

Житомирський державний університет імені Івана Франка
Фізико-математичний факультет
Кафедра комп'ютерних наук та інформаційних технологій

ІНСТРУКТИВНО-МЕТОДИЧНІ МАТЕРІАЛИ ДО ЛАБОРАТОРНИХ ЗАНЯТЬ

**обов'язкової освітньої компоненти
«Бази даних»**

для підготовки здобувачів першого
(бакалаврського) рівня вищої освіти

Галузь знань	01 Освіта /Педагогіка
Спеціальність	015 Професійна освіта
Спеціалізація	015.39 Цифрові технології
Освітня програма	Професійна освіта (Цифрові технології)
Факультет	Фізико-математичний

Укладач: к.пед.н., доцент Горобець Сергій
Розглянуто та схвалено на засіданні
кафедри комп'ютерних наук та
інформаційних технологій
Протокол від «08» червня 2022 р. № 18
Завідувач кафедри _____
Ярослава СІКОРА

Житомир 2022

*Рекомендовано до друку вченою радою Житомирського державного
університету імені Івана Франка
(протокол №15 від «26» серпня 2022 р.)*

Рецензенти:

КОРОТУН Ольга – кандидат педагогічних наук, доцент кафедри комп'ютерних наук, заступник декана факультету інформаційно-комп'ютерних технологій Державного університету «Житомирська політехніка».

МАЄВСЬКИЙ Олександр – кандидат технічних наук, доцент, доцент кафедри комп'ютерних технологій і моделювання систем Поліського національного університету.

КРИВОНОС Олександр – кандидат педагогічних наук, доцент, доцент кафедри комп'ютерних наук та інформаційних технологій Житомирського державного університету імені Івана Франка.

Горобець С. М.

Інструктивно-методичні матеріали до лабораторних занять обов'язкової освітньої компоненти «Бази даних» для підготовки здобувачів першого (бакалаврського) рівня вищої освіти спеціальності «015 Професійна освіта». Житомир: Вид-во ЖДУ імені Івана Франка, 2022. 45 с.

© Горобець С. М., 2022

© Житомирський державний університет
імені Івана Франка, 2022

ЗМІСТ

Критерії оцінювання занять		3
Лабораторна робота № 1	Інфологічне (концептуальне) моделювання БД	5
Лабораторна робота № 2	Даталогічне проектування. Нормалізація відношень	8
Лабораторна робота № 3	Основні та похідні операції реляційної алгебри	13
Лабораторна робота № 4	Налаштування СУБД та створення бази даних	19
Лабораторна робота № 5	Використання мови SQL для відображення інформації	22
Лабораторна робота № 6	Агрегатні функції	26
Лабораторна робота № 7	З'єднання таблиць та групові операції	29
Лабораторна робота № 8	Використання підзапитів мовою SQL	33
Лабораторна робота № 9	Проектування звітності	36
Лабораторна робота № 10	Команди модифікації даних	39
Лабораторна робота № 11	Захист інформації в СУБД	43

КРИТЕРІЇ ОЦІНЮВАННЯ ЗАНЯТЬ

Оцінювання здобувачів вищої освіти здійснюється відповідно до «Положення про критерії та порядок оцінювання навчальних досягнень здобувачів вищої освіти Житомирського державного університету імені Івана Франка згідно з Європейською кредитною трансферно-накопичувальною системою» https://zu.edu.ua/offic/ocinjuvannya_zvo.pdf.

Оцінювання навчальних досягнень здобувачів вищої освіти за всіма видами навчальних робіт проводиться за поточним, модульним та підсумковим контролем.

Кожен здобувач вищої освіти має виконати обов'язкові завдання, передбачені інструктивно-методичними матеріалами до лабораторних занять, методичними рекомендаціями до організації самостійної та індивідуальної роботи здобувачів вищої освіти, силабусом, навчальною та робочою програмою освітньої компоненти.

Картка обліку виконання завдань здобувачем вищої освіти
 ПІБ здобувача вищої освіти _____ Група _____

№ лабораторного заняття	Виконання завдання	Відповіді на контрольні питання
	80 балів	20 балів
1		
2		
3		
4		
5		
6		
7		
8 (ПМК 1)	100	
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		
26 (ПМК 2)	100	

Лабораторна робота № 1

Тема: Інфологічне (концептуальне) моделювання БД

Мета: закріпити та поглибити знання студентів, які отримані ними на лекціях та під час самостійної роботи з питань розробки інфологічної моделі бази даних.

Завдання: розробити інфологічну (концептуальну) модель бази даних для обраної предметної області.

Час виконання завдання: 4 год.

План заняття

1. Виконання завдання згідно з інструкцією.
2. Захист роботи та відповіді на контрольні питання.

Рекомендації до виконання

1.1. Вибір предметної області для проектування БД

Спектр використання баз даних досить широкий і має міждисциплінарний характер – це державне управління, науково-дослідницька діяльність, банківська та страхова справа, логістика, управління складським господарством на підприємствах, геоінформаційні системи, бібліотечна справа та створення різноманітних електронних каталогів, управління контентом інтернет-сайтів тощо.

Розглянемо основні поняття і терміни, які використовуються в теорії баз даних.

База даних – це сукупність структурованих, логічно взаємопов'язаних даних, що зберігаються та використовуються спільно. Це потужний засіб для збирання та впорядкування даних, який призначений для задоволення інформаційних потреб користувачів.

Кожна база даних описує певну предметну область, тобто певну визначену частину реального світу.

На першому етапі виконання лабораторної роботи необхідно визначитись з предметною областю для проектування власної бази даних. Пропонується розробити базу даних інтернет-магазину за обраним та узгодженим з викладачем асортиментом товарів. За бажанням, студенти можуть обрати іншу предметну область із переліку, наведеного в табл. 1.1.

Таблиця 1.1.

Перелік тем для проектування БД

Варіант	Тема
1.	База даних абонентів Інтернет-провайдера.
2.	База даних відділу контролю якості готової продукції.
3.	База даних замовлень робіт з ландшафтного дизайну.
4.	База даних замовлень робіт з ремонту будівлі.
5.	База даних контролю за рухом складських запасів.
6.	База даних майстерні з ремонту побутової техніки.
7.	База даних обліку амортизації виробничого обладнання
8.	База даних обліку дзвінків в мережі мобільного оператора.
9.	База даних обліку замовлень на виробництво продукції.

10.	База даних обліку збуту продукції для фермерського господарства.
11.	База даних обліку звернень громадян до органу місцевого самоврядування.
12.	База даних обліку літератури в бібліотеці.
13.	База даних обліку робочого часу працівників.
14.	База даних обліку транспортних перевезень.
15.	База даних обліку фінансових операцій співвласників багатоквартирного будинку.
16.	База даних підприємства з виробництва бетону.
17.	База даних підрозділу постачання сировини для кондитерського цеху.
18.	База даних підрозділу постачання сировини для меблевого виробництва.
19.	База даних планування та обліку навантаження викладачів.
20.	База даних сервісного центру з ремонту мобільних телефонів.
21.	База даних системи контролю за доступом до режимних об'єктів.
22.	База даних системи моніторингу якості продукції.
23.	База даних страхової компанії.
24.	База даних обліку виконаних робіт (для обраної та узгодженої з викладачем сфери послуг).

1.2. Інфологічне (концептуальне) проектування БД.

На першому етапі проектування бази даних необхідно визначитись з групами об'єктів, інформація про які буде в ній зберігатися. Ці групи однотипних об'єктів можуть бути будь-якої природи та мають назву «сутності».

При складанні інфологічної моделі необхідно визначитись з переліком сутностей.

Під час опису сутностей потрібно дослідити зв'язки між ними та зазначити їх тип.

При аналізі сутностей предметної області також визначають обмеження, що накладаються на зв'язки між сутностями та на діапазони допустимих значень, що будуть зберігатись у БД.

У звіті про виконання лабораторної роботи необхідно навести графічну діаграму типу сутність-зв'язок (ER-діаграму) із зазначенням всіх сутностей та зв'язків між ними (рис.1.1).

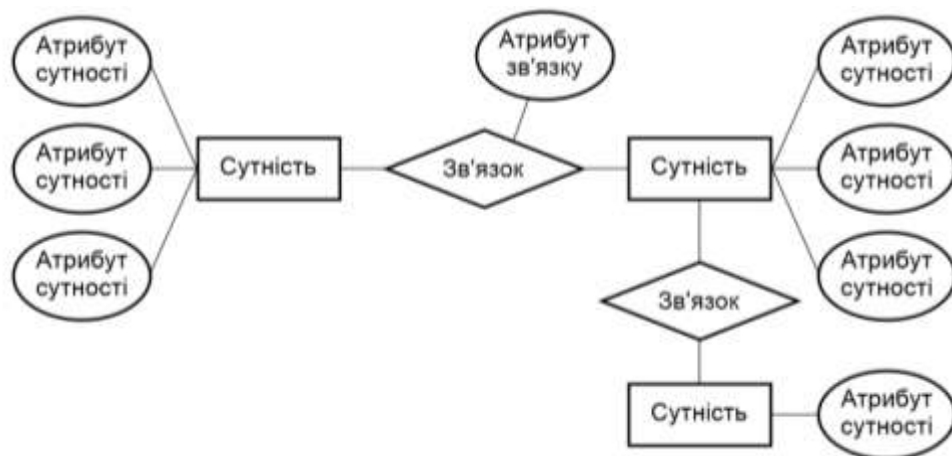


Рис. 1.1. Приклад оформлення діаграми типу сутність-зв'язок

Для автоматизації та спрощення процесу підготовки таких моделей можна використовувати он-лайн сервіси типу diagrams.net та lucidchart.com.

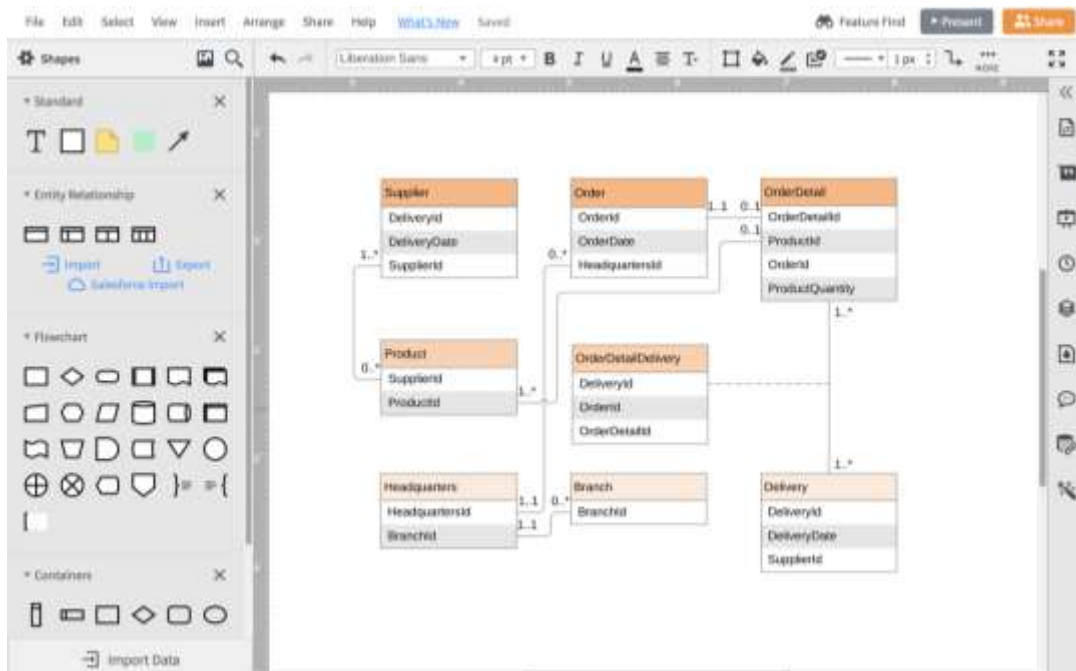


Рис. 1.2. Приклад оформлення ER-діаграми за допомогою он-лайн сервісу lucidchart.com

Контрольні питання

1. Які основні завдання вирішуються шляхом проектування баз даних?
2. Назвіть основні етапи проектування бази даних та коротко опишіть їх.
3. Яку інформацію містить в собі інфологічна (концептуальна, семантична) модель БД?
4. З якою метою створюють ER-модель бази даних?
5. Які нотації (графічні зображення) інфологічних моделей найбільш популярні? Наведіть приклади умовних позначень елементів моделі в різних нотаціях.

