Міністерство освіти і науки України
Національний університет «Запорізька політехніка»

Програма,
методичні вказівки та контрольні завдання

з дисципліни «Бази даних»

для студентів спеціальності
121 «Інженерія програмного забезпечення
заочної форми навчання

2021

Програма, методичні вказівки та контрольні завдання з дисципліни «Бази даних» для студентів спеціальності 121 «Інженерія програмного забезпечення» заочної форми навчання / Уклад.: С.К. Корнієнко . – Запоріжжя: НУ «Запорізька політехніка», 2021. – 26 с.

Укладач: С.К.Корнієнко, доцент, к.т.н.

Рецензент: Г.М.Шило, професор, к.т.н.

Відповідальний
за випуск: С.О.Субботін, професор, д.т.н.

Затверджено
на засіданні кафедри
"Програмних засобів"

Протокол №1
від 18.08.2020 р.

ЗМІСТ

1 Мета і задачі дисципліни, її місце в навчальному процесі 4
1.1 Мета викладання дисципліни 4
1.2 Задачі вивчення дисципліни 4
1.3 Зв'язок з іншими дисциплінами 4
1.4 Загальні вказівки 4
2 Програма дисципліни та методичні вказівки з її вивчення 6
2.1 Основи побудови банків інформації 6
2.2 Концептуальне проектування баз даних 14
2.3 Реляційна модель даних 17
2.4 Захист даних 20
3 Курсовий проект 23
3.1 Мета курсового проектування 23
3.2 Тематика курсових проектів 23
4 Контрольна робота 24
4.1 Загальні положення 24
4.2 Зміст контрольної роботи 24
Література 26

1 МЕТА І ЗАДАЧІ ДИСЦИПЛІНИ, ЇЇ МІСЦЕ В НАВЧАЛЬНОМУ ПРОЦЕСІ

1.1 Мета викладання дисципліни

Метою даної дисципліни є вивчення загальних принципів побудови та функціонування баз даних, а також надбання практичних навичок розробки та налагодження відповідного програмного забезпечення.

1.2 Задачі вивчення дисципліни

В наслідку вивчення дисципліни студенти повинні:

– розуміти проблеми, які виникають під час побудови та використанні сучасних банків даних;

– вивчити основні підходи та загальні принципи проектування баз даних на концептуальному, логічному та фізичному рівнях;

– придбати навички експлуатації сучасних СКБД .

1.3 Зв’язок з іншими дисциплінами

Вивчення даної дисципліни базується на знаннях, одержаних при вивченні дисциплін "Структури та організація даних у ЕОМ". Отримані знання будуть використовуватися при вивченні дисциплін "Основи автоматизованого проектування складних об'єктів та систем", а також у курсовому та дипломному проектуванні.

1.4 Загальні вказівки

Дана дисципліна вивчається у четвертому семестрі й містить 10 лекційних годин, 8 годин лабораторних робіт і 4 години практичних занять, а також виконання контрольної роботи та курсового проекту.

Працювати над дисципліною треба систематично. При цьому необхідно користуватися не тільки основною, а й додатковою літературою.

Під час самостійної опрацювання матеріалу слід також враховувати, що контрольні завдання складені таким чином, щоб результат їхнього виконання був покладений у підвалини виконання лабораторного практикуму.

Підведення підсумків вивчення дисципліни здійснюється у вигляді іспиту. До складення іспиту допускаються студенти, які успішно виконали лабораторний практикум і всі контрольні завдання.

2 ПРОГРАМА ДИСЦИПЛІНИ ТА МЕТОДИЧНІ ВКАЗІВКИ З ЇЇ ВИВЧЕННЯ

2.1 Основи побудови банків інформації

2.1.1 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 пойменована структурована сукупність взаємозв'язаних даних, що відносяться до конкретної предметної області та призначаються для задовольняння 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.

2.1.2 Архітектура системи баз даних. Система управління базою даних. Мова опису даних. Мова маніпулювання даними. Функціонування СКБД . Адміністрування даних. Словник даних. Колектив спеціалістів банку даних.

