється можливість забезпечення мір безпеки баз даних.

1.1.2 Архітектура автоматизованого банку даних. Система управління базою даних. Мова опису даних. Мова маніпулювання даними. Функціонування СКБД. Адміністрування даних. Словник даних. Колектив спеціалістів банку даних.

Автоматизований банк даних (АБД) - це організаційно-технічна (людино-машинна) система, що містить у собі (рисунок 1.1) комплекс спеціальних методів та засобів для підтримки динамічної інформаційної моделі предметної області та забезпечує зберігання, пошук та видавання інформації у вигляді, зручному користувачам.

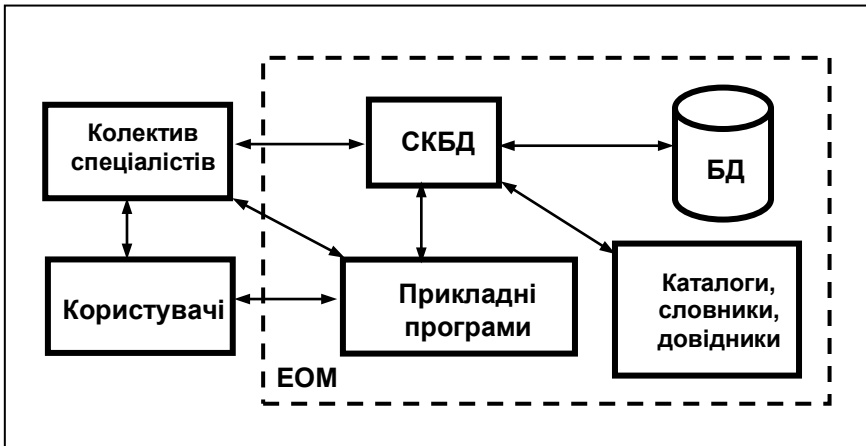


Рисунок 1.1 - Структура автоматизованого банку даних

СКБД - це комплекс програмних та мовних засобів, що забезпечує завантаження інформації до бази даних, реорганізацію та супроводження бази, пошук і перетворення інформації для забезпечення роботи програм користувачів БД.

СКБД грає центральну роль у функціонуванні автоматизованого банку даних. Вона володіє засобами безпосереднього доступу до БД та надає кожній прикладній програмі інтерфейс із БД.

Архітектурно СКБД складається із двох великих компонент: **мови описання даних і мови маніпулювання даними** (рисунок 1.2).

У процесі роботи прикладних програм і термінальних користувачів база даних змінюється. Зазвичай у БД існують достатньо жорсткі обмеження на можливості маніпулювання даними, які відображають закономірності предметної області.

Ці обмеження дозволяють заздалегідь виконати описання незмінних властивостей даних БД. Таке описання отримало назву **описання ст рукт ури даних** або **схема бази даних**.



Рисунок 1. 2 - Архітектура СКБД

МОД - це мова високого рівня декларативного типу, призначена для створення схеми бази даних. За допомогою МОД описуються типи даних, що повинні зберігатися у БД або вибиратися із БД, їх структура та зв'язки між собою.

Вхідні текст (описання даних), що написані на цій мові, після трансляції відображаються до управляючих таблиць, що містять:

У ряді систем розрізняють завдання всієї схеми БД (**МОД**) та завдання її частин - підсхем (**МОП**). Ці мови можуть мати деякі відмінності.

Потрібно особливо відмітити, що МОД дозволяє створити не саму базу даних, а лише її описання. Фактична структура фізичного зберігання даних відома тільки СКБД.

ММД - (або **мова запитів до БД**) являє засіб, який застосовується прикладними програмами або користувачами для виконання операцій над БД. У ньому можуть бути, наприклад, наступні команди:

- виконати запис даного у конкретне поле;
- здійснити вибірку з бази конкретного даного, що задовольняє заданим умовам;
- знайти у базі позицію даного та помістити туди його нове значення або вилучити;
- здійснити сортування даних за заданими критеріями;
- визначити середнє значення даних, що відповідають заданим умовам і т.д.

Запит користувача або прикладної програми до СКБД, що

ініціює виконання операції (сукупності операцій) по обробці даних при зберіганні логічної цілісності БД, називається ***т ранзакцією***.

Література: /6, с.19-29; 7, с.37-63/

1.1.3 Поняття предметної області та проблемної середи. Три рівні уявлення даних. Модель даних. Модель предметної області.

База даних - це деяким чином організована сукупність даних, які відображають стан об'єктів будь-якої предметної області та відносин між цими об'єктами.

Предметною областю називається сукупність об'єктів, предметів реального світу, що розглядаються у межах даного контексту (теорії, моделі, банка інформації та інші). Поняття предмета реального світу в даному визначенні трактується широко. Це може бути об'єкт, процес, явище і т.д.

При описуванні предметної області (ПО) дані прийнято представляти у вигляді трирівневої схеми (рисунок 1.3), запропонованої комітетом **ANSI-SPARK**.



Рисунок 1.3 - Трирівневе представлення даних

Трирівнева модель включає:

- зовнішнє представлення (з точки зору кінцевого користувача та прикладного програміста);
- концептуальне (з точки зору адміністратора);
- внутрішнє (із позиції системного програміста).

Література: /1, с.20-22; 6, с.36-43/

1.1.4 Адміністрування баз даних

Адміністратором бази даних (АБД) називається особа, відповідальна за виконання функції адміністрування бази даних.

АБД повинен координувати дії, що пов'язані із збиранням відомостей, проектуванням та експлуатації БД, а також із забезпеченням захисту даних. АБД повинен враховувати як перспективні, так і поточні інформаційні вимоги предметної області.

До основних функцій АБД можна віднести наступні:

- Визначення концептуальної схеми.
АБД визначає, які саме дані необхідно зберігати в БД про предметну область, тобто виконує концептуальне проектування бази даних.
- Визначення внутрішньої схеми.
Адміністратор виконує також фізичне проектування БД, тобто визначає, яким чином повинні бути представлені дані в БД.
- Взаємодія з користувачем.
АБД допомагає користувачам при складанні зовнішніх схем, а також зіставляє ці схеми з концептуальною схемою БД.
- Визначення правил безпеки та цілісності.
- Визначення процедур резервного копіювання та поновлення.

Література: /1, с.25-27; 6, с.22-45/

1.1.5 Технологія клієнт-сервер. Поняття технології відкритого доступу до даних.

При роботі у комп'ютерній мережі БД може розміщуватися на файл-сервері або декількох файл-серверах. У якості файл-сервера може бути використаний або спеціально виділений комп'ютер, або одна з об'єднаних у мережу найбільш потужних машин.

Функції файл-сервера містяться в основному у зберіганні БД та забезпеченні доступу до них користувачів, що працюють на різних комп'ютерах. Ці функції забезпечуються, як правило, тією ж СКБД, яка працює і на комп'ютерах користувачів.

При невеликих об'ємах даних ця схема в цілому задовольняє всім сучасним вимогам, але зі збільшенням кількості комп'ютерів у мережі або ростом БД починають виникати проблеми, пов'язані з різким падінням продуктивності. Це пов'язане із збільшенням об'єму даних, що передаються по мережі, оскільки вся обробка інформації виконується на комп'ютері користувача.

При використанні технології клієнт-сервер систему баз даних можна розглядати як систему з дуже простою структурою, що складається з двох частин: **Front-end** (набору *клієнтів*) та **Back-end** (*сервера*) (рисунк 1.4).

- **Сервер** - це саме СКБД. Він знаходиться на спеціально виділених комп'ютерах та підтримує всі основні функції СКБД: визначення даних, обробка даних, захист і цілісність даних, адміністрування, тощо.
- **Клієнти** - це різноманітні додатки, що знаходяться на комп'ютері користувача та забезпечують інтерактивний, легкий у використанні, звичай графічний інтерфейс.

При технології клієнт-сервер клієнтський додаток (**front-end**) формує запит до сервера БД (**back-end**), на якому виконуються всі команди. Результати виконання команд посилаються потім клієнту для використання або розглядання.

У теперішній час у якості засобів створення клієнтських додатків можуть використовуватись **Visual FoxPro**, **Visual Basic**, **Access**. В якості сервера можуть використовуватись, наприклад, **Oracle** або **MS SQL Server**.



Рисунок 1.4 - Архітектура клієнт-сервер

Важливим етапом у побудові додатку клієнт-сервер є встановлення зв'язку клієнтського додатку з джерелом даних, що знаходяться на сервері БД. У теперішній час різноманітні засоби розробки використовують декілька технологій забезпечення доступу до даних. Загальновизнаним стандартом є технологія відкритого доступу до даних **ODBC (Open Database Connectivity)**.

ODBC - це загальне визначення мови та набір протоколів. **ODBC** дозволяє додатку клієнта, написаному, наприклад, на **Access** або **Visual FoxPro**, працювати з командами та функціями, що їх підтримує сервер.

Одна з головних цілей створення **ODBC** - приховати складність з'єднання із сервером та по мірі можливості

автоматизувати виконання багатьох процедур, пов'язаних із отриманням даних. **ODBC** вимагає від розробника тільки ім'я джерела даних. При цьому функції драйвера, адреси серверів, мережі та шлюзи, приховані від користувача.

Література: /1, с.198-210; 6, с.48-54, 564-589; 7, с.97-100/

1.1.6 Життєвий цикл інформаційної системи. Етапи проектування банків даних

Життєвий цикл інформаційної системи складається з трьох основних стадій:

- "паперове" проектування;
- програмна реалізація;
- експлуатація.

Основними етапами проектування баз даних являються **інфологічне** та **дат алогічне** проектування, останнє підрозділяється на **логічне** та **фізичне** проектування.

Задача **інфологічного етапу** проектування БД - отримання семантичних (змістових) моделей, що відображають інформаційний зміст конкретної ПО.

Задача **логічного етапу** проектування - організація даних, виділених на попередньому етапі проектування у форму, прийняту у конкретній СКБД.

Задача **фізичного етапу** проектування - вибір раціональної структури зберігання даних і методів доступу до неї, виходячи з методів та засобів, які представляються розробнику конкретною СКБД.

Контрольні питання:

- Поняття організації інформаційних масивів та бази даних?
- З яких компонентів складається АБнД ?
- Яке призначення має словник даних АБнД?

- Місце СКБД в архітектурі АБД.
- Яка різниця між МОД і ММД?
- Яким чином функціонує СКБД?
- Що таке *предметна область*?
- Склад і призначення тривірневої моделі даних.
- Навіщо використовується адміністрування БД?
- Яка різниця між технологіями файл-сервер і клієнт-сервер?
- Поняття технології відкритого доступу до даних.
- Етапи життєвого циклу інформаційної системи.

1.2 Концептуальне проектування баз даних

1.2.1 Мета й задачі концептуального проектування. Основні підходи до проектування баз даних.

В залежності від змісту вхідної інформації, покладеної в основі проектування, розрізняють підходи *"від предметної області і"* та *"від запиту"*.

Підхід *"від предметної області і"* (об'єктний підхід) означає опис об'єктів та зв'язків між ними безвідносно до потреб користувачів.

У підході *"від запиту"* (функціональний підхід) основним джерелом інформації про ПО являються запити користувачів (задачі).

Окремо взятий жоден із указаних підходів не дає достатньої інформації для проектування раціональної БД. Доречно сумісне застосування обох підходів із ведучим положенням об'єктного підходу.

1.2.2 Інструментальні засоби концептуального моделювання: сутності, атрибути, зв'язки.

На етапі концептуального проектування у якості неформальної моделі ПО використовується модель типу *"сутність - зв'язок"* (*ER – Entity-Relation*), що була запропонована Ченом у 1976 р. Конструктивними елементами моделі служать сутності, атрибути та зв'язки. Основним конструктивним елементом являється сутність.