Рекомендована література

1. Верес О. М., Пасічник В. В. Системи баз даних та знань. Книга 1: організація баз даних та знань. Київ: Магнолія, 2019. 440 с.
2. Гайна Г. А. Основи проектування баз даних. Київ: Кондор, 2018. 208 с.
3. Демиденко М. А. Введення в сучасні бази даних: навч. посіб. НТУ «Дніпровська політехніка». Дніпро, 2020. 38 с.
4. Мулеса О. Ю. Інформаційні системи та реляційні бази даних: навч. посіб. Ел. видання, 2018. 118 с. URL: https://dSPACE.uzhnu.edu.ua/jspui/bitstream/lib/19776/1/%D0%9C%D1%83%D0%BB%D0%B5%D1%81%D0%B0_%D0%91%D0%94.pdf

Лабораторна робота № 2

Тема: Даталогічне проектування БД. Нормалізація відношень.

Мета: закріпити та поглибити знання студентів, які отримані ними на лекціях та під час самостійної роботи з питань розробки даталогічної моделі бази даних.

Завдання: розробити даталогічну модель бази даних для обраної предметної області. Провести процедуру нормалізації відношень.

Час виконання завдання: 6 год.

План заняття

1. Виконання завдання згідно з інструкцією.
2. Захист роботи та відповіді на контрольні питання.

Рекомендації до виконання

2.1. Даталогічне (логічне) проектування БД.

Зазвичай, сутності описують за допомогою таблиць, де властивості екземплярів даної групи об'єктів представлені у вигляді множини атрибутів. Отже, атрибути – це властивості, що описують сутність. Множина атрибутів повинна однозначно визначати кожен екземпляр сутності, тобто бути унікальною.

Для спрощення ідентифікації окремих екземплярів сутностей використовують поняття «первинний ключ». Це поле чи сукупність полів, що містять унікальні значення, за якими однозначно визначають той чи інший екземпляр сутності.

Попередньо розроблену інфологічну (концептуальну) модель БД необхідно перевести у даталогічну (схему бази даних).

Для кожної сутності необхідно призначити окрему таблицю та вказати для неї атрибути, зв'язки та ключові поля.

Для атрибутів треба вказати назви полів, типи даних, обмеження для даних, синоніми полів (якщо планується їх використання).

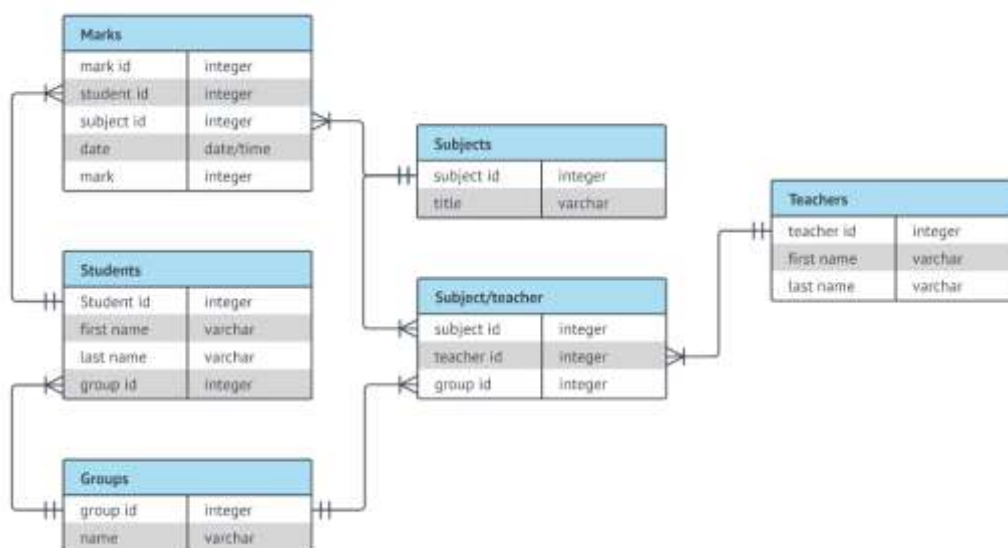


Рис. 2.1. Приклад оформлення схеми БД за допомогою онлайн сервісу lucidchart.com

При створенні схеми бази даних використовуйте умовні позначення для типу зв'язків та ключових полів.

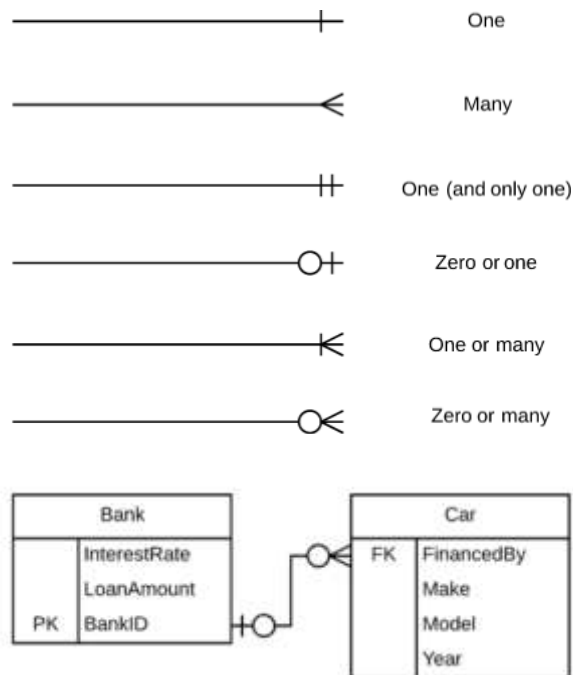


Рис. 2.2. Умовні позначення для типу зв'язків та ключових полів

Як альтернативу он-лайн редакторам, за бажанням, можна використати оф-лайн програмні продукти, наприклад, потужний програмний пакет Erwin Data Modeler, що дозволяє гнучко проектувати та документувати бази даних будь-якого масштабу та складності (рис.2.3).

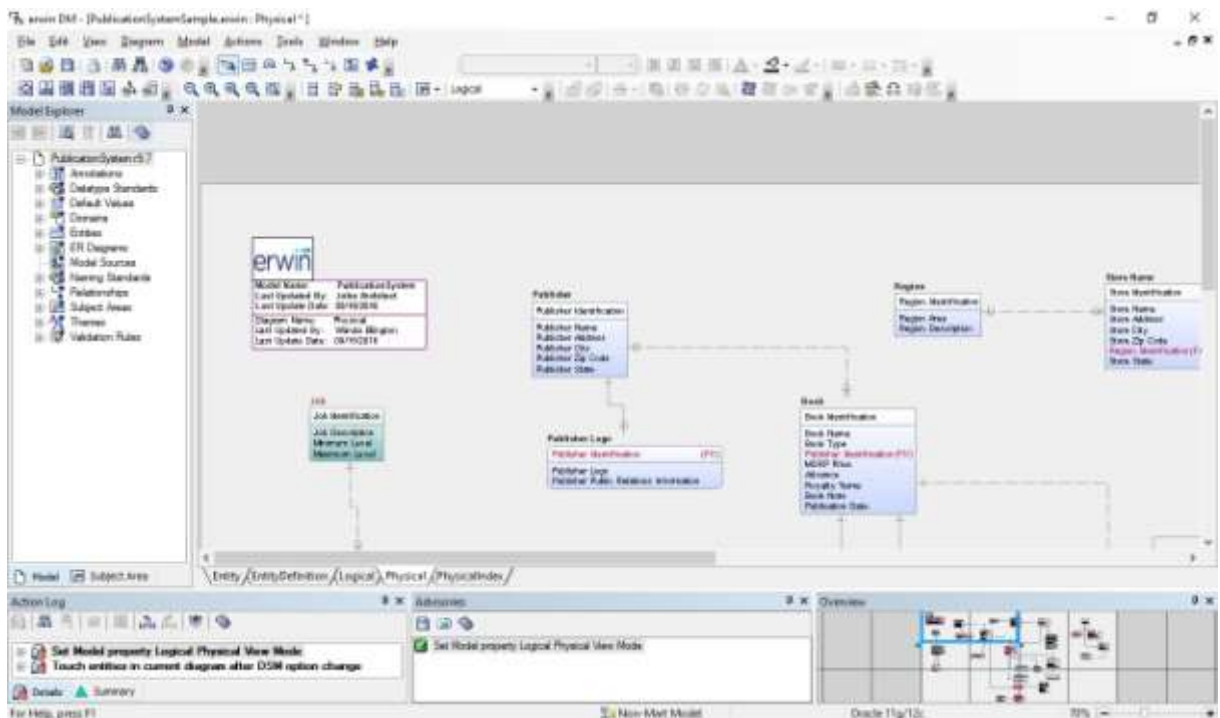


Рис. 2.3. Інтерфейс Erwin Data Modeler

2.2. Нормалізація БД.

Для створення оптимальної структури зберігання даних та для зменшення їх надмірності необхідно проаналізувати функціональні залежності у розробленій вище схемі бази даних.

Для цього покроково перевірте схему БД на виконання умов нормалізації. Перевірку потрібно проводити не менше ніж для умов четвертої нормальної форми.

При необхідності проведіть декомпозицію відношень, перепризначення ключових полів та переоцінку належності атрибутів до певних типів даних і доменів.

2.3 Приклад опису структури таблиць

В якості предметної області оберемо інтернет-магазин, який здійснює продаж електроніки, наприклад, смартфонів.

Створення БД дозволить вирішувати багато поточних задач організації, зокрема:

- автоматизувати облік;
- підвищити ефективність.

Для досягнення навчальної мети достатньо розробити спрощену структуру, яка налічує мінімально необхідну кількість сутностей. Для обраної предметної області такими сутностями можуть бути: «Товар», «Співробітники», «Клієнти» і «Замовлення».

Для прикладу, в якості системи управління базою даних оберемо Microsoft Access. Подальшу розробку та опис атрибутів визначених сутностей будемо здійснювати з урахуванням специфіки створення баз даних для СУБД Microsoft Access.

Атрибути сутностей «Товар», «Співробітники», «Клієнти» і «Замовлення» наведені в табл.2.1–2.4.

Таблиця 2.1.

Атрибути сутності «Товар»

Назва атрибута (поля)	Тип даних	Опис атрибута
Код_тов	лічильник	Ключове поле. Унікальний ідентифікатор товару.
Виробник	текстовий	Назва торгової марки виробника.
Модель	текстовий	Найменування моделі смартфона.
Пам_операт	числовий (довге ціле)	Обсяг оперативної пам'яті смартфона в ГБ.
Пам_вбудов	числовий (довге ціле)	Обсяг вбудованої пам'яті смартфона в ГБ.
Диспл_діаг	числовий (одинарне з плаваючою точкою)	Розмір діагоналі дисплею в дюймах.
Диспл_тип	текстовий	Тип конструкції дисплею. Джерело рядків: "IPS"; "OLED"; "AMOLED"; "SuperAMOLED"; "TN"
Диспл_пікс	текстовий	Кількість пікселів по ширині та висоті екрану. Маска введення даних: 9999" x "9999
Кам_фронт	числовий (одинарне з плаваючою точкою)	Найбільша кількість мегапікселів в блоці фронтальної камери.
Кам_осн	числовий (одинарне з плаваючою точкою)	Найбільша кількість мегапікселів в блоці основної камери.
Висота	числовий (одинарне з плаваючою точкою)	Висота смартфона в мм.
Ширина	числовий (одинарне з плаваючою точкою)	Ширина смартфона в мм.
Товщина	числовий (одинарне з плаваючою точкою)	Товщина смартфона в мм.
Акум	числовий (довге ціле)	Значення ємності акумулятора в мАгод.
Колір	текстовий	Назва кольору смартфона.

Ціна	числовий (подвійне з плаваючою точкою)	Ціна смартфона в грн. з копійками. Формат з роздільниками розрядів. Кількість десяткових знаків – 2.
Фото_товар	Поле об'єкта OLE	Фотографія товару

Таблиця 2.2.

Атрибути сутності «Співробітники»

Назва атрибута (поля)	Тип даних	Опис атрибута
Код_співр	лічильник	Ключове поле. Унікальний ідентифікатор співробітника.
Прізвище	текстовий	Прізвище співробітника.
Ім_я	текстовий	Ім'я співробітника.
По_батькові	текстовий	По-батькові співробітника.
Тел	текстовий	Номер телефону співробітника. Маска введення даних: "тел. +38 "000" "000\-\00\-\00
Паспорт	текстовий	Номер паспорта співробітника.
Ідент_код	числовий (довге ціле)	Номер ідентифікаційного коду співробітника.
Адр_місто	текстовий	Назва міста, де проживає співробітник.
Адр_вул	текстовий	Назва вулиці, де проживає співробітник.
Адр_буд	текстовий	Номер будинку, де проживає співробітник.
Адр_кв	текстовий	Номер квартири, де проживає співробітник.
Дата_народж	дата/час (короткий формат дати)	Дата народження співробітника у форматі dd.mm.yyyy.
Дата_прийому	дата/час (короткий формат дати)	Дата прийому на роботу співробітника у форматі dd.mm.yyyy.
Посада	текстовий	Посада співробітника.
Права_водія	логічний (так/ні)	Наявність у співробітника прав водія.
Фото_співр	Поле об'єкта OLE	Фотографія співробітника

Таблиця 2.3.