Система баз даних (***СБД***) - це організаційно-технічна (людино-машинна) система (рис. 2.1), що містить у собі комплекс спеціальних методів та засобів для підтримки динамічної інформаційної моделі предметної області та забезпечує зберігання, пошук та видавання інформації у вигляді, зручному користувачам.

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.

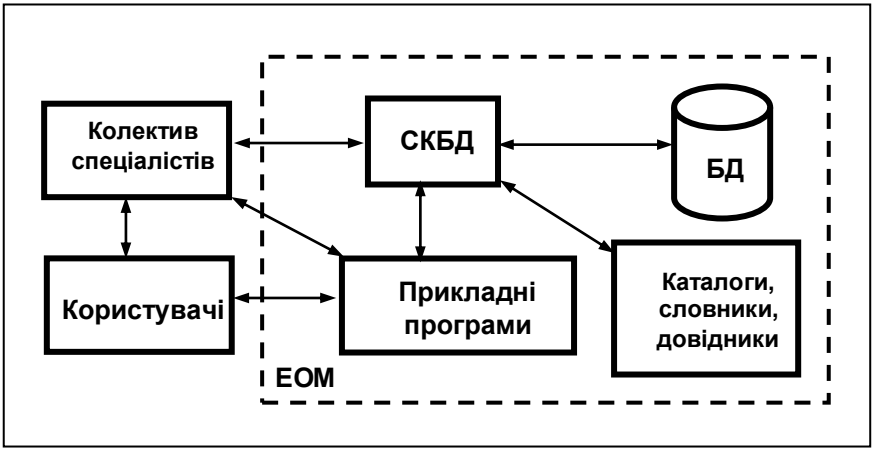

Рисунок 2.1 - Архітектура системи баз даних

СКБД грає центральну роль у функціонуванні системи баз 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 2.2).

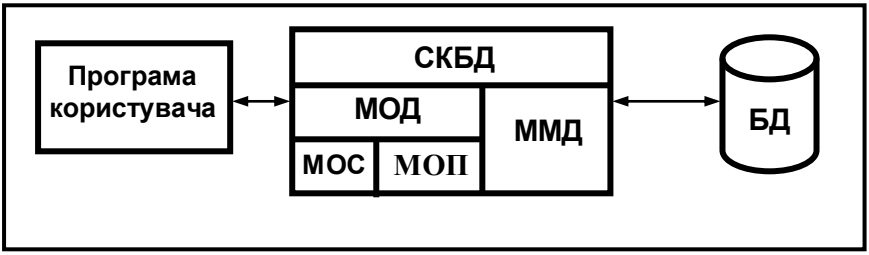

Рисунок 2.2 - Архітектура СКБД

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 структури даних*** або ***схема бази даних***.

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 (***МОС***) 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 по заданим критеріям;
– 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 ***транзакцією.***

2.1.3 Поняття предметної області та проблемної середи. Три рівні подання 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 предмету 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, запропонованої комітетом ***ANSI-SPARK***, яка включає (рис. 2.3):

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

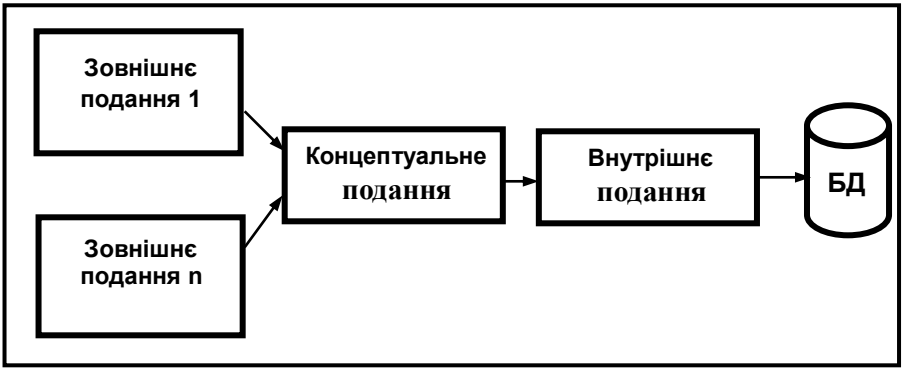

Рисунок 2.3 - Трирівневе подання даних

2.1.4 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 такі:

– 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 із 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.

