

Горобець С. М.

ІНСТРУКТИВНО-МЕТОДИЧНІ МАТЕРІАЛИ
ДО ЛАБОРАТОРНИХ ЗАНЯТЬ
ОСВІТНЬОЇ КОМПОНЕНТИ

«БАЗИ ДАНИХ»

для підготовки здобувачів першого
(бакалаврського) рівня вищої освіти

Видання друге, перероблене та доповнене

Житомир, 2026

*Рекомендовано до друку вченою радою
Житомирського державного університету імені Івана Франка
(протокол №12 від 26.06.2026 р.)*

Рецензенти:

БРОДСЬКИЙ Юрій – кандидат технічних наук, доцент, доцент кафедри комп'ютерної інженерії та кібербезпеки Державного університету «Житомирська політехніка».

ТОПОЛЬНИЦЬКИЙ Павло – 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 освітньої компоненти «Бази даних» для підготовки здобувачів першого (бакалаврського) рівня вищої освіти. Видання друге, перероблене та доповнене. Житомир: Вид-во ЖДУ імені Івана Франка, 2026. 65 с.

© Горобець С. М., 2026

© Житомирський державний університет
імені Івана Франка, 2026

ЗМІСТ

Критерії оцінювання занять		4
Лабораторна робота № 1	Формулювання завдання на проектування БД. Концептуальне (інфологічне) моделювання бази даних	5
Лабораторна робота № 2	Проектування реляційних БД методом нормалізації відношень	13
Лабораторна робота № 3	Основні та похідні операції реляційної алгебри	16
Лабораторна робота № 4	Налаштування СУБД та створення бази даних. Загальна структура мови SQL	18
Лабораторна робота № 5	Використання мови SQL для відображення інформації	31
Лабораторна робота № 6	Агрегатні функції	34
Лабораторна робота № 7	З'єднання таблиць та використання групових операцій	36
Лабораторна робота № 8	Використання підзапитів мовою SQL	39
Лабораторна робота № 9	Проектування звітності	42
Лабораторна робота № 10	Команди модифікації даних	46
Лабораторна робота № 11	Проектування інтерфейсу користувача (головної кнопкової форми)	49
Лабораторна робота № 12	Технології доступу та захист інформації в СУБД. Розподілені БД	51
Лабораторна робота № 13	Програмний доступ до бази даних засобами Python	54
Лабораторна робота № 14	Реалізація транзакційних операцій у базі даних засобами Python	61

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.

Лабораторна робота № 1

Тема: Формулювання завдання на проектування БД. Концептуальне (інфологічне) моделювання бази даних

Мета: закріпити основні поняття дисципліни «Бази даних» та поглибити теоретичні знання щодо розробки концептуальної моделі бази даних.

Завдання: обрати предметну область, розробити концептуальну (інфологічну) модель бази даних, побудувати ER-діаграму.

Час виконання завдання: 4 год.

План роботи

1. Виконати завдання згідно з методичними вказівками.
2. Відповісти на контрольні питання.
3. Оформити звіт та захистити лабораторну роботу.

Завдання до виконання

1.1. Перший етап проектування бази даних. Вибір предметної області.

Розвиток комп'ютерних технологій, пов'язаних із зберіганням та обробкою даних, призвели до появи в 60-х рр. ХХ ст. поняття «база даних» (БД) та спеціалізованого програмного забезпечення, яке отримало назву «системи управління базами даних» (СУБД).

База даних – це структурована сукупність логічно пов'язаних даних, що відображують стан об'єктів певної предметної області та зв'язки між ними, зберігаються та використовуються спільно.

Структурування даних – це введення правил про способи представлення даних, яке дозволяє впорядкувати інформацію за різними ознаками та швидко отримувати дані з будь-яким поєднанням таких ознак.

Предметна область – це частина реального світу, яка вивчається з метою організації управління з перспективою подальшої автоматизації. При цьому база даних містить відомості лише про ту частину предметної області, яка важлива для користувачів даної БД. Отже, предметною областю називають сферу застосування конкретної бази даних. Приклади різних предметних областей наведено в табл. 1.1.

Об'єктами предметної області можуть бути підприємства, книжки, квитки, студенти, товари і т.п. Об'єкт має певний набір параметрів, кожен з яких має значення. Між деякими об'єктами існують зв'язки.

Бази даних є потужним і зручним інструментом автоматизованої обробки великих масивів інформації. Саме тому вони широко використовуються в різних сферах, наприклад, у виробничій, науково-дослідній, освітній, банківській, страховій діяльності, торгівлі, логістиці, охороні здоров'я, державному управлінні та інших галузях.

Системи управління базами даних – це комплекс мовних та програмних засобів, призначених для створення, ведення, захисту та обробки баз даних.

Прикладне програмне забезпечення баз даних – це програми та системи, що розробляються і створюються компаніями-розробниками програмного забезпечення або самими користувачами для вирішення конкретних прикладних завдань.

Для проектування своєї бази даних спочатку необхідно визначитись із предметною областю. Варіанти завдань обираються з табл. 1.1 згідно з номером студента в журналі академічної групи.

Таблиця 1.1.

Варіанти завдань для проектування бази даних

Варіант	Тема
1.	База даних бібліотеки
2.	База даних страхової компанії
3.	База даних служби таксі
4.	База даних готелю
5.	База даних ресторану
6.	База даних лікувального закладу
7.	База даних залізничного вокзалу
8.	База даних аеропорту
9.	База даних будівельної організації
10.	База даних туристичної компанії
11.	База даних рекламної агенції
12.	База даних абонентів Інтернет-провайдера
13.	База даних відділу кадрів організації
14.	База даних транспортного підприємства
15.	База даних для ведення земельного кадастру
16.	База даних приймальної комісії закладу вищої освіти
17.	База даних складського обліку організації
18.	База даних агентства нерухомості
19.	База даних для підтримки екзаменаційної сесії в університеті
20.	База даних служби кур'єрської доставки
21.	База даних замовлень ремонтних робіт будівель
22.	База даних обліку абонентів у мережі мобільного оператора
23.	База даних клієнтів банку
24.	База даних організації з обслуговування житлово-комунального господарства
25.	База даних організації з ремонту побутової техніки
26.	База даних для пошуку та бронювання залізничних квитків
27.	База даних обліку лікарських засобів аптеки
28.	База даних сервісного центру МВС для реєстрації транспортних засобів
29.	База даних обліку збуту продукції сільськогосподарського підприємства
30.	База даних підприємства з ремонту комп'ютерної техніки

1.2. Концептуальне (інфологічне) проектування БД.

Після вибору та аналізу предметної області необхідно створити загальний опис своєї бази даних, використовуючи звичайну мову, математичні формули, графіки, таблиці та інші засоби. Такий опис називають концептуальною (або інфологічною) моделлю даних. Вона відображує предметну область у вигляді сукупності інформаційних об'єктів та їх структурних зв'язків.

При складанні концептуальної моделі потрібно визначити сутності предметної області, атрибути цих сутностей, встановити зв'язки між ними та зазначити їх тип.

Сутність – це множина об’єктів, інформація про які буде зберігатися в базі даних. Вони мають суттєве значення для даної предметної області та однаковий набір параметрів. Кожна сутність повинна мати наступні властивості: унікальний ідентифікатор; один чи декілька атрибутів, які або належать цій сутності, або наслідуються через зв’язок з іншими сутностями; мати сукупність атрибутів, які однозначно ідентифікують кожний екземпляр сутності.

Атрибут – це поіменована характеристика сутності. Він повинен мати чітке смислове значення та називатися в однині. Кожен атрибут зберігає інформацію про певну властивість сутності й повинен бути унікальним.

Зв’язок – це з’єднання двох чи більше сутностей, при якому кожен екземпляр першої сутності пов’язаний з довільною кількістю екземплярів другої сутності, а кожен екземпляр другої сутності пов’язаний лише з одним екземпляром першої сутності. Зв’язок є логічним відношенням між сутностями і називається дієсловом.

При аналізі сутностей бажано визначитись з можливими обмеженнями, що накладаються на зв’язки між сутностями та на діапазони допустимих значень, що будуть зберігатися у базі даних.

Для представлення концептуальної моделі даних найбільш широко використовують графічну ER-діаграму – модель «сутність-зв’язок», яка наочно представляє зв’язок між сутностями (рис. 1.1). Таку діаграму необхідно навести у звіті про виконання лабораторної роботи.

Зазвичай сутність та її атрибути позначаються іменниками, а зв’язки між сутностями позначаються дієсловами.

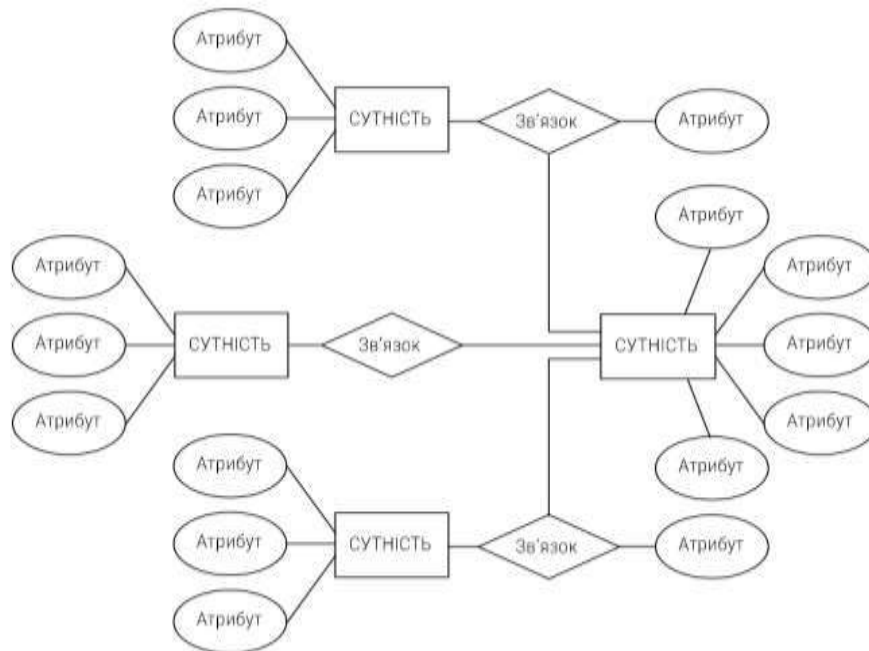


Рис. 1.1 – Приклад оформлення ER-діаграми (сутність-зв'язок)

Для спрощення процесу підготовки моделей сутність-зв'язок та автоматизації їх створення зручно використовувати програми чи он-лайн сервіси типу Diagrams.net, Lucidchart.com, GitMind, Gliffy, Visual Paradigm, SqlDBM, DBDiagram.io, QuickDBD, Creately, ERDPlus та ін.

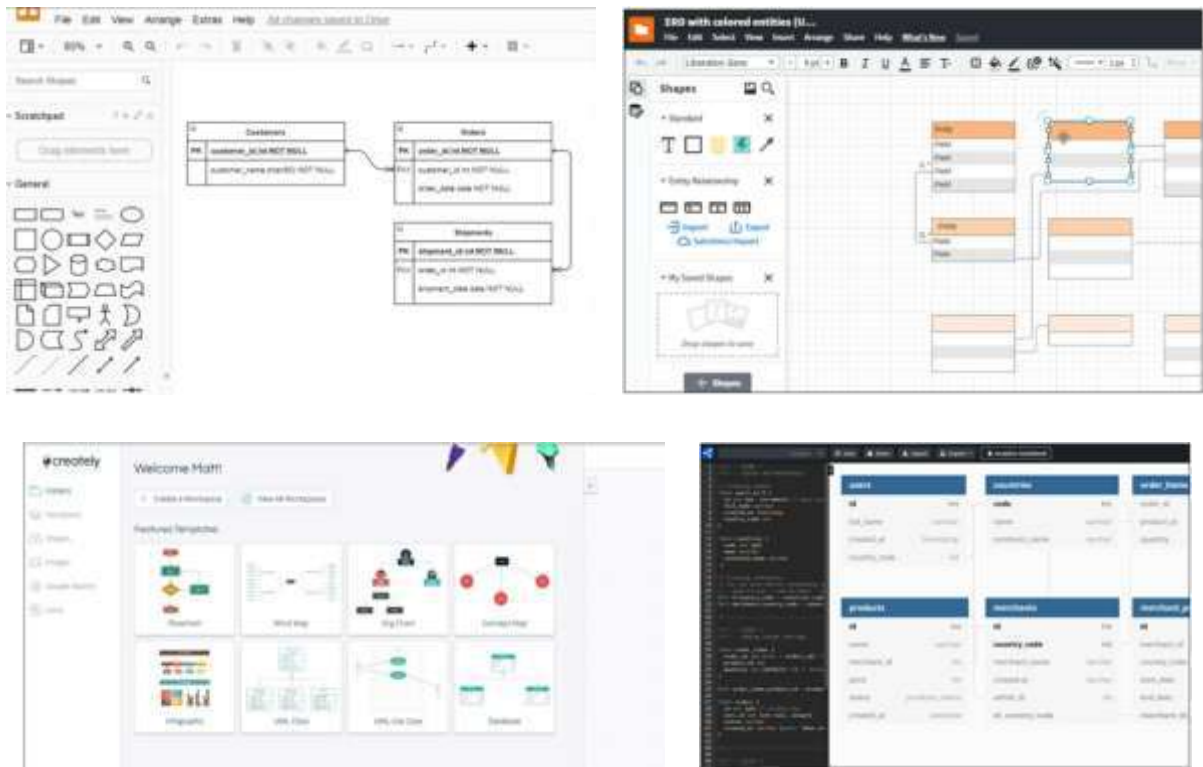


Рис. 1.2 – Приклади оформлення ER-діаграм за допомогою он-лайн сервісів Diagrams.net, Lucidchart.com, Creately.com та DBDiagram.io

1.3. Приклад побудови ER-діаграми для заданої предметної області

В якості предметної області оберемо компанію, що займається морськими вантажними перевезеннями.

Діяльність компанії організована наступним чином – вона виконує перевезення товарів за попередніми замовленнями клієнтів.

Клієнт (замовник) може зробити необмежену кількість замовлень, але при цьому кожне замовлення повинно містити хоча б один товар. Одне замовлення може включати в себе продукти різного типу. Також для кожного замовлення обов’язково фіксується дата його створення.

В описі замовлення на перевезення зазначено товари та їх кількість.

Для транспортування товарів компанія використовує кораблі з приписаною до них командою. Команда складається з робітників організації. Кожен робітник займає певну посаду та отримує визначену заробітну плату.

Перевезення здійснюються за маршрутами, які мають власну назву, порт відправлення та порт прибуття.

Клієнти (замовники) мають змогу відстежувати рух замовлених товарів за допомогою перегляду статусу доставки.

Для прикладу, в якості системи управління базою даних оберемо Microsoft SQL Server. Для написання керуючих скриптів будемо використовувати діалект мови SQL Transact-SQL.