Атрибути сутності «Клієнти»

Назва атрибута (поля)	Тип даних	Опис атрибута
Код_клієнта	лічильник	Ключове поле. Унікальний ідентифікатор організації, яка купує товар.
Назва_орг	текстовий	Назва організації, яка купує товар.
Адр_місто	текстовий	Назва міста, де розташована організація.
Адр_вул	текстовий	Назва вулиці, де розташована організація.
Адр_буд	текстовий	Номер будинку, де розташована організація.
Код_ЄДРПОУ	текстовий	Код ЄДРПОУ організації. Маска введення даних: 00000000
Р_рахунок	текстовий	Номер розрахункового рахунку організації. Маска введення даних: 9999999999
Прізвище	текстовий	Прізвище контактної особи.
Ім_я	текстовий	Ім'я контактної особи.
По-батькові	текстовий	По-батькові контактної особи.
Тел	текстовий	Номер телефону клієнта. Маска введення даних: "тел. +38 "000" "000\-\00\-\00
Ел_пошта	текстовий	Електронна пошта контактної особи.

Таблиця 2.4.

Атрибути сутності «Замовлення»

Назва атрибута (поля)	Тип даних	Опис атрибута
Номер_замовл	лічильник	Ключове поле. Унікальний ідентифікатор замовлення.
Дата_замовл	дата/час (повний формат дати)	Дата та час замовлення у форматі dd.mm.yyyy hh:mm:ss. Значення за замовчуванням: =Now()
Код_співр	числовий (довге ціле)	Зовнішнє ключове поле. Підстановка: SELECT [Співробітники].[Код_співр], [Співробітники].[Прізвище], [Співробітники].[Ім_я], [Співробітники].[По- батькові] FROM Співробітники ORDER BY [Прізвище], [Ім_я];
Код_клієнта	числовий (довге ціле)	SELECT [Клієнти].[Код_клієнта], [Клієнти].[Назва_орг], [Клієнти].[Адр_місто], [Клієнти].[Прізвище], [Клієнти].[Ім_я], [Клієнти].[По-батькові], [Клієнти].[Тел] FROM Клієнти ORDER BY [Назва_орг], [Адр_місто], [Прізвище];
Код_тов	числовий (довге ціле)	Зовнішнє ключове поле. Підстановка: SELECT Товар.[Код_тов], [Товар].[Модель], [Товар].[Виробник], Товар.[Пам_операт], Товар.[Пам_вбудов], Товар.[Диспл_діаг], Товар.[Диспл_тип], Товар.[Диспл_пікс], Товар.[Кам_фронт], Товар.[Кам_осн], [Товар].[Висота], [Товар].[Ширина], [Товар].[Товщина], Товар.[Акум], [Товар].[Колір], [Товар].[Ціна] FROM Товар ORDER BY [Виробник], [Модель];
Кількість	числовий (довге ціле)	Кількість замовлених товарів.
Дата_оплати	дата/час (короткий формат дати)	Дата і час оплати замовлення у форматі dd.mm.yyyy.
Дата_відправки	дата/час (короткий формат дати)	Дата відправки замовлення у форматі dd.mm.yyyy.
Відправлено	логічний	Відмітка про те, що товар відправлено

Контрольні питання

1. Для чого розробляють даталогічну модель бази даних?
2. З якою метою проводять нормалізацію відношень в моделі БД?
3. Назвіть переваги та недоліки реляційних моделей даних.
4. Які види ключів ви знаєте? Дайте їм характеристику.
5. Поясніть зміст понять «атрибут», «кортеж», «домен», «відношення».

Рекомендована література

1. Верес О. М., Пасічник В. В. Системи баз даних та знань. Книга 1: організація баз даних та знань. Київ: Магнолія, 2019. 440 с.

2. Демиденко М. А. Введення в сучасні бази даних: навч. посіб. НТУ «Дніпровська політехніка». Дніпро, 2020. 38 с.
3. Мулеса О. Ю. Інформаційні системи та реляційні бази даних: навч. посіб. Електронне видання, 2018. 118 с. URL: https://dspace.uzhnu.edu.ua/jspui/bitstream/lib/19776/1/%D0%9C%D1%83%D0%BB%D0%B5%D1%81%D0%B0_%D0%91%D0%94.pdf
4. Шпортко О. В., Шпортко Л. В. Розробка баз даних в СУБД Microsoft Access 2010/2013. Практикум: навч. посіб. для студентів вищих навчальних закладів. Рівне, 2016. 184 с.
5. Ярцев В. П. Організація баз даних та знань: навч. посіб. Київ. ДУТ, 2018. 214 с.

Лабораторна робота № 3

Тема: Основні та похідні операції реляційної алгебри

Мета: закріпити та поглибити знання студентів, які отримані ними на лекціях та під час самостійної роботи з питань використання основних та похідних операцій реляційної алгебри.

Завдання: виконати операції реляційної алгебри над множинами даних.

Час виконання завдання: 4 год.

План заняття

1. Виконання завдання згідно з інструкцією.
2. Захист роботи та відповіді на контрольні питання.

Рекомендації до виконання

Реляційна алгебра – замкнута система операцій над відносинами в реляційній моделі даних.

3.1. Навести не менше 5 прикладів використання операції проекції із застосуванням констант, виразів ($=$, $>$, $<$, $<=$, $>=$) та логічних операторів (AND, OR, NOT) для таблиці свого варіанту бази даних.

Проекція (projection):

$$\sigma_{a_1, \dots, a_n}(R)$$

A1	A2	A3	A4	A5	A6

Операція проєкції є унарною (обробляє дані з одного відношення) і повертає вертикальну підмножину відношення R, яка створена за допомогою отримання значень вказаних атрибутів і виключення з результату рядків-дублікатів.

Приклад виконання операції проєкції:

Відношення R			Проекція $\pi_{A,C}(R)$	
A	B	C	A	C
a	b	c	a	c
c	a	d	c	d
c	b	d		

3.2. Навести не менше 5 прикладів використання операції вибірки із застосуванням констант, виразів ($=$, $>$, $<$, \leq , \geq) та логічних операторів (AND, OR, NOT) для таблиці свого варіанту бази даних.

Вибірка (selection):

$\sigma_{\text{ПРЕДИКАТ}}(R)$

A1	A2	A3	A4	A5	A6

Операція вибірки також працює з одним відношенням R і визначає результуюче відношення, яке містить тільки ті кортежі (рядки) відношення R, які задовольняють заданій умові (*предикату*).

Приклад виконання операції вибірки:

Відношення R			Вибірка $\sigma_{C=d}(R)$		
A	B	C	A	B	C
a	b	c	c	a	d
c	a	d	c	b	d
c	b	d			

3.3. Навести не менше 2 прикладів використання операції декартового добутку відношень для своєї бази даних.

Декартовий добуток (cartesian product):

$$R \times S$$

a
b

 \times

1
2
3

 $=$

a	1
a	2
a	3
b	1
b	2
b	3

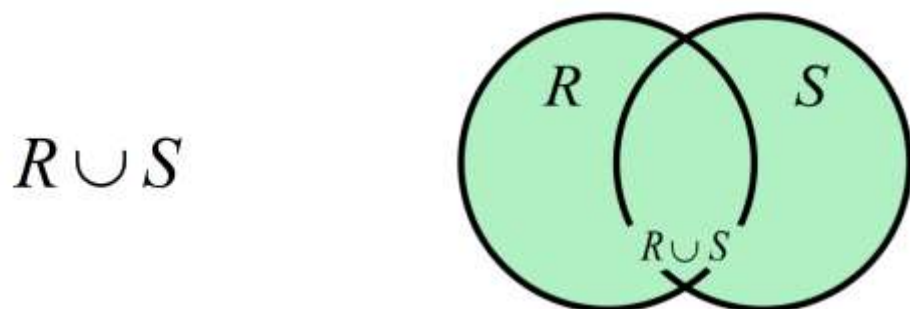
Операція декартового добутку створює нове відношення, що є результатом зчеплення (конкатенації) кожного рядка (кортежу) з відношення R з кожним рядком з відношення S.

Приклад виконання операції декартового добутку:

Відношення R		Відношення S			Декартовий добуток R×S				
A	B	C	D	E	A	B	C	D	E
1	4	g	h	a	1	4	g	h	a
2	5	a	b	c	1	4	a	b	c
3	6				2	5	g	h	a
					2	5	a	b	c
					3	6	g	h	a
					3	6	a	b	c

3.4. Навести не менше 2 прикладів використання операції об'єднання відношень для своєї бази даних. При проведенні операцій прослідкувати виконання вимоги на сумісність відношень по об'єднанню (необхідна попередня перевірка на співпадіння схем відношень).

Об'єднання (union):



При об'єднанні відношень R і S, з кортежами I та J відповідно, в результаті їх конкатенації можна одержати одне відношення з максимальною кількістю кортежів (I+J), причому кортежі-дублікати виключені.

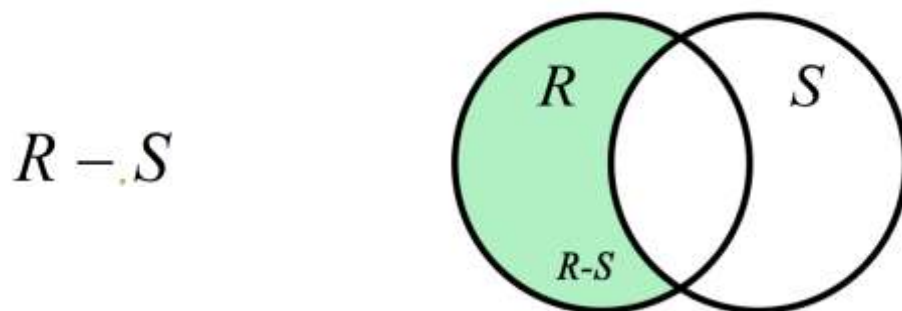
Об'єднання відношень можливе тільки в тому випадку, якщо співпадають їх схеми, тобто якщо вони мають однакову кількість атрибутів із співпадаючими доменами, тобто відношення повинні бути сумісні по об'єднанню.

Приклад виконання операції об'єднання:

Відношення R			Відношення S			Об'єднання $R \cup S$		
A	B	C	A	B	C	A	B	C
a	b	c	g	h	a	a	b	c
c	a	d	a	b	c	c	a	d
c	h	c	h	d	d	c	h	c
						g	h	a
						h	d	d

3.5. Навести не менше 2 прикладів використання операції об'єднання відношень для своєї бази даних. При проведенні операцій прослідкувати виконання вимоги на сумісність відношень (необхідна попередня перевірка на співпадіння схем відношень).

Різниця (set difference):



Різниця двох відношень R і S складається з кортежів, які є у відношенні R, але відсутні у відношенні S. Причому відношення R і S повинні бути сумісні.

Приклад виконання операції різниці:

Відношення R			Відношення S			Різниця $R-S$		
A	B	C	A	B	C	A	B	C
a	b	c	g	h	a	c	a	d
c	a	d	a	b	c	c	h	c
c	h	c	h	d	d			

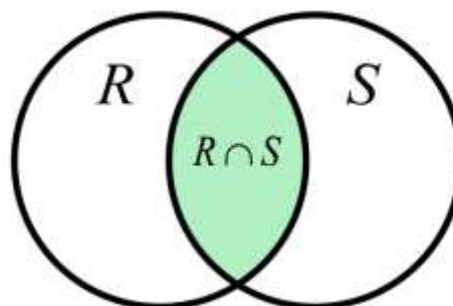
На підставі п'яти основних операцій можна отримати додаткові операції – природне з'єднання, перетин та поділ.

3.6. Навести не менше 2 прикладів використання операції перетину відношень для своєї бази даних. При проведенні операцій прослідкувати виконання вимоги на

сумісність відношень по об'єднанню (необхідна попередня перевірка на співпадіння схем відношень).

Перетин (intersection):

$$R \cap S$$



Операція перетину визначає відношення, яке містить кортежі, присутні як у відношенні R, так і у відношенні S. Відношення R і S повинні бути сумісні по об'єднанню.

Перетин можна сформулювати і на основі оператора різниці множин:

$$R \cap S = R - (R - S)$$

Відношення R			Відношення S			Перетин $R \cap S$		
A	B	C	A	B	C	A	B	C
a	b	c	g	h	a	a	b	c
c	a	d	a	b	c	c	a	d
c	h	c	c	a	d			
d	r	t	g	u	v			

3.7. Навести не менше 2 прикладів використання операції ділення відношень для своєї бази даних.

