

Міністерство освіти і науки України
Таврійський національний університет імені В.І. Вернадського
Навчально-науковий інститут муніципального управління та міського
господарства
Кафедра інженерних систем та технологій

ОРГАНІЗАЦІЯ БАЗ ДАНИХ

МЕТОДИЧНІ ВКАЗІВКИ ДО ВИКОНАННЯ КУРСОВОГО ПРОЕКТУ

Форма навчання: очна (денна) і заочна

Спеціальність (освітньо-професійна програма):

122 Комп'ютерні науки (Комп'ютерні науки)

Спеціальність (освітньо-професійна програма):

123 Комп'ютерна інженерія (Комп'ютерна інженерія)

Упорядник: Лісовець Сергій Миколайович, доцент кафедри інженерних систем та технологій, к.т.н., доцент

Рецензенти: Барилко Сергій Віталійович, професор кафедри інформаційно-вимірювальних технологій Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського», д.т.н., доцент;
Кисельов Володимир Борисович, директор навчально-наукового інституту муніципального управління та міського господарства Таврійського національного університету імені В.І. Вернадського, д.т.н., професор

Рекомендовано до друку Вченою радою Таврійського національного університету імені В.І. Вернадського. Протокол № 8 від 27.03.2025 р.

Рекомендовано навчально-методичною радою навчально-наукового інституту муніципального управління та міського господарства Таврійського національного університету імені В.І. Вернадського. Протокол № 1 від 27.02.2025 р.

Організація баз даних: Методичні вказівки до виконання курсового проекту (очна (денна) і заочна форми навчання) / упор.: Лісовець С.М. – К.: ТНУ, 2025. – 24 с.

В методичних вказівках до виконання курсового проекту розглянуто питання доступу до SQL-баз даних, використовуючи такі технології роботи з SQL-базами даних Microsoft Corporation, як Microsoft SQL Server і Microsoft SQL Server Management Studio. Виконання курсового проекту передбачає створення і заповнення даними таблиці SQL-бази даних, проведення розрахунків над такими даними, а також здійснення доступу до таких даних з використанням функцій, збережуваних процедур, представлень і курсорів. Курсовий проект спрямований на закріплення існуючих і на розвиток нових теоретичних і практичних знань з дисципліни “Організація баз даних”.

ЗМІСТ

ЗМІСТ	3
ВСТУП.....	4
1. ПОСТАНОВКА ЗАДАЧІ.....	5
2. ЗАВДАННЯ НА КУРСОВИЙ ПРОЕКТ	6
3. ВИМОГИ ДО СТРУКТУРИ І ПРАВИЛ ОФОРМЛЕННЯ КУРСОВОГО ПРОЕКТУ	8
4. ПРИКЛАД ВИКОНАННЯ КУРСОВОГО ПРОЕКТУ	10
БІБЛІОГРАФІЧНІ ПОСИЛАННЯ.....	23
ДОДАТОК А. ЗОВНІШНІЙ ВИГЛЯД ТИТУЛЬНОГО АРКУШУ	24

ВСТУП

В сучасному світі об'єм даних, який використовується в побуті, в промисловості і в інших областях діяльності людини, невідомо збільшується. Одночасно з цим розширюється перелік типів даних, які використовуються: зокрема, виникає необхідність зберігати аудіо- і відеодані в різних форматах. Відповідно, розвиваються засоби доступу до таких даних, причому вимоги до їх продуктивності, надійності, захищеності тощо теж постійно зростають. Це стосується як апаратної, так і програмної складових таких засобів.

Одним з засобів надійного зберігання даних є бази даних, а високопродуктивного доступу до них є системи керування такими базами даних. Серед них перші місця за багатьма властивостями займають SQL-бази даних і відповідні системи керування ними. Технологія роботи з SQL-базами даних відома кілька останніх десятиліть – вона є достатньо зрілою, але одночасно з цим вона постійно розвивається. Сучасні SQL-бази даних дозволяють оперувати практично будь-якими даними, які використовуються в сучасному інформаційному просторі. Така “універсальність” SQL-баз даних дозволяє використовувати їх в різних технічних системах – наприклад, під час виконання виробничих процесів часто виникає необхідність в обробці великих об'ємів виробничої інформації [10, 20].

Незважаючи на наявність на ринку SQL-баз даних різних виробників, одним з лідерів такого ринку є Microsoft Corporation, яка представляє свою систему керування SQL-базами даних Microsoft SQL Server, що має багато різних виконань (наприклад, для настільних комп'ютерів, для невеликих серверів або для Web-серверів). Для доступу до SQL-баз даних використовується мова запитів SQL, яка є стандартизованою (зокрема, згідно із стандартом SQL:2023). Microsoft Corporation має свою версію мови запитів SQL, яка має назву Transact-SQL. Основою всіх мов запитів SQL є реляційна модель, яка представляє собою математичну модель, основними складовими якої є теорія множин і логіка предикатів. SQL-запити визначають, що треба виконати, а безпосередньо системи керування SQL-базами даних визначають, як це буде виконано.

Таким чином, курсовий проект полягає в дослідженні основ доступу до SQL-баз даних, використовуючи такі технології роботи з SQL-базами даних Microsoft Corporation, як Microsoft SQL Server і Microsoft SQL Server Management Studio.

Він спрямований на закріплення існуючих і на розвиток нових теоретичних і практичних знань з дисципліни “Організація баз даних”.

1. ПОСТАНОВКА ЗАДАЧІ

Будь-який достатньо великий об'єм даних, особливо якщо він утримує цінну інформацію, потребує ефективних засобів керування. Такі дані можуть бути отримані з різних джерел: наприклад, вони можуть представляти собою результат вимірювання протягом визначеного часу параметрів виробничого процесу або