Визначимо перелік основних сутностей:

- 1) Customers;
- 2) Orders;

- 3) Orders_Statutes;
- 4) Orders_Details;
- 5) Products;
- 6) Routes;
- 7) Ports;
- 8) Ships;
- 9) Ships_Types;
- 10) Crews;
- 11) Employees;
- 12) Positions.

Сутність Customers має зв'язок типу «один до багатьох» із сутністю Orders, оскільки декілька замовників можуть мати декілька замовлень.

Сутність Orders_Statutes має зв'язок типу «один до багатьох» із сутністю Orders, оскільки у багатьох замовлень може бути однаковий статус.

Сутність Orders пов'язана із сутністю Orders_Details зв'язком «один до багатьох», оскільки одне замовлення може міститися у багатьох деталях замовлень.

Сутність Products пов'язана із сутністю Orders_Details зв'язком «один до багатьох», оскільки один продукт може міститися у багатьох деталях замовлень.

Сутність Routes має зв'язок типу «один до багатьох» із сутністю Orders, оскільки по одному маршруту можуть відправлятися багато замовлень.

Сутність Ports має зв'язок типу «один до багатьох» із сутністю Routes, оскільки одні й ті самі порти можуть бути вказані в багатьох маршрутах.

Сутність Ships має зв'язок типу «один до багатьох» із сутністю Orders, оскільки одні й ті самі кораблі можуть бути вказані в багатьох замовленнях.

Сутність Ships_Types має зв'язок типу «один до багатьох» із сутністю Ships, оскільки в багатьох кораблях може бути однаковий тип.

Сутність Crews має зв'язок типу «один до багатьох» із сутністю Ships, оскільки одна команда може бути закріплена за декількома кораблями.

Сутність Crews має зв'язок типу «один до багатьох» із сутністю Employees, оскільки в одній команді може міститися багато робітників.

Сутність Positions має зв'язок типу «один до багатьох» із сутністю Employees, оскільки більше одного працівника можуть займати одну й ту саму посаду.

Перелік сутностей та їх атрибутів наведено у таблиці 2.1.

Таблиця 2.1.

Назва сутності	Опис сутності	Назва атрибуту
Customers	Замовники, які замовляють товари	id first_name second_name birth_date address phone_number email

Продовження таблиці 2.1

Orders	Замовлення	id customer_id route_id ship_id status_id shipment_price created_at description
Orders_States	Статуси замовлень	id title description
Orders_Details	Товар, замовлення та кількість товарів	id order_id product_id quantity
Products	Товари, які є на зберіганні	id name description price unit
Routes	Маршрути, за якими рухаються кораблі	id start_port_id destination_port_id name
Ports	Порти, в які виконується доставка	id name location capacity
Ships	Кораблі, на яких здійснюється доставка	id crew_id ship_type_id name displacement last_maintenance_date
Ships_Types	Типи кораблів	id type description
Crews	Екіпажи кораблів	id name creation_date
Employees	Працівники компанії	id crew_id position_id first_name second_name birth_date address phone_number email hire_date salary
Positions	Посади, які можуть займати працівники	id title description

За вихідними даними побудуємо ER-діаграму (рис. 1.3).

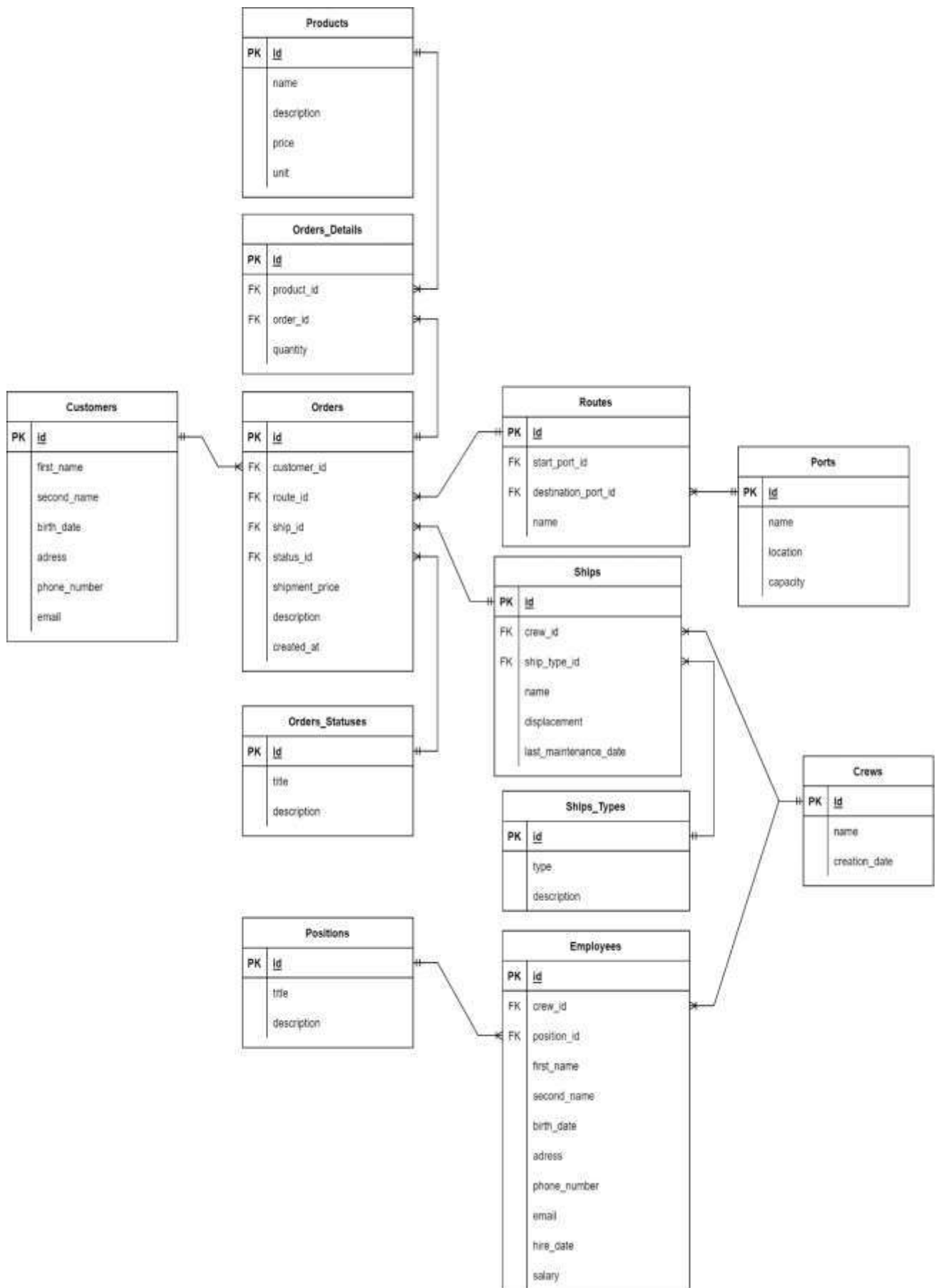


Рис. 1.3 – ER-діаграма

Контрольні питання

1. Дайте визначення поняттям: база даних, предметна область, СУБД, сутність, атрибут, зв'язок, концептуальна модель бази даних.
2. Надайте коротку характеристику основним етапам проектування БД.
3. Яку інформацію містить в собі інфологічна модель БД та які існують способи її представлення?
4. Назвіть основні типи зв'язків між сутностями та наведіть їх графічне позначення.
5. Наведіть приклади умовних позначень елементів концептуальної моделі БД в найбільш популярних нотаціях.

Рекомендована література

1. Босько В.В., Константинова Л.В., Поліщук Л.І., Коноплицька-Слободенюк О.К. Бази даних: навч. посібн. Кропивницький: ЦНТУ, 2024. 226 с.
2. Мулеса О., Варга Я. Інформаційні системи та реляційні бази даних: навч. посібник. Ужгород, 2023. 132 с.
3. Верес О. М., Пасічник В. В. Системи баз даних та знань. Книга 1: організація баз даних та знань. Київ: Магнолія, 2019. 440 с.
4. Ярцев В. П. Організація баз даних та знань: навч. посіб. Київ: ДУТ, 2018. 214 с.

Лабораторна робота № 2

Тема: Проектування реляційних БД методом нормалізації відношень

Мета: закріпити та поглибити теоретичні знання щодо розробки даталогічної моделі бази даних.

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

Час виконання завдання: 4 год.

План роботи

1. Виконати завдання згідно з методичними вказівками.
2. Відповісти на контрольні питання.
3. Оформити звіт та захистити лабораторну роботу.

Завдання 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 виділити 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).

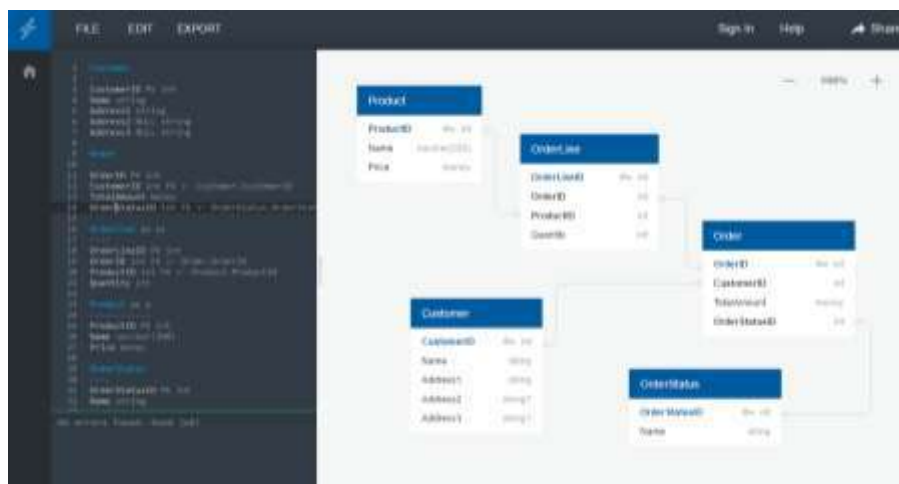


Рис. 2.1 – Приклад оформлення схеми БД за допомогою он-лайн сервісу app.quickdatabasediagrams.com

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 визначення необхідних відношень, первинних та зовнішніх ключів, обмежень цілісності для спроектованих відношень.

Для майбутньої бази даних морських перевезень проведемо нормалізацію таблиць мінімум до третьої нормальної форми. Для всіх полів таблиць проведемо декомпозицію та спрощення. Для кожного неключового атрибуту визначимо нетранзитивну залежність від первинного ключа. Кінцевий результат перетворень представлено на рис. 2.2.

Контрольні питання

1. Яка мета розробки даталогічної моделі бази даних?
2. Для чого здійснюють нормалізацію відношень в моделі бази даних?
3. Які переваги та недоліки представлення даних у вигляді реляційної моделі БД?
4. Опишіть поняття реляційного ключа та охарактеризуйте його різновиди.
5. Дайте визначення поняттям «кортеж», «домен», «відношення».

Рекомендована література

1. Босько В.В., Константинова Л.В., Поліщук Л.І., Коноплицька-Слободенюк О.К. Бази даних: навч. посібн. Кропивницький: ЦНТУ, 2024. 226 с.
2. Мулеса О., Варга Я. Інформаційні системи та реляційні бази даних: навч. посібник. Ужгород, 2023. 132 с.
3. Верес О. М., Пасічник В. В. Системи баз даних та знань. Книга 1: організація баз даних та знань. Київ: Магнолія, 2019. 440 с.
4. Демиденко М. А. Введення в сучасні бази даних: навч. посіб. НТУ «Дніпровська політехніка». Дніпро, 2020. 38 с.

Лабораторна робота № 3

Тема: Основні та похідні операції реляційної алгебри

Мета: закріпити та поглибити теоретичні знання щодо використання основних та похідних операцій реляційної алгебри.

Завдання: виконати операції реляційної алгебри над множинами даних.

Час виконання завдання: 4 год.

План роботи

1. Виконати завдання згідно з методичними вказівками.
2. Відповісти на контрольні питання.
3. Оформити звіт та захистити лабораторну роботу.

Завдання до виконання

Розглянути основні операції реляційної алгебри та виконати наведені нижче завдання.

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

3.2. Навести приклад використання операції вибірки із застосуванням констант, виразів ($=$, $>$, $<$, $<=$, $>=$) та логічних операторів (AND, OR, NOT) для таблиці свого варіанту бази даних.

3.3. Навести приклад використання операції декартового добутку відношень для своєї бази даних. Зверніть увагу на достовірність отриманих результатів. У звіті зазначте чи всі отримані результати цієї операції відповідають об'єктивній реальності.

3.4. Навести 2 приклади 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).

3.5. Навести 2 приклади використання операції різниці відношень для своєї бази даних. При проведенні операцій зверніть увагу на вимогу сумісності відношень (необхідна попередня перевірка на співпадіння схем відношень).

На підставі п'яти основних операцій можна отримати додаткові похідні операції – природне з'єднання, перетин та поділ. Розглянути похідні операції реляційної алгебри та виконати наведені нижче завдання.

3.6. Навести приклад 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).

3.7. Навести приклад використання операції ділення відношень для своєї бази даних.

3.8. Навести 2 приклади використання операції природного з'єднання відношень для своєї бази даних.

Контрольні питання

1. Які операції з множинами дозволяє виконати реляційна алгебра?
2. Дайте визначення операціям перейменування, проекції вибірки та наведіть приклади.
3. Дайте визначення операціям декартового добутку об'єднання, різниці, та перетину відношень та наведіть приклади.
4. Яке призначення операцій з'єднання відношень?
5. Опишіть різницю між такими операціями з'єднання, як природне з'єднання, напівз'єднання, тета-з'єднання.

Рекомендована література

1. Булатецький В. В., Булатецька Л. В. 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». ВНУ ім. Лесі Українки. Луцьк: ВНУ ім. Лесі Українки, 2020. 36 с.

2. Мулеса О., Варга Я. Інформаційні системи та реляційні бази даних: навч. посібник. Ужгород, 2023. 132 с.

3. Проектування реляційних баз даних: навч. посібн. в електр. вигляді / В. О. Нелюбов, Ю. Ю. Білак. А. М. Реблян. Ужгород: ДВНЗ «УжНУ», 2023. 60 с. URL: <https://dspace.uzhnu.edu.ua/server/api/core/bitstreams/c66b090b-b87e-41e5-b4ac-e3a5a58da81c/content>

4. Робота з базами даних: навч. посіб. для здобувачів ступеня бакалавра за освіт. програмою «Інформаційно-комунікаційні технології» спец. 172 Електронні комунікації та радіотехніка / Л.С. Глоба, С.В. Суліма, М.А. Скулиш; КПІ ім. Ігоря Сікорського. 2-е вид. Київ: КПІ ім. Ігоря Сікорського, 2024. 532 с. URL: <https://ela.kpi.ua/bitstreams/ae45b0ef-ff71-4d39-90e2-930756c3d130/download>

Лабораторна робота № 4

Тема: Налаштування СУБД та створення бази даних.

Загальна структура мови SQL

Мета: закріпити та поглибити знання студентів, щодо налаштування СУБД та створення бази даних.

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

Час виконання завдання: 6 год.