Ділення (division):

Нехай задано відношення зі схемою $R(M,N)$. Образом реляційного відношення R за кортежем $t_1 \in R[M]$ називається така множина кортежів $t_2 \in R[N]$, для яких зчеплення (t_1, t_2) належить відношенню R. Образ R за кортежем t_1 позначається $I_R(t_1)$ і формально визначається за формулою:

$$I_R(t_1) = \{ t_2 \mid t_2 \in R[N] \ \& \ (t_1, t_2) \in R \}$$

Приклад виконання операції ділення:

Відношення R			
A	B	C	D
a	b	c	b
a	b	g	h
c	f	g	h
c	f	c	b
a	v	c	b
c	v	g	h

Відношення S	
C	D
c	b
g	h

Ділення R/S	
A	B
a	b
c	f

3.7. Навести не менше 2 прикладів використання операції природного з'єднання відношень для своєї бази даних.

Природне з'єднання (natural join)

$R \bowtie S$	R			S		$R \bowtie S$
	A	B		B	C	
	a	1		1	x	a
	b	2		1	y	a
				3	z	

Операція з'єднання є похідною від операції декартового добутку, оскільки вона еквівалентна операції вибірки з декартового добутку двох операндів-відношень тих кортежів, які задовольняють умові (формулі вибірки), вказаній в предикаті з'єднання.

Природним з'єднанням називається з'єднання по еквівалентності двох відношень R і S, виконане по всіх загальних атрибутах х, з результатів якого виключається по одному екземпляру кожного загального атрибуту.

Приклад виконання операції природного з'єднання:

Відношення R			Відношення S			З'єднання $R \bowtie S$				
A	B	C	A	D	E	A	B	C	D	E
a	b	c	g	h	a	c	a	d	b	c
c	a	d	c	b	c	c	h	c	b	c
c	h	c	h	d	d	g	b	d	h	a
g	b	d								

Контрольні питання

1. Які операції з множинами в межах реляційної моделі дозволяє виконати реляційна алгебра?
2. Які особливості виконання унарних операцій?

3. Як виконують операцію зовнішнього з'єднання (outer join).
4. Як виконують операцію напівз'єднання (semi-join)?
5. Як виконують операцію тета-з'єднання (θ -join)?

Рекомендована література

1. Верес О. М., Пасічник В. В. Системи баз даних та знань. Книга 1: організація баз даних та знань. Київ: Магнолія, 2019. 440 с.
2. Мулеса О. Ю. Інформаційні системи та реляційні бази даних: навч. посіб. Електронне видання, 2018. 118 с. URL: https://dspace.uzhnu.edu.ua/jspui/bitstream/lib/19776/1/%D0%9C%D1%83%D0%BB%D0%B5%D1%81%D0%B0_%D0%91%D0%94.pdf
3. Реляційна алгебра. Реляційне числення: методичні вказівки для підготовки до контрольної роботи з нормативних навчальних дисциплін «Бази даних та розподілені інформаційно-аналітичні системи», «Організація баз даних та знань»/ В. В. Булатецький, Л. В. Булатецька; ВНУ ім. Лесі Українки. Луцьк: ВНУ ім. Лесі Українки, 2020. 36 с.

Лабораторна робота № 4

Тема: Налаштування СУБД та створення бази даних

Мета: закріпити та поглибити знання студентів, які отримані ними на лекціях та під час самостійної роботи з питань налаштування СУБД та створення бази даних.

Завдання: обрати та налаштувати СУБД, створити структуру власної бази даних та внести початкові дані.

Час виконання завдання: 4 год.

План заняття

1. Виконання завдання згідно з інструкцією.
2. Захист роботи та відповіді на контрольні питання.

Рекомендації до виконання

4.1. Обрати одну з наведених СУБД для створення та управління базою даних за обраною темою:

- Microsoft SQL Server (SQL Server Management Studio);
- Microsoft Access;
- PostgreSQL (pgAdmin чи PostgreSQL Web Client);
- MySQL (MySQL Workbench) або MariaDB.

Найбільш простим варіантом реалізації бази даних буде використання продукту Microsoft Access.

4.2. Встановити обрану СУБД та налаштувати її роботу.

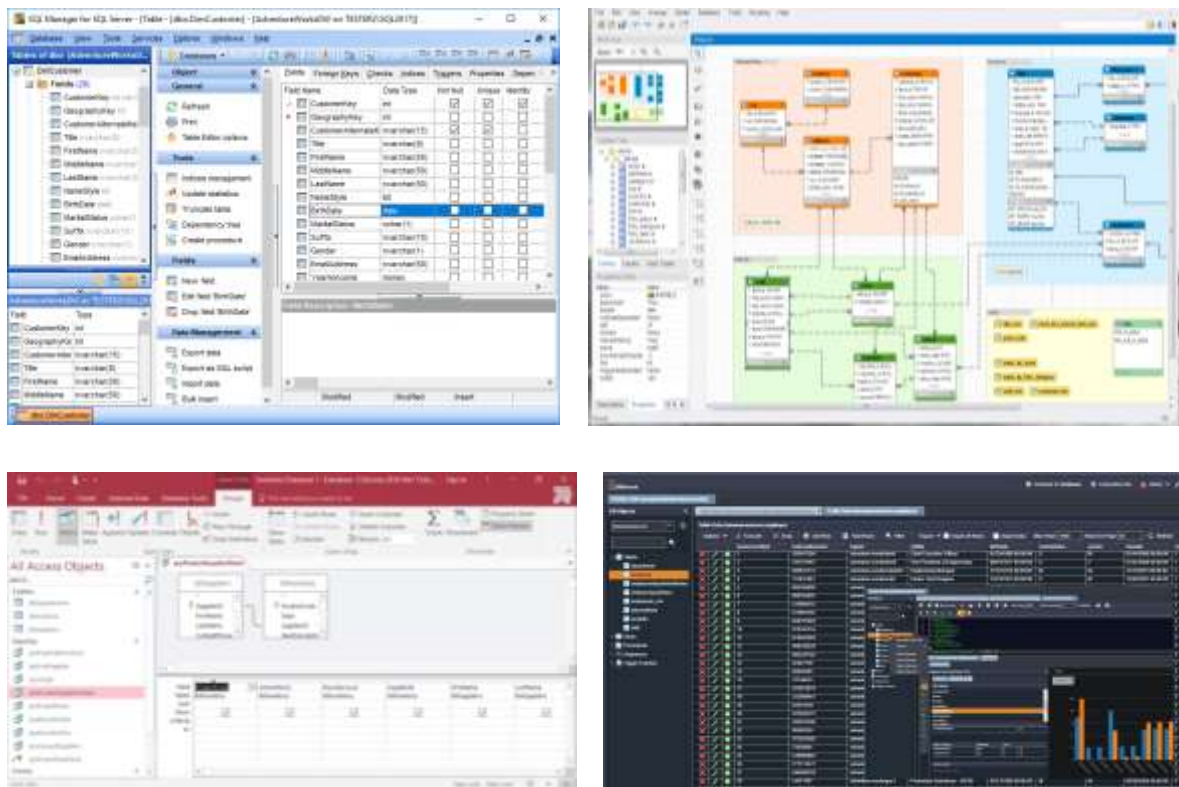


Рис. 4.1. Інтерфейси клієнтських додатків для роботи з СУБД: SQL Server Management Studio, MySQL Workbench, Microsoft Access, PostgreSQL Web Client.

4.3. Використовуючи команди мови SQL, створити структуру власної бази даних.

Зверніть увагу на те, що різні СУБД можуть використовувати різні діалекти мови SQL з певними особливостями написання команд. Більш детальну інформацію можна знайти у довідкових матеріалах розробників та користувачів обраної СУБД.

Приклад створення таблиць «Співробітники» та «Замовлення» в режимі конструктора в СУБД Access згідно з даними, наведеними в табл. 2.2 і табл. 2.4, наведено на рис. 4.2.

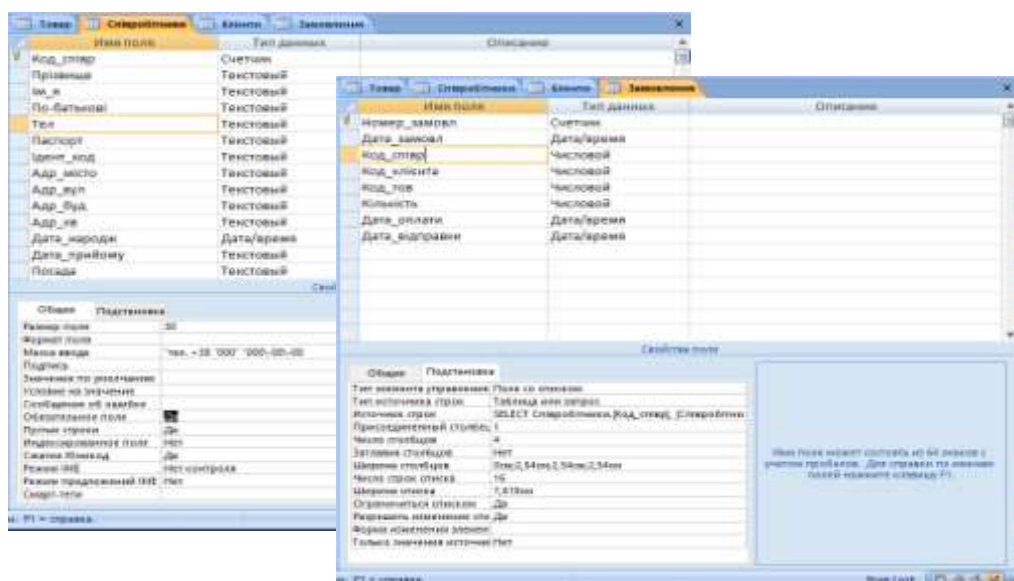


Рис. 4.2. Приклад створення таблиць в режимі конструктора в СУБД Access

Наступним етапом є налаштування зв'язків між створеними таблицями. Для цього на вкладці «Робота з базами даних» обираємо команду «Схема даних». Важливо зазначити, що при створенні зв'язків необхідно встановити позначку «Забезпечення цілісності даних» для кожного зв'язку, як це показано на рис. 4.3.

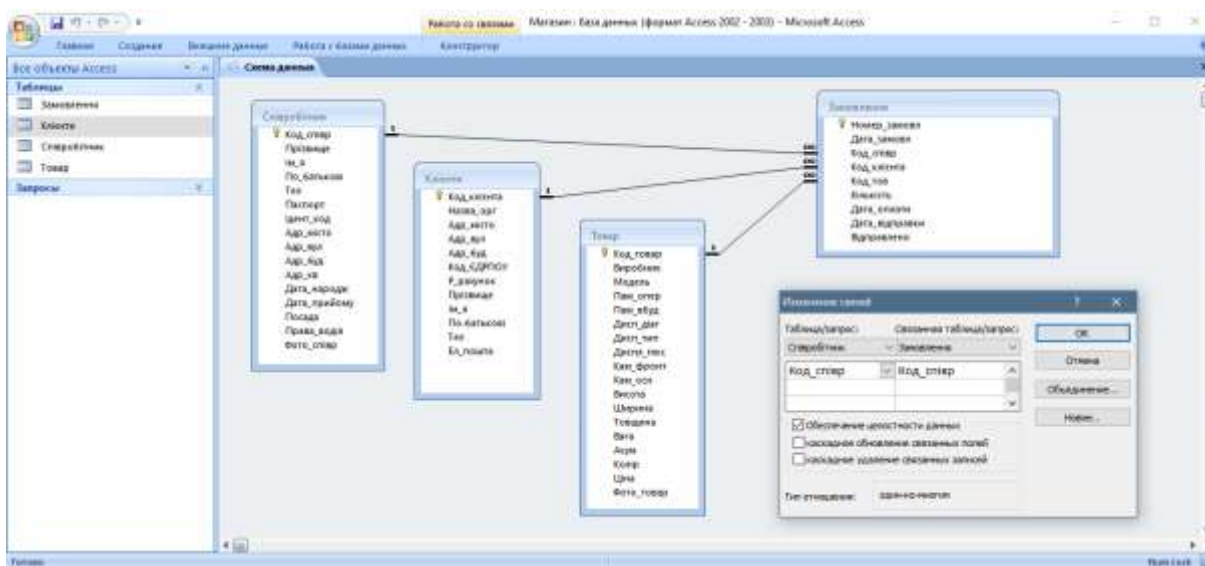


Рис. 4.3. Схема даних

4.4. Всі створені таблиці необхідно заповнити початковими даними. Для навчальних цілей достатньо такої мінімально необхідної кількості позицій:

- 10 товарів в таблиці «Товар»;
- 3 особи в таблиці «Співробітники»;
- 5 назв організацій в таблиці «Клієнти»;
- 50 замовлень в таблиці «Замовлення».

Після створення та заповнення бази даних можна здійснювати різноманітні операції, які дозволяють знаходити та обирати дані згідно з певними умовами; сортувати та групувати їх; виконувати арифметичні обчислення та проводити логічні операції.