2.1.5 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 використання 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***) (рисунок 2.4).

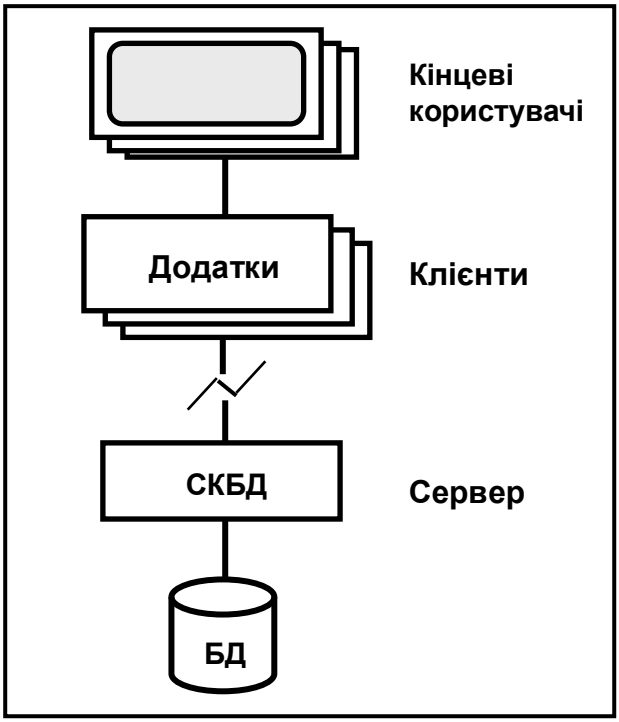

Рисунок 2.4 - Архітектура клієнт-сервер

- ***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 і т.д
- ***Клієнти*** - це різноманітні додатки, що знаходяться на комп'ютері користувача та забезпечують інтерактивний, легкий у використані, зазвичай 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.***

Важливим етапом у побудові додатку клієнт-сервер є встановлення зв'язку клієнтського додатку із 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 і шлюзи приховані від користувача.

2.1.6 Життєвій 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 і***нфологічне*** та ***даталогічне*** 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 та методів доступу до неї, виходячи із методів та засобів, які надані розробнику конкретної СКБД .

Контрольні питання

1 Поняття організації інформаційних масивів та бази даних?
2 З яких компонентів складається СБД ?
3 Яке призначення має словник даних СБД?
4 Місце СКБД в архітектурі СБД.
5 Яка різниця між МОД і ММД?
6 Яким чином функціонує СКБД ?
7 Що таке *предметна область*?
8 Склад і призначення трирівневої моделі даних.
9 Навіщо використовується адміністрування БД?
10 Яка різниця між технологіями файл-сервер і клієнт-сервер?
11 Поняття технології відкритого доступу до даних.
12 Етапи життєвого циклу інформаційної системи.

2.2 Концептуальне проектування баз даних

2.2.1 Мета і задачі концептуального проектування. Основні підходи до проектування баз даних.

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

Підхід "***від предметної області***" (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 з вказаних підходів не дає достатньої інформації для проектування раціональної БД. Доречно сумісне застосування обох підходів із ведучим положенням об'єктного підходу.

2.2.2 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 пойменована 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.

2.2.3 Методика концептуального проектування. Моделювання локальних подань.

При моделюванні використовуються такі 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 функціональній задачі або окремому користувачу. Але воно може відповідати й цілій незалежній області даних, яка використовується декількома додатками.

2.2.4 Об'єднання моделей локальних уявлень. Синтез концептуальної моделі. Приклад концептуального проектування БД.

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** і **АДРЕСА**.

Узагальненням називається абстракція даних, яка дозволяє трактувати клас різних подібних об'єктів як один пойменований 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 узагальнені підкреслюється загальна природа об'єктів, а їх індивідуальні відмінності ігноруються.