План роботи

1. Виконати завдання згідно з методичними вказівками.
2. Відповісти на контрольні питання.
3. Оформити звіт та захистити лабораторну роботу.

Завдання до виконання

4.1. Для виконання завдання даної роботи та всіх наступних лабораторних робіт необхідно обрати одну з сучасних СУБД, яка дозволить створити базу даних та керувати нею.

Прикладом таких популярних сучасних СУБД є MySQL, PostgreSQL, Microsoft Access, Microsoft SQL Server, MariaDB.

Однією з найбільш поширених безкоштовних СУБД є розроблений компанією Microsoft сервер Microsoft SQL Server. Основною утилітою для керування цією СУБД є SQL Server Management Studio (SSMS), що використовується для адміністрування та конфігурування всіх компонентів Microsoft SQL Server (рис. 4.1).

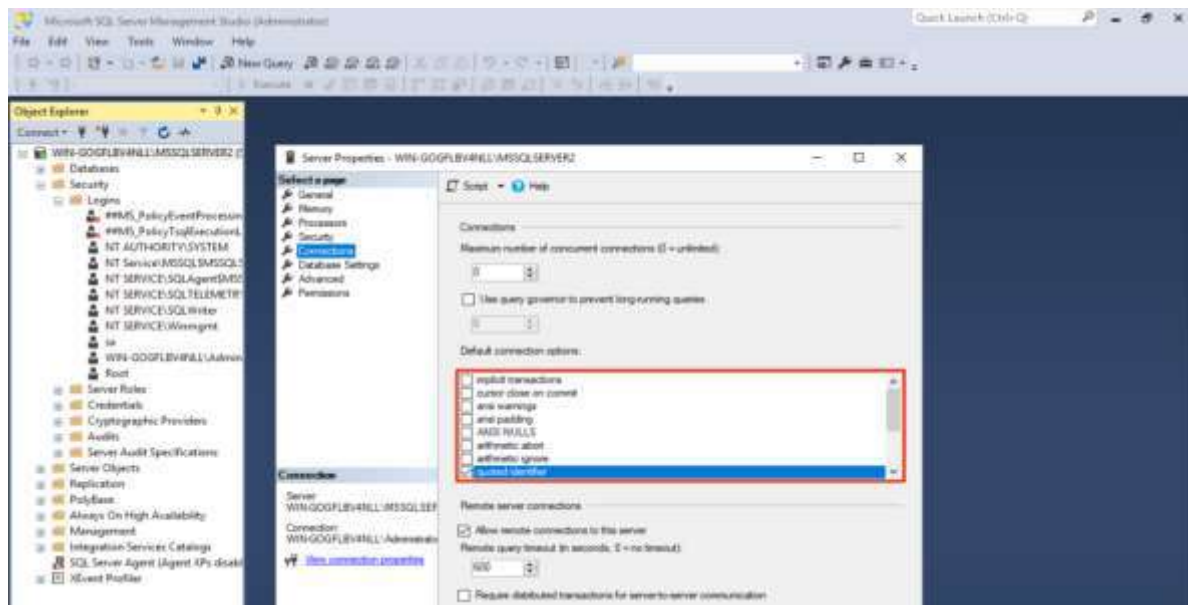


Рис. 4.1 – Інтерфейс Microsoft SQL Server Management Studio

Дана програма дозволяє не тільки переглядати, а й повністю керувати всіма

об'єктами сервера. Утиліта має редактор скриптів та графічну оболонку, яка працює з об'єктами бази даних та всіма налаштуваннями SQL сервера. Для версії сервера Express існує безкоштовний додаток SQL Server Management Studio Express.

Система управління базами даних Microsoft Office Access (рис. 4.2) входить до складу дуже популярного пакету Microsoft Office. Дана СУБД має значний набір функціональних можливостей, які дозволяють не тільки створити власну базу даних, керувати нею, а й зв'язати із зовнішніми додатками.

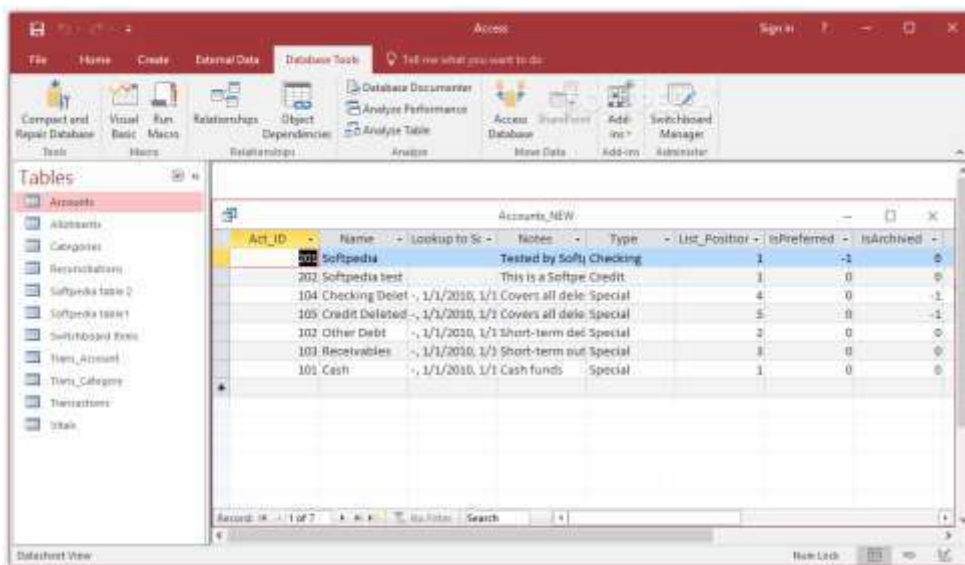


Рис. 4.2 – Інтерфейс Microsoft Office Access

Окрім підтримки мови SQL, дана СУБД дозволяє використовувати програми на мові VBA. Завдяки такій інтеграції адміністратор БД має додаткові можливості для керування вмістом бази даних та роботою з користувачами.

Вільно поширювана реляційна СУБД MySQL на сьогодні є одним із лідерів на ринку баз даних для веб-орієнтованих додатків. Особливістю СУБД MySQL є те, що вона підтримує різні типи таблиць. Завдяки цьому при роботі з таблицями є можливість реалізувати як підтримку повнотекстового пошуку, так і підтримку транзакцій тільки на рівні окремих записів.

Найчастіше MySQL використовується як сервер, до якого звертаються мережеві клієнти. Але до дистрибутиву інтегрована бібліотека внутрішнього сервера, встановлення якого дозволяє додавати сервер MySQL в будь-які автономні програми. Для зручного керування сервером MySQL можна використати програмний інструмент MySQL Workbench (рис. 4.3).

MySQL Workbench є вільно поширюваним додатком і дозволяє значно спростити процес проектування та використання БД у середовищі MySQL.

СУБД PostgreSQL має особливість – вона є об'єктно-реляційною. Дана система вміє працювати з бітовими послідовностями, мережевими адресами, багатовимірними масивами даних та іншими складними інформаційними структурами. PostgreSQL підтримує роботу з JSON, XML та з базами NoSQL.

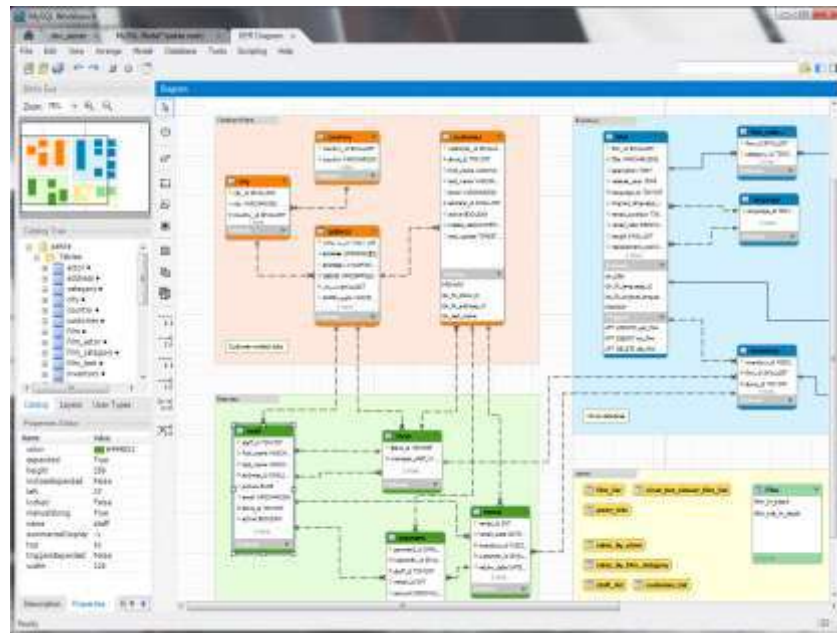


Рис. 4.3 – Інтерфейс MySQL Workbench

До найбільш зручних графічних інтерфейсів для СУБД PostgreSQL відносять TablePlus, DBeaver, pgAdmin, DataGrip, Navicat, Beekeeper Studio, HeidiSQL та інші (рис. 4.4).

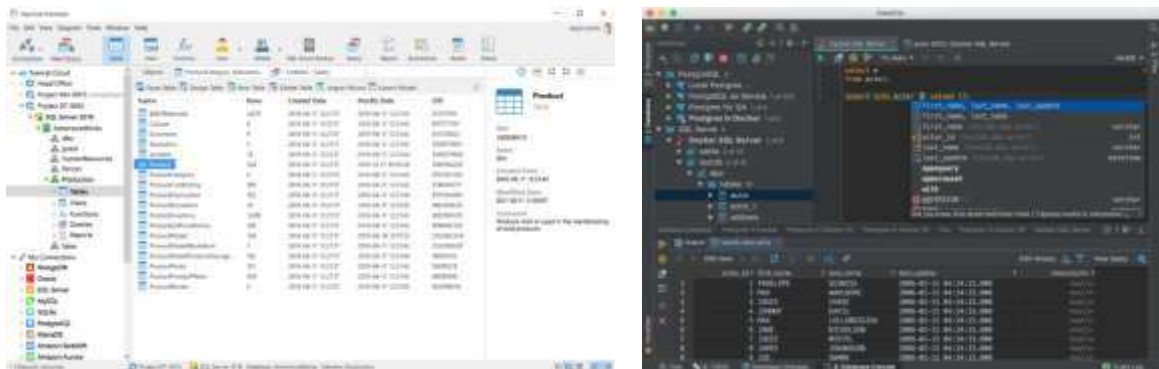


Рис. 4.4 – Інтерфейси DBeaver та DataGrip

4.2. Проінсталивати обрану систему управління базами даних та налаштувати її роботу.

4.3. Використовуючи мову SQL написати послідовність команд, яка дозволить створити об'єкти вашої бази даних та налаштувати зв'язки між ними.

Оскільки різні СУБД використовують свої діалекти мови SQL, то синтаксис команд треба уточнювати у довідкових матеріалах розробників обраної СУБД.

Деякі СУБД, наприклад, Microsoft Office Access дозволяють спростити розробку структури БД через використання графічного інтерфейсу та режиму «Конструктор».

4.4. Розглянемо приклад створення об'єкти бази даних морських перевезень та налаштуємо зв'язки між цими об'єктами.

Додатково визначимо умови на проектування БД морських перевезень:

- 1) Усі робітники та замовники повинні мати унікальні номер телефону та електронну пошту.
- 2) Видалити можна лише записи, на які немає посилань у інших таблицях.
- 3) При видаленні покупця з бази даних видаляються також усі його замовлення.
- 4) Заборонено додавати маршрути, в яких порт відправки співпадає з пунктом призначення.
- 5) Усі маршрути повинні мати унікальні пари – пункти відправки та призначення.
- 6) Кількість продуктів у замовленні повинна бути додатньою.
- 7) Атрибути, які обов'язково необхідно заповнити, мають обмеження NOT NULL.

Інструкції мовою Transact-SQL можуть мати наступний вигляд:

```
USE master  
DROP DATABASE IF EXISTS SeaShippingCompany
```

```
CREATE DATABASE SeaShippingCompany  
GO
```

```
USE SeaShippingCompany
```

```
CREATE TABLE Customers(  
    id INT IDENTITY(1, 1),  
    first_name nvarchar(50) NOT NULL,  
    second_name nvarchar(50) NOT NULL,  
    birth_date date NOT NULL,  
    [address] nvarchar(250) NOT NULL,  
    phone_number nvarchar(20) NOT NULL UNIQUE,  
    email nvarchar(50) NOT NULL UNIQUE,  
  
    CONSTRAINT PK_Customers_id PRIMARY KEY(id)  
);
```

```
CREATE TABLE Products(  
    id INT IDENTITY(1, 1),  
    name nvarchar(50) NOT NULL,  
    description nvarchar(500) NOT NULL,  
    price money NOT NULL,  
    unit int NOT NULL,  
  
    CONSTRAINT PK_Product_id PRIMARY KEY(id)  
)
```

```
CREATE TABLE [Ports](  
    id INT IDENTITY(1, 1),  
    name nvarchar(50) NOT NULL,  
    location nvarchar(100) NOT NULL,  
    capacity real NOT NULL,  
  
    CONSTRAINT PK_Port_id PRIMARY KEY(id)
```

```

)

CREATE TABLE Routes(
    id INT PRIMARY KEY IDENTITY(1, 1),
    start_port_id INT FOREIGN KEY REFERENCES Ports(id) NOT NULL,
    destination_port_id INT FOREIGN KEY REFERENCES Ports(id) NOT NULL,
    name nvarchar(100) NOT NULL

    CONSTRAINT UQ_Routes_constr UNIQUE (start_port_id,destination_port_id),
    CONSTRAINT UQ_Routes_strt CHECK (start_port_id!=destination_port_id)

)

CREATE TABLE Positions(
    id INT PRIMARY KEY IDENTITY(1, 1),
    title nvarchar(50) NOT NULL UNIQUE,
    description nvarchar(450) NOT NULL
)

CREATE TABLE Crews(
    id INT PRIMARY KEY IDENTITY(1, 1),
    name nvarchar(50) NOT NULL,
    creation_date DATE NOT NULL
)

CREATE TABLE Employees(
    id INT IDENTITY(1, 1),
    crew_id INT FOREIGN KEY REFERENCES Crews(id),
    position INT FOREIGN KEY REFERENCES Positions(id),
    first_name nvarchar(50) NOT NULL,
    second_name nvarchar(50) NOT NULL,
    birth_date date NOT NULL,
    [address] nvarchar(250) NOT NULL,
    phone_number nvarchar(20) NOT NULL UNIQUE,
    email nvarchar(50) NOT NULL UNIQUE,
    hire_date DATE NOT NULL,
    salary money NOT NULL

    CONSTRAINT PK_Employees_id PRIMARY KEY(id)
)

CREATE TABLE Ships_Types(
    id INT PRIMARY KEY IDENTITY(1, 1),
    type nvarchar(50) NOT NULL,
    description nvarchar(500) NOT NULL
)

CREATE TABLE Ships(
    id INT IDENTITY(1, 1),
    crew_id INT FOREIGN KEY REFERENCES Crews(id) NOT NULL,
    ship_type_id INT FOREIGN KEY REFERENCES Ships_Types(id) NOT NULL,
    name nvarchar(30) NOT NULL,