Контрольні питання

1. Які СУБД можна встановити на сервер під управлінням ОС Linux?
2. Які СУБД зручно використовувати при розробці мобільних додатків під ОС Android?
3. Назвіть особливості нереляційних баз даних (NoSQL).
4. Якими типами даних може оперувати обрана вами СУБД?
5. Яким чином налаштовуються зв'язки між таблицями в обраній вами СУБД?

Рекомендована література

1. Access developer documentation. URL: <https://docs.microsoft.com/en-us/office/client-developer/access/access-home>
2. Мулеса О. Ю. Інформаційні системи та реляційні бази даних: навч. посіб. Електронне видання, 2018. 118 с. URL: https://dspace.uzhnu.edu.ua/jspui/bitstream/lib/19776/1/%D0%9C%D1%83%D0%BB%D0%B5%D1%81%D0%B0_%D0%91%D0%94.pdf
3. Шпортько О. В., Шпортько Л. В. Розробка баз даних в СУБД Microsoft Access 2010/2013. Практикум: навч. посіб. для студентів вищих навчальних закладів. Рівне, 2016. 184 с.

Лабораторна робота № 5

Тема: Використання мови SQL для відображення інформації

Мета: закріпити та поглибити знання студентів, які отримані ними на лекціях та під час самостійної роботи з питань використання команди SELECT мови SQL.

Завдання: використовуючи команду SELECT, створити запити за заданими умовами.

Час виконання завдання: 4 год.

План заняття

1. Виконання завдання згідно з інструкцією.
2. Захист роботи та відповіді на контрольні питання.

Рекомендації до виконання

Для опису, модифікації та управління даними використовують мову програмування SQL (англ. structured query language – мова структурованих запитів). Простота та декларативність цієї мови зумовили широке використання інструкцій SQL при роботі з базами даних.

Найбільш популярною інструкцією мови SQL є інструкція SELECT. Загальна структура цієї інструкції для Microsoft Access має вигляд:

```
SELECT [предикат] { * | таблиця.* | [таблиця.]поле1 [AS псевдонім1] [,
[таблиця.]поле2 [AS псевдонім2] [, ...]]}
FROM табличний_вираз [, ...] [IN зовнішня_база_даних]
[WHERE... ]
[GROUP BY... ]
[HAVING... ]
[ORDER BY... ]
[WITH OWNERACCESS OPTION]
```

5.1. Використовуючи інструкцію SELECT, створити 3 запита на виведення всіх записів з обраних вами таблиць (одна таблиця – один запит) Наприклад, вивести всі записи, що зберігаються в таблицях «Товар» та «Співробітники».

No	Виробник	Модель	Пам_о	Пам_в	Диск_л	Диск_тип	Дискл_пкс	Кам_фр	Кам_ос	Висота	Ширини	Товщи	Вага	Акум	Колір	Ціна	Фото
1	Xiaomi	Redmi 10A	2	32	6,53	IPS	720x1600	5	13	165	77	9	184	5000	чорний	5079	
2	Xiaomi	Note 10S	6	128	6,43	AMOLED	1080 x 2400	13	64	160	74	8	179	5000	білий	8892	
3	Samsung	Galaxy M23	4	128	6,6	IPS	1080 x 2400	8	50	165	77	8	198	5000	білий	10280	
4	OnePlus	Nord 2	8	128	6,43	AMOLED	1080 x 2400	32	50	159	73	8	183	4500	чорний	16550	
5	Samsung	Galaxy S20	6	128	6,5	Super AMOLED	1080 x 2400	32	12	160	75	8	190	4500	рожевий	19000	
6	Google	Pixel 6a	6	128	6,1	AMOLED	1080 x 2400	8	12	152	72	9	176	4410	білий	21926	

Рис. 5.1. Фрагмент таблиці «Товар»

5.2. За допомогою інструкції SELECT створити 3 запита на виведення окремих атрибутів обраної вами сутності.

Наприклад, можна вивести ПІБ та посади всіх співробітників.

```
SELECT Прізвище, Ім_я, По_батькові, Посада, Код_співр
FROM Співробітник;
```

5.3. Створити 3 запита на виведення даних без повторів значень.

Наприклад, можна вивести список виробників товарів.

```
SELECT DISTINCT Виробник
FROM Товар;
```

5.4. Створити 3 запита на виведення даних з подвійною умовою.

Наприклад, вивести коди і моделі товарів Samsung білого кольору.

```
SELECT Товар.Код_товар, Товар.Виробник, Товар.Модель, Товар.Колір
FROM Товар
WHERE (((Товар.Виробник)="Samsung") AND ((Товар.Колір)="білий"));
```

Код_товар	Виробник	Модель	Колір
3	Samsung	Galaxy M23	білий

Рис. 5.2. Результат виконання запиту до задачі 5.4

5.5. Створити 3 запита на виведення даних з подвійною умовою, використанням логічних операторів (AND, OR, NOT) та реляційних операторів (=, <>, >, <, >=, <=, !=, !<).

Наприклад, вивести код, модель, виробника, обсяг оперативної пам'яті, колір та ціну товару за умови, що виробник будь-який, але не Samsung; обсяг оперативної пам'яті більше 6; колір білий або чорний і ціна не перевищує 12 тис.


```
SELECT Код_товар, Модель, Виробник, Пам_опер, Колір, Ціна
FROM Товар
WHERE ((Not (Виробник)="Samsung")) AND ((Пам_опер)>=6) AND ((Колір)="білий" Or
(Колір)="чорний") AND ((Ціна)<=12000));
```

5.6. Створити 3 запита на використання спеціального оператора IN.

Наприклад, вивести всі параметри всіх моделей телефонів Samsung, OnePlus та Google.

```
SELECT *
FROM Товар
WHERE ((Виробник)="Samsung" OR "OnePlus" OR "Google");
```

5.7. Створити 3 запита на використання спеціального оператора BETWEEN.

Наприклад, вивести код товару, виробника, модель, ціну та діагональ дисплею за умови, що ціна знаходиться в межах від 8 до 15 тис., а діагональ дисплею – від 6,4 до 6,6 дюймів.

```
SELECT Код_товар, Виробник, Модель, Ціна, Дисп_діаг
FROM Товар
WHERE (((Ціна) BETWEEN 8000 AND 15000) AND ((Дисп_діаг) BETWEEN 6.4 AND
6.6));
```

5.8. Створити 3 запита на використання службового слова LIKE.

Наприклад, вивести технічні характеристики всіх моделей, у назві яких присутнє слово "Galaxy".

5.9. Створити 3 запита на використання службового слова LIKE зі спеціальними символами підстановки, наведеними в табл. 5.1.

Наприклад, вивести дані всіх клієнтів, назва яких починається з "ФОП" і вони зареєстровані в м. Житомирі.

```
SELECT Клієнти.Код_клієнта, Клієнти.Назва_орг, Клієнти.Адр_місто, Клієнти.Прізвище,
Клієнти.Ім_я, Клієнти.[По-батькові], Клієнти.Тел
FROM Клієнти
WHERE (((Клієнти.Назва_орг) Like "ФОП*") AND ((Клієнти.Адр_місто)="Житомир"));
```

5.10. Створити 3 запита на використання оператора IS NULL для пошуку порожніх значень у комірках таблиць.

Наприклад, знайти товари, для яких не повністю заповнені всі атрибути.

Символи підстановки

Тип співпадання	Зразок	Співпадає (True)	Не співпадає (False)
Декілька знаків	a*a	aa, aBa, aBBBa	aBC
	ab	abc, AABb, Xab	aZb, bac
Спеціальний знак	a[*]a	a*a	aaa
Декілька знаків	ab*	abcdefg, abc	cab, aab
Одиночний знак	a?a	aaa, a3a, aBa	aBBBa
Одиночна цифра	a#a	a0a, a1a, a2a	aaa, a10a
Діапазон знаків	[a-z]	f, p, j	2, &
Поза діапазоном	[!a-z]	9, &, %	b, a
Не цифра	[!0-9]	A, a, &, ~	0, 1, 9
Комбінований вираз	a[!b-m]#	An9, az0, a99	abc, aj0

Контрольні питання

1. Якими командами можна знайти та вивести дані, що належать до певної множини значень?
2. Використання якої команди дозволить при виведенні даних приховати дублікати записів?
3. Напишіть команду для пошуку символу % всередині довільної послідовності символів.
4. Для чого використовують інструкцію WHERE?
5. Яким набором виразів можна замінити використання спеціального оператора IN?

Рекомендована література

1. Beaulieu A. Learning SQL. Second Edition. O'Reilly Media, Inc., 2009. 338 p. URL: <https://media.oaipdf.com/pdf/64103754-f42b-438c-99e3-59d6a294b2ff.pdf>
2. Верес О. М., Пасічник В. В., Берко А. Ю. Системи баз даних та знань. Книга 2: системи управління базами даних та знань. Київ: Магнолія, 2019. 584 с.
3. Мулеса О. Ю. Інформаційні системи та реляційні бази даних: навч. посіб. Електронне видання, 2018. 118 с. URL: https://dspace.uzhnu.edu.ua/jspui/bitstream/lib/19776/1/%D0%9C%D1%83%D0%BB%D0%B5%D1%81%D0%B0_%D0%91%D0%94.pdf
4. Ярцев В. П. Організація баз даних та знань: навч. посіб. Київ. ДУТ, 2018. 214 с.

Лабораторна робота № 6

Тема: Агрегатні функції

Мета: закріпити та поглибити знання студентів, які отримані ними на лекціях та під час самостійної роботи з питань використання агрегатних функцій.

Завдання: використовуючи агрегатні функції мови SQL, створити запити за заданими умовами.

Час виконання завдання: 4 год.

План заняття

1. Виконання завдання згідно з інструкцією.
2. Захист роботи та відповіді на контрольні питання.

Рекомендації до виконання

Агрегатні функції розраховують єдине значення для колекції вхідних значень. В якості колекцій даних може виступати множина (set), мультимножина (multiset) або список (list).

В програмі MS Access більшість агрегатних функцій розташовані в розділі «Вбудовані функції => Статистичні» (рис. 6.1).

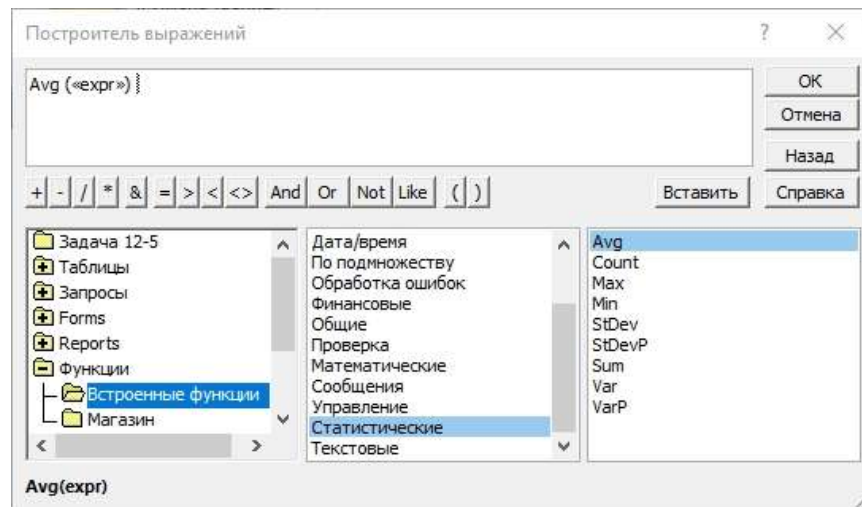


Рис. 6.1. Вікно для побудови виразів в програмі Access

Функція AVG – обчислює середнє арифметичне значення для заданого набору значень (для поля). При розрахунках ігноруються порожні (NULL) поля. Разом з ключовим полем ABS повертає абсолютне значення середнього значення поля (наприклад: ABS AVG ([назва_стовпця])).

Функція COUNT – обчислює загальну кількість записів, які отримані через запит. Обробляє значення, записані у будь-якому типі даних, зокрема текстовому. При розрахунках ігноруються порожні (NULL) поля.

Для визначення загальної кількості рядків у таблиці достатньо використати простий вираз COUNT(*). Для обчислення кількості не порожніх записів одночасно в

декількох полях використовують амперсанд (&), наприклад: COUNT ('Дата_оплати & Дата_відправки').

Функція FIRST або LAST – повертає перший або останній запис з набору даних, які отримані через запит. Якщо додатково не застосувати впорядкування (ключове слово ORDER BY), то отриманий результат буде довільним першим або останнім значенням в наборі, оскільки записи запит повертає у випадковому порядку.

Функція MIN або MAX – повертає найменше або найбільше значення з набору даних, які отримані через запит. Цими агрегатними функціями можна обробляти не лише константи, а й поля, в яких використовуються обчислення чи інші функції (окрім інших агрегатних функцій).

Функція SUM – повертає суму всіх чисел з набору значень, які отримані через запит. При розрахунках ігноруються порожні (NULL) поля. З її допомогою можна обробляти не тільки константи, а й поля, в яких використовуються обчислення або інші функції.