Контрольні питання

1 Які існують основні підходи до концептуального проектування баз даних?
2 Які конструктивні елементи має ER-модель?
3 Що називається *сутністю*?
4 Яка різниця між *типом сутності* та *екземпляром сутності*?
5 Які типи зв'язків існують?
6 Що таке *локальне уявлення*?
7 Як об'єднуються локальні уявлення?

2.3 Реляційна модель даних

2.3.1 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:

– припустимою структурою (організацією) даних;
– кількістю припустимих операцій над даними;
– обмеженнями цілісності (семантичної).

Модель даних – це сукупність правил формування структури даних в 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) введена ***Коддом*** (***Kodd E.E.)*** в 1970 році. В основі РМД лежить математичне поняття відношення (***relation***).

Відношення - це підмножина декартова 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 $\mathbf{D_1, D_2, D_3, ..., D_K}$

$$\mathbf{D = D_1 \times D_2 \times ... \times D_K}$$

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

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

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

або у матричному вигляді

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

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

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

Число всіх різних кортежів $\mathbf{d=<d_1,d_2,d_3,...,dn>}$, які утворюють відношення **Rn**, 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 $\mathbf{A_1, A_2, ... , A_K}$, позначається $\mathbf{R(A_1,A_2,...,A_K)}$.

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 з***в'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 в 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.

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

Відношення реляційної БД містять як структурну, так і семантичну (смислову) 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 оброку 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 ***B*** відношення ***R*** функціонально залежить від атрибута ***A*** того ж

відношення, якщо в кожний момент часу кожному значенню атрибута ***A*** відповідає тільки одне значення ***атрибута B***, зв'язаного з ***A*** у відношенні ***R***:

$$f : A \rightarrow B.$$

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

Контрольні питання

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

2.4 Захист даних

Відновлення бази даних. Транзакції. Відновлення транзакції. 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 баз даних означає відновлення самої бази даних, тобто повернення бази даних у *вірний стан*, якщо який-небудь збій зробив поточний стан невірним або підозрілим.

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. 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***. 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 (дампу) 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***.

Виникаючі 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)

Термін ***безпека*** відноситься до захисту даних від несанкціонованого доступу, зміні 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.

Контрольні питання

1 Які методи захисту даних використовують у базах даних?
2 Що називається транзакцією?
3 Як у базах даних підтримується цілісність даних?
4 Яким чином відбувається адміністрування бази даних?

3 КУРСОВИЙ ПРОЕКТ

3.1 Мета курсового проектування

Мета виконання курсового проекту - набути практичних навичок у галузі проектування інформаційного забезпечення автоматизованих систем різного призначення.

В процесі курсового проектування студенти повинні:

– узагальнити, закріпити та поглибити знання, отримані під час вивчення дисципліни, використати їх для обґрунтованого прийняття проектних рішень;

– ознайомитись з методикою та практично закріпити навички комплексної розробки інформаційних систем в цілому та їх базових компонентів: програмного, інформаційного, лінгвістичного та математичного забезпечень;

– придбати досвід в оформленні відповідної проектної документації та складанні пояснювальних та розрахункових записок.

3.2 Тематика курсових проектів

Тематика курсових проектів повинна бути актуальною, відповідати сучасному стану розвитку науки та техніки, враховувати реальні потреби виробництва.

Приклади тем курсових проектів

– Автоматизована система обліку контрактів.
– Автоматизована система з продажу автомобілів.
– Автоматизована система складського обліку.
– Інформаційно-пошукова система "Бібліотека".
– Автоматизована система комерційної діяльності;
– Автоматизована система розрахунку заробітної плати.
– Автоматизація комерційної діяльності комп’ютерного центру.
– Автоматизація обліку господарської діяльності будівельно-ремонтної фірми.
– Розробка АРМу робітника планово-економічної служби.

За узгодженням із викладачем допускається самостійний вибір теми студентом.

4 КОНТРОЛЬНА РОБОТА

4.1 Загальні положення

В процесі вивчення дисципліни «Бази даних» студенти заочної форми навчання повинні виконати контрольну роботу.

Основна ціль її виконання – оволодіння навичками практичного проектування баз даних.