```

```

displacement real NOT NULL,
last_maintenance_date DATE NOT NULL,

CONSTRAINT PK_Ships_id PRIMARY KEY(id)
)

CREATE TABLE Orders_Statuses(
id INT PRIMARY KEY IDENTITY(1, 1),
title nvarchar(40) NOT NULL,
description nvarchar(200) NOT NULL
)

CREATE TABLE Orders(
id INT PRIMARY KEY IDENTITY(1, 1),
customer_id INT FOREIGN KEY REFERENCES Customers(id) ON DELETE
CASCADE NOT NULL,
route_id INT FOREIGN KEY REFERENCES Routes(id) NOT NULL,
ship_id INT FOREIGN KEY REFERENCES Ships(id) NOT NULL,
status_id INT FOREIGN KEY REFERENCES Orders_Statuses(id) NOT NULL,
shipment_price money NOT NULL,
description nvarchar(200) NOT NULL,
created_at datetime
)

CREATE TABLE Orders_Details(
id int PRIMARY KEY IDENTITY(1, 1),
order_id int FOREIGN KEY REFERENCES Orders(id) ON DELETE CASCADE
NOT NULL,
product_id int FOREIGN KEY REFERENCES Products(id) NOT NULL,
quantity int NOT NULL
)

```

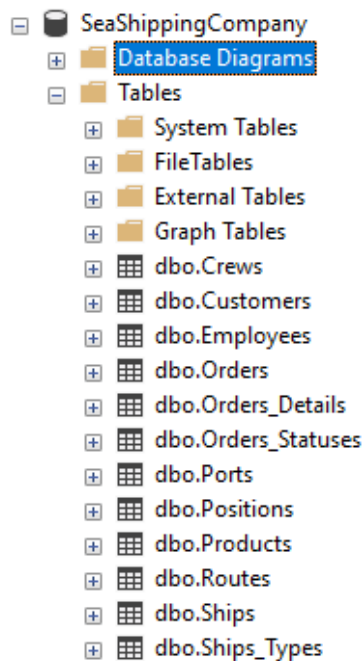


Рис. 4.5 – Результат виконання SQL-коду в середовищі Microsoft SQL Server Management Studio

4.5. Додавання тригерів

Створити тригер для запису в таблицю БД значення поточної дати та часу внесення запису.

Наприклад, створимо тригер для таблиці Orders, який буде спрацьовувати після виконання команди INSERT та встановлювати у поле “created_at” значення поточної дати та часу.

```
GO
CREATE OR ALTER TRIGGER InsertIntoOrdersTrigger
ON Orders
AFTER INSERT
AS
BEGIN
    UPDATE Orders
    SET created_at = GETDATE()
    WHERE id IN (SELECT id FROM inserted)
END
```

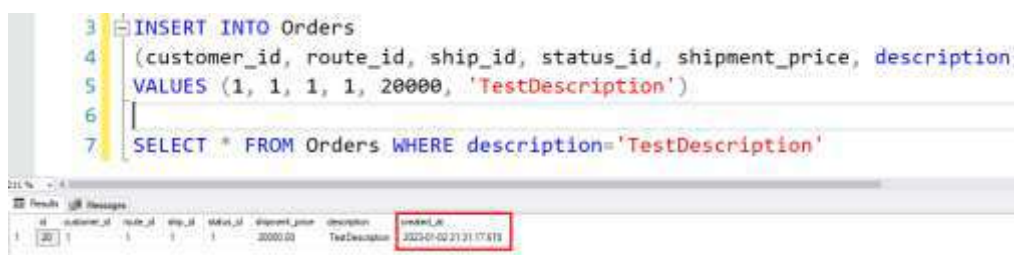


Рис. 4.6 – Приклад роботи тригера InsertIntoOrdersTrigger

4.6. Заповнити даними створені таблиці.

В пп. 4.6.1 – 4.6.12 наведено частину коду для вставки в таблицю на прикладі декількох перших значень.

4.6.1 Заповнення таблиці Customers

```
INSERT INTO Customers
VALUES
```

('Edwina', 'Hughes', '1971-4-5', '5581 Blanda Corner, Suite 956, 11838, North Norberto, Ohio, United States', '+118760366874', 'edwina_hughes71@gmail.com'),

('Aimil', 'Kelly', '1962-5-1', '74846 Kunde Haven, Apt. 025, 70583-7845, Hillardmouth, Oklahoma, United States', '+122317109585', 'aimil_kelly62@gmail.com'),

...

id	first_name	second_name	birth_date	address	phone_number	email
1	Edwina	Hughes	1971-04-05	5581 Blanda Corner, Suite 956, 11838, North Norberto, Ohio, United States	+118760366874	edwina_hughes71@gmail.com
2	Airel	Kelly	1962-05-01	74846 Kunde Haven, Apt. 025, 70583-7845, Hillardmouth, Oklahoma, United States	+122317109985	airel_kelly62@gmail.com
3	Biddy	James	1953-11-10	908 Dietrich Streets, Apt. 049, 79886-5066, South Lauryville, North Dakota, United States	+188131766406	biddy_james53@gmail.com
4	Marietta	Gutierrez	1953-04-20	84281 Bianka Mission, Apt. 736, 70328-4378, Ejahton, Idaho, United States	+122918221166	marietta_gutierrez53@gmail.com
5	Mama	Kim	1981-02-03	03951 Collier Creek, Suite 716, 03083-7549, East Corneliusborough, Arkansas, United States	+158157888905	mama_kim81@gmail.com
6	Heidi	Parker	1952-07-06	13362 Wittheiser Lodge, Apt. 209, 15045, Crookchester, Texas, United States	+124833792077	heidi_parker52@gmail.com
7	Andie	Ramirez	1993-07-20	352 Jerod Brooks, Apt. 469, 15999-0477, Lake Jordberg, New Mexico, United States	+142937938320	andie_ramirez93@gmail.com
8	Chelley	Moore	1989-04-09	74325 Hiram Vile, Apt. 267, 72134-9126, Amaldoshire, Pennsylvania, United States	+191401932293	chelley_moore89@gmail.com
9	Kansa	Garcia	1990-03-13	3438 Cars Landing, Apt. 254, 14804, Theodorehaven, Hawaii, United States	+116965303746	kansa_garcia90@gmail.com
10	Madelena	Hill	1973-09-05	28933 Roine Coves, Apt. 740, 01490, Lake Otis, Maine, United States	+186525779159	madelena_hill73@gmail.com
11	Ava	Diaz	1988-05-07	5239 Ratley Oval, Apt. 777, 86379-9729, South Micheal, Wisconsin, United States	+103430216629	ava_diaz88@gmail.com
12	Laurena	Chavez	1985-05-16	2485 Treutel Pines, Suite 332, 01326, East Elizabeth, Oklahoma, United States	+176483478745	laurena_chavez85@gmail.com
13	Carlene	Alvarez	1964-02-06	29792 Sidney Glens, Apt. 996, 94278, Arkundingsbury, West Virginia, United States	+131185934259	carlene_alvarez64@gmail.com
14	Deanne	Castillo	1978-08-13	70972 Wunsch Plains, Apt. 508, 97677, Lake Miltonberg, Massachusetts, United States	+176060573989	deanne_castillo78@gmail.com
15	Jome	Diaz	1985-07-09	67453 Satna Ferry, Apt. 847, 72111-2226, New Anshiside, Minnesota, United States	+102607264703	jome_diaz85@gmail.com

Рис. 4.7 – Таблиця Customers після вставки даних

4.6.2. Заповнення таблиці Positions

INSERT INTO Positions
VALUES

('Captain', 'The Captain or Master is the official representative of the company and acts as the manager of the vessel. The Master must also comply with all the pertinent regulations regarding the operation of the ship. In addition to these duties, the Master must ensure the safety of the ship, crew, cargo and the environment.'),

...

id	title	description
1	Captain	The Captain or Master is the official representative ...
2	Chief Mate	The Chief Mate is the second in command. As suc...
3	Second Mate	The Second Mate is the ship's navigator and is res...
4	Third Mate	The Third Mate is the most junior officer in the Dec...
5	Able-Bodied Seaman	The Able-Bodied Seaman takes on this position aft...
6	Ordinary Seaman	The Ordinary Seaman (O/S) occupies the most juni...
7	Chief Engineer	The Chief Engineer is the official representative of t...
8	Second Engineer	The Second Engineer reports to the Chief Engineer...
9	Third Engineer	The Third Engineer is an Officer of the Watch and i...
10	Fourth Engineer	The Fourth Engineer is the most junior officer in the...
11	Engine Room Rating	The Engine Room Rating is responsible for assistin...
12	Electrician	The Electrician is under the supervision of the Chie...
13	Head Tunnelman	The Head Tunnelman is responsible for supervising...
14	Tunnelman	The Tunnelman reports to and acts as instructed b...
15	Mobile Utility Employee	The Mobile Utility Employee is a floater within the fl...
16	Chief Cook	The Chief Cook reports to the Master and is respon...
17	Second Cook	The Second Cook reports to the Chief Cook and is ...

Рис. 4.8 – Таблиця Positions після вставки даних

4.6.3 Заповнення таблиці Crews

INSERT INTO Crews
VALUES

('Gossamer Crew', '2016-1-6'),

('Canton Crew', '2018-4-7'),

('Prescot Crew', '2020-3-6'),

('Isle Crew', '2016-9-18'),

...

Results		Messages	
	id	name	creation_date
1	1	Gossamer Crew	2016-01-06
2	2	Canton Crew	2018-04-07
3	3	Prescot Crew	2020-03-06
4	4	Isle Crew	2016-09-18
5	5	Blackbum Crew	2020-11-13
6	6	Polyphemus Crew	2017-03-14
7	7	Icelanders Crew	2018-08-08
8	8	Birkenhead Crew	2020-06-03
9	9	Victorian Crew	2018-05-18
10	10	Kinsale Crew	2019-08-18

Рис. 4.9 – Таблиця Crews після вставки даних

4.6.4. Заповнення таблиці Employees

INSERT INTO Employees
VALUES

(1, 1, 'Dix', 'Reyes', '1955-3-16', '622 Kessler Corners, Apt. 903, 16973, East Loma, Nevada, United States', '+150633312343', 'dix_reyes55@gmail.com', '2004-8-8', 10808),

(1, 2, 'Farica', 'Gonzales', '1972-2-15', '336 Orion Spurs, Apt. 959, 97116-8943, Bergnaumfurt, Colorado, United States', '+188513036331', 'farica_gonzales72@gmail.com', '2009-4-6', 8749),

Results		Messages									
	id	crew_id	position	first_name	second_name	birth_date	address	phone_number	email	hire_date	salary
1	1	1	1	Dix	Reyes	1955-03-16	622 Kessler Corners, Apt. 903, 16973, East Loma, Ne...	+150633312343	dix_reyes55@gmail.com	2004-08-08	10808.00
2	2	1	2	Farica	Gonzales	1972-02-15	336 Orion Spurs, Apt. 959, 97116-8943, Bergnaumfur...	+188513036331	farica_gonzales72@gmail.com	2009-04-06	8749.00
3	3	1	3	Sidora	Morgan	1951-05-10	11524 Jerry Harbor, Apt. 124, 10384, Jocoyview, Ma...	+135028453061	sidora_morgan51@gmail.com	2002-01-20	8140.00
4	4	1	4	Beatrice	Mendoza	1972-02-18	69337 Bobbie Extension, Apt. 634, 44495-1510, Vita...	+177109552889	beatrice_mendoza72@gmail.com	2008-07-16	6369.00
5	5	1	5	Maggie	Hall	1965-10-03	7706 Madie Stream, Suite 804, 85117, South Giovan...	+113131357975	maggie_hall69@gmail.com	2002-05-16	6778.00
6	6	1	6	Hermie	Green	1967-07-09	0376 Witting Ferry, Apt. 361, 01793-5688, Jesseboro...	+179373265099	hermie_green67@gmail.com	2008-05-05	5299.00
7	7	1	7	Martha	Richardson	1977-11-15	3144 Richie Junction, Apt. 298, 21392, Diegoberg, ...	+166481702593	martha_richardso77@gmail.com	2004-07-05	5709.00
8	8	1	8	Elyn	Wood	1955-08-05	7332 Geirvanny Pass, Apt. 480, 52137, East Donnes...	+143043712764	elyn_wood59@gmail.com	1999-01-10	5527.00
9	9	1	9	Abigale	Cruz	1976-11-04	4429 Milton Valley, Suite 681, 82659-5343, Port Hym...	+148989676900	abigale_cruz76@gmail.com	2008-05-14	5856.00
10	10	1	10	Carin	Green	1959-03-09	978 Jacinta Extension, Apt. 831, 42600-1919, South...	+184697172158	carin_green69@gmail.com	2003-10-02	5751.00
11	11	1	11	Donna	Ramirez	1961-05-05	813 Reilly Forks, Apt. 605, 27103, New Angelstad, ...	+145096904511	donna_ramirez61@gmail.com	2004-06-15	5217.00
12	12	1	12	Condie	Davis	1978-11-12	919 Daniel Square, Apt. 121, 57115, South Adobeto...	+128366622502	condie_davis78@gmail.com	2003-02-04	5689.00
13	13	1	13	Mark	Mitchell	1953-10-12	93901 Adaline Station, Apt. 377, 74594-8386, East M...	+135472741394	mark_mitchell53@gmail.com	1999-05-08	5349.00
14	14	1	14	Aurie	Ramirez	1958-03-12	2072 Ova Stevenson, Suite 318, 24391-0511, Marav...	+176924208793	aurie_ramirez58@gmail.com	2001-01-14	5214.00

Рис. 4.10 – Таблиця Employees після вставки даних

4.6.5. Заповнення таблиці Ships_Types

INSERT INTO Ships_Types
VALUES

('Container ship', 'It is the most common mode of sea freight transport. As the name suggests, these vessels are designed to carry standard 20', 40' and 45' containers. They can accommodate most dry-load transport. Their capacity ranges from 85 TEUs (twenty equivalent units) to 15,000+ TEUs. The biggest container vessel is the Emma Maersk. Gantry cranes are used to load and unload the boxes.'),

...

Results		Messages	
id	type	description	
1	1	Container ship	It is the most common mode of sea freight transport. As the name suggests, these vessels are designed to carry standard 20', 40' and 4...
2	2	General cargo ship	They mostly carry packaged goods but do not have space for containers. They use their own built-in cranes for loading and unloading ...
3	3	Dry bulk carriers	They are structured specifically to transport solid non-packaged loose dry cargo in bulk quantities. Such carriers are then used for bag...
4	4	Multi-purpose vessels	They can carry a combination of all kinds of goods (general cargo, liquid), thanks to separate containers and storage systems. They ar...
5	5	Reefer ship	They are designed for the carriage of frozen/temperature-controlled cargoes, mainly in refrigerated containers. Food and perishable go...