Функція STDEV – обчислює середньоквадратичне (стандартне) відхилення величин, які отримані через запит, від їх середнього значення. Можна вказати не тільки перелік значень, а й посилання на масив. Середньоквадратичне (стандартне) відхилення обчислюється за допомогою методу «n-1».

Функція VARP або VAR – обчислює значення дисперсії для генеральної сукупності або для вибірки з генеральної сукупності значень, які отримані через запит. Якщо запит повертає менше двох значень, то функції VARP та VAR повертають Null-значення, що означає неможливість обчислення дисперсії.

6.1. Створити 3 запита на використання спеціального оператора AVG з виведенням результату в поіменованій стовпчик, використавши інструкцію AS.

Наприклад, вивести середнє значення вартості смартфонів з розміром оперативної пам'яті від 6 до 8 гігабайт та діагоналлю від 6,4 до 6,6 дюймів в стовпчик з ім'ям "Середня ціна".

```
SELECT AVG (Товар.Ціна) AS Середня_ціна
FROM Товар
WHERE (((Товар.Пам_опер) BETWEEN 6 And 8) AND ((Товар.Дисп_діаг) BETWEEN 6.4
AND 6.6));
```

6.2. Створити 3 запита на використання спеціального оператора COUNT з виведенням результату в поіменованій стовпчик, використавши інструкцію AS.

Наприклад, вивести в стовпчик з ім'ям "Кількість моделей білого кольору" кількість моделей смартфонів білого кольору.

```
SELECT COUNT (Товар.Колір) AS Кількість_моделей_білого_кольору
FROM Товар
WHERE (((Товар.Колір)="білий"));
```

6.3. Створити 3 запита на використання спеціального оператора COUNT в поєднанні з інструкціями DISTINCT та AS.

Наприклад, вивести в стовпчик з ім'ям "Кількість клієнтів" кількість клієнтів, які замовляли товари в попередньому році.

```
SELECT COUNT(Замовлення.Код_клієнта) AS Кільк_клієнтів_в_попередн_році  
FROM Замовлення  
WHERE (Year(Замовлення.Дата_замовл)=Year(Date())-1);
```

6.4. Створити 3 запита на використання спеціальних операторів MIN і MAX в поєднанні з умовами "WHERE" для інших атрибутів.

Наприклад, вивести інформацію про модель найдешевшого смартфона з розміром оперативної пам'яті від 6 до 8 гігабайт та діагоналлю від 6,4 до 6,6 дюймів в стовпчик з ім'ям "Мінімальна ціна".

```
SELECT Min(Товар.Ціна) AS Мінімальна_ціна  
FROM Товар  
WHERE (((Товар.Пам_опер) BETWEEN 6 And 8) AND ((Товар.Дисп_діаг) BETWEEN 6.4  
AND 6.6));
```

6.5. Створити 3 запита на використання спеціального оператора SUM в поєднанні з умовами "WHERE" для інших атрибутів.

Наприклад, вивести інформацію про загальну кількість телефонів, проданих за попередній рік.

```
SELECT SUM([Замовлення].[Кількість]) AS Загальн_кільк  
FROM Замовлення  
WHERE (((Замовлення.Відправлено)=True)  
AND (Year(Замовлення.Дата_замовл)=Year(Date())-1));
```

Контрольні питання

1. Які типи даних використовуються у вашій БД?
2. Опишіть загальну структуру команди SELECT.
3. З якою метою використовують функцію AS?
4. Як знайти та вивести поточну дату, місяць, день, час?
5. Виконання яких логічних та арифметичних операцій над даними підтримує обрана вами СУБД?

Рекомендована література

1. Access developer documentation. URL: <https://docs.microsoft.com/en-us/office/client-developer/access/access-home>
2. Beaulieu A. Learning SQL. Second Edition. O'Reilly Media, Inc., 2009. 338 p. URL: <https://media.oaipdf.com/pdf/64103754-f42b-438c-99e3-59d6a294b2ff.pdf>
3. Верес О. М., Пасічник В. В., Берко А. Ю. Системи баз даних та знань. Книга 2: системи управління базами даних та знань. Київ: Магнолія, 2019. 584 с.
4. Статистичні функції (довідка): Microsoft Підтримка. URL: <https://support.microsoft.com/uk-ua/office/%D1%81%D1%82%D0%B0%D1%82%D0%B8%D1%81%D1%82%D0%B8%D1%87%D0%BD%D1%96-%D1%84%D1%83%D0%BD%D0%BA%D1%86%D1%96%D1%97-%D0%B4%D0%BE%D0%B2%D1%96%D0%B4%D0%BA%D0%B0-624dac86-a375-4435-bc25-76d659719ffd>
5. Ярцев В. П. Організація баз даних та знань: навч. посіб. Київ. ДУТ, 2018. 214 с.

Лабораторна робота № 7

Тема: З'єднання таблиць та групові операції

Мета: закріпити та поглибити знання студентів, які отримані ними на лекціях та під час самостійної роботи з питань конкатенації та групових операцій над даними та з'єднання таблиць.

Завдання: використовуючи оператори JOIN та GROUP BY, створити запити за заданими умовами.

Час виконання завдання: 4 год.

План заняття

1. Виконання завдання згідно з інструкцією.
2. Захист роботи та відповіді на контрольні питання.

Рекомендації до виконання

JOIN – операція з'єднання двох таблиць, що утворює нову тимчасову таблицю (з'єднану таблицю).

CROSS JOIN – операція з'єднання за умовами декартового добутку. Після виконання даної операції ми отримаємо таблицю, що буде являти собою набір всіх можливих поєднань рядків двох таблиць.

INNER JOIN – операція з'єднання двох таблиць, при якій, кожен рядок з першої таблиці зіставляється з кожним рядком другої таблиці, після чого відбувається перевірка умови. Якщо умова істинна, рядки потрапляють у результуючу таблицю.

LEFT JOIN – операція з'єднання двох таблиць, яка повертає всі записи лівої

таблиці доповнені відповідними кортежами з правої таблиці. У випадку, якщо в правій таблиці немає збігу, то запис буде доповнено значеннями NULL.

RIGHT JOIN – операція з'єднання двох таблиць, яка додає до записів правої таблиці відповідні кортежі лівої таблиці або значення NULL.

FULL JOIN – операція, яка поєднує записи двох таблиць за відповідним критерієм. Записи для яких не було знайдено збігу будуть доповнені значеннями NULL.

Речення **ORDER BY** використовують для впорядкування результатів запиту за зростанням чи зменшенням значень.

Речення **GROUP BY** дозволяє визначати підмножину значень в певному полі (або групі полів) в термінах іншого поля, і застосовувати функцію агрегування до підмножини.

7.1. Створити 2 запита на використання оператора **INNER JOIN**, поєднавши дані з двох таблиць.

Наприклад, вивести інформацію про виробника, назву моделі та колір замовлених телефонів.

```
SELECT Замовлення.Номер_замовл, Замовлення.Дата_замовл, Товар.Виробник,  
Товар.Модель, Товар.Колір  
FROM Товар INNER JOIN Замовлення ON Товар.Код_товар = Замовлення.Код_тов;
```

7.2. Створити 2 запита на використання оператора **INNER JOIN**, поєднавши дані з трьох таблиць.

Наприклад, вивести інформацію про номер замовлення із зазначенням прізвища та ім'я відповідального співробітника і назви організації клієнта, який зробив замовлення.

```
SELECT Замовлення.Номер_замовл, Замовлення.Дата_замовл, Співробітник.Прізвище,  
Співробітник.Ім_я, Клієнти.Назва_орг, Клієнти.Адр_місто  
FROM Співробітник INNER JOIN (Клієнти INNER JOIN Замовлення ON  
Клієнти.Код_клієнта = Замовлення.Код_клієнта) ON Співробітник.Код_співр =  
Замовлення.Код_співр;
```

7.3. Створити 2 запита на використання оператора **ORDER BY** для подвійного впорядкування даних у двох поєднаних таблицях.

Наприклад, вивести інформацію про замовлені моделі телефонів з їх упорядкуванням за кольором та ціною.

```
SELECT Замовлення.Номер_замовл, Замовлення.Дата_замовл, Товар.Виробник,  
Товар.Модель, Товар.Колір, Товар.Ціна  
FROM Товар INNER JOIN Замовлення ON Товар.Код_товар = Замовлення.Код_тов
```

```
ORDER BY Товар.Колір ASC, Товар.Ціна DESC;
```

7.4. Створити 2 запита на використання оператора INNER JOIN та умови WHERE з діапазоном даних про дату або час з подальшим упорядкуванням за обраним атрибутом.

Наприклад, знайти номери замовлень, які за останні два тижні надійшли від організацій, розташованих в м. Житомир, та впорядкувати замовлення за назвою організації.

```
SELECT Замовлення.Номер_замовл, Замовлення.Дата_замовл, Клієнти.Назва_орг,  
Клієнти.Адр_місто  
FROM Клієнти INNER JOIN Замовлення ON Клієнти.Код_клієнта =  
Замовлення.Код_клієнта  
WHERE (((Замовлення.Дата_замовл)>=(Now()-14)  
AND (Замовлення.Дата_замовл)<Now())) AND ((Клієнти.Адр_місто)="Житомир"))  
ORDER BY Клієнти.Назва_орг;
```

7.5. Створити 2 запита на використання оператора GROUP BY спільно з агрегатними функціями.

Наприклад, вивести інформацію про загальну кількість замовлень, оформлених кожним менеджером.

```
SELECT Співробітник.Прізвище, Count([Замовлення].[Номер_замовл]) AS  
Кількість_замовлень  
FROM Співробітник INNER JOIN Замовлення ON Співробітник.Код_співр =  
Замовлення.Код_співр  
GROUP BY Співробітник.Прізвище;
```

7.6. Створити 3 запита на використання оператора GROUP BY спільно з математичними операціями.

Наприклад, використовуючи дані таблиці «Замовлення» розрахувати показники, що характеризують ефективність роботи кожного співробітника, які обслуговували замовлення. Для цього можна розрахувати загальну суму та середню вартість замовлень, що припадають на кожного співробітника.

```
SELECT Замовлення.Код_співр, SUM([Товар].[Ціна]*[Замовлення].[Кількість]) AS  
Загальна_сума, AVG([Товар].[Ціна]*[Замовлення].[Кількість]) AS Середня_вартість  
FROM Товар INNER JOIN Замовлення ON Товар.Код_товар = Замовлення.Код_тов  
GROUP BY Замовлення.Код_співр;
```

7.7. Створити 3 запита на використання оператора GROUP BY з умовою WHERE,

реченням HAVING, агрегатною функцією та впорядкуванням результату.

Наприклад, використовуючи дані таблиці «Замовлення», знайти загальну кількість замовлень для кожної моделі телефону білого кольору. Вивести тільки ті моделі, яких продано більше 3 шт., та впорядкувати список за спаданням значень.

```
SELECT Sum(Замовлення.Кількість) AS [Кількість_всього], Товар.Виробник,  
Товар.Модель, Товар.Колір  
FROM Товар INNER JOIN Замовлення ON Товар.Код_товар = Замовлення.Код_тов  
WHERE ((Товар.Колір)="білий")  
GROUP BY Товар.Виробник, Товар.Модель, Товар.Колір  
HAVING (Sum(Замовлення.Кількість)>3)  
ORDER BY Sum(Замовлення.Кількість) DESC;
```

Контрольні питання

1. Чим застосування умови WHERE відрізняється від застосування умови HAVING?
2. Порівняйте результат від застосування операторів RIGHT JOIN та LEFT JOIN.
3. Які особливості застосування оператора GROUP BY?
4. Який з операторів – WHERE чи HAVING інструкції SELECT буде виконаний першим?
5. Який з операторів інструкції SELECT в прикладі 7.6 буде виконаний останнім?

Рекомендована література

1. Access developer documentation. URL: <https://docs.microsoft.com/en-us/office/client-developer/access/access-home>
2. Beaulieu A. Learning SQL. Second Edition. O'Reilly Media, Inc., 2009. 338 p. URL: <https://media.oaipdf.com/pdf/64103754-f42b-438c-99e3-59d6a294b2ff.pdf>
3. Верес О. М., Пасічник В. В., Берко А. Ю. Системи баз даних та знань. Книга 2: системи управління базами даних та знань. Київ: Магнолія, 2019. 584 с.
4. Мулеса О. Ю. Інформаційні системи та реляційні бази даних: навч. посіб. Електронне видання, 2018. 118 с. URL: https://dspace.uzhnu.edu.ua/jspui/bitstream/lib/19776/1/%D0%9C%D1%83%D0%BB%D0%B5%D1%81%D0%B0_%D0%91%D0%94.pdf
5. Ярцев В. П. Організація баз даних та знань: навч. посіб. Київ. ДУТ, 2018. 214 с.