Користувач описує ті об'єкти ПО, що його зацікавили, за допомогою сутностей, потім визначає властивості сутностей, використовуючи атрибути, та, нарешті, описує відповідності між сутностями за допомогою зв'язків.

Сутність - це збиральне поняття, деяка абстракція реально існуючого об'єкта навколишнього світу, процесу чи явища.

Із сутністю пов'язані два поняття: тип сутності та екземпляр сутності. **Тип сутності** визначає набір однорідних об'єктів, що виступають як єдине ціле, а **екземпляр сутності** відноситься до конкретного об'єкта у наборі. Наприклад, типом сутності може бути **СТУДЕНТ**, а екземпляром - Петренко, Сидоренко та інші.

Атрибут - це поійменована характеристика сутності. Атрибут являється засобом, за допомогою якого визначаються властивості сутності.

Для ідентифікації конкретних екземплярів сутностей використовуються спеціальні атрибути - **ідентифікатори (ключі)**. Це може бути один чи декілька атрибутів, значення яких дозволяють беззаперечно розрізняти один екземпляр сутності від іншого. Наприклад, ідентифікуючий атрибут (ключ) **ТАБЕЛЬНИЙ_НОМЕР**.

Зв'язки виступають у моделі в якості засобу, за допомогою якого представляються стосунки між двома і більш сутностями. Розрізняють три види зв'язків, що позначаються **1:1** ("**один до одного**"), **1:M** ("**один до багатьох**") та **M:N** ("**багато до багатьох**").

Під час аналізу можливе виявлення трьох типів зв'язку:

- сутність-сутність;
- сутність-атрибут;
- атрибут-атрибут.

1.2.3 Методика концептуального проектування. Моделювання локальних уявлень.

При моделюванні використовуються наступні основні правила:

- використовуються тільки три типи конструктивних елементів - сутності, атрибути й зв'язки;
- в окремому проектному представленні кожний компонент інформації моделюється лише одним конструктивним

елементом, тобто необхідно уникати надмірності у використанні конструктивних елементів.

Вибір локального представлення залежить від масштабів предметної області. Для зручності проектування в окремому локальному представленні бажано використовувати 6-7 типів сутностей. Частіше за все локальне представлення відповідає окремому зовнішньому додатку, наприклад, окремий функціональний задачі або окремому користувачу. Але воно може відповідати й цілій незалежній області даних, яка використовується декількома додатками.

1.2.4 Об'єднання моделей локальних уявлень. Синтез концептуальної моделі

Після етапу формування локальних представлень здійснюється їх об'єднання в єдину глобальну інформаційну структуру. Ця структура не тільки взаємно відповідає кожному представленню, але й представляє інтегровану БД у стиснутому вигляді.

Існує три основні концепції об'єднання: *ідентичність*, *агрегація* та *узагальнення*.

Два і більше елементів у моделі *ідентичні*, якщо вони мають однакове семантичне (змістове) значення. Для ідентичних елементів не є, однак, обов'язковим наявність однакового синтаксичного представлення.

Агрегація дозволяє розглядати зв'язок між елементами моделі як новий елемент більш високого рівня. Наприклад, елемент **СЛУЖБОВЕЦЬ** може бути сприйнятий як агрегація елементів **ІМ'Я**, **НОМЕР_СТРАХОВОГО_ПОЛІСА** й **АДРЕСА**.

Узагальненням називається абстракція даних, яка дозволяє трактувати клас різних подібних об'єктів як один поименований узагальнений тип об'єкта. В узагальненні підкреслюється загальна природа об'єктів.

Наприклад, клас об'єктів {СОБАКА, КІШКА, СЛОН} може бути узагальнений в об'єкт ТВАРИНА. В цьому узагальненні підкреслюється загальна природа об'єктів, а їх індивідуальні відмінності ігноруються.

Конт рольні пит ання:

- Які існують основні підходи до концептуального проектування баз даних?
- Які конструктивні елементи має ER-модель?
- Що називається *сут ніст ю*?
- Яка різниця між *т ипом сут ност і* та *екземпляром сут ност і*?
- Які типи зв'язків існують?
- Що таке *локальне уявлення*?
- Як об'єднуються локальні уявлення?

1.3 Реляційна модель даних

1.3.1 Поняття моделі даних. Реляційна модель даних. Визначення відношення. Властивості та типи відношень. Основи реляційної алгебри. Операції над відношеннями

Система баз даних підтримує у пам'яті ЕОМ модель предметної області (ПО). Однак результат моделювання залежить не тільки від ПО, але й від СКБД, яка використовується, оскільки кожна система надає свій інструментарій для відображення ПО. Цей інструментарій прийнято називати ***моделлю даних***.

Модель даних, яку підтримує СКБД на логічному рівні, визначається трьома компонентами:

- допустимою структурою (організацією) даних;
- кількістю припустимих операцій над даними;
- обмеженнями цілісності (семантичної).

Модель даних – це сукупність правил формування структури даних у базах даних, операцій над ними, а також обмежень цілісності, яка визначає допустимі зв'язки та значення даних.

У теперішній час відрізняють три основні типи моделей даних: ***ієрархічну, мереживу та реляційну***.

Реляційна модель даних (РМД) уведена ***Коддом (Kodd E.E.)*** у

1970 році. В основі РМД лежить математичне поняття відношення (*relation*).

Відношення - це підмножина декартового добутку доменів.

Домен - це численність елементів, наприклад, численність цілих чисел, численність назв міст, прізвищ студентів і т.д.

Кортеж - це упорядкований набір взаємозв'язаних величин (даних).

Декартовим добутком k доменів $D_1, D_2, D_3, \dots, D_k$

$$D = D_1 \times D_2 \times \dots \times D_k$$

називається множина всіх кортежів виду $d = (d_1, d_2, d_3, \dots, d_k)$ довжини k , де d_1 - елемент із D_1 , d_2 - елемент із D_2 , та d_k - елемент із D_k .

Припустимо, $D_1 = (1, 2, 3)$; $D_2 = (a, b, c)$. Тоді

$$D_1 \times D_2 = \{(1, a), (1, b), (1, c), (2, a), (2, b), (2, c), (3, a), (3, b), (3, c)\}$$

або у матричному вигляді

$$D_1 \times D_2 = \begin{vmatrix} 1, a & 1, b & 1, c \\ 2, a & 2, b & 2, c \\ 3, a & 3, b & 3, c \end{vmatrix}$$

Декартовий добуток дозволяє отримати всі можливі комбінації елементів початкових множин (елементів доменів, які розглядаються).

Підмножина декартова добутку R_n називається **n -місним відношенням**. Число n (кількість атрибутів або стовпців) називається **мірою** відношення.

Число всіх різних кортежів $d = \langle d_1, d_2, d_3, \dots, d_n \rangle$, які утворюють відношення R_n , називається **потужністю** цього відношення.

Міра відношення зазвичай не змінюється після утворення відношення, але потужність буде колихатися по мірі додавання нових або знищення старих кортежів.

Кожне відношення в БД зберігається в окремому файлі та надається у вигляді двомірної таблиці, в якій рядок є кортеж (запис). Стовпці відношень називаються атрибутами. Атрибути іменуються.

Список імен атрибутів відношення називається схемою відношення. Схема відношення R , яка має атрибути $A_1, A_2, \dots, A_K$, позначається $R(A_1, A_2, \dots, A_K)$.

Реляційна база даних (РБД) - це набір екземплярів кінцевих відношень. Схему РБД можна уявити у вигляді сукупності схем відношень.

Відношення реляційної БД, у залежності від змісту, поділяють на два класи: **об'єктні відношення** та **зв'язні відношення**.

Об'єктне відношення зберігає дані про об'єкти (екземпляри сутності). В об'єктному відношенні один з атрибутів однозначно ідентифікує окремий об'єкт. Такий атрибут має назву **ключовий**, або **первинний атрибут**. Решта атрибутів функціонально залежать від цього ключа. Ключі бувають прості або складові.

В об'єктному відношенні не повинно бути рядків з однаковими ключами, тобто не повинно бути дублювання об'єктів. Це **основне обмеження реляційної моделі** для забезпечення цілісності даних.

Зв'язне відношення зберігає ключі двох або більше об'єктних відношень, тобто по ключам встановлюються зв'язки між об'єктами відношень.

Реляційною алгеброю (алгеброю відношень) називають систему операцій маніпулювання відношеннями, кожний оператор якої в якості операндів має одне або більше відношень.

До таких операції належать: **об'єднання**, **перехрещення**, **різниця**, **декартовий добуток**, **ділення**, **проекція**, **з'єднання** та **селекція**.

При виконанні операції об'єднання, перехрещення та різниці відношення, які приймають у них участь, повинні задовольняти властивості сумісності по об'єднанню.

Два відношення **сумісні по об'єднанню** (сумісні за типом - К.Дейт), якщо вони мають одну й ту же кількість атрибутів і однакові типи відповідних атрибутів.

Література /1, с.47-65; 5; 6, с.79-200/

1.3.2 Типи функціональних залежностей. Нормалізація баз даних. Проектування реляційних баз даних.

Відношення реляційної БД містять як структурну, так і семантичну (сміслову) інформацію. Структурна інформація задається схемою відношень, а семантична виявляється функціональними зв'язками між атрибутами, які враховуються в схемі.

Склад атрибутів відношень БД повинен задовольняти двом основним вимогам:

- між атрибутами не повинно бути небажаних функціональних залежностей;
- групування атрибутів повинне забезпечувати мінімальне дублювання даних, забезпечувати їх обробку та оновлення без труднощів.

Задовольняння цих вимог досягається нормалізацією відношень. **Нормалізація відношень** - це покроковий оборотний процес декомпозиції (розкладу) початкових відносин БД на інші, більш дрібні та прості відношення. При цьому встановлюються (з'ясовуються) всі можливі функціональні залежності.

Апарат нормалізації був розроблений Коддом. В ньому визначаються різні **нормальні форми** (усього п'ять). Кожна нормальна форма обмежує типи припустимих функціональних залежностей відношення.

Термін "**функціональна залежність**" має таке значення: атрибут ***V*** відношення ***R*** функціонально залежить від атрибута ***A*** того ж відношення, якщо в кожний момент часу кожному значенню атрибута ***A*** відповідає тільки одне значення **атрибута *V***, зв'язаного з ***A*** у відношенні ***R***:

$$f : A \rightarrow V.$$

Наприклад, **ПОСАДА** → **ОКЛАД**; **ТАБ._НОМЕР** → **ПРИЗВИЩЕ** та інші.

Література /1, с.104-120; 3, с.48-52; 5; 6, с. 259-299/

Контрольні питання:

- Що входить до поняття *моделі даних*?
- Визначення відношення.
- Які типи відношень існують?
- Яка різниця між первинним, вторинним та зовнішніми ключами?
- Що таке *зв'язне відношення*?
- Які існують операції реляційної алгебри?
- Що таке *функціональна залежність*?
- Що таке *нормальна форма* та які види нормальних форм існують?

1.4 Захист даних

Відновлення бази даних. Транзакції. Відновлення транзакцій. Відновлення системи. Паралелізм. Проблеми паралелізму. Блокування об'єктів. Вирішення проблем паралелізму. Безпека. Вибіркове та обов'язкове управління доступом. Шифрування даних. Цілісність даних. Обмеження цілісності. Потенційні та зовнішні ключі.

Відновлення в системі управління базами даних означає, в першу чергу, відновлення самої бази даних, тобто повернення бази даних у *вірний стан*, якщо який-небудь збій зробив поточний стан невірним або підозрілим. Основний принцип, на якому будується таке відновлення, - це **надмірність**, яка реалізується на фізичному рівні.

Транзакція – це логічна одиниця роботи, яка виконується без порушення логічної цілісності бази даних. Якщо в процесі виконання транзакції виникла помилка виконання (наприклад, арифметичне переповнення та інші), то система, яка підтримує процес транзакції, гарантує повернення до первинного стану. Таким чином, транзакція або виконується повністю, або повністю скасовується.