Рис. 4.11 – Таблиця Ships_Types після вставки даних

4.6.6. Заповнення таблиці Ships

INSERT INTO Ships

VALUES

(1, 1, 'Southwold', 50000, '2021-4-8'),
 (2, 1, 'Broom', 20000, '2021-5-12'),
 (3, 2, 'Spindrift', 100000, '2020-9-1'),
 (4, 2, 'Leeuwin', 120000, '2021-9-19'),

...

Results		Messages				
id	crew_id	ship_type_id	name	displacement	last_maintenance_date	
1	1	1	Southwold	50000	2021-04-08	
2	2	1	Broom	20000	2021-05-12	
3	3	2	Spindrift	100000	2020-09-01	
4	4	2	Leeuwin	120000	2021-09-19	
5	5	3	Blazer	150000	2019-06-13	
6	6	3	Mimico	170000	2021-10-10	
7	7	4	Aster	80000	2021-09-18	
8	8	4	Haman	10000	2021-05-17	
9	9	5	Puma	250000	2021-08-17	
10	10	5	Daisy	280000	2019-01-05	

Рис. 4.12 – Таблиця Ships після вставки даних

4.6.7. Заповнення таблиці Ports

INSERT INTO Ports

VALUES

('Port of Shanghai', 'Shanghai China', 744),
 ('Port of Singapore', 'Singapore', 537.6),
 ('Port of Tianjin', 'Tianjin China', 476),
 ('Port of Guangzhou', 'Guangzhou China', 460),
 ('Port of Ningbo', 'Ningbo-Zhoushan, China', 453),

...

	id	name	location	capacity
1	1	Port of Shanghai	Shanghai China	744
2	2	Port of Singapore	Singapore	537,6
3	3	Port of Tianjin	Tianjin China	476
4	4	Port of Guangzhou	Guangzhou China	460
5	5	Port of Ningbo	Ningbo-Zhoushan, China	453
6	6	Port of Rotterdam	Rotterdam, The Netherlands	441,5
7	7	Port of Suzhou	Zhangjiagang, China	428
8	8	Port of Qingdao	Qingdao, China	400
9	9	Port of Dalian	Dalian, China	303
10	10	Port of Busan	Busan, South Korea	298

Рис. 4.13 – Таблиця Ports після вставки даних

4.6.8. Заповнення таблиці Routes

INSERT INTO Routes

VALUES

(1, 2, 'Route A'),

(1, 3, 'Route B'),

(1, 4, 'Route C'),

(1, 6, 'Route D'),

...

	id	start_port_id	destination_port_id	name
1	1	1	2	Route A
2	2	1	3	Route B
3	3	1	4	Route C
4	4	1	6	Route D
5	5	2	3	Route E
6	6	2	4	Route F
7	7	2	5	Route G
8	8	12	6	Route H

Рис. 4.14 – Таблиця Routes після вставки даних

4.6.9. Заповнення таблиці Products

INSERT INTO Products

VALUES

('Baseball top', 'A top that usually has contrasting sleeves - often three quarter length.', 15, 500),

('Bolero', 'A cropped and fitted jacket or cardigan.', 20, 1000),

...

	id	name	description	price	unit
1	1	Baseball top	A top that usually has contrasting sleeves - often three quarter length.	15,00	250
2	2	Bolero	A cropped and fitted jacket or cardigan.	20,00	1000
3	3	Bodycut	A top that fastens at the crotch, to give a streamlined, tucked-in look.	100,00	1000
4	4	Corset	Corsets were traditionally undergarments that emphasized the waist. Nowadays they are usually worn as decorative tops, or as part of a gown.	125,00	1000
5	5	Duster	Dusters are half cardigan and half coat, but are always long in length. They sometimes have a slit up the back, which is a nod to their original purpose of keeping dust off.	150,00	800
6	6	Henley top	Henley shirts have round, collarless necks, which is fastened by a few buttons at the front.	120,00	650
7	7	Hoodie	A sweater with a hood to keep you comfy-cozy.	75,00	1000
8	8	Peplum top	A top with an extra ring of fabric at the waist to add flare and emphasize an hour glass figure.	150,00	950
9	9	Shrug	A cropped knitted cardigan that is usually open at the front, or loosely tied.	85,00	1000
10	10	Wheat	It is one of the most important crops in the world. It is a source of protein and dietary fiber. Wheat is mostly used for making bread, chapatis, and other baked goods in India.	95,00	5000
11	11	Rice	Rice is a staple food in different parts of the world and has to top our cereal food list. People consume rice because it's healthy and very filling and an ideal gluten-free meal.	125,00	5000
12	12	Corn	Corn is a very popular cereal among Americans and one of the major on cereal list, but it's also used in many parts of the world. It is one of the oldest types of cereals known.	90,00	2000
13	13	Oats	One of the commonest cereals grown in the world, which has a high content of protein. It comes in either a hard and soft variety in a lot of sizes. Pats are a	10,00	8000


Рис. 4.15 – Таблиця Products після вставки даних

4.6.10. Заповнення таблиці Orders_Statuses

```
INSERT INTO Orders_Statuses  
VALUES
```

('Awaiting Payment', 'Customer has completed the checkout process, but payment has yet to be confirmed. Authorize only transactions that are not yet captured have this status'),

...



id	title	description
1	Awaiting Payment	Customer has completed the checkout process, but payment has yet to be confirmed. Authorize onl...
2	Awaiting Fulfillment	Customer has completed the checkout process and payment has been confirmed
3	Awaiting Shipment	Order has been pulled and packaged and is awaiting collection from a shipping provider.
4	Partially Shipped	Only some items in the order have been shipped.
5	Shipped	Order has been shipped, but receipt has not been confirmed.
6	Completed	Order has been shipped/picked up, and receipt is confirmed.
7	Declined	Seller has marked the order as declined.
8	Refunded	Seller has used the Refund action to refund the whole order.

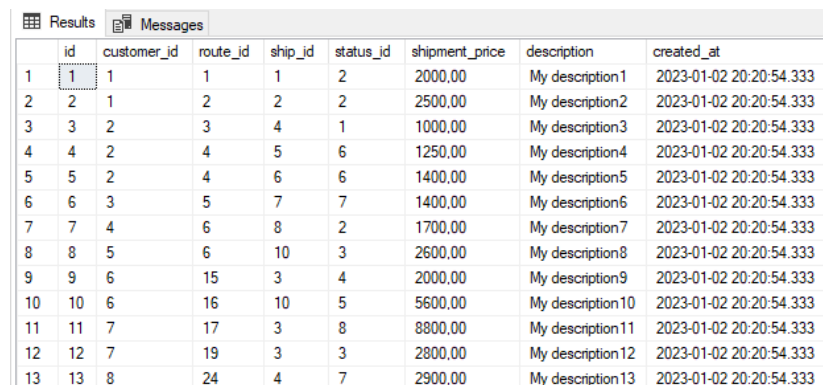
Рис. 4.16 – Таблиця Orders_Statuses після вставки даних

4.6.11. Заповнення таблиці Orders

```
INSERT INTO Orders  
(customer_id, route_id, ship_id, status_id, shipment_price, description)  
VALUES
```

(1, 1, 1, 2, 2000, 'My description1'),
(1, 2, 2, 2, 2500, 'My description2'),
(2, 3, 4, 1, 1000, 'My description3'),
(2, 4, 5, 6, 1250, 'My description4'),

...



id	customer_id	route_id	ship_id	status_id	shipment_price	description	created_at	
1	1	1	1	2	2000.00	My description1	2023-01-02 20:20:54.333	
2	2	1	2	2	2500.00	My description2	2023-01-02 20:20:54.333	
3	3	2	3	4	1000.00	My description3	2023-01-02 20:20:54.333	
4	4	2	4	5	6	1250.00	My description4	2023-01-02 20:20:54.333
5	5	2	4	6	6	1400.00	My description5	2023-01-02 20:20:54.333
6	6	3	5	7	7	1400.00	My description6	2023-01-02 20:20:54.333
7	7	4	6	8	2	1700.00	My description7	2023-01-02 20:20:54.333
8	8	5	6	10	3	2600.00	My description8	2023-01-02 20:20:54.333
9	9	6	15	3	4	2000.00	My description9	2023-01-02 20:20:54.333
10	10	6	16	10	5	5600.00	My description10	2023-01-02 20:20:54.333
11	11	7	17	3	8	8800.00	My description11	2023-01-02 20:20:54.333
12	12	7	19	3	3	2800.00	My description12	2023-01-02 20:20:54.333
13	13	8	24	4	7	2900.00	My description13	2023-01-02 20:20:54.333

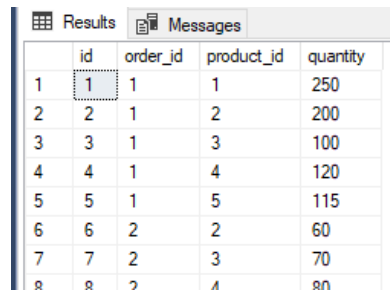
Рис. 4.17 – Таблиця Orders після вставки даних

4.6.12. Заповнення таблиці Orders_Details

```
INSERT INTO Orders_Details  
VALUES
```

(1, 1, 250), (1, 2, 200), (1, 3, 100), (1, 4, 120), (1, 5, 115),

(2, 2, 60), (2, 3, 70), (2, 4, 80), (2, 5, 90), (2, 6, 85), (2, 7, 25),
(3, 16, 60), (3, 15, 70), (3, 14, 80),



	id	order_id	product_id	quantity
1	1	1	1	250
2	2	1	2	200
3	3	1	3	100
4	4	1	4	120
5	5	1	5	115
6	6	2	2	60
7	7	2	3	70
8	8	2	4	80

Рис. 4.18 – Таблиця Orders_Details після вставки даних

Контрольні питання

1. Наведіть коротку характеристику найбільш поширених сучасних СУБД.
2. Якими типами даних може оперувати обрана вами СУБД?
3. Як налаштовувати зв'язки між таблицями в обраній СУБД?
4. За допомогою якого програмного інструменту можна зручно керувати сервером SQL?
5. Наведіть приклади і коротку характеристику об'єктно-реляційних СУБД.

Рекомендована література

1. Access developer documentation. URL: <https://docs.microsoft.com/en-us/office/client-developer/access/access-home>
2. Мулеса О., Варга Я. Інформаційні системи та реляційні бази даних: навч. посібник. Ужгород, 2023. 132 с.
3. MySQL Documentation. URL: <https://dev.mysql.com/doc/>
4. Основи роботи з базами даних із використанням СУБД Microsoft SQL Server: навчально-методичний посібник для студентів спеціальностей F2 «Інженерія програмного забезпечення» та F3 «Комп'ютерні науки» / А. М. Копп, Д. Л. Орловський. Харків: НТУ «ХП», 2026. 349 с.
5. Остапченко К. Б. Бази даних. Комп'ютерний практикум: навч. посібник. Київ: КПІ ім. Ігоря Сікорського, 2022. 151 с.

Лабораторна робота № 5

Тема: Використання мови SQL для відображення інформації

Мета: закріпити та поглибити знання студентів щодо використання команди SELECT мови SQL.

Завдання: використовуючи команду SELECT, створити запити згідно з умовами завдання.

Час виконання завдання: 4 год.

План роботи

1. Виконати завдання згідно з методичними вказівками.
2. Відповісти на контрольні питання.
3. Оформити звіт та захистити лабораторну роботу.

Завдання до виконання

5.1. Використовуючи інструкцію SELECT, створити 2 запити на виведення всіх записів з обраних вами таблиць (одна таблиця – один запит) Наприклад, вивести всі записи, що зберігаються в таблицях Orders та Orders_Statuses.

5.2. За допомогою інструкції SELECT створити 2 запити на виведення окремих атрибутів обраної вами сутності.

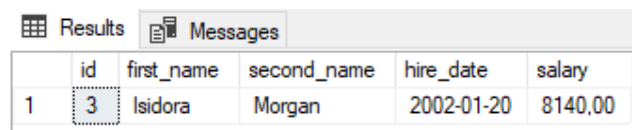
Наприклад, можна вивести всі атрибути сутності Employees.

5.3. Створити 2 запити на виведення даних без повторів значень.

5.4. Створити 2 запити на виведення даних з подвійною умовою.

Наприклад, виведемо коди, імена, прізвища, дати найму та заробітні плати працівників які прийняті на роботу 2002-01-20 та мають ім'я Isidora .

```
SELECT id, first_name, second_name, hire_date, salary
FROM Employees
WHERE (hire_date = '2002-01-20') AND (first_name = 'Isidora')
```



	id	first_name	second_name	hire_date	salary
1	3	Isidora	Morgan	2002-01-20	8140,00

Рис. 5.1 – Результат виконання запиту до завдання 5.4

5.5. Створити 2 запити на виведення даних з подвійною умовою, використанням логічних операторів (AND, OR, NOT) та реляційних операторів (=, <>, >, <, >=, <=, !=, !<).

Наприклад, вивести код, ім'я та прізвище працівника за умови, що він не служить у команді з кодом 5, його ім'я 'Donni' або Maisey' і заробітна плата більше ніж 5400 одиниць.

```
SELECT id, crew_id, first_name, second_name, salary
FROM Employees
WHERE (first_name = 'Donni' OR first_name='Maisey') AND salary > 5400 AND
crew_id != 5
```

5.6. Створити 2 запита на використання спеціального оператору IN.

Наприклад, вивести дані про всіх працівників з іменами Donni, Maisey, Lyndy, Cally.

```
SELECT *
FROM Employees
WHERE first_name IN ('Donni', 'Maisey', 'Lyndy', 'Cally')
```

5.7. Створити 2 запита на використання спеціального оператору BETWEEN.

Наприклад, вивести коди, імена, прізвища, дати найму та заробітні плати працівників за умови, що заробітна плата знаходиться в межах від 8 до 10 тисяч одиниць.

```
SELECT id, first_name, second_name, hire_date, salary
FROM Employees
WHERE salary BETWEEN 8000 AND 10000
```

5.8. Створити 2 запита на використання службового слова LIKE.

5.9. Створити 2 запита на використання службового слова LIKE зі спеціальними символами підстановки, наведеними в табл. 5.1.

Таблиця 5.1

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

Наприклад, вивести коди, імена, прізвища, дати найму та заробітні плати працівників ім'я яких починається з літери 'A'.

```
SELECT id, first_name, second_name, hire_date, salary
FROM Employees
WHERE first_name LIKE 'A%'
```

5.10. Створити 2 запити на використання оператора IS NULL для пошуку порожніх значень у комірках таблиць.

Наприклад, знайти працівників, для яких не вказаний номер телефону.

```
SELECT id, first_name, second_name, hire_date, salary
FROM Employees
WHERE phone_number IS NULL
```

Контрольні питання

1. Дайте визначення поняттю SQL та охарактеризуйте структуру мови SQL.
2. Які основні типи даних підтримує мова SQL?
3. Опишіть синтаксис та можливості команди вибірки даних з таблиці.
4. З якою метою використовують інструкцію WHERE?
5. Яким чином можна замінити використання оператора IN?