Лабораторна робота № 8

Тема: Використання підзапитів мовою SQL

Мета: закріпити та поглибити знання студентів, які отримані ними на лекціях та під час самостійної роботи з питань використання підзапитів мовою SQL.

Завдання: використовуючи оператор SELECT, створити підзапити за заданими умовами.

Час виконання завдання: 6 год.

План заняття

1. Виконання завдання згідно з інструкцією.
2. Захист роботи та відповіді на контрольні питання.

Рекомендації до виконання

8.1 Створити 2 запита з використанням одного підзапиту, обмеження та агрегатної функції.

Наприклад, вивести інформацію про всі смартфони з діагоналлю більше 6 дюймів, ціна яких нижче за середню для таких смартфонів.

```
SELECT Товар.*  
FROM Товар  
WHERE (Товар.Дисп_діаг >= 6) AND (Ціна <  
    (SELECT AVG(Ціна)  
    FROM Товар)  
);
```

8.2. Створити 2 запита з використанням подвійного підзапиту.

Наприклад, вивести інформацію про клієнтів, які замовили телефон моделі Xiaomi Note 10S.

```
SELECT Прізвище  
FROM Клієнти WHERE Клієнти.Код_клієнта IN  
    (SELECT Код_клієнта  
    FROM Замовлення WHERE Замовлення.Код_тов IN  
        (SELECT Код_товар  
        FROM Товар WHERE Модель = "Note 10S")  
    )  
;
```

8.3. Створити 2 запита з пошуку другої позиції в списку за вказаним атрибутом.

Наприклад, вивести назву моделі смартфона, що займає другу позицію серед найбільш дорогих смартфонів.

```

SELECT Модель
FROM Товар
WHERE Ціна =
    (SELECT MAX(Ціна)
     FROM Товар
     WHERE Ціна NOT IN (
         SELECT MAX(Ціна)
         FROM Товар)
    );

```

8.4. Створити 2 запити з пошуку екземплярів сутності, які відповідають певній умові, з груп сутностей.

Наприклад, використовуючи дані таблиці «Товар», для кожного значення обсягу оперативної пам'яті знайти моделі смартфонів з мінімальною ціною.

```

SELECT Товар.Виробник, Товар.Модель, Товар.Пам_операт, Товар.Ціна, Товар.Колір
FROM Товар
WHERE ((Товар.Ціна) IN (
    SELECT MIN(Товар.Ціна)
    FROM Товар
    GROUP BY Товар.Пам_операт))
ORDER BY Товар.Пам_операт;

```

Задача 12-3					
Виробник	Модель	Пам_опе	Ціна	Колір	
Sony	Xperia 5	6	30 500,00	чорний	
Xiaomi	Redmi K40	8	12 100,00	білий	
*					

Рис. 8.1. Результат виконання запиту до задачі 8.4

8.5. Створити 2 запити з пошуку значення атрибуту однієї сутності за умови введення користувачем діапазону даних параметра іншої сутності. За умови, що сутності зв'язані між собою через ключове поле.

Наприклад, вивести прізвища клієнтів, які купували смартфони заданого діапазону вартості (вартість у якості параметру задає користувач при активації запиту).

```

SELECT Клієнти.Прізвище, Клієнти.Ім_я, Клієнти.[По-батькові]
FROM Клієнти
WHERE Клієнти.Код_клієнта IN (
    SELECT DISTINCT Замовлення.Код_клієнта
    FROM Замовлення
    WHERE Замовлення.Код_тов IN(
        SELECT Товар.Код_товар

```

```

FROM Товар
WHERE (((Товар.Ціна) BETWEEN ['Введіть початкове значення ціни']
      And ['Введіть кінцеве значення ціни']))
)
);

```

8.6. Створити 2 запити з пошуку найбільш популярного запису в списку за введеним користувачем параметром.

Наприклад, вивести назву найбільш популярної моделі смартфона заданого кольору (колір у якості параметру задає користувач при активації запиту).

Для спрощення коду запиту його необхідно розділити на два взаємодіючих запита.

Запит 8.6.

```

SELECT Товар.Виробник, Товар.Модель
FROM Товар
WHERE Товар.Код_товар IN(
  SELECT QW.Код_тов
  FROM QW
  WHERE QW.Кількість_Продаж=
    (SELECT MAX(QW.Кількість_Продаж)
     FROM QW));

```

Допоміжний запит QW.

```

SELECT Замовлення.Код_тов, SUM(Замовлення.Кількість) AS Кількість_Продаж
FROM Замовлення
WHERE Замовлення.Код_тов IN (
  SELECT Товар.Код_товар
  FROM Товар
  WHERE Товар.Колір = ["Введіть колір"])
GROUP BY Замовлення.Код_тов;

```

Контрольні питання

1. Які можливості надає використання підзапитів?
2. Яким чином оформити питання до користувача на введення даних у вигляді параметрів?
3. В якій послідовності виконуватимуться операції у прикладі завдання 8.6?
4. Які можливості надає використання оператора IN?
5. Як виключити межі діапазону значень, вказаних у реченні BETWEEN з діапазону допустимих значень?

Рекомендована література

1. Access developer documentation. URL: <https://docs.microsoft.com/en-us/office/client-developer/access/access-home>
2. Beaulieu A. Learning SQL. Second Edition. O'Reilly Media, Inc., 2009. 338 p. URL: <https://media.oaipdf.com/pdf/64103754-f42b-438c-99e3-59d6a294b2ff.pdf>
3. Верес О. М., Пасічник В. В., Берко А. Ю. Системи баз даних та знань. Книга 2: системи управління базами даних та знань. Київ: Магнолія, 2019. 584 с.
4. Shin T. 5 Common SQL Interview Problems for Data Scientists. URL: <https://towardsdatascience.com/5-common-sql-interview-problems-for-data-scientists-1bfa02d8bae6>
5. Ярцев В. П. Організація баз даних та знань: навч. посіб. Київ. ДУТ, 2018. 214 с.

Лабораторна робота № 9

Тема: Проектування звітності

Мета: закріпити та поглибити знання студентів, які отримані ними на лекціях та під час самостійної роботи з питань підготовки та створення звітів.

Завдання: використовуючи вбудовані інструменти СУБД, підготувати та оформити звіти за заданими умовами.

Час виконання завдання: 6 год.

План заняття

1. Виконання завдання згідно з інструкцією.
2. Захист роботи та відповіді на контрольні питання.

Рекомендації до виконання

9.1. Зазвичай умовний оператор в SQL реалізується за допомогою конструкції CASE:

```
CASE
  WHEN condition_1 THEN result_1
  WHEN condition_2 THEN result_2
  [WHEN ...]
  [ELSE else_result]
END
```

Проте в СУБД MS Access умовний оператор реалізовано за допомогою конструкції SWITCH. Тому, при виконанні завдання уважно перегляньте технічну документацію до обраної вами СУБД на предмет правил використання умовного оператора.

Завдання 9.1: Створити 2 запита з використанням умовного оператора з не менш, ніж двома умовами для обчислення.

Наприклад, обчислити ціну товару зі знижкою за умови, що при покупці 2-х товарів знижка дорівнює 5%, а при покупці 3-х і більше товарів знижка складає 15%.

Зазвичай рішення цієї задачі можна записати, як набір команд:

```
SELECT Замовлення.Номер_замовл, Замовлення.Кількість,
CASE
    WHEN Замовлення.Кількість >= 3 THEN 15
    WHEN Замовлення.Кількість = 2 THEN 5
    ELSE 0
END AS Знижка,
Товар.Ціна, [Ціна]-[Знижка]*[Ціна]/100 AS Ціна_зі_знижкою
FROM Товар INNER JOIN Замовлення ON Товар.Код_товар = Замовлення.Код_тов;
```

Для СУБД MS Access рішення задачі буде виглядати наступним чином:

```
SELECT Замовлення.Номер_замовл, Замовлення.Кількість,
SWITCH (
    Замовлення.Кількість >= 3, 15,
    Замовлення.Кількість >= 2, 5,
    TRUE, 0
) AS Знижка,
Товар.Ціна, [Ціна]-[Знижка]*[Ціна]/100 AS Ціна_зі_знижкою
FROM Товар INNER JOIN Замовлення ON Товар.Код_товар = Замовлення.Код_тов;
```

9.2. Опублікувати у вигляді документів, придатних для комфортного перегляду та друку, результати виконання попередньо створених 10 запитів на ваш вибір. Звіти повинні включати такі елементи: таблиці, фотографії, графіки та діаграми.

Оскільки різні СУБД мають в своєму арсеналі засоби для виведення підготовлених даних у вигляді звітів, ознайомитись з технічною документацією до СУБД та оформити відібрані запити у вигляді звітів.

Так, СУБД MS Access має в своєму розпорядженні «Конструктор звітів» (рис. 9.1).



Рис 9.1. Приклад макету звіту в СУБД MS Access

СУБД Microsoft SQL Server має власну службу звітів SQL Server Reporting Services (рис. 9.2).



Рис 9.1. Приклад звіту SQL Server Reporting Services (СУБД Microsoft SQL Server)

Для виведення даних з СУБД MySQL можна скористатись інструментами від різних сторонніх розробників, наприклад, безкоштовними Microsoft Power BI, Google Data Studio або Apache Superset.

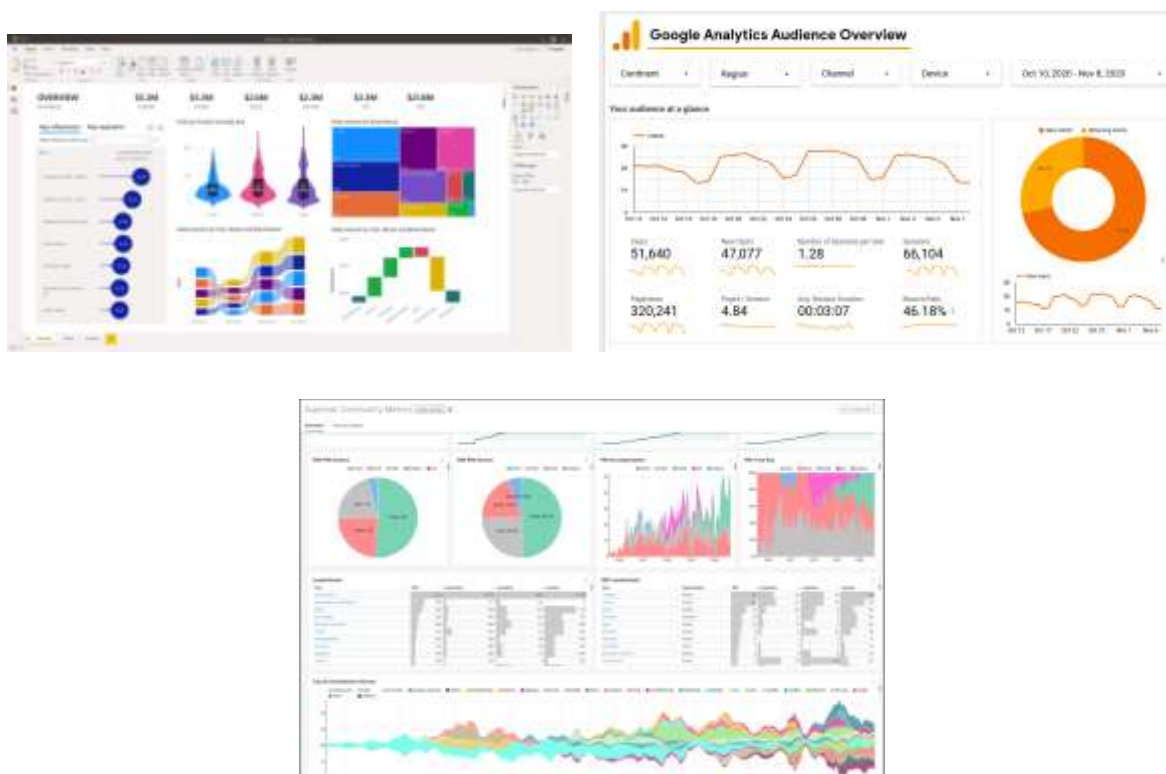


Рис 9.1. Звіти в програмах Microsoft Power BI, Google Data Studio та Apache Superset.

Контрольні питання

1. Наведіть загальну схему для написання умовного оператора в обраній вами СУБД.
2. Що повертає умовного оператора в обраній вами СУБД, якщо жодна з умов не буде виконана?
3. Як організувати обчислення та виведення розрахунків проміжних та загальних підсумків в обраному вами інструменті створення звітів?
4. Які формати використовує обраний вами інструмент створення звітів для збереження звітів?
5. Назвіть сфери використання звітів.