Якщо транзакція не досягне запланованого завершення, система забезпечить для неї виконання операції **відновлення транзакції**.

Відновлення системи виконується як частина процедури

перезавантаження (також називається *перезавантаження/відновлення*). При цьому система, аналізуючи останній **запис контрольної точки**, система визначає, яку роботу необхідно виконати повторно, а яку відмінити. Записи контрольних точок заносяться до **журналу (файлу) реєстрації** у відповідній послідовності.

Система також забезпечує відновлення носіїв інформації шляхом відновлення бази даних зі спеціалізованої резервної копії (дамп) з наступним використанням журналу та повторним виконанням роботи, яка була завершена після збереження резервної копії.

Термін "**паралелізм**" означає можливість одночасної обробки в СКБД багатьох транзакцій з доступом до одних і тих же даних, причому в один і той же час.

Для таких систем для коректної обробки паралельних транзакцій без виникнення конфліктних ситуацій необхідно використовувати деякий **метод управління паралелізмом**.

Виникаючи при цьому проблеми можуть бути розв'язані шляхом використання **методу блокування**. Існують два основних типи блокування: з **можливістю взаємного доступу** (S-блокування) та без **можливості взаємного доступу** (X-блокування)

Терміни "**безпека**" та "**цілісність**" у контексті обговорення баз даних часто використовуються разом, хоча це зовсім різні поняття.

Термін "**безпека**" відноситься до захисту даних від несанкціонованого доступу, зміни або руйнуванню даних, а "**цілісність**" – до точності або істинності даних.

Інакше їх можна описати наступним чином:

- під **безпекою** розуміють, що користувачам *дозволяється* виконувати деякі дії;
- під **цілісністю** розуміють, що ці дії виконуються *коректно*.

Безпека забезпечується підсистемою безпеки СКБД, котра перевіряє відповідність усіх запитів правилам безпеки, які зберігаються в системному каталозі.

Обмеження цілісності підрозділяються на:

- обмеження цілісності атрибутів;
- обмеження цілісності відношень;
- обмеження цілісності бази даних.

Література /1, с.216,246; 276-285; 6, с.354-430; 7, с.566-642/

Контрольні питання:

- Які методи захисту даних використовують у базах даних?
- Що називається транзакцією?
- Як у базах даних підтримується цілісність даних?
- Яким чином відбувається адміністрування бази даних?

2 ПРОЕКТУВАННЯ БАЗ ДАНИХ

2.1 Життєвий цикл інформаційної системи

Життєвий цикл інформаційної системи (ІС) можна розбити на три загальні стадії:

- *“наперове” проектування;*
- *програмна реалізація;*
- *експлуатація.*

На стадії проектування ІС проектувальник повинен виконати наступну роботу:

- а) обстежити предметну область автоматизації.
- б) розробити концептуальну модель предметної області.
- в) розробити технологію обслуговування ІС, тобто визначити порядок зберігання даних у БД, частоту та формати вводу-виводу даних, правила роботи всіх груп користувачів. Проект повинен забезпечувати простоту та зручність експлуатації ІС, захист даних від некоректних поновлень користувачами та від зруйнувань при збоях комп'ютера.
- г) вибрати інструментальні засоби (конкретну СКБД) для реалізації.
- д) перевірити коректність проекту. Він повинен адекватно, на відповідному рівні детальності, відображати предметну область (ПО). Виділеним об'єктам та процесам ПО повинні відповідати дані та процедури обробки в комп'ютері.

На стадії програмної реалізації необхідно виконати такі пункти:

- а) описати засобами СКБД і ввести до комп'ютера схему усіх відношень;
- б) розробити інтерфейси користувачів з БД. Необхідно розробити екранні форми для вводу та відображення даних, зручні екранні засоби звернення та доступу до даних БД , порядок уводу та поновлення даних; визначення розмірів та складу порцій даних, які одночасно відбиваються на екрані, порядок їх розміщення. Кожна

картинка на екрані повинна забезпечувати максимальну інформативність і зручність сприйняття, будувати звичне для користувача середовище;

- в) розробити програмне забезпечення ІС для всіх додатків;
- г) заповнити ІС даними для відлагодження та відлагодити її;
- д) провести тестування системи та скоректувати технологію її обслуговування;
- є) скласти необхідні інструкції до системи.

Стадія експлуатації починається з наповнення системи реальними даними, після чого безпосередньо використовується ІС, підтримується її функціонування. Стадія експлуатації може вмішувати вдосконалення системи та розробку нових додатків у випадку розвитку та зміни предметної області.

2.2 Концептуальне проектування

2.2.1. Технологія аналізу предметної області

Аналіз предметної області являє собою перший етап проектування БД будь-якого типу і закінчується побудовою інформаційної структури (концептуальної моделі). На цьому етапі аналізуються потреби користувачів, вибираються інформаційні об'єкти та їх характеристики, й на підставі проведеного аналізу структурується предметна область. Аналіз ПО є загальним етапом незалежним від програмного та технічного середовища, в яких буде реалізовуватися БД.

Аналіз предметної області доцільно розбити на три фази:

- *аналіз концептуальних вимог та інформаційних потреб;*
- *виявлення інформаційних об'єктів і зв'язків між ними;*
- *побудова концептуальної моделі ПО та проектування концептуальної схеми БД.*

На етапі аналізу концептуальних вимог та інформаційних потреб необхідно вирішити такі завдання:

- *аналіз вимог користувачів до БД (концептуальних вимог);*
- *виявлення існуючих задач по обробці інформації (перспективних додатків);*
- *документування результатів аналізу.*

Вимоги користувачів до БД, що розробляється, являє собою список запитів із зазначенням їх інтенсивності та обсягів даних.

На цьому етапі також з'ясовуються вимоги до виводу, поновленню та корекції інформації. Вимоги користувача уточнюються та доповнюються при аналізі існуючих та перспективних напрямлень.

2.2.2 Виявлення інформаційних об'єктів та зв'язків між ними

Друга фаза аналізу предметної області складається з вибору інформаційних об'єктів, завдання необхідних якостей для кожного об'єкта, виявлення зв'язків між об'єктами, визначення обмежень, які накладаються на інформаційні об'єкти, типи зв'язків між ними, характеристики інформаційних об'єктів.

Під час вибору інформаційних об'єктів бажано відповісти на ряд запитань:

- На які класи можна розподілити дані, що підлягають зберіганню у БД ?
- Яке ім'я можна надати кожному класу даних?
- Які найбільш цікаві характеристики (з точки зору користувача) кожного класу даних можна виділити?
- Які імена можна присвоїти вибраним наборам характеристик?

Виділення інформаційних об'єктів - процес інтерактивний. Він здійснюється на підставі аналізу інформаційних потоків та інтерв'ювання споживачів. Характеристики інформаційних об'єктів визначаються тими ж методами.

Далі виділяються зв'язки між інформаційними об'єктами. Під час цього процесу потрібно спромогтися відповісти на наступні запитання:

- Які типи зв'язків між інформаційними об'єктами?
- Яке ім'я можна присвоїти кожному типу зв'язків?

- Які можливі типи зв'язків можна використовувати?
- Чи мають сенс які-небудь комбінації типів зв'язків?

При необхідності потрібно завдати обмеження на об'єкти, їх характеристики та зв'язки, даючи відповіді на такі запитання:

- Яка область значень для числових характеристик?
- Які функціональні залежності між характеристиками одного інформаційного об'єкта?
- Який тип відбиття відповідає кожному типу зв'язків?

Під обмеженням цілісності (внутрішньої єдності), як правило, розуміють логічні обмеження, які пред'являються до даних. Обмеження цілісності - це така властивість, яку ми задаємо для будь-якого інформаційного об'єкта або його характеристик і яка повинна зберігатися для кожного стану об'єкта.

Наприклад, твердження, що оклад професора ВНЗ більш ніж оклад доцента, може розглядатися як обмеження цілісності лише у тому випадку, якщо воно слухне в будь-який момент і не залежить від процедури підвищення заробітної плати, або зміни, наприклад, тарифної сітки.

В загальному випадку розглядаються три види обмежень: внутрішні, явні та змішані. Внутрішні обмеження в основному пов'язані з моделлю даних, яка підтримується СКБД.

2.2.3 Побудова інформаційної структури

Заключна фаза аналізу ПО - проектування її інформаційної структури (концептуальної схеми, або концептуальної моделі). Концептуальна модель використовується для структурування ПО з урахуванням не тільки інформаційних інтересів користувачів системи, але й інформаційних потреб самої предметної області.

В рамках кожної БД концептуальні вимоги узагальнюються в концептуальну модель, яка виражається абстрактними засобами, що дозволяє побачити весь інформаційний зміст предметної області.

При концептуальному проектуванні як основна модель використовується модель типу "сутність - зв'язок" (ER-модель). На мові ER-моделі інформаційний об'єкт називають сутністю. При цьому

розрізняють тип сутності та екземпляр сутності. Під типом сутності розуміють набір подібних об'єктів, які виступають як ціле. Поняття “екземпляр сутності” відноситься до конкретного об'єкта.

Наприклад:

тип сутності - СЛУЖБОВЕЦЬ,

екземпляр сутності - ПЕТРЕНКО, ІВАНЕНКО та ін.

Сутність слід визначати поймаєнованими характеристиками, які називаються атрибути. Найменування повинно бути унікальним для кожного екземпляру сутності, хоча воно може повторюватись для різних типів сутностей.

В кожному наборі атрибутів, які характеризують сутність, необхідно вибрати ключеві атрибути, тобто атрибути, які однозначно ідентифікують кожний екземпляр сутності.

Після визначення сутностей та атрибутів виявляються залежності між двома і більше сутностями, сутностями та атрибутами, зв'язки атрибутів між собою.

Далі визначається, які зв'язки необхідні, а які надмірні. Для виявлення зв'язків визначаються їх характеристики, кожний тип зв'язку ” сутність - сутність”, “сутність - атрибут”, “ атрибут - атрибут” іменується.

Моделювання ПО закінчується графічним оформленням усіх виявлених сутностей, зв'язків між ними, атрибутів та складанням усіх специфікацій.

При моделюванні предметної області проектувальник розбиває її на ряд локальних областей, моделює кожне локальне представлення, а потім їх об'єднує.

Вибір локального представлення залежить від масштабів ПО. Для зручності проектування в окремому локальному представленні бажано використовувати 6 - 7 типів сутей. Частіше всього локальне представлення відповідає окремому зовнішньому додатку, наприклад, окремому функціональному завданню або окремому користувачу. Але воно може відповідати й цілій незалежній області даних, яка використовується декількома додатками.

При виборі області для локального представлення проектувальник відшукує компромісні рішення, оскільки вузька область призводить до зниження рівня інтеграції даних та їх

дробленню, а широка область - до нечіткості й складності проектування.

2.3. Логічне проектування

Основне завдання логічного проектування - це розробка логічної схеми, яка зорієнтована на обрану БД. Процес логічного проектування складається із таких етапів:

- Вибір конкретної СКБД;
- Відображення концептуальної схеми на логічну схему БД;
- Проектування запитів;
- Конструювання форм вводу та виводу;
- Проектування звітної документації.

2.3.1. Вибір СКБД

При виборі СКБД, яка реалізує конкретну БД, виділяють два підходи до її оцінювання. Перший підхід пов'язаний з поглядом користувача на конкретну систему, а другий пов'язаний із продуктивністю системи. Беручи в рахунок цю систему, розглянемо сім груп параметрів для характеристики та подальшого вибору СКБД.

Дані параметри не претендують на вичерпну повноту, проте дають достатньо повну картину про можливості СКБД:

1. Загальні характеристики:

- тип комп'ютера;
- максимальне число записів у файлі;
- максимальна ємність файлу;
- максимальне число копій в запису;
- максимальне число символів на запис;
- максимальне число індексів на файл;
- максимальне число таблиць та операцій об'єднання;
- максимальне число файлів, які доступні для однієї команди;

- максимальне число одночасно відкритих файлів;
- максимальне число файлів в БД;
- максимальне число записів у БД;
- максимальне число змінних;
- максимальне число символів у одному полі;
- імпорт - експорт даних;
- тип внутрішньої моделі даних;
- фірма - виробник;
- версія.

2. Керування файлами та пошук:

- тип зв'язку;
- модифікація декількох файлів;
- двунаправлене поєднання таблиць;
- мова маніпулювання даними;
- тип пошуку.

3. Засоби підтримки додатків:

- каталог даних;
- генератор додатків, процедурна мова і підпрограми;
- макроси;
- відлагоджувач;
- система підтримки виконання;
- шифрування програм та даних;
- розмежування доступу;
- графіка;
- графіка кольорова, текстовий редактор;
- статистика.

4. Ввід та підтримка цілісності:

- керування за допомогою команд;
- перевірка цілісності за таблицею;
- перевірка унікальності ключа;

- перевірка за датою;
- незалежність даних.

5. Звіти:

- звіти за декількома файлами;
- зберігання форматів звітів;
- видача звіту на екран;
- видача звіту на магнітний носій;
- поля, які обчислюються;
- перевизначення формату даних;
- генератор звітів;
- підсумкові поля;
- максимальна ширина звіту;
- звіт у матричній формі.

6. Операційне середовище:

- тип операційної системи;
- ємність потребуємої оперативної пам'яті;
- мова системи.

7. Додаткові відомості:

- наявність варіанту мережі;
- вартість.

2.3.2 Відображення концептуальної схеми на логічну схему

БД

Для відбиття інформаційної схеми, отриманої на етапі концептуального проектування, на логічну схему БД необхідно знати:

- розмір БД (число сутностей та атрибутів);
- частоту звернень до БД (число виконання додатків за одиницю

часу);

- правила побудови логічних схем БД за допомогою обраної СКБД;
- перелік додатків з вказівками на використані дані;
- конфігурацію, швидкодію та ємність пам'яті, яка надається у розпорядження проектувальнику технічних засобів.

При відображенні інформаційної схеми кожний прямокутник схеми для ER-діаграми предметної області відображається у таблицю, яка являє собою одне відношення. При цьому потрібно враховувати обмеження на розмір таблиць, котрі накладає конкретна СКБД.

Припустимо, що на етапі концептуального проектування було складено дворівневе меню.

I рівень

1. Споживач
2. Вироб
3. Склад
4. Виробник
5. Заявка

II рівень

- | | |
|-----------------------|---------------|
| 1. Споживач | 2. Вироб |
| 1.1. Назва | 2.1. Назва |
| 1.2. Адреса | 2.2. Вартість |
| 1.3. Телефон | 2.3. Артикул |
| 1.4. Профіль магазину | 2.4. Шифр |
| | 2.5. Матеріал |
| 3. Склад | 4. Виробник |
| 3.1. № складу | 4.1. Назва |
| 3.2. Адреса | 4.2. Адреса |
| 3.3. Телефон | 4.3. Телефон |
| 3.4. Профіль | 4.4. Профіль |
| 5. Заявка | |
| 5.1. № заявки | |

5.2. Дата

5.3. Назва виробу

Інформаційні об'єкти (сутності) будуть відповідати назвам першого рівня меню.

Зробимо відбиття концептуальної схеми на реляційну схему та зробимо оцінювання потрібної ємності пам'яті для зберігання БД. Як результат отримаємо наступні відношення:

1. Відношення СПОЖИВАЧ

назва	адреса	телефон	профіль магазину
20	30	7	23

На одну таблицю необхідно $20+30+7+23=80$ байт. Всього споживачів 200, тобто на зберігання всіх таблиць з даним відношенням потрібно $80 \times 200 = 16000$ байт.

2. Відношення ВИРОБ.

назва	вартість	артикул	шифр	матеріал
30	5	5	5	15

На одну таблицю необхідно $30+5+5+5+15=60$ байт. Всього виробів 400, тобто на зберігання всіх таблиць з даним відношенням потрібно $60 \times 400 = 24000$ байтів.

3. Відношення СКЛАД.

№ складу	адреса	телефон	профіль
5	18	7	30

Одна таблиця займає 60 байт. Всього складів 40, разом $60 \times 40 = 2400$ байт.

4. Відношення ВИРОБНИК

назва	адреса	телефон	профіль
15	18	7	30

Одна таблиця займає 70 байт. Всього виробників 15, разом $70 \times 15 = 1050$ байт.

5. Відношення ЗАЯВКА

№ заявки	дата	назва виробу
5	6	30

Одна таблиця займає 41 байт. Всього заявок 1000, разом $41 \times 1000 = 41000$ байт.

Отже, в першому наближенні на зберігання БД необхідно $16000 + 2400 + 1050 + 41000 = 84450$ байт.

Після визначення всіх відношень необхідно провести нормалізацію БД, яка полягає в тому, щоб зробити декомпозицію похідних відношень БД на більш прості відношення. Нормалізація ліквідує небажані функціональні залежності між атрибутами та забезпечує мінімальне дублювання даних за рахунок раціонального групування атрибутів.

2.4 Оптимізація запитів

Як основні стратегії оптимізації можна запропонувати такі:

1. Виконувати операцію селекції як можна раніше. Це перетворення запитів більше, чим щось інше, впливає на економію часу виконання, оскільки воно зменшує кількість проміжних обчислень.

2. Доцільно спочатку обробити файли перед виконанням з'єднання (або, що еквівалентно, декартового добутку з наступною селекцією). За процедуру попередньої обробки доцільно використовувати сортування файлів та їх індексування.

3. Шукати загальні підвирази у виразах. Якщо результат

загального підвиразу являє собою невелике відношення, яке можна прочитати із зовнішньої пам'яті за значно менший час, ніж потрібно для його обчислення, доцільно спочатку обчислити цей загальний підвираз.

4. Збирати каскади селекції та проекції. Послідовність таких операцій, кожна з яких вимагає тільки одного операнда, може виконуватися одночасно, за один перегляд файлу операнда.

5. Комбінувати деякі селекції з попереднім декартовим добутком та виконувати замість них з'єднання. Поєднання може бути виконано ефективніше декартова добутка тих же самих відношень.

3 РОЗРОБКА ІНТЕРФЕЙСА КОРИСТУВАЧА

Інтерфейс системи "людина-комп'ютер" забезпечує безпосередню взаємодію користувача з комп'ютером при рішенні задач. Це дає можливість користувачу визначати, які завдання слід зробити активними в даний момент, як передавати їм дані для обробки та приймати результат. З точки зору програмного забезпечення до складу інтерфейсу входять дві складові: набір процесів вводу-виводу та процес діалогу.

Користувач обчислювальної системи взаємодіє з інтерфейсом, надсилаючи через нього вхідні дані та приймаючи вихідні. Процеси, які пов'язані з виконанням завдань викликаються інтерфейсом у потрібний момент часу. Тому, з точки зору користувача, інтерфейс - це основний процес, а процеси, що виконують різні завдання, являються неосновними, або, як називають їх багато авторів, фоновими.

Людина, як активний суб'єкт керування, є джерелом глобальних помилок у системі "людина-комп'ютер" внаслідок обмеженості своїх психофізичних можливостей. Тому комплексний підхід до розробки інтерфейсу користувача повинен бути орієнтований на урахування функціональних, психологічних, соціальних та естетичних факторів.

3.1 Загальні принципи розміщення інформації на екрані

Густина розміщення даних - поняття суб'єктивне. Вона залежить від конкретного користувача та задачі. Наприклад, дані на екрані, які призначені для роботи досвідченого оператора, розміщуються тісніше, ніж дані, які призначені для роботи недосвідченого користувача. Але існують деякі правила, які регулюють густину розміщення даних на екрані за допомогою інтервалів:

- залишайте порожнім приблизно половину екрану;
- залишайте порожнім рядок після кожного п'ятого рядка таблиці ;
- залишайте 4 або 5 пропусків (Space) між стовпчиками

таблиці .

Не треба механічно слідувати цим правилам. Вони визначають лише загальні принципи. Іноді можна порушувати ці принципи, але потрібно це робити розумно.

Якщо дані виводяться на декілька екранів, то ділити їх потрібно так, щоб зберігся чіткий кордон розділу (якщо такий є). Не можна порушити логічні зв'язки в групі даних , треба також розташовувати на одному екрані дані , необхідні для прийняття конкретного рішення.

На будь-якому екрані логічно зв'язані дані повинні представлятися окремою групою. Цього можна досягти, залишаючи по декілька пропусків з кожного боку групи або проводячи горизонтальні чи вертикальні лінії, або використовуючи різні атрибути для полів різних груп .

Фрагменти тексту повинні розташовуватися на екрані так, щоб погляд користувача сам переміщався по екрану в потрібному напрямку.

Кількість блоків повинна бути достатньою. Зміст полів не повинен наближатися до краю екрану, а розташовуватися біля горизонтальних або вертикальних осів.

Меню, маючи відносно невеликий об'єм інформації, повинно бути розташоване у верхній лівій частині екрану.

Щоб підкреслити симетрію, зміст або назви полів, які відносяться до однієї групи, повинні вирівнюватися по вертикалі. За можливістю необхідно вирівнювати всі логічно зв'язані групи даних.

Потрібно визначити точку початку відліку. Природніше почати з верхнього лівого кута та переміщуватись зліва направо та згори вниз (напрямок "головної діагоналі"). Хоча розташування матеріалу на екрані може задавати іншу послідовність переміщення, загальноприйняту послідовність слід міняти тільки обґрунтовано. Один і той же тип інформації повинен з'являтися завжди в одному й тому ж місці.

Якщо розташування даних потрібно узгодити як у рамках однієї програми, так і між декількома програмами, їх необхідно розташовувати за єдиним шаблоном, як показано на рис. 3.1.

Верхні два або три рядки екрану зарезервовані для відображення заголовку та стану системи . Назва часто описує пакет програм або конкретну задачу цього пакету , який виводить дані на екран. Поточну дату та час можна вивести в праву частину екрану. Прийнято також присвоювати кожному екрану свій номер. Якщо

користувач має деякі проблеми, то цей номер дозволить обслуговуючому персоналу системи краще уявити собі суть проблеми. Назва показує, де в рамках системи знаходиться користувач.

В ієрархічно побудованому меню область для виводу стану системи може показувати пункти меню верхнього рівня. Крім того, в області стану системи може виводитись підтвердження про те, що система працездатна.

Заголовок та дані про стан системи
Верхня область для виводу повідомлень
Основна область
Нижня область для виводу повідомлень
Опис функціональних клавіш та "запасний люк"

Рисунок 3.1 - Приблизний шаблон екрану

Області в верхній та нижній частинах екрану, призначені для виводу підказок користувачеві або індикації особливих станів, використовувати необов'язково. Підказки, які визначають способи роботи з інформацією на екрані, розташовуються у верхній частині екрану, а підказки, які пояснюють, що треба робити далі, - в нижній.

Будь-яку з цих областей можна використовувати для відображення довідкової інформації та повідомлень про помилки. Прийнято відображати довідкові повідомлення у верхню частину екрану, а повідомлення про помилки - у нижню. Повідомлення, які вимагають відповіді користувача, краще розташовувати в нижній частині екрану.

Основна область екрану містить головну інформацію. Для меню - це список варіантів. Заголовок меню може розташовуватись або в верхній, або в основній частині екрану. При заповненні форми - ця частина екрану використовується для виводу назв та значень полів. У змішаних діалогах, наприклад, в електронних таблицях, в основній області розташовані чарунки таблиці.

Один або два рядки в нижній частині екрану відводяться для виводу назв керуючих клавіш та їх функцій, які відносяться до будь-якого екрану. Області під повідомлення призначені для виводу інформації, яка відноситься до основної частини даного екрану. Іноді ця область екрану має назву "запасний люк", бо в цій області описаний засіб переривання процесу обробки.

3.2 Виділення інформації на екрані

Виділення інформації - це використання таких атрибутів, які дозволяють звернути увагу користувача до конкретної області екрану. Зрозуміло, що увага користувача може бути звернена на обмежену кількість областей, тому виділення багатьох областей може його тільки заплутати.

Спочатку розроблювач повинен розташувати інформацію на екрані без виділення полів, потім він повинен продивитись, який ефект має виділення того чи іншого поля. Якщо система виводить тільки знайому користувачеві інформацію, великої потреби в виділених полях немає.

До атрибутів полів відносять наступні: колір символів, колір фону, рівень яскравості та режим мигтіння. Вимкнув будь-який з них, ми досягнемо різного ефекту. Часто їх сполучення може викликати непередбачений ефект. Щоб не знизити вплив цих засобів, завжди необхідно використовувати мінімальну їх кількість. Декілька атрибутів одночасно потрібно використовувати лише в тому разі, якщо треба звернути особливу увагу до поля.

Блимаючий фон - це самий сильний засіб привертання уваги. Різні кольори спектру сприймаються по-різному. Області, фон яких представлений більш теплими відтінками в червоній області спектру, уявляються крупнішими, ніж області, колір фону яких зміщений в блакитну область. Області екрану на білому фоні або на фоні кольору, який знаходиться у середній області спектру, уявляються яскравішими та легше сприймаються при різному зовнішньому освітленні. Найбільший контраст між двома областями екрану досягається, коли колір фону однієї з них чорний або близький до одного з кінців спектру, а другого - білий або взятий з середини спектру. Ті ж розмірковування вірні й для відношення між кольорами переднього

плану та фону .

Користувачам подобається кольорове зображення. Люди можуть розрізняти тисячі відтінків кольору, але одночасно працювати лише з декількома. Неможливо переносити правила кольорового друку на дисплейні форми. Основний ефект від кольорового друку матеріалу досягається завдяки використанню напівтонів та відтінків кольору. Більшість кольорових моніторів не може відображати всієї гами кольорових відтінків, можливих на папері. Крім того, на різних екранах один й той же колір має різні відтінки. Деякі комбінації кольору, наприклад, блакитний колір символів на червоному фоні, небажані для очей. Кращий спосіб довідатись, який вигляд буде мати колір, - це подивитись його на екрані.

Узагальнені рекомендації по роботі з кольором:

- ✓ *Використовуйте мінімальну кількість кольорів, не більш трьох або чотирьох на одному екрані.*
- ✓ *Для великих прямокутників використовуйте колір фону.*
- ✓ *Використовуйте яскраві кольори для виділення даних і більш спокійні тони для фону.*
- ✓ *Для виділення двох областей використовуйте для однієї чорний колір або колір із одного краю спектру, а для іншої області візьміть білий колір або колір з середини спектру.*
- ✓ *Використовуйте колір у відповідності з уявленнями користувача про нього.*
- ✓ *Проекспериментуйте з різними кольоровими відтінками на реальному екрані.*

4 ЛАБОРАТОРНІ РОБОТИ

Лабораторна робота № 1 "Створення та редагування електронної таблиці"

1 Мета роботи

Метою роботи є ознайомлення з пакетом MS Excel, а також надбання початкових навичок у створенні та редагуванні електронних таблиць.

2 Завдання до лабораторної роботи

2.1 Ознайомитися зі змістом основних пунктів головного меню, а також призначенням клавіш стандартної панелі приладів та панелі форматування.

2.2 У відповідності до індивідуального завдання сформуванати структуру таблиці та виконати необхідні дії щодо її редагування.

2.3 Ввести дані до таблиці.

3 Основні теоретичні відомості та вказівки щодо виконання роботи

3.1 Книги, сторінки, адреса чарунок.

В якості первинного документа для зберігання та обробки даних **Excel** використовує сторінку, що називають також електронною таблицею. Сторінка є частиною книги. Кожну сторінку можна представити із власною назвою у окремому вікні.

На одній сторінці розміщуються 256 стовпців та 16384 рядків (в **Excel 97** – 65536 рядків). Перетин рядків і стовпців утворює чарунки, які призначені для вводу до них відповідної інформації.

Адресу чарунки можна задати двома способами:

- у стилі **A1** для позначення стовпців використовують літери;
- у стилі **R1C1** рядки та стовпці позначаються цифрами, де **R1** –

номер рядку, а **C1** – номер стовпця.

Вибір стилю позначення чарунок виконується за допомогою опції **“Общие”** підпункту **“Параметры”** пункту **“Сервис”** головного меню.

3.2 Виділення чарунок

Виділення в **Excel** використовується перед виконанням будь-яких операцій над даними: копіювання, переміщення, вилучання та інші.

Для **виділення стовпця** необхідно натиснути ліву кнопку миші на його заголовок.

Рядок виділяється натисненням на його номер.

Дві або більше чарунок сторінки утворюють діапазон. Чарунки діапазону можуть бути як суміжними, так і несуміжними.

Діапазон суміжних чарунок визначається адресами лівої верхньої і правої нижньої чарунок, між якими ставлять двокрапку (наприклад, **A3:C4**).

Діапазон чарунок можна виділити наступним чином:

- мишею, переміщуючи її з натиснутою лівою кнопкою;
- мишею, вказавши на першу виділену чарунку і, після натиснення клавіші **[Shift]**, на останню. Для переміщення до останньої чарунки при виділенні великого діапазону можна використовувати полоси прокрутки ;
- з клавіатури, тримаючи затиснуту клавішу **[Shift]** та розширюючи клавішами керування курсором зону виділення.

Для виділення несуміжних чарунок чи несуміжного діапазону чарунок необхідно виділити першу чарунку або діапазон чарунок, після, тримаючи натиснуту клавішу **[Ctrl]**, виділити останні чарунки у діапазоні.

3.3 Вставка чарунок

Вставити рядок можна за допомогою:

- пункту **“Вставка”** головного меню;
- команди **“Добавить”** контекстного меню виділеного рядка. Для цього необхідно натиснути мишкою на номері рядка, перед

яким повинна бути виконана вставка, потім натиснути праву кнопку миші і в контекстному меню обрати команду **"Добавить"**.

- клавіші **[Ctrl]** та знаку **"+"**.
Рядок, який був виділений перед вставленням, як і всі наступні рядки, зміститься вниз на одну позицію.

Формули та функції, що містять посилання на зміщення чарунки, будуть автоматично модифіковані з урахуванням вставки

Для вставки кількох рядків необхідно попередньо виділити стільки рядків, скільки треба вставити.

Вставка стовпців у таблицю виконується аналогічно.

3.4 Вилучання чарунок

Вилучити рядок можна за допомогою:

- пункту **"Правка"** головного меню;
- команди **"Удалить"** контекстного меню виділеного рядка. Для цього необхідно натиснути мишку на номері рядку, який вилучається, а потім натиснути праву кнопку миші та у контекстному меню обрати команду **"Удалить"**;
- клавіш **[Ctrl]** та знаку **"_"**.

Усі рядки, що знаходяться під вилученим рядком, перемістяться вгору на одну позицію.

Формули та функції, що містять посилання на зміщені чарунки, будуть автоматично актуалізовані (модифіковані) з урахуванням вставки.

Для вилучання кількох рядків необхідно попередньо виділити стільки рядків, скільки потрібно вилучити.

Якщо необхідно вилучити тільки зміст виділених рядків (без вилучання самих чарунок), то необхідно натиснути клавішу **[Delete]** або у контекстному меню (правою кнопкою миші) обрати команду **"Очистить содержимое"**.

Вилучання стовпців та областей чарунок виконується аналогічно.

3.5 Копіювання, вставка та вилучання інформації

Копіювання або перенесення змісту чарунки (групи чарунок) можна виконати за допомогою буфера обміну, використовуючи кнопки стандартної панелі пристроїв (таким чином, як у редакторі **Word**), або за допомогою клавіатури.

В першому випадку звичайним чином виділяються чарунки, потім за допомогою кнопок **"Копировать в буфер"** або **"Удалить в буфер"** зміст цих чарунок переміщується в буфер, після чого за допомогою кнопок **"Вставка из буфера"** зміст буфера копіюється до активної чарунки.

Таблиця 3.1 - Комбінації клавіш

Комбінації клавіш для обробки чарунок:	
Копіювання виділеної області в буфер обміну	[Ctrl+C] або [Ctrl+Insert]
Вирізання та розташування виділеної області у буфер обміну	[Ctrl+X] або [Shift+Delete]
Вставка змісту буфера обміну в активну чарунку	[Ctrl+V] або [Shift+Insert]

Перехід у режим редагування змісту чарунки виконується клавішею **F2**.

Для копіювання змісту виділеної чарунки у сусідні чарунки необхідно показчик миші, що має форму широкого білого хрестика, помістити на чорний квадратик у правому нижньому кутку виділеної чарунки. Як тільки показчик змінить форму на суцільний чорний хрестик, необхідно натиснути ліву кнопку миші і, не відпускаючи її, "протягнути" по чарункам, в які копіюється інформація.

При цьому, якщо в початковій чарунці зберігався текст або число, вони копіюються без зміни, якщо формула – вона контекстно модифікується, тобто змінюється у відповідності з адресою нової чарунки.

3.6 Вилучання тільки постійних значень

Для зберігання формул та вилучання тільки постійних значень (констант) необхідно:

- ✓ Виділити область.
- ✓ У головному меню вибрати пункт **"Правка"** та у ньому команду **"Перейти"** (**Ctrl+G** або **F5**).
- ✓ Натиснути кнопку **"Виделить"**.
- ✓ Активувати перемикач **"Константы"**.
- ✓ Натиснути кнопку **"ОК"**. При цьому Excel маркірує усі чарунки виділеної області, що містять константи. Чарунки з формулами не виділяються.
- ✓ У меню **"Правка"** викликати команду **"Все"** із підменю **"Очистить"** або натиснути клавішу [**Delete**].

Аналогічним чином, використовуючи один із перемикачів діалогового вікна **"Выделение группы ячеек"**, можна вилучати інші частини виділеної області.

4 Контрольні запитання

- ✓ Як виділити область чарунок за допомогою миші?
- ✓ Як виділити область чарунок за допомогою клавіатури?
- ✓ Як додатково виділити другу область за допомогою миші чи клавіатури?
- ✓ Склад та призначення панелі інструментів Excel?
- ✓ Як налагодити панель інструментів?
- ✓ Виклик та призначення контекстного меню.
- ✓ Яким чином можна вказати адресу чарунки?
- ✓ Як скопіювати зміст чарунки?
- ✓ Як вилучити чарунку (групу чарунок)?
- ✓ Як вилучити зміст чарунок?
- ✓ Як вилучити з чарунок постійну інформацію, зберігши формули?
- ✓ Як можна перейти від одного стилю позначення чарунок до іншого?

Лабораторна робота №2 “Робота з формулами та функціями”

1 Мета роботи

Метою роботи є надбання практичних навичок організації обчислень за допомогою пакету **MS EXCEL**.

2 Завдання на лабораторну роботу

2.1 Ознайомитись зі змістом розділу 3 наданих методичних вказівок, а також з роботою ***Майстра функцій*** стандартної панелі інструментів.

2.2 У відповідності до індивідуального завдання розробити електрону таблицю і виконати відповідні обчислення.

2.3 Оформити результат обчислень у вигляді діаграми.

2.4 Для розробленої таблиці виконати сортування даних по окремим полям.

2.5 Відфільтрувати дані відповідно до одержаного завдання.

3 Основні теоретичні відомості та вказівки по виконанню роботи

3.1 Формули та функції

Формула - це вказівка для Excel виконати визначені обчислення. Формула може містити числові значення, адреси чарунок, функції, об'єднані відповідними математичними та логічними операторами.

Якщо формула повинна містити посилання на чарунок, або декілька чарунок, то її можна ввести без клавіатури. Для цього потрібно встановити курсор вводу у тому місці формули, у якому потрібно розмістити посилання, і натиснути на мишку на чарунці, яка повинна бути вказана у посиланні. Аналогічно можна вставити посилання на декілька чарунок.

Функції призначені для виконання стандартних обчислювань. Excel пропонує більше 200 вбудованих функцій, які можна вживати

окремо, або включати у формули.

Перед кожною функцією, яка не являється частиною формули, повинен стояти знак “дорівнює”. Після імені функції необхідно вказати у дужках числа або посилання на чарунки, використанні у обчисленнях.

3.2 Оператори для позначення головних операцій

Для позначення головних операцій використовуються оператори:

- **Арифметичні** оператори використовуються при роботі з числами. У результаті виконання арифметичних операцій завжди одержується число.
- Оператори **порівняння** використовуються для порівняння двох чисел. У результаті виконання операції порівняння одержується логічне значення **істина** або **брехня**.
- **Текстовий** оператор вживається для означення операцій об'єднання декількох послідовностей символів у одну послідовність.
- **Адресні** оператори вживаються при посиланні на чарунки.

Таблиця 3.1 – Клавiші для правки чарунок або рядку формул

Операція	Клавiші
Почати формулу	Знак “дорівнює”
Перейти до режиму правки змісту чарунки	F2
Очистити рядок формул після вказівки чарунки	<i>Backspace</i>
Скопіювати формулу (значення) верхньої чарунки у поточну чарунку або в рядок формул	Ctrl + '

Таблиця 2.2 – Оператори, які використовуються в Excel

<i>Арифметичний оператор</i>	<i>Значення</i>
+	Додавання
-	Віднімання, або від’ємне число, наприклад, -1
/	Ділення
*	Множення
%	Відсоток
^	Зведення у ступень
<i>Оператор порівняння</i>	<i>Значення</i>
=	Дорівнює
>	Більше
<	Менше
>=	Більше, або дорівнює
<=	Менше, або дорівнює
<>	Не дорівнює
<i>Текстовий оператор</i>	<i>Значення</i>
&	Об’єднання послідовностей символів у одну послідовність
<i>Адресні оператори</i>	<i>Значення</i>
Діапазон (двокрапка)	Посилання на всі чарунки діапазону, включаючи межі
Об’єднання (кома)	Посилання на об’єднання чарунок діапазонів
Перехрестя (пробіл)	Посилання на загальні чарунки діапазонів

3.3 Посилання на чарунку

У формулі може бути вказано посилання на іншу чарунку або діапазони чарунок. Чарунка, яка містить формулу, називається **залежною** чарункою. Її значення залежить від значення іншої чарунки. Формула може повернути інше значення, якщо змінити чарунку, на яку формула посилається.

Частина формули, що являє собою адресу чарунки та

посилання на дану чарунку незалежно від положення формули, називається **абсолютним посиланням** на чарунку. Абсолютне посилання, окрім номера стовпця та номера рядка, додатково містить символ долара \$, наприклад **\$A\$1**, **\$B\$1**. Абсолютне посилання не змінюється при перенесенні або копіюванні формул у інше місце.

Якщо положення чарунки вказується відносно активної чарунки (чарунки вводу), то таке посилання називається **відносним**. У такому випадку положення потрібної чарунки визначається відносною відстанню у рядках та стовпцях до неї від чарунки вводу. При зміні положення чарунки вводу, змінюються і чарунки, з яких беруться дані.

ПРИКЛАДИ RC – чарунка вводу (активна чарунка):

RC[-1] - чарунка, що знаходиться зліва від чарунки вводу;

RC[2] – друга праворуч чарунка від чарунки вводу;

R[1]C – чарунка, яка знаходиться знизу від чарунки вводу;

R[-3]C – третя зверху чарунка від чарунки вводу.

При абсолютному посиланні вказується точна адреса чарунки в таблиці, а при відносному посиланні вказується відстань від чарунки вводу.

При копіюванні формули (див. Лабораторну роботу №1), яка містить відносні адреси у стилі A1, ці адреси змінюються. При копіюванні чарунки, яка містить посилання RC[1] у стилі R1C1, на один рядок вниз, то посилання залишається таке саме (продовжує вказувати на одну чарунку праворуч).

Якщо, наприклад, чарунка, що містить посилання A1, копіюється на один рядок вниз, то адреса змінюється на A2. У стилі R1C1 відбувається те ж саме, але адреси чарунок не змінюються, змінюється чарунка, відносно якої вони обчислюється.

Щоб посилатися на діапазон чарунок, треба ввести посилання на лівий верхній кут діапазону (натиснути мишку на потрібній чарунці), поставити у формулі двокрапку (:), а потім – посилання на правий нижній кут діапазону. Якщо формула посилається на декілька діапазонів чарунок, то діапазони відокремлюються один від одного

знаком “;”, наприклад, (A1:A10; B2:B6; C17).

Можливо посилатися на чарунки, які знаходяться на інших сторінках книги або в інших книгах. Посилання на чарунки інших книг називаються **зовнішніми посиланнями**. При цьому ім'я файлу робочої книги заключається у квадратні дужки, а ім'я листа відокремлюється від імені чарунки знаком оклику, наприклад, [Book2]Лист1!\$C\$3.

Зовнішнє посилання можна задати також наступним чином:

- ✓ завантажити книгу, на чарунку якої повинно бути виконано посилання;
- ✓ виділити чарунку і скопіювати у буфер її зміст;
- ✓ перейти на книгу, яка повинна містити посилання;
- ✓ виділити чарунку, у якій буде міститися посилання, та викликати контекстне меню (правою кнопкою мишки);
- ✓ обрати пункт “**Специальная вставка**”;
- ✓ у вікні, що з'явилося, натиснути на кнопку “**Вставить связь**”

При використанні у формулі декількох функцій, вони відокремлюються символами “.”, “пробіли” у формулі не ставляться, за винятком тексту.

3.4 Логічна функція **ЯКЩО**

Часто у розрахунках використовується логічна функція **ЯКЩО**, працююча наступним чином:

= **ЯКЩО(Умова**; виконується одна операція при відповіді “**так**”; виконується інша операція при відповіді “**ні**”).

ПРИКЛАД У формулі =ЯКЩО(RC[-1]>0;”Плюс”;”Мінус”) виконується перевірка знака числа у сусідній зліва чарунці та в залежності від знаку у поточну чарунку записується текст “Плюс” або “Мінус”

Замість операцій, що виконуються при істинності або брехливості умови, можна використовувати вкладені функції ЯКЩО.

Увага! Кількість вкладених функцій ЯКЩО не повинна перевищувати 7.

3.5 Функція автосумування

Функції можна вводити прямо у рядок формул, тобто безпосередньо у чарунку або за допомогою Майстра функцій на стандартній панелі інструментів. Крім того, для зручності роботи виділена функція автосумування, яку можна викликати за допомогою відповідної кнопки “ Σ ” також на стандартній панелі. Функцію автосумування можна викликати також комбінацією клавіш [**ALT+=**].

3.6 Автоматичне заповнення чарунок

При необхідності вводу послідовностей числових значень, отриманих шляхом додавання до числа певного значення (кроку), немає необхідності вводити всі значення з клавіатури. **Функція автоматичного заповнення** допоможе прискорити цю процедуру:

- ✓ Введіть два перших значення послідовності та виділіть область, що їх містить.
- ✓ Встановіть покажчик миші на маркері заповнення, розташованому у правому нижньому куті виділеної області.
- ✓ Пересунути покажчик миші при натиснутій лівій кнопці через чарунки, які необхідно заповнити.
- ✓ Після виділення необхідної області відпустіть ліву кнопку миші.
- ✓ Визначивши крок за першими двома значеннями, Excel автоматично заповнить інші чарунки виділеної області елементами послідовності.

При частому застосуванні функцій, Майстер функцій може бути викликаним за допомогою клавіш [**Shift+F3**].

3.7 Виправлення помилок

Якщо при виконанні розрахунків у чарунці з'являється повідомлення **#ЗНАЧ!**, **#ИМЯ**, або інші, які починаються із символу #, то це означає, що програма не може знайти початкові дані або у формулі припущені помилки.

При вводі формули Excel автоматично перетворює у прописні всі літери назв та посилань, якщо вони введені без помилок.

Вводіть назви функцій у формулах, не використовуючи прописні літери, щоб швидко знаходити помилки вводу.

При вводі складних формул велика ймовірність помилки. Клавiша [F9] дозволяє її легко локалізувати.

Активізуйте чарунку, у котрій знаходиться формула. Виділіть праву частину формули (наприклад, вираз, що визначає суму) та натисніть клавішу [F9]. Якщо цей вираз буде перетворено у числове значення, помилку треба шукати у іншій частині формули.

Таким чином можна визначити частину формули, яка містить помилку. Перевіряючи частини формули, враховуйте дужки. Отримати початковий вид формули можна клавішею [Esc].

Наприклад, ми ввели формулу: $=A1*A2*(A3-A4)/A5-A6*A7$.

При перевірці цю формулу можна розбити на три частини: $(A1*A2)$; $(A3-A4)$; $(A5-A6*A7)$. Цю формулу можна розбити і на більші частини, однак доцільно перевіряти мінімальні частини формули.

За допомогою клавіші [F9] можна швидко знайти помилку у формулі. Клавіша [Esc] встановлює початковий вигляд формули.

У випадку видачі повідомлення про помилку локалізувати її можна також за допомогою опції **“Источник ошибки”** підпункту **“Зависимости”** пункту **“Сервис”** головного меню.

4 Контрольні запитання

- ✓ Що таке зовнішнє посилання?
- ✓ У чому різниця між абсолютним та відносним посиланнями?
- ✓ Якою буде адреса C3 в стилі R1C1?
- ✓ Якою буде адреса \$A\$4 в стилі R1C1?
- ✓ Якою буде адреса R3C8 в стилі A1?
- ✓ Як буде виглядати адреса R[-3]C[1] в стилі A1 відносно чарунки R4C1?
- ✓ Як можна локалізувати помилки у формулах?
- ✓ Як можна викликати Майстра функцій, як з ним працювати?
- ✓ Яким чином можна задати функцію автосумування чарунок стовпця або рядка?
- ✓ Як працює логічна функція ЯКЩО?
- ✓ Як здійснити посилання на чарунки іншої таблиці або книги?
- ✓ Як здійснити сортування або фільтрацію даних?

Лабораторна робота № 3 "Макроси, звідні таблиці, консолідація даних"

1 Мета роботи

Метою роботи є отримання практичних навиків створення та використання макросів та роботи зі звідними таблицями за допомогою пакета **MS Excel**.

2 Завдання до лабораторної роботи

2.1 Ознайомитися зі змістом розділу 3 методичних вказівок.

2.2 У відповідності до індивідуального завдання створити макроси та перевірити їх виконання.

2.3 На основі індивідуальної таблиці створити звідну таблицю.

2.4 Створити декілька ідентичних по формі, але різних за змістом таблиць та зробити консолідацію даних.

3 Основні теоретичні відомості та вказівки до виконання роботи

3.1 Робота з макросами

Макрос - це макропрограма, яка написана на мові **Visual Basic for Application (VBA)** та призначена для виконання сукупності операцій, які часто використовуються (форматування таблиць, настройка печаті і т.д.). Виклик макроса робиться комбінацією клавіш, пунктом меню або кнопкою панелі інструментів.

Існує два способи створення макросів:

- ✓ використати вбудований засіб запису макросів;
- ✓ ввести **VBA**-код макроса безпосередньо у редакторі **Visual Basic**.

Один з найбільш важливих етапів у створенні макроса - спланувати, що цей макрос повинен робити. Після включення запису макроса кожна дія, яку виконує користувач, буде записана. Якщо дії

сплановані неточно, макрос може вміщувати багато непотрібних операцій, що може призвести до загальмування роботи макроса або навіть до появи небажаних побічних ефектів.

Для запису макроса треба виконати наступні дії:

- ✓ відкрити потрібну робочу книгу та активізувати робочий лист, де буде записаний макрос;
- ✓ обрати команду **Сервис⇒Макрос⇒Начать запись**. З'явиться діалогове вікно Запис макроса;
- ✓ ввести ім'я та опис макроса, а також сполучення клавіш;
- ✓ обрати місце зберігання макроса. В даному випадку обирайте **Эта книга** зі списку **Сохранить в**;
- ✓ клацнути на кнопці **ОК**, щоб почати запис макроса. Діалогове вікно зачиниться, та з'явиться панель інструментів **Остановить запись**;
- ✓ виконати заплановану дію;
- ✓ клацнути на кнопці **Остановить запись** панелі інструментів **Остановить запись** або обрати команду **Сервис⇒Макрос⇒Остановить запись**.

Призначення макроса кнопці панелі інструментів

Для цього спочатку на панелі інструментів необхідно створити налагоджуєму кнопку, а потім цій кнопці назначити потрібний макрос:

- ✓ оберіть команду **Сервис ⇒Настройка**;
- ✓ клацнути на вкладці **Команды**. Із списку **Категории** оберіть елемент **Макросы**;
- ✓ пересуньте зображення налагоджуваної кнопки у потрібне місце панелі інструментів;
- ✓ не зачиняючи вікно **Настройка**, клацніть правою кнопкою миші по налагоджуваній кнопці. З'явиться контекстне меню;
- ✓ оберіть команду **Назначить макрос**. З'явиться діалогове вікно **Назначить макрос**;
- ✓ оберіть макрос, який треба назначити для кнопки, а потім клацніть на кнопці **ОК**.

Малюнок кнопки та її підпис можна у будь-який момент

відредагувати. Для цього треба визвати вікно **Настройка** та, не зачиняючи його, клацнути правою кнопкою миші по налагоджуємій кнопці; у з'явившомуся контекстному меню обрати відповідні пункти.

Увага! При зберіганні макроса *обов'язково* вказувати *Эта книга* зі списку *Сохранить в.*

3.2 Звідні таблиці

3.2.1 Поняття звідної таблиці

Звідні таблиці є дуже потужним інструментом в арсеналі засобів швидкого аналізу складних даних. Звідні таблиці дозволяють підводити підсумки та різноманітними способами аналізувати дані в робочих листах та таблицях. Звідними таблиці називаються тому, що користувач може легко змінити розміщення полів в них та продивитись інформацію з іншої точки зору.

Звідна таблиця є своєрідним уявленням набору даних. Поля, які маютьесь в початкових даних, можуть сумуватись різноманітними способами. Крім того, в звідних таблицях можна створювати обчислюємі поля, визначати складні правила додавання або обмежити певними вимогами дані звідної таблиці, щоб представити їх різними способами, що дозволяє значно спростити аналіз складних даних.

Звідні таблиці можна створювати із декількох джерел. За умовчанням звідна таблиця створюється зі списку або бази даних Excel. Крім того, можна створити звідну таблицю із зовнішнього джерела даних, із декількох консолідованих діапазонів або з іншої звідної таблиці.

3.2.2 Створення звідної таблиці

Роздивимось створення звідної таблиці на основі початкової таблиці, представленої на рисунку 3.1.

Microsoft Excel - Приклад

Файл Правка Вид Вставка Формат Сервис Данные Окно ?

Arial Cyr 12 Ж К Ч А %

I20 =

Ведомость № 10
поставки товаров на склад

материал	код матер.	поставщик	дата прих.	Цена в \$	Цена в грн	кол-во	НДС	сумма	Несклада
краска	666	АО мотор	01.02.97	11	39,49	10	7,898	473,88	7
олифа	7777	АО мотор	07.мар	5	17,95	15	3,59	323,1	9
паркет	887	ООО промкаб	12.сен	15	53,85	100	10,77	6462	7
цемент	543	ЧП стройка	04.май	4	14,36	50	2,872	861,6	1
кирпич	87	МП дом	10.июн	6	21,54	75	4,308	1938,6	3
обои	342	АО склад	01.май	8	28,72	25	5,744	861,6	3
обои	111	АО склад	01.май	9	32,31	25	6,462	969,3	3
панели	456	МП дом	03.июл	11	39,49	80	7,898	3791,04	4
сумма в пр.								15681,1	
сумма в USD								4368	

курс \$	3,59
дата	01.06.98
%уценки	10

Лист1 Лист2 Лист4 Лист3 Лист6 Лист5 Лист7

Рисунок 3.1 – Початкова таблиця

Процес створення звідної таблиці включає чотири етапи роботи засобів *Майстра звідних таблиць*.

Для запуску Майстра оберіть команду *Данные⇒Сводная таблица*. З'явиться вікно *Мастер сводных таблиц - шаг 1 из 4* (рисунок 3.2).

В цьому вікні треба обрати джерело даних, котрі необхідно проаналізувати за допомогою звідної таблиці. Можна обрати одну з існуючих можливостей:

- *В списке або базі даних Microsoft Excel.* Встановіть цей перемикач, якщо дані вже знаходяться у робочому листі.
- *У зовнішньому джерелі даних.* Ця опція запускає програму *Microsoft Query* для витягання даних із зовнішньої бази даних.

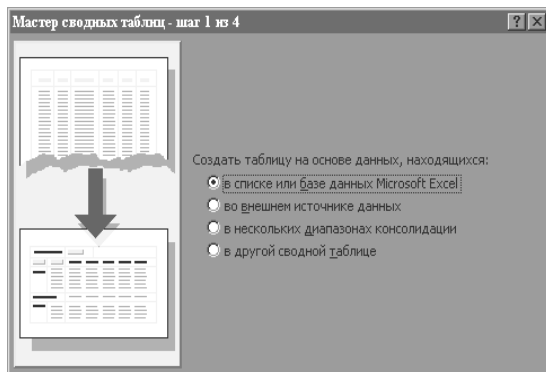


Рисунок 3.2 - Перший крок Майстра звідних таблиць

- **У декількох діапазонах консолідації.** Якщо є декілька робочих листів, котрі організовані однаковим чином та мають однакові заголовки полів, то можна скористатись даною опцією для того, щоб консолідувати та проаналізувати дані за допомогою звідних таблиць.
- **У іншій звідній таблиці.** У звідній таблиці зберігається копія аналізуємих даних для підвищення ефективності та більш економного використання пам'яті. Якщо необхідно створити декілька звідних таблиць на основі одного й того ж набору даних, використовуйте цей перемикач, щоб не допускати дублювання даних та не використовувати зайвих системних ресурсів.

На другому кроці треба обрати діапазон чарунок, на основі яких буде створена звідна таблиця. Для цього необхідно на початковому листі виділити потрібний діапазон та у вікні **Мастер сводних таблиць - шаг 2 из 4** (рисунок 3.3) натиснути кнопку **Далее**.

Якщо перед запуском засобів Майстра звідних таблиць була активізована одна з чарунок списку, то на другому кроці діапазон буде виділений автоматично.

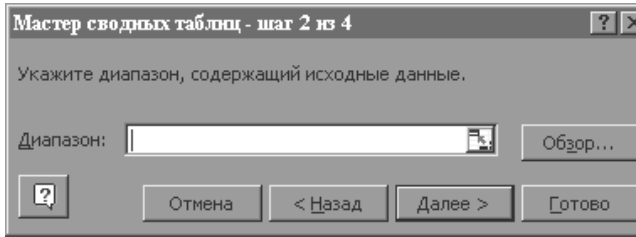


Рисунок 3.3 - Другой krok Мастера сводных таблиц

З'явиться діалогове вікно *Мастер сводных таблиц - шаг 3 из 4* (рисунок 3.4), у якому необхідно прийняти основні рішення по розташуванню полів у звідній таблиці. При цьому треба пересунути назви полів з правої області діалогового вікна у відповідні області звідної таблиці. Після цього натиснути кнопку *Далее*.

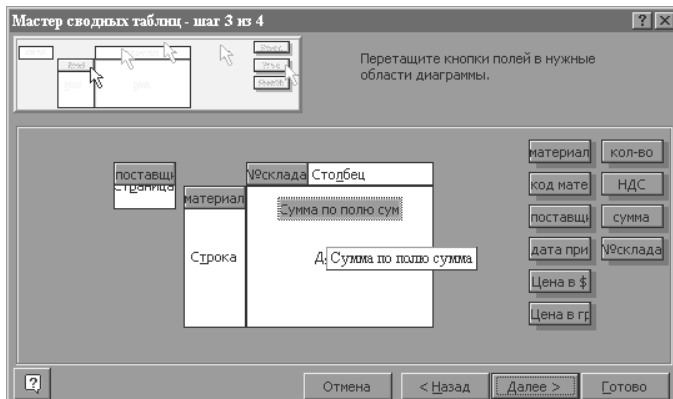


Рисунок 3.4 - Вікно формування структури звідної таблиці

На останньому кроці Майстра звідних таблиць необхідно визначити, де буде розташована звідна таблиця (рисунок 3.5).

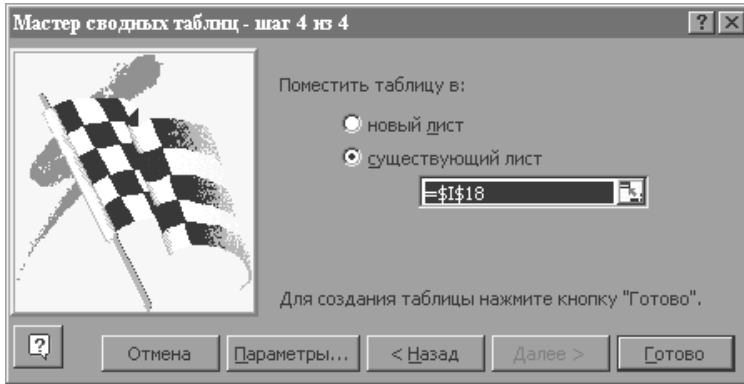


Рисунок 3.5 - Четвертый krok Майстра звідних таблиць

Остаточный вид одного з варіантів звідної таблиці зображений на рисунку 3.6.

Microsoft Excel - Пример1

Файл Правка Вид Вставка Формат Сервис Данные Окно ?

Arial Cyr 10 Ж К Ч А Масштаб

A3 = Сумма по полю сумма

	A	B	C	D	E	F	G
1	поставщик	(Все)					
2							
3	Сумма по полю сумма	Насклада					
4	материал	1	3	4	7	9	Общий итог
5	кирпич		1938,6				1938,6
6	краска				473,88		473,88
7	обои		1830,9				1830,9
8	олифа					323,1	323,1
9	панели			3791,04			3791,04
10	паркет				6482		6482
11	цемент	861,6					861,6
12	Общий итог	861,6	3769,5	3791,04	6935,88	323,1	15681,12
13							
14							
15							

Лист1 Лист9 Лист2 Лист4 Лист3 Лист6 Лист5

Рисунок 3.6 - Звідна таблиця

В отриманій звітній таблиці по стовпцям згруповані дані відповідають складам, рядки відповідають матеріалам. Цю таблицю можна уявити у більш компактному вигляді. Для цього необхідно виконати подвійне клацання по виділеній кнопці *Месклада* та у з'явившомуся діалоговому вікні *Вычисление поля сводной таблицы* (рисунок 3.7) обрати опцію *Ориентация* ⇨ *По листам*.

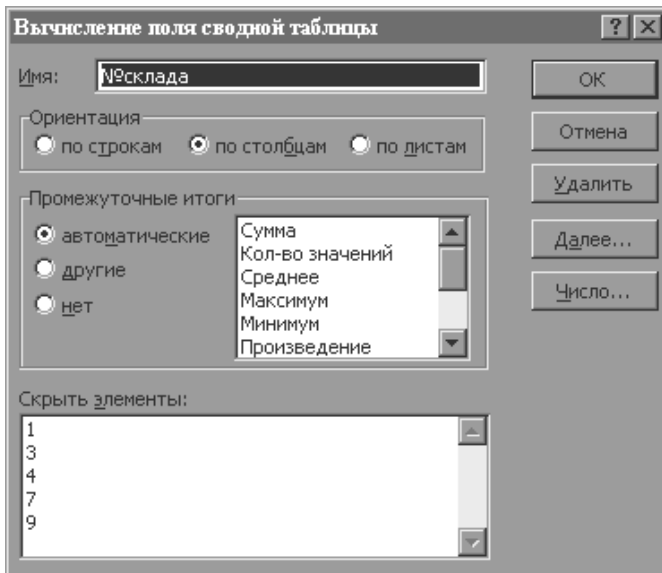


Рисунок 3.7 - Вікно Обчислення полів звітної таблиці

Після цього з'явиться нова кнопка *Номер склада*, а всі стовпці звітної таблиці згорнуться у один підсумковий. У цьому стовпці можна продивитися підсумкову інформацію як по окремим складам, обираючи номер складу, так і загальний підсумок по всім складам, обираючи "Все". Аналогічно можна згорнути в один рядок рядки матеріалів з можливістю продивлятися інформацію як по всім, так і по окремим матеріалам.

Якщо після створіння звітної таблиці були внесені зміни в початкову таблицю, то для їх фіксації у звітній таблиці необхідно в пункті *Данные* головного меню обрати опцію *Обновить данные*.

3.2.3 Додавання обчислюємих полів

Можна створювати нові поля звідної таблиці на основі обчислювань та додавати їх до звідної таблиці. Обчислюємі поля базуються на існуючих полях та на введеній формулі. Щоб здійснити таке перетворення, виконайте наступні дії:

- ✓ Клацніть правою кнопкою миші з середини звідної таблиці. Оберіть з контекстного меню команду **Формулы** ⇒ **Вычисляемое поле**. З'явиться діалогове вікно **Вставка вычисляемого поля** (рисунок 3.8).

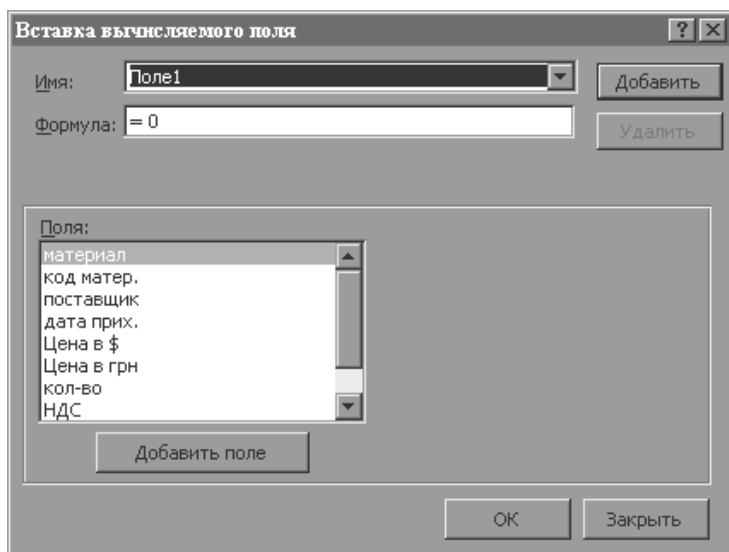


Рисунок 3.8 - Вікно вставки обчислюємого поля

- ✓ Введіть ім'я нового поля в полі **Имя**.
- ✓ В полі **Формула** введіть формулу, яка буде використовуватися для обчислювання значень в новому полі. В цьому полі можна використовувати будь-які функції **Excel**. Для додавання назви поля звідної таблиці в формулу виділіть це поле в списку **Поле** та клацніть на кнопці **Добавить поле**.

- ✓ Клацніть на кнопці **Добавить** для створення нового поля.

Наприклад, можна створити нове обчислюєме поле за допомогою формули **=СРЗНАЧ(Количество)** для обчислювання середньої кількості товарів в кожному замовленні, а потім використовувати це нове поле в звітній таблиці.

3.3 Консолідація даних

Консолідація даних використовується у тих випадках, коли треба об'єднати в одну таблицю інформацію, яка зберігається у різних листах **Excel**. Наприклад, якщо дані об'ємах продажу по філіалам знаходяться на різних листах та повинні бути зведені воедино для створення звіту, треба використати можливість об'єднання у єдиній таблиці даних з декількох діапазонів консолідації.

Для консолідації даних з декількох таблиць необхідно:

- ✓ Відкрити новий лист.
- ✓ Виконати команду **Данные** ⇒ **Консолидация...** З'явиться діалогове вікно **Консолидация** (рис. 3.9).

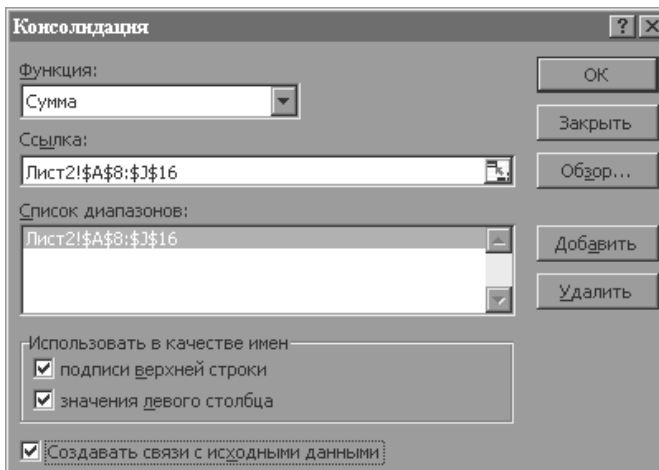


Рисунок 3.9 - Вікно Консолідації

- ✓ У вікні **Функція** обрати функцію, яка буде виконуватися над чарунками таблиці, яка створюється.
- ✓ У вікні **Ссылка** вказати перший діапазон консолідації. Для цього, не зачиняючи вікно **Консолидация**, перейти на лист, який містить першу із об'єднуємих таблиць, виділити необхідний діапазон і у вікні **Консолидация** натиснути кнопку **Добавить**.
- ✓ Аналогічні дії виконати для останніх початкових таблиць.
- ✓ Встановити прапорці **“Подписи верхней строки”**, **“Значения левого столбца”** та **“Создавать связи с исходными данными”**.



4 Контрольні питання

- ✓ Що собою представляє макрос?
- ✓ Для чого використовуються макроси?
- ✓ Які бувають макроси?
- ✓ Послідовність запису макроса.
- ✓ Як задати кнопку для виклику макроса?
- ✓ Що уявляє собою звідна таблиця?
- ✓ Для чого використовуються звідні таблиці?
- ✓ Послідовність дій по створенню звідних таблиць.
- ✓ Які види джерел даних використовуються при створенні звідних таблиць?
- ✓ Яким чином виконують згортку інформації у звідних таблицях?
- ✓ Як оновлюються дані у звідній таблиці після внесення змін у початкову таблицю?
- ✓ Що таке обчислюємі поля, як вони створюються?
- ✓ Як консолідувати дані, які зберігаються в різних таблицях?

ЛІТЕРАТУРА

1. Базы данных: модели, разработка, реализация/ Т.С.Карпова. – СПб:Питер, 2001. – 304 с.
2. Боуман Дж., Эмерсон С., Дарновски М. Практическое руководство по SQL. - К: Диалектика, 1997. - 320 с.
3. Горев А., Ахаян Р., Макашарипов С. Эффективная работа с СУБД. - С.-Петербург:Питер, 1997. - 700 с.
4. Грабер М. Введение в SQL. - М: Лори, 1996. - 379 с.
5. Джексон Г. Проектирование реляционных баз данных для использования с микроЭВМ. - М.: Мир, 1991.-252с.
6. Дейт К. Введение в системы баз данных. - К:Диалектика,1998. - 781с.
7. Конноли Т., Бегг К., Страчан А. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. – Москва, С.-Петербург, Киев: Изд. дом "Вильямс", 1999. – 1120 с.



Додаток А Словник основних термінів

Адміністратор бази даних – особа, яка відповідає за виконання функцій *адміністрування* бази даних, тобто координацію дій по зборі відомостей, проектуванню та експлуатації бази даних, а також по забезпеченню захисту даних.

Альтернативний ключ – атрибут (або група атрибутів), які не співпадають із первинним ключем і унікально ідентифікують кожний рядок у таблиці.

Атрибут – інформаційне відображення властивостей об'єкту.

База даних – поєднана структурована сукупність взаємозв'язаних даних, які відносяться до конкретної предметної області та призначені для задоволення інформаційних потреб багатьох користувачів.

Безпека БД - захист від несанкціонованого доступу, зміни або руйнування даних.

Домен – набір значень елементів даних одного типу, який відповідає поставленим умовам.

Екземпляр сутності – опис конкретного об'єкту в наборі.

Зв'язок – функціональна залежність між сутностями.

Зв'язне відношення – відношення, яке зберігає ключі двох або більше об'єктних відношень, за якими встановлюється зв'язок між цими відношеннями.

Ключовий елемент даних – елемент, по якому можна визначити значення інших елементів даних.

Курсор (віртуальна таблиця) – об'єкт, який не містить власних даних. Це свого роду віртуальна таблиця, що не існує як незалежний об'єкт в базі даних і вміст якої береться з інших таблиць шляхом виконання запиту.

Нормалізація відношень – процес побудови оптимальної структури таблиць і зв'язків у реляційній базі даних.

Об'єкт – елемент інформаційної системи, інформацію про який зберігається.

Об'єктне відношення – відношення, яке зберігає дані про об'єкти (екземпляри сутності) предметної області.

Первинний ключ – атрибут (або група атрибутів), які однозначно ідентифікують кожний рядок у таблиці.

Посилальна цілісність – забезпечення співвідношення значень зовнішнього ключа екземпляра дочірній сутності значенням первинного ключа в батьківській сутності.

Предметна область – частина реальної системи, яка представляє цікавість для даного дослідження.

Словник даних – централізоване сховище відомостей про об'єкти, елементи даних, які їх складають, взаємозв'язки між об'єктами, їх джерела, значення, використання та формати представлення.

Ступінь відношення – кількість атрибутів (стовпців) відношення.

Сутність – це збиральне поняття, деяка абстракція реально існуючого об'єкта навколишнього світу, процесу або явища.

Таблиця (відношення) – деяка регулярна структура, яка складається з кінцевого набору однотипних записів.

Тип сутності – набір однорідних об'єктів, які виступають як єдине ціле.

Транзакція – це логічна одиниця роботи, яка виконується без порушення логічної цілісності бази даних. Якщо в процесі виконання транзакції виникла помилка виконання, то система, яка підтримує процес транзакції, гарантує повернення до первинного стану.

Тригер – попередньо визначена дія або послідовність дій, які автоматично здійснюються під час виконання операцій оновлення, додавання або видалення даних.

ODBC (Open Database Connectivity) – відкритий доступ до баз даних – загальне визначення мови та набір протоколів, які дозволяють клієнтському додатку працювати з командами та функціями, що підтримуються сервером.

OLE (Object Linking and Embedding) – зв'язування та впровадження об'єктів – технологія, яка дозволяє використовувати в додатку об'єкти, розроблені в іншому додатку. OLE-об'єктами можуть бути зву, рисунки, діаграми, відеокліпи тощо.

OLE-сервер – програма, яка може надавати іншим програмам можливість використання своїх об'єктів.

SQL (Structured Query Language) – мова структурованих запитів, офіційний стандарт мови для роботи з реляційними базами даних.