Рекомендована література

1. Босько В.В., Константинова Л.В., Поліщук Л.І., Коноплицька-Слободенюк О.К. Бази даних: навч. посібн. Кропивницький: ЦНТУ, 2024. 226 с.
2. Основи роботи з базами даних із використанням СУБД Microsoft SQL Server: навчально-методичний посібник для студентів спеціальностей F2 «Інженерія програмного забезпечення» та F3 «Комп'ютерні науки» / А. М. Копп, Д. Л. Орловський. Харків: НТУ «ХПІ», 2026. 349 с.
3. Проектування реляційних баз даних: навч. посібн. в електр. вигляді / В. О. Нелюбов, Ю. Ю. Білак, А. М. Реблян. Ужгород: ДВНЗ «УжНУ», 2023. 60 с. URL: <https://dspace.uzhnu.edu.ua/server/api/core/bitstreams/c66b090b-b87e-41e5-b4ac-e3a5a58da81c/content>
4. Salter K. Fundamentals of Database Systems. Toronto Academic Press, 2024. 272 p.
5. Taylor A. SQL All-in-One For Dummies, 4th Edition. John Wiley & Sons, Inc., 2024. 803 p.
6. SQL Server technical documentation. URL: <https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver16>

Лабораторна робота № 6

Тема: Агрегатні функції

Мета: закріпити та поглибити знання студентів щодо використання агрегатних функцій.

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

Час виконання завдання: 4 год.

План роботи

1. Виконати завдання згідно з методичними вказівками.
2. Відповісти на контрольні питання.
3. Оформити звіт та захистити лабораторну роботу.

Завдання до виконання

6.1. Створити 2 запити на використання спеціального оператора AVG з виведенням результату в поіменованій стовпчик, використавши інструкцію AS.

Наприклад, вивести середнє значення вартості товарів кількість у наявності яких в межах від 200 до 1000.

```
SELECT AVG (Products.price) AS average_price  
FROM Products  
WHERE unit BETWEEN 200 AND 1000;
```

6.2. Створити 2 запити на використання спеціального оператора COUNT з виведенням результату в поіменованій стовпчик, використавши інструкцію AS.

Наприклад, вивести в стовпчик з ім'ям "employees_counter" кількість працівників, номер команди яких 5.

```
SELECT COUNT(crew_id) AS employees_counter  
FROM Employees  
WHERE crew_id = 5
```

6.3. Створити 2 запити на використання спеціального оператора COUNT в поєднанні з інструкціями DISTINCT та AS.

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

```
SELECT DISTINCT COUNT(Orders.customer_id) AS Customers_previous_year  
FROM Orders  
WHERE (Year(Orders.created_at)=Year(GETDATE()-1));
```

6.4. Створити 2 запити на використання спеціальних операторів MIN і MAX в поєднанні з умовами "WHERE" для інших атрибутів.

Наприклад, в стовпчик з ім'ям "max_salary" вивести максимальну заробітну плату працівників п'ятої команди.

```
SELECT MAX(salary) AS max_salary
FROM Employees
WHERE crew_id = 5
```

6.5. Створити 2 запити на використання спеціального оператора SUM в поєднанні з умовами "WHERE" для інших атрибутів.

Наприклад, вивести суму яку необхідно виплатити працівникам команди під номером 5.

```
SELECT SUM(salary) AS sum_salary
FROM Employees
WHERE crew_id = 5
```

Контрольні питання

1. Назвіть типи даних, які використовуються в базі даних, що проектується.
2. Надайте опис загальної структури оператора SELECT.
3. Для чого використовують оператор AS?
4. За допомогою яких команд можна знайти та вивести поточну дату та час?
5. Назвіть логічні та арифметичні операції, виконання яких підтримує обрана вами СУБД.

Рекомендована література

1. Босько В.В., Константинова Л.В., Поліщук Л.І., Коноплицька-Слободенюк О.К. Бази даних: навч. посібн. Кропивницький: ЦНТУ, 2024. 226 с.
2. Основи роботи з базами даних із використанням СУБД Microsoft SQL Server: навчально-методичний посібник для студентів спеціальностей F2 «Інженерія програмного забезпечення» та F3 «Комп'ютерні науки» / А. М. Копп, Д. Л. Орловський. Харків: НТУ «ХПІ», 2026. 349 с.
3. Остапченко К. Б. Бази даних. Комп'ютерний практикум: навч. посібник. Київ: КПІ ім. Ігоря Сікорського, 2022. 151 с.
4. Проектування реляційних баз даних: навч. посібн. в електр. вигляді / В. О. Нелюбов, Ю. Ю. Білак. А. М. Реблян. Ужгород: ДВНЗ «УжНУ», 2023. 60 с. URL: <https://dspace.uzhnu.edu.ua/server/api/core/bitstreams/c66b090b-b87e-41e5-b4ac-e3a5a58da81c/content>
5. SQL Server technical documentation. URL: <https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver16>

Лабораторна робота № 7

Тема: З'єднання таблиць та використання групових операцій

Мета: закріпити та поглибити знання студентів 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 роботи

1. Виконати завдання згідно з методичними вказівками.
2. Відповісти на контрольні питання.
3. Оформити звіт та захистити лабораторну роботу.

Завдання до виконання

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

Наприклад, вивести імена та прізвища замовників та коди їх замовлень.

```
SELECT
  first_name,
  second_name,
  Orders.id AS OrderId
FROM Orders
INNER JOIN Customers ON customer_id = Customers.id
```

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

Наприклад, вивести інформацію про номер замовлення із зазначенням назви маршруту та корабля який здійснює перевезення.

```
SELECT
  Orders.id AS order_id,
  Routes.name AS route_name,
  Ships.name ship_name
FROM Orders
INNER JOIN Routes ON Routes.id = Orders.route_id
INNER JOIN Ships ON Ships.id = Orders.ship_id
```

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

Наприклад, вивести інформацію про робітників з їх упорядкуванням за заробітною платою та прізвищем.

```
SELECT id, first_name, second_name, salary
FROM Employees
```

```
ORDER BY salary DESC, second_name ASC
```

7.4. Створити 2 запити 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 надійшли за останні два тижні та посортувати їх за датою створення.

```
SELECT
  Orders.id,
  Orders.created_at
FROM Orders
INNER JOIN Customers ON Customers.id = Orders.customer_id
WHERE Orders.created_at >= GETDATE()-14
ORDER BY created_at DESC
```

7.5. Створити 2 запити на використання оператора GROUP BY спільно з агрегатними функціями. Вивести результати поєднавши не менше ніж з дві таблиці.

Наприклад, вивести інформацію про загальну кількість чи загальну суму замовлень, зроблених кожним покупцем.

```
SELECT
  customer_id,
  COUNT(Orders.id) AS orders_count
FROM Orders
INNER JOIN Customers ON Customers.id = Orders.customer_id
GROUP BY customer_id
```

7.6. Створити 2 запити на використання оператора GROUP BY спільно з математичними операціями.

Наприклад, використовуючи дані таблиць «Orders_Details» та «Products» розрахувати вартість усіх продуктів, що містяться у замовленні.

```
SELECT SUM(quantity * Products.price)
FROM Orders_Details
INNER JOIN Products ON product_id = Products.id
GROUP BY order_id
```

7.7. Створити 2 запити на використання оператора GROUP BY з умовою WHERE, реченням HAVING, агрегатною функцією та впорядкуванням результату.

Наприклад, використовуючи дані таблиці «Order_Details», кожне замовлення, в якому міститься продукт Rice на суму більше ніж 10000 одиниць, посортувати по спаданню за ціною продукту.

```
SELECT order_id, SUM(quantity * Products.price) AS product_cost
FROM Orders_Details
```

```
INNER JOIN Products ON Orders_Details.product_id = Products.id
WHERE Products.name = 'Rice'
GROUP BY order_id
HAVING SUM(quantity * Products.price) > 10000
ORDER BY product_cost DESC
```

Контрольні питання

1. Чим умова WHERE відрізняється від умови HAVING?
2. Чим відрізняється результат застосування команд RIGHT JOIN та LEFT JOIN?
3. Назвіть особливості використання оператора GROUP BY?
4. При виконанні інструкції SELECT який з операторів буде виконаний першим – WHERE чи HAVING?
5. Який з операторів інструкції SELECT в прикладі 7.7 буде виконаний останнім?

Рекомендована література

1. Мулеса О., Варга Я. Інформаційні системи та реляційні бази даних: навч. посібник. Ужгород, 2023. 132 с.
2. Основи роботи з базами даних із використанням СУБД Microsoft SQL Server: навчально-методичний посібник для студентів спеціальностей F2 «Інженерія програмного забезпечення» та F3 «Комп'ютерні науки» / А. М. Копп, Д. Л. Орловський. Харків: НТУ «ХПІ», 2026. 349 с.
3. Проектування реляційних баз даних: навч. посібн. в електр. вигляді / В. О. Нелюбов, Ю. Ю. Білак. А. М. Реблян. Ужгород: ДВНЗ «УжНУ», 2023. 60 с. URL: <https://dspace.uzhnu.edu.ua/server/api/core/bitstreams/c66b090b-b87e-41e5-b4ac-e3a5a58da81c/content>
4. Taylor A. SQL All-in-One For Dummies, 4th Edition. John Wiley & Sons, Inc., 2024. 803 p.
5. SQL Server technical documentation. URL: <https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver16>

Лабораторна робота № 8

Тема: Використання підзапитів мовою SQL

Мета: закріпити та поглибити знання студентів щодо використання підзапитів мовою SQL.

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

Час виконання завдання: 4 год.

План роботи

1. Виконати завдання згідно з методичними вказівками.
2. Відповісти на контрольні питання.
3. Оформити звіт та захистити лабораторну роботу.

Завдання до виконання

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

Наприклад, вивести інформацію про всі товари вартість яких менша за середню.

```
SELECT Products.id, name, description, unit
FROM Products
WHERE (Products.price < (
    SELECT AVG(price) FROM Products)
);
```

8.2. Створити 2 запита з використанням подвійного підзапиту.

Наприклад, вивести інформацію про клієнтів, які замовили телефон товар 'Corn'.

```
SELECT * FROM Customers
WHERE Customers.id IN (
    SELECT customer_id
    FROM Orders
    WHERE Orders.id IN (
        SELECT order_id FROM Orders_Details
        WHERE product_id IN
            (SELECT id FROM
                Products WHERE name = 'Corn')
    )
);
```

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

Наприклад, вивести дані товару, що займає другу позицію серед найбільш дорогих.

```
SELECT
```

```

        id,
        name,
        price,
        unit
FROM Products
WHERE price = (
                SELECT MAX(price)
                FROM Products
                WHERE price NOT IN
                    (
                        SELECT MAX(price)
                        FROM Products
                    )
            );

```

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

Наприклад, використовуючи дані таблиці «Ships» та «Ships_Types», для кожного типу корабля знайти назви кораблів з мінімальною вантажопідйомністю.

```

SELECT
    Ships_Types.type AS ship_type,
    name as ship_name,
    displacement
From Ships
INNER JOIN Ships_Types ON Ships.ship_type_id = Ships_Types.id
WHERE displacement IN (
    SELECT MIN(displacement)
    FROM Ships
    GROUP BY ship_type_id
)
ORDER BY ship_type;

```

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 імена та прізвища клієнтів, які купували продукти заданого діапазону вартості (вартість у якості параметру задає користувач при активації запити).

```

SELECT
Customers.first_name,
Customers.second_name
FROM Customers
WHERE Customers.id IN (

```

```

SELECT DISTINCT Orders.customer_id
FROM Orders
WHERE Orders.id IN
(
    SELECT order_id FROM Orders_Details
    WHERE Orders_Details.product_id
    IN
        (
            SELECT Products.id
            FROM Products
            WHERE Products.price BETWEEN ['Введіть початкове
значення ціни'] AND ['Введіть кінцеве значення ціни']
        )
    )
);

```

Контрольні питання

1. Що таке підзапити і для чого вони використовуються?
2. Яким чином створити запит з параметрами?
3. В якій послідовності виконуватимуться операції у прикладі завдання 8.4?
4. З якою метою використовують оператор IN?
5. Яким чином при написанні команди BETWEEN виключити межі діапазону значень з множини допустимих значень?

Рекомендована література

1. Мулеса О., Варга Я. Інформаційні системи та реляційні бази даних: навч. посібник. Ужгород, 2023. 132 с.
2. Основи роботи з базами даних із використанням СУБД Microsoft SQL Server: навчально-методичний посібник для студентів спеціальностей F2 «Інженерія програмного забезпечення» та F3 «Комп'ютерні науки» / А. М. Копп, Д. Л. Орловський. Харків: НТУ «ХП», 2026. 349 с.
3. Остапченко К. Б. Бази даних. Комп'ютерний практикум: навч. посібник. Київ: КПІ ім. Ігоря Сікорського, 2022. 151 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. Taylor A. SQL All-in-One For Dummies, 4th Edition. John Wiley & Sons, Inc., 2024. 803 p.
6. SQL Server technical documentation. URL: <https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver16>

Лабораторна робота № 9
Тема: Проектування звітності

Мета: закріпити та поглибити знання студентів щодо підготовки та створення звітів.

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

Час виконання завдання: 4 год.

План роботи

1. Виконати завдання згідно з методичними вказівками.
2. Відповісти на контрольні питання.
3. Оформити звіт та захистити лабораторну роботу.

Завдання 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
```

Умовні оператори та синтаксис команд для різних СУБД можуть дещо відрізнятися. 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 Orders.Order_id, Order.Qty,
CASE
  WHEN Order.Qty >= 3 THEN 15
  WHEN Order.Qty = 2 THEN 5
  ELSE 0
END AS Discount,
Goods.Price, [Price]-[Discount]*[Price]/100 AS Price_Discount
FROM Goods INNER JOIN Order ON Goods.id = Orders.Order_id;
```

Для СУБД MS Access рішення такої ж самої задачі буде виконуватись із за

діянням оператору SWITCH і буде виглядати наступним чином:

```
SELECT Orders.Order_id, Order.Qty,  
SWITCH (  
    WHEN Order.Qty >= 3 THEN 15  
    WHEN Order.Qty = 2 THEN 5  
    TRUE, 0  
) AS Discount,  
Goods.Price, [Price]-[Discount]*[Price]/100 AS Price_Discount  
FROM Goods INNER JOIN Order ON Goods.id = Orders.Order_id;
```

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 Google Data Studio, Microsoft Power BI або Apache Superset (рис 9.2 – 9.4).

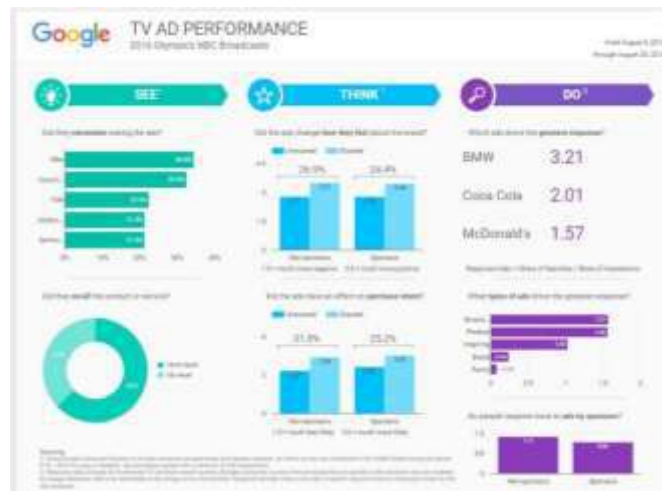


Рис 9.2 – Звіт в програмі Google Data Studio



Рис 9.3 – Звіт в програмі Microsoft Power BI



Рис 9.4 – Звіт в програмі Apache Superset.

Контрольні питання

1. Що називається звітом та які вам відомі способи створення звітів?
2. Для чого використовуються звіти?
3. У яких форматах дозволяє зберегти звіт обраний вами інструмент створення звітів?
4. Яким чином здійснюється обрахунок та виведення результатів обчислень проміжних та загальних підсумків в обраному вами інструменті створення звітів?
5. Як відбувається написання умовного оператора в обраній вами СУБД?

Рекомендована література

1. Мулеса О., Варга Я. Інформаційні системи та реляційні бази даних: навч. посібник. Ужгород, 2023. 132 с.
2. Основи роботи з базами даних із використанням СУБД Microsoft SQL Server: навчально-методичний посібник для студентів спеціальностей F2 «Інженерія програмного забезпечення» та F3 «Комп'ютерні науки» / А. М. Копп, Д. Л. Орловський. Харків: НТУ «ХПІ», 2026. 349 с.
3. Остапченко К. Б. Бази даних. Комп'ютерний практикум: навч. посібник. Київ: КПІ ім. Ігоря Сікорського, 2022. 151 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. Taylor A. SQL All-in-One For Dummies, 4th Edition. John Wiley & Sons, Inc., 2024. 803 p.
6. SQL Server technical documentation. URL: <https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver16>

Лабораторна робота № 10
Тема: Команди модифікації даних

Мета: закріпити та поглибити знання студентів щодо використання команд модифікації даних.

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

Час виконання завдання: 4 год.

План роботи

1. Виконати завдання згідно з методичними вказівками.
2. Відповісти на контрольні питання.
3. Оформити звіт та захистити лабораторну роботу.

Завдання до виконання

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

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

```
INSERT INTO Employer (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 ("Петров", "Петро", "Петрович", "тел. +38 077 555-55-55",  
"AB11001100", 121212121, "Житомир", "Бердичівська", "40", "1",  
"01.01.1980", "10.06.2000", "стажер", TRUE);
```

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

Наприклад, змінити свою посаду зі стажера на директора.

```
UPDATE Employer  
SET Посада = "директор"  
WHERE Прізвище = "Петров" AND Ім_я = "Петро";
```

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

Наприклад, скопіювати таблицю "Orders" в таблицю "Orders_Archive".

```
SELECT * INTO Orders_Archive  
FROM Orders;
```

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

Наприклад, видалити весь вміст таблиці "Orders_Archive".

```
DELETE FROM Orders_Archive;
```

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

Наприклад, наповнити таблицю "Orders_Archive" інформацією з таблиці "Orders".

```
INSERT INTO Orders_Archive
  (Orders_id, Data, id_employer, id_client, id_good, qty, data_pay)
SELECT Orders_id, Data, id_employer, id_client, id_good, qty, data_pay
FROM Orders;
```

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

Наприклад, з таблиці "Employers" видалити запис про конкретну особу.

```
DELETE FROM Employers
WHERE S_name = "Петров" AND Name = "Петро";
```

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 "Orders_Archive" видалити 2 замовлення, що були зроблені клієнтами з Житомира.

```
DELETE FROM
  (SELECT TOP 2 *
   FROM Orders_Archive
   WHERE Id_client IN
     (SELECT Id_client FROM Clients
      WHERE Addr_city = '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 "Orders"). Величину відсотка задати як параметр, який вводить менеджер.

```
UPDATE Goods SET Price = (Price*(1-["Вкажіть величину знижки"]/100))
WHERE Id_goods NOT IN
  (SELECT Id_goods
   FROM Goods);
```

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

```
DROP TABLE Orders_Archive;
```

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

Наприклад, додати до таблиці Orders стовпець із назвою Additional_Description.

```
ALTER TABLE Orders  
ADD Additional_Description TEXT(30) NOT NULL;
```

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

Наприклад, видалити колонку Additional_Description із таблиці Orders.

```
ALTER TABLE Orders  
DROP COLUMN Additional_Description;
```

Контрольні питання

1. Чим відрізняються оператори SELECT INTO та INSERT INTO SELECT?
2. Опишіть синтаксис та можливості команд видалення таблиць та їх змісту.
3. Опишіть синтаксис та можливості команд перейменування таблиць.
4. Опишіть синтаксис та можливості команд вставки, оновлення та видалення даних з таблиці.
5. Яким чином в обраній вами СУБД можна додати дані в поле типу «лічильник»?

Рекомендована література

1. Босько В.В., Константинова Л.В., Поліщук Л.І., Коноплицька-Слободенюк О.К. Бази даних: навч. посібн. Кропивницький: ЦНТУ, 2024. 226 с.
2. Мулеса О., Варга Я. Інформаційні системи та реляційні бази даних: навч. посібник. Ужгород, 2023. 132 с.
3. Основи роботи з базами даних із використанням СУБД Microsoft SQL Server: навчально-методичний посібник для студентів спеціальностей F2 «Інженерія програмного забезпечення» та F3 «Комп'ютерні науки» / А. М. Копп, Д. Л. Орловський. Харків: НТУ «ХПІ», 2026. 349 с.
4. Остапченко К. Б. Бази даних. Комп'ютерний практикум: навч. посібник. Київ: КПІ ім. Ігоря Сікорського, 2022. 151 с.
5. Shin T. 5 Common SQL Interview Problems for Data Scientists. URL: <https://towardsdatascience.com/5-common-sql-interview-problems-for-data-scientists-1bfa02d8bae6>
6. SQL Server technical documentation. URL: <https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver16>

Лабораторна робота № 11

Тема: Проектування інтерфейсу користувача

Мета: закріпити та поглибити знання студентів щодо проектування інтерфейсу користувача.

Завдання: спроектувати інтерфейс користувача.

Час виконання завдання: 4 год.

План роботи

1. Виконати завдання згідно з методичними вказівками.
2. Відповісти на контрольні питання.
3. Оформити звіт та захистити лабораторну роботу.

Завдання до виконання

11.1. Використовуючи сучасні програми для проектування та прототипування інтерфейсів додатків, розробити не менше двох інтерфейсів користувача БД. При проектуванні врахувати права користувачів БД та їх функціональні обов'язки.

Звернути увагу на те, що, на відміну від СУБД Microsoft Office Access (рис. 11.1), не всі сучасні СУБД мають вбудовані інструменти для створення інтерфейсів користувачів. Тому для виконання даного завдання можна використати безкоштовний он-лайн сервіс Figma (рис. 11.2).

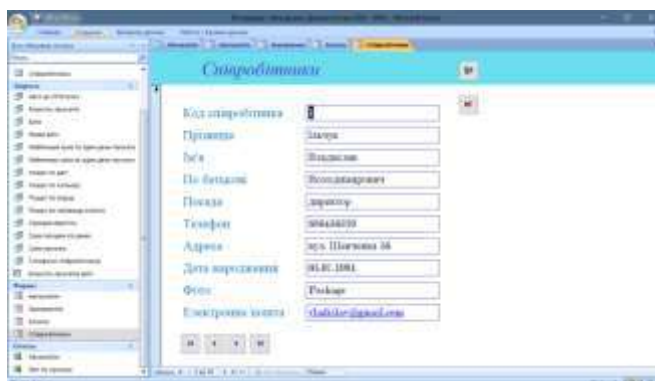


Рис. 11.1 – Приклад форми для введення даних (СУБД Microsoft Office Access)



Рис. 11.2 – Приклад інтерфейсу користувача створеного в додатку Figma

Контрольні питання

1. З якою метою проектують інтерфейс користувача?
2. Назвіть ознаки досконалого інтерфейсу користувача.
3. Які фактори потрібно враховувати при проектуванні інтерфейсів баз даних для мобільних додатків?
4. Які типи інтерфейсів ви можете назвати? Наведіть їх короткий опис.
5. Як ви розумієте термін «UI usability»?

Рекомендована література

1. Access developer documentation. URL: <https://docs.microsoft.com/en-us/office/client-developer/access/access-home>
2. Microsoft. Створення форми в Access. URL: <https://support.microsoft.com/uk-UA/Access/create-a-form-in-access>
3. Microsoft. Створення запиту, форми та звіту в Access URL: <https://support.microsoft.com/uk-ua/access/training/create-a-query-form-or-report-in-access>
4. Офіційний сайт Figma. URL: <https://www.figma.com/>
5. Керівництво користувача засобом прототипування та проектування дизайну FIGMA. URL: https://msn.khnu.km.ua/pluginfile.php/418503/mod_resource/content/2/%D0%94%D0%BE%D0%B4%D0%B0%D1%82%D0%BA%D0%B8.pdf
6. Робота з базами даних: навч. посіб. для здобувачів ступеня бакалавра за освіт. програмою «Інформаційно-комунікаційні технології» спец. 172 Електронні комунікації та радіотехніка / Л.С. Глоба, С.В. Суліма, М.А. Скулиш; КПІ ім. Ігоря Сікорського. 2-е вид. Київ: КПІ ім. Ігоря Сікорського, 2024. 532 с. URL: <https://ela.kpi.ua/bitstreams/ae45b0ef-ff71-4d39-90e2-930756c3d130/download>

Лабораторна робота № 12

Тема: Технології доступу та захист інформації в СУБД. Розподілені БД

Мета: закріпити та поглибити знання студентів щодо доступу та захисту даних, в тому числі розподілених баз даних.

Завдання: налаштувати права доступу та ролі користувачів в мережевих БД.

Час виконання завдання: 4 год.

План роботи

1. Виконати завдання згідно з методичними вказівками.
2. Відповісти на контрольні питання.
3. Оформити звіт та захистити лабораторну роботу.

Завдання до виконання

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

12.1. 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, для СУБД Microsoft SQL Server при роботі з базою даних SeaShippingCompany програмний код може мати вигляд:

```
GO
CREATE LOGIN Administrator
WITH PASSWORD = 'qwerty',
DEFAULT_DATABASE = SeaShippingCompany
GO
USE SeaShippingCompany
CREATE USER Administrator
FOR LOGIN Administrator
GO
USE SeaShippingCompany
GRANT SELECT TO Administrator
GRANT INSERT TO Administrator
GRANT UPDATE TO Administrator
GRANT DELETE TO Administrator
```

12.2. В обраній вами СУБД створіть 2 нових користувача. Налаштуйте їх дозволи на роботу з різними об'єктами бази даних.

Наприклад, створимо користувач ProductManager, який має права на роботу лише з таблицею Products.

```

GO
CREATE LOGIN ProductManager
WITH PASSWORD = 'qwerty',
DEFAULT_DATABASE = SeaShippingCompany
GO
USE SeaShippingCompany
CREATE USER ProductManager
FOR LOGIN ProductManager
GO
USE SeaShippingCompany
GRANT SELECT ON [dbo].[Products] TO ProductManager
GRANT INSERT ON [dbo].[Products] TO ProductManager
GRANT DELETE ON [dbo].[Products] TO ProductManager
GRANT UPDATE ON [dbo].[Products] TO ProductManager

```

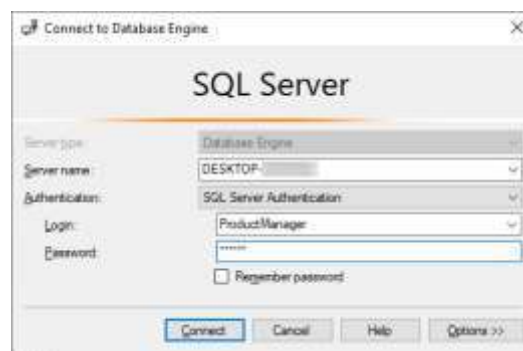


Рис. 12.1. Авторизація користувача ProductManager у базі даних

Створимо обліковий запис користувача OrdersManager, який має повний доступ до таблиць в яких зберігається інформація про замовлення, а саме: Orders та Orders_Details. Також надамо йому дозвіл на вибірку даних з таблиці Products.

```

GO
CREATE LOGIN OrdersManager
WITH PASSWORD = 'qwerty',
DEFAULT_DATABASE = SeaShippingCompany
GO
USE SeaShippingCompany
CREATE USER OrdersManager
FOR LOGIN OrdersManager
GO
USE SeaShippingCompany
GRANT SELECT ON [dbo].[Orders] TO OrdersManager
GRANT INSERT ON [dbo].[Orders] TO OrdersManager
GRANT DELETE ON [dbo].[Orders] TO OrdersManager
GRANT UPDATE ON [dbo].[Orders_Details] TO OrdersManager
GRANT SELECT ON [dbo].[Products] TO OrdersManager

```

12.3. Використовуючи SQL команду REVOKE, обмежте створений у завданні 12.2 запис користувача в правах на деякі операції.

12.4. Використовуючи SQL команду GRANT передайте користувачу, створеному у завданні 12.2 права на деякі операції.

Контрольні питання

1. Поясніть основні задачі адміністрування баз даних.
2. Для чого потрібно розподіляти користувачів БД за ролями?
3. За допомогою яких команд можна обмежити права користувача?
4. Назвіть можливі ролі користувачів в обраній вами СУБД.
5. Назвіть характерні властивості розподілених баз даних?

Рекомендована література

1. SQL Server technical documentation. URL: <https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver16>
2. Мулеса О., Варга Я. Інформаційні системи та реляційні бази даних: навч. посібник. Ужгород, 2023. 132 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/>
4. Основи роботи з базами даних із використанням СУБД Microsoft SQL Server: навчально-методичний посібник для студентів спеціальностей F2 «Інженерія програмного забезпечення» та F3 «Комп'ютерні науки» / А. М. Копп, Д. Л. Орловський. Харків: НТУ «ХПІ», 2026. 349 с.
5. Робота з базами даних: навч. посіб. для здобувачів ступеня бакалавра за освіт. програмою «Інформаційно-комунікаційні технології» спец. 172 Електронні комунікації та радіотехніка / Л.С. Глоба, С.В. Суліма, М.А. Скулиш; КПІ ім. Ігоря Сікорського. 2-е вид. Київ: КПІ ім. Ігоря Сікорського, 2024. 532 с. URL: <https://ela.kpi.ua/bitstreams/ae45b0ef-ff71-4d39-90e2-930756c3d130/download>

Лабораторна робота № 13

Тема: Програмний доступ до бази даних засобами Python

Мета: ознайомитися з механізмами програмного доступу до реляційної бази даних засобами мови Python та набути практичних навичок виконання SQL-запитів із прикладного застосунку.

Завдання: розробити Python-застосунок для програмної взаємодії з власною базою даних, створеною в попередніх лабораторних роботах.

Розроблений застосунок повинен:

- підключатися до MS SQL Server;
- виконувати операції вибірки, створення та модифікації даних;
- використовувати параметризовані запити.

Час виконання завдання: 8 год.

План роботи

1. Виконати завдання згідно з методичними вказівками.
2. Відповісти на контрольні питання.
3. Оформити звіт та захистити лабораторну роботу.

Завдання до виконання

13.1. Перед початком виконання лабораторної роботи потрібно встановити Python. Завантажити Python можна з офіційного сайту [Python.org](https://python.org).

Під час встановлення **обов'язково увімкнути опцію:** *Add Python to PATH*.

Перевірити правильність установки можна в PowerShell або командному рядку (cmd) за допомогою команд:

- Перевірка Python:
python --version

Якщо Python встановлено правильно, у терміналі з'явиться напис, наприклад Python 3.12.0. Якщо видає помилку, то Python не встановлено або не додано до системної змінної PATH.

- Перевірка менеджера пакетів (pip):
pip --version

У відповідь ви маєте побачити версію pip та шлях до нього. Якщо ця команда видає помилку (навіть коли *python --version* працює), спробуйте універсальну команду:
python -m pip --version

Якщо проблема все ще не вирішена, виконайте команду:

python -m ensurepip --default-pip
або
python -m ensurepip --upgrade

13.2. Також, необхідно встановити середовище розробки (IDE).

Рекомендується використовувати [Visual Studio Code](https://visualstudio.microsoft.com/) або [PyCharm](https://www.jetbrains.com/pycharm/).

Інструкції по їх використанню можна завантажити з адрес:
<https://code.visualstudio.com/docs/python/python-tutorial> - Налаштування Python у VS Code

<https://www.jetbrains.com/help/pycharm/quick-start-guide.html> - Getting Started with PyCharm

13.3. Для роботи з MS SQL Server необхідно встановити драйвер ODBC.

Інструкція по встановленню Microsoft ODBC Driver for SQL Server:
<https://learn.microsoft.com/en-us/sql/connect/odbc/download-odbc-driver-for-sql-server?view=sql-server-ver17>

Взаємодія із БД відбувається за допомогою бібліотеки **pyodbc**.

Встановити її можна за допомогою команди:

```
pip install pyodbc
```

13.4. Створити новий Python проєкт та підключитись до БД.

Структура проєкту може мати наступний вигляд:

```
database_lab13/  
|  
|— requirements.txt (для управління залежностями)  
|— database.py (для отримання підключення до БД)  
|— test_connection.py(файл із перевіркою підключення до БД)  
|— select.py (із завданням на SELECT)  
...  
|— join.py (із останнім завданням, на JOIN).
```

Файл requirements.txt:

```
≡ requirements.txt  
1  pyodbc  
2
```

Файл database.py (для отримання підключення до БД):

```
import pyodbc
```

```
def get_connection():  
    connection = pyodbc.connect(  
        'DRIVER={ODBC Driver 17 for SQL Server};'  
        'SERVER=localhost;'  
        'DATABASE=ShippingCompanyDB;'  
        'UID=sa;'  
        'PWD=password'  
    )  
    return connection
```

Файл test_connection.py (файл із виконаним завданням)

```
from database import get_connection

try:
    connection = get_connection()
    print("Підключення успішне!")
    connection.close()

except Exception as error:
    print("Помилка підключення:")
    print(error)
```

Увага! Запропонована структура проекту та лінійний код (без функцій/класів) є рекомендованими для спрощення виконання роботи.

Якщо ви володієте Python на більш просунутому рівні, ви можете (і це вітається) організувати проєкт за власним бажанням: використовувати об'єктно-орієнтований підхід (ООП), розділити логіку на функції або реалізувати патерн Repository / DAO. Головна умова — виконання всіх функціональних завдань лабораторної роботи.

13.5. Створити новий файл: «select.py». Використовуючи курсор, надіслати 2 запити на виведення усіх записів із обраних вами таблиць (одна таблиця – один запит). Результат вивести у консоль.

Приклад вибірки всіх записів таблиці «Ships»:

```
from database import get_connection

with get_connection() as connection:
    cursor = connection.cursor()

    query = """
    SELECT id, name, displacement
    FROM Ships
    """

    cursor.execute(query)

    ships = cursor.fetchall()

    for ship in ships:
        print(ship)

    cursor.close()
```

13.6. В новому файлі «parametrized_select.py» виконати 2 запити на витягування даних із умовою WHERE. З метою запобігання SQL ін'єкції використати параметризовані запити. Результат вивести в консоль.

Приклад виведення інформації про кораблі із водотоннажністю більше за задане значення:

```
from database import get_connection

with get_connection() as connection:
    cursor = connection.cursor()

    min_displacement = 5000

    query = """
    SELECT id, name, displacement
    FROM Ships
    WHERE displacement > ?
    """

    cursor.execute(query, min_displacement)

    ships = cursor.fetchall()

    for ship in ships:
        print(ship)

    cursor.close()
```

13.7. Додати 3 записи до обраної вами таблиці (не менше 5 колонок із різними типами даних). Запит повинен бути захищений від SQL Injection за допомогою параметризації. Файл – «insert.py».

Приклад додавання 1 запису до таблиці «Customers»:

```
from database import get_connection

with get_connection() as connection:
    cursor = connection.cursor()

    query = """
    INSERT INTO Customers
    (first_name, second_name, birth_date,
    address, phone_number, email)
    VALUES (?, ?, ?, ?, ?, ?)
    """

    values = (
        "Ivan",
```

```

        "Petrenko",
        "1995-10-11",
        "Kyiv",
        "+380991112233",
        "ivan@gmail.com"
    )

    cursor.execute(query, values)

    connection.commit()

    print("Клієнта додано!")

    cursor.close()

```

Важливо! Без рядка `connection.commit()` дані не будуть збережені.

13.8. Оновити запис в обраній вами таблиці за допомогою команди UPDATE. Код розмістити у файлі «update.py».

Приклад оновлення зарплати співробітника:

```

from database import get_connection

with get_connection() as connection:
    cursor = connection.cursor()

    query = """
    UPDATE Employees
    SET salary = ?
    WHERE id = ?
    """

    cursor.execute(query, 35000, 5)

    connection.commit()

    print("Дані оновлено!")

    cursor.close()

```

13.9. Створити запит на видалення запису по ID із обраної вами таблиці. Також реалізувати видалення усіх записів, що відповідають заданому критерію (наприклад: усіх клієнтів яким менше 18 років). Код для із обома запитами зберегти у файлі «delete.py».

Приклад видалення клієнта за заданим ID:

```

from database import get_connection

with get_connection() as connection:
    cursor = connection.cursor()

    query = """
DELETE FROM Customers
WHERE id = ?
"""

    cursor.execute(query, 10)

    connection.commit()

    print("Запис видалено!")

    cursor.close()

```

13.10. Побудувати запит із використанням операції JOIN для отримання результатів з БД. Операція з'єднання повинна бути між двома або більше таблицями. Файл – «join.py».

Приклад отримання списку замовлень разом із клієнтами та кораблями:

```

from database import get_connection

with get_connection() as connection:
    cursor = connection.cursor()

    query = """
SELECT
    o.id,
    c.first_name,
    c.second_name,
    s.name,
    o.shipment_price
FROM Orders o
JOIN Customers c
    ON o.customer_id = c.id
JOIN Ships s
    ON o.ship_id = s.id
"""

    cursor.execute(query)
    orders = cursor.fetchall()

    for order in orders:
        print(order)

    cursor.close()

```

Контрольні питання

1. Що таке pyodbc?
2. Для чого використовується курсор?
3. Яка різниця між fetchone() та fetchall()?
4. Навіщо використовувати параметризовані запити?
5. Що таке SQL Injection?
6. Як реалізувати SELECT із параметром у Python?
7. Навіщо закривати курсор та підключення до БД?

Рекомендована література

1. Real Python – Working with Databases in Python. URL: <https://realpython.com/python-sql-libraries/>
2. Python Documentation – sqlite3 module (Python DB API). URL: <https://docs.python.org/3/library/sqlite3.html>
3. W3Schools – Python MySQL Database Tutorial. URL: https://www.w3schools.com/python/python_mysql_getstarted.asp
4. Робота з базами даних: навч. посіб. для здобувачів ступеня бакалавра за освіт. програмою «Інформаційно-комунікаційні технології» спец. 172 Електронні комунікації та радіотехніка / Л.С. Глоба, С.В. Суліма, М.А. Скулиш; КПІ ім. Ігоря Сікорського. 2-е вид. Київ: КПІ ім. Ігоря Сікорського, 2024. 532 с. URL: <https://ela.kpi.ua/bitstreams/ae45b0ef-ff71-4d39-90e2-930756c3d130/download>
5. Палагін В. В., Палагіна О. А., Зорін О. С. Основи Python та програмування електронних систем: навч. посіб. Черкаси: ЧДТУ, 2024. 216 с.
6. Методичні вказівки до виконання лабораторних робіт з дисципліни «Програмування, бази даних і знань» для студентів 122 спеціальності «Комп'ютерні науки» / А. О. Лисенко, О. О. Лисенко. Харків: НТУ «ХПІ», 2024. 86 с.

Лабораторна робота № 14

Тема: Реалізація транзакційних операцій у базі даних засобами Python

Мета: Ознайомитися з механізмами транзакційної обробки даних у реляційних базах даних та набути практичних навичок реалізації транзакцій у Python–застосунках при роботі з MS SQL Server.

Завдання: реалізувати консольний застосунок із транзакційною обробкою даних в межах комплексного багатокрокового бізнес-процесу.

Розроблений застосунок повинен:

- використовувати транзакції при модифікації даних;
- реалізовувати завдання в межах багатокрокових бізнес-операцій;
- виконувати відкат змін (rollback) у разі помилки.

Час виконання завдання: 8 год.

План роботи

1. Виконати завдання згідно з методичними вказівками.
2. Відповісти на контрольні питання.
3. Оформити звіт та захистити лабораторну роботу.

Завдання до виконання

14.1. Транзакційний INSERT. Реалізувати бізнес процес створення нового замовлення або іншої сутності, який буде містити декілька операцій вставки нових даних в різні таблиці (не менше 2 варіантів).

Наприклад, можна вирішити наступне завдання:

Для бізнес-процесу «створення замовлення» за умовами бізнес-процесу потрібно виконати наступні дії:

- розрахувати суму доставки;
- створити запис у таблиці «Orders»;
- додати замовлені товари в таблицю «Orders_Details»;

Усі операції повинні бути виконані в рамках однієї транзакції.

```
from database import get_connection

def create_order(customer_id, route_id, ship_id, status_id, products):
    with get_connection() as connection:
        try:
            cursor = connection.cursor()

            order_query = """
            INSERT INTO Orders (
                customer_id, route_id, ship_id,
                status_id, shipment_price, created_at
            )
            OUTPUT INSERTED.id
            VALUES (?, ?, ?, ?, ?, GETDATE())
            """
```

```

shipment_price = 0 # calculate shipment price logic

cursor.execute(
    order_query, customer_id, route_id,
    ship_id, status_id, shipment_price
)

order_id = cursor.fetchone()[0]

details_query = """
INSERT INTO Orders_Details
(order_id, product_id, quantity)
VALUES (?, ?, ?)
"""

for product_id, quantity in products:
    cursor.execute(details_query, order_id, product_id, quantity)

connection.commit()

print("Замовлення успішно створено!")

except Exception as error:
    connection.rollback()

    print("Помилка!")
    print(error)

finally:
    cursor.close()

```

Під час виконання вищенаведеного коду в рамках однієї бізнес-операції (транзакції) виконується:

- INSERT у таблицю Orders (створення замовлення);
- INSERT у таблицю Orders_Details (додавання товарів).

Якщо будь-який INSERT завершиться помилкою, то виконається *connection.rollback()* і всі зміни будуть скасовані.

14.2. Тестування роботи механізму rollback.

При умові, що транзакція – це набір операцій, які мають бути виконані або всі разом, або жодної з них, протестуйте роботу механізму відкату транзакції.

Штучно створіть помилку на етапі, коли бізнес-процес вже частково виконаний, але перед тим, як він виконається повністю і відпрацює *commit()*.

Наприклад:

```

with get_connection() as connection:
    try:
        cursor.execute("INSERT ...") # перша частина бізнес-процесу
        # ...
        raise Exception("Тестова помилка")
        # ...
        cursor.execute("INSERT...") # друга частина бізнес-процесу
        connection.commit()

    except Exception as error:
        connection.rollback()

    finally:
        cursor.close()

```

Перевірте дані в БД після виконання вищенаведеного коду впевнитесь, що ніяких змін даних не відбулось, оскільки при виникненні помилки транзакція була відмінена, разом із усіма вже виконаними змінами. Це можливо завдяки транзакції та її властивості – атомарності.

Щоб побачити, до яких наслідків призвела б відсутність транзакцій, замініть метод *rollback()* у блоці *except* на метод *commit()*.

При запуску такої програми виконається лише перша частина операцій бізнес-процесу. Друга частина буде пропущена через помилку, в результаті чого дані в БД частково зміняться, але перестануть бути консистентними.

На прикладі замовлення: буде створено запис у таблиці «Orders», але інформація про конкретні товари із замовлення буде втрачена, оскільки в таблиці «Order_Details» не було створено жодних записів.

14.3. Реалізувати транзакційний UPDATE. Для цього обрати бізнес-процес, який охоплює модифікацію даних одразу в декількох таблицях (не менше 2 варіантів).

Прикладом виконання цього завдання може бути наступна умова:

Бізнес-процес зміни маршруту доставки одночасно повинен здійснити дві модифікації:

- змінити маршрут;
- перерахувати вартість доставки.

```

with get_connection() as connection:

    cursor = connection.cursor()

    try:
        cursor.execute(update_route)
        cursor.execute(update_price)

```

```

        connection.commit()

except Exception:
    connection.rollback()

finally:
    cursor.close()

```

14.4. Реалізувати транзакційний DELETE, що охоплює операції по видаленню записів одразу із декількох таблиць (не менше 2 варіантів).

Прикладом виконання цього завдання може бути бізнес-процес по видаленню замовлення. При видаленні замовлення потрібно:

- видалити усі записи із таблиці Orders_Details пов'язані із замовленням;
- видалити відповідний заданому order_id запис із таблиці Orders .

Програмний код для виконання такої транзакції може виглядати наступним чином:

```

with get_connection() as connection:
    cursor = connection.cursor()

    try:
        cursor.execute(
            "DELETE FROM Orders_Details WHERE order_id=?",
            order_id
        )

        cursor.execute(
            "DELETE FROM Orders WHERE id=?",
            order_id
        )

        connection.commit()

    except Exception:
        connection.rollback()

    finally:
        cursor.close()

```

Контрольні питання

1. Що таке транзакція?
2. Для чого використовується метод commit()?
3. Для чого використовується метод rollback()?
4. Що означає термін «атомарність»?

5. Що таке ACID?
6. Чому CRUD-операції можуть вимагати транзакцій?
7. Що станеться, якщо помилка виникне до виконання методу `commit()`?
8. Які переваги реалізації багатокрокових бізнес-процесів через використання транзакцій у прикладних системах?