Рекомендована література

1. Access developer documentation. URL: <https://docs.microsoft.com/en-us/office/client-developer/access/access-home>
2. Верес О. М., Пасічник В. В., Берко А. Ю. Системи баз даних та знань. Книга 2: системи управління базами даних та знань. Київ: Магнолія, 2019. 584 с.
3. Створення простого звіту. URL: <https://support.microsoft.com/uk-ua/office/%D1%81%D1%82%D0%B2%D0%BE%D1%80%D0%B5%D0%BD%D0%BD%D1%8F-%D0%BF%D1%80%D0%BE%D1%81%D1%82%D0%BE%D0%B3%D0%BE-%D0%B7%D0%B2%D1%96%D1%82%D1%83-408e92a8-11a4-418d-a378-7f1d99c25304>
4. Змінення та редагування звіту. URL: <https://support.microsoft.com/uk-ua/office/%D0%B7%D0%BC%D1%96%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F-%D1%82%D0%B0-%D1%80%D0%B5%D0%B4%D0%B0%D0%B3%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F-%D0%B7%D0%B2%D1%96%D1%82%D1%83-2837aea7-0ee5-4c81-9f0f-5cf324c2c3bd>
5. Створення групового запиту. URL: <https://docs.microsoft.com/ru-ru/office/client-developer/access/desktop-database-reference/create-user-or-group-statement-microsoft-access-sql>

Лабораторна робота № 10

Тема: Команди модифікації даних

Мета: закріпити та поглибити знання студентів, які отримані ними на лекціях та під час самостійної роботи з питань використання команд модифікації даних.

Завдання: використовуючи оператор модифікації даних, створити запити за заданими умовами.

Час виконання завдання: 6 год.

План заняття

1. Виконання завдання згідно з інструкцією.
2. Захист роботи та відповіді на контрольні питання.

Рекомендації до виконання

10.1. Створити 2 запита на використання оператора INSERT INTO, додавши нові дані до двох різних таблиць.

Наприклад, додати себе, як співробітника компанії на посаду стажера.

```
INSERT INTO Співробітник (Прізвище, Ім_я, По_батькові, Тел, Паспорт, Ідент_код,  
                        Адр_місто, Адр_вул, Адр_буд, Адр_кв, Дата_народж,  
                        Дата_прийому, Посада, Права_водія)  
VALUES ("Горобець", "Сергій", "Миколайович", "тел. +38 077 555-55-55", "AB11001100",  
        121212121, "Житомир", "Бердичівська", "40", "1", "01.01.1980", "10.06.2000",  
        "стажер", TRUE);
```

10.2. Створити 2 запита із використанням оператора UPDATE, змінивши дані, які були додані в завданні 10.1.

Наприклад, змінити свою посаду зі стажера на директора.

```
UPDATE Співробітник  
SET Посада = "директор"  
WHERE Прізвище = "Горобець" AND Ім_я = "Сергій";
```

10.3. Написати запит із використанням оператора SELECT INTO, за допомогою якого створити архів (копію) однієї з таблиць.

Наприклад, скопіювати таблицю "Замовлення" в таблицю "Архів_Замовлень".

```
SELECT * INTO Архів_Замовлень  
FROM Замовлення;
```

10.4. Створити запит, за допомогою якого очистити вміст таблиці, створеної в завданні 10.3, використавши для цього оператор DELETE.

Наприклад, очистити вміст таблиці "Архів_Замовлень".

```
DELETE FROM Архів_Замовлень;
```

10.5. Створити запит, в якому за допомогою інструкції INSERT INTO SELECT повторно наповнити інформацією таблицю, очищену в завданні 10.4.

Наприклад, наповнити таблицю "Архів_Замовлень" інформацією з таблиці "Замовлення".

```
INSERT INTO Архів_Замовлень
(Номер_замовл, Дата_замовл, Код_співр, Код_клієнта, Код_тов,
Кількість, Дата_оплати, Дата_відправки, Відправлено)
SELECT Номер_замовл, Дата_замовл, Код_співр, Код_клієнта,
Код_тов, Кількість, Дата_оплати, Дата_відправки, Відправлено
FROM Замовлення;
```

10.6. Створити 2 запити із використанням команди DELETE для видалення записів із таблиці.

Наприклад, з таблиці "Співробітник" видалити запис про себе.

```
DELETE FROM Співробітник
WHERE Прізвище = "Горобець" AND Ім_я = "Сергій";
```

10.7. Створити 2 запити, в яких за допомогою інструкції DELETE видалити фіксовану кількість записів, що відповідають заданій умові. Умову вказати у підзапиті.

Наприклад, з таблиці "Архів_Замовлень" видалити 2 замовлення, що були зроблені клієнтами з Житомира.

```
DELETE FROM
(SELECT TOP 2 *
FROM Архів_Замовлень
WHERE Код_клієнта IN
(SELECT Код_клієнта FROM Клієнти
WHERE Адр_місто = 'Житомир')
);
```

10.8. Створити 2 запити на внесення змін до таблиці за допомогою команди UPDATE із використанням підзапиту та параметру.

Наприклад, зменшити на задану величину відсотка ціну всіх товарів, які жодного разу не були замовлені (відсутні в таблиці "Замовлення"). Величину відсотка задати як параметр, який вводить користувач.

```
UPDATE Товар SET Ціна = (Ціна*(1-["Вкажіть величину знижки"]/100))
WHERE Код_товар NOT IN
(SELECT Код_тов
FROM Замовлення);
```

10.9. Створити запит на видалення таблиці за допомогою команди DROP TABLE. Наприклад, видалити таблицю "Архів_Замовлень".

```
DROP TABLE Архів_Замовлень;
```

10.10. Створити дві таблиці за допомогою інструкції CREATE TABLE. Кожна таблиця має містити не менш, ніж 4 поля відповідного типу даних та мати первинний ключ.

Наприклад, створити таблицю "Постачальники" із полями "Код_постачальника", "Назва_організації", "Адр_місто", "Телефон", "Електронна_пошта". Код постачальника використовувати в якості первинного ключа.

```
CREATE TABLE Постачальники  
(Код_постачальника INTEGER NOT NULL PRIMARY KEY,  
Назва_організації TEXT(50) NOT NULL,  
Адр_місто TEXT(20) NOT NULL,  
Телефон TEXT(14) NOT NULL);
```

10.11. Створити 2 запити з використанням команди ALTER TABLE ADD, за допомогою яких додати до таблиць нову колонку.

Наприклад, додати до таблиці "Постачальники" стовпець із назвою "Електронна_пошта".

```
ALTER TABLE Постачальники  
ADD Електронна_пошта TEXT(30) NOT NULL;
```

10.12. Створити 2 запити із використанням команди ALTER TABLE DROP, за допомогою яких видалити деякі колонки із таблиці.

Наприклад, видалити колонку "Телефон" із таблиці "Постачальники".

```
ALTER TABLE Постачальники  
DROP COLUMN Електронна_пошта;
```

Контрольні питання

1. Поясніть різницю між операторами SELECT INTO та INSERT INTO SELECT.
2. Як видалити заповнену таблицю з бази даних?
3. В якій послідовності будуть виконані команди запиту, наведеному в прикладі

завдання 10.7?

4. Як за допомогою оператора INSERT внести дані лише в деякі задані стовпчики таблиці?

5. Як відреагує обрана вами СУБД на спробу додати дані в поле з автоінкрементом (поле типу "лічильник") за допомогою оператора INSERT?

Рекомендована література

1. Access developer documentation. URL: <https://docs.microsoft.com/en-us/office/client-developer/access/access-home>

2. Beaulieu A. Learning SQL. Second Edition. O'Reilly Media, Inc., 2009. 338 p. URL: <https://media.oaipdf.com/pdf/64103754-f42b-438c-99e3-59d6a294b2ff.pdf>

3. Верес О. М., Пасічник В. В., Берко А. Ю. Системи баз даних та знань. Книга 2: системи управління базами даних та знань. Київ: Магнолія, 2019. 584 с.

4. Ярцев В. П. Організація баз даних та знань: навч. посіб. Київ. ДУТ, 2018. 214 с.

Лабораторна робота № 11

Тема: Захист інформації в СУБД

Мета: закріпити та поглибити знання студентів, які отримані ними на лекціях та під час самостійної роботи з питань доступу та захисту даних, що зберігаються в базі даних.

Завдання: налаштувати права доступу та ролі користувачів.

Час виконання завдання: 4 год.

План заняття

1. Виконання завдання згідно з інструкцією.
2. Захист роботи та відповіді на контрольні питання.

Рекомендації до виконання

Перед початком виконання завдань лабораторної роботи обов'язково створіть резервну копію вашої бази даних на випадок помилкових дій із зміною прав користувачів.

11.1. В обраній вами СУБД створіть нового користувача і додайте його обліковий запис в групу адміністраторів. Встановіть для нього всі необхідні права для роботи з об'єктами вашої бази даних.

Наприклад, в СУБД Microsoft Access, починаючи з 2013 р., можна використовувати SQL команду CREATE USER з певними параметрами. В ранніх версіях Microsoft Access подібну процедуру можна виконати через меню «Робота с базами даних – Пользователи и разрешения – Пользователи и группы» (рис. 11.1.) або

через редагування файлів типу mdw (*\Application Data\Microsoft\Access\System.mdw).

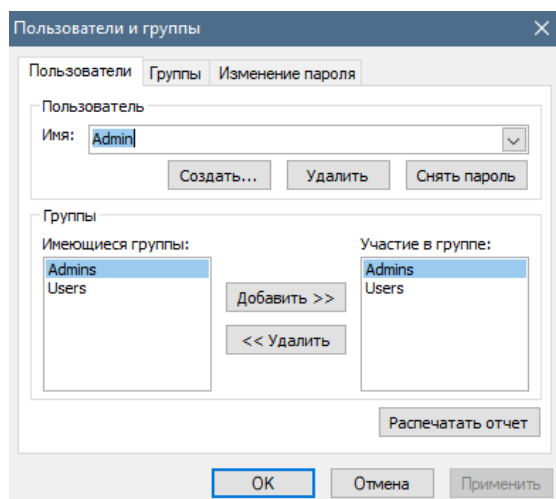


Рис. 11.1. Діалогове вікно для додавання нових користувачів (Microsoft Access 2007)

11.2. В обраній вами СУБД створіть нову групу користувачів. Встановіть для цієї групи окремі дозволи.

Наприклад, в СУБД Microsoft Access починаючи з 2013 р., можна використовувати SQL команду CREATE GROUP з певними параметрами. Врахуйте те, що назви груп та користувачів не можуть мати однакові імена.

11.3. В обраній вами СУБД створіть нового користувача і додайте його в групу користувачів, створену в завданні 11.2. Налаштуйте його дозволи на роботу з різними об'єктами бази даних.

Наприклад, в СУБД Microsoft Access ранніх версій можна використовувати діалогове вікно «Работа с базами данных – Пользователи и разрешения – Разрешения» (рис. 11.2.) або майстер «Мастер защиты на уровне пользователей...».

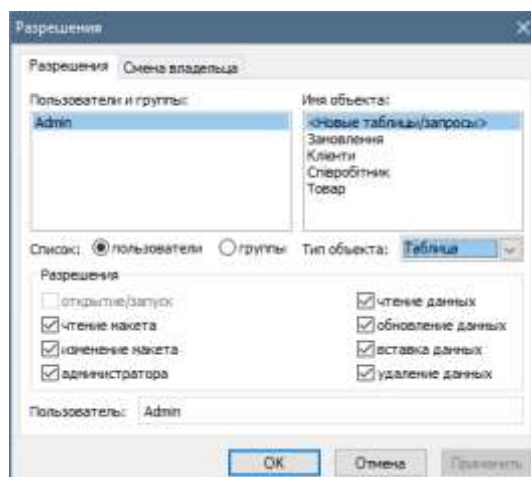


Рис. 11.2. Діалогове вікно для налаштування дозволів користувачів (Microsoft Access 2007)

11.4. Використовуючи SQL команду REVOKE, обмежте створений у завданні 11.3 запис користувача в правах на деякі операції.

11.5. Використовуючи SQL команду GRANT передайте користувачу, створеному у завданні 11.3 права на деякі операції.

Контрольні питання

1. Чому важливо розподіляти користувачів бази даних за ролями?
2. Які дозволи можна видати за допомогою команди GRANT?
3. Які права користувача можна обмежити за допомогою команди REVOKE?
4. Які шаблонні групи (ролі) користувачів пропонує обрана вами СУБД?
5. За допомогою яких команд можна видалити групу або користувача бази даних?

Рекомендована література

1. Access developer documentation. URL: <https://docs.microsoft.com/en-us/office/client-developer/access/access-home>
2. Гайна Г. А. Основи проектування баз даних. Київ: Кондор, 2018. 208 с.
3. Kelley K. Brian. Understanding GRANT, DENY, and REVOKE in SQL Server. URL: <https://www.mssqltips.com/sqlservertip/2894/understanding-grant-deny-and-revoke-in-sql-server/>