Приступати до виконання контрольних робіт слід лише після детального вивчення теоретичного матеріалу та виконання лабораторних робіт згідно з відповідними методичними вказівками.

Виконані контрольні роботи повинні бути представлені на перевірку в суворій відповідності до графіку навчального процесу. При наявності зауважень рецензента слід виконати всі виправлення та доповнення так, щоб при повторному рецензуванні можна було б співвіднести попереднє та нове рішення.

Захист контрольних робіт відбувається безпосередньо на комп'ютері.

Результати виконання кожної контрольної роботи оформлюються у вигляді звіту, який виконується комп'ютерним способом і повинен містити роздрук вихідних і кінцевих даних (таблиці, форми, звіти та інші).

Робота повинна бути датована та підписана виконавцем.

4.2 Зміст контрольної роботи

При виконанні контрольної роботи слід вибрати предметну область, узгодивши її з викладачем, виконати проектування схеми БД і наповнити таблиці БД реальними даними.

В процесі виконання контрольної роботи необхідно дати відповіді на нижче наведені питання, проілюструвавши їх відповідними копіями екрану.

<u>Питання до контрольної роботи:</u>

– побудова концептуальної моделі предметної області;
– специфікація структури таблиці, властивості полів;

– редагування бази даних: переміщення по таблиці, редагування записів, вилучання даних, копіювання даних, зміна типу поля, індикація та заглушення стовпців;

– сортування та фільтрація записів;

– встановлення зв’язків між таблицями. підтримка цілісності даних;

– складання простих запитів до бази даних;

– вибірка інформації за складними умовами відбору записів;

– призначення операторів Like, In, Between;

– зв’язування таблиць у запитах;

– створення у запиті обчислювального поля;

– створення групових запитів: підсумкові функції, групові операції для груп записів, визначення критерію для групового запиту, складання виразу для групових операцій;

– створення перехресних запитів: визначення критерію для перехресного запиту, використання майстра перехресного запиту;

– створення параметричного запиту;

– створення запитів на оновлення;

– побудова запитів на створення таблиці;

– побудова запиту на добавлення записів;

– побудова запиту на вилучання записів;

– майстер запитів на пошук повторюваних записів;

– майстер запитів на пошук повторюваних записів, які не мають підлеглих;

– створення форм даних за допомогою майстра форм;

– створення форм даних за допомогою конструктора форм;

– призначення елементів управління, які використовуються у формах;

– приєднані, вільні та обчислювальні елементи управління;

– створення підлеглих форм;

– створення форм із вкладками;

– створення кнопкової форми;

– використання OLE-об’єктів у формах;

– створення діалогових вікон;

– створення простих та складових звітів.

Крім того, треба виконати та належним чином оформити індивідуальне завдання, одержане від викладача.

ЛІТЕРАТУРА

Основна

1. Корнієнко С.К. Системи баз даних: організація та проектування: Навч. Посібник. – Запоріжжя: ЗНТУ, 2006. – 252 с.

2. Корнієнко С.К. Проектування інформаційного забезпечення автоматизованих систем. Навч. Посібник. – Запоріжжя: ЗНТУ, 2015. – 210 с.

Додаткова

1. Бази даних. Методичні вказівки до лабораторних робіт. Частина 1 для студентів спеціальності 121 «Інженерія програмного забезпечення» усіх форм навчання / Уклад. С.К. Корнієнко – Запоріжжя: НУ «Запорізька політехніка», 2021. – 46 с.

2. Бази даних. Методичні вказівки до лабораторних робіт. Частина 2 для студентів спеціальності 121 «Інженерія програмного забезпечення» усіх форм навчання / Уклад. С.К. Корнієнко – Запоріжжя: НУ «Запорізька політехніка», 2021. – 30 с.

2. Методичні вказівки до виконання курсового проекту з дисципліни «Бази даних» для студентів спеціальності 121 «Інженерія програмного забезпечення» усіх форм навчання / Уклад. С.К. Корнієнко. – Запоріжжя: НУ «Запорізька політехніка», 2021. – 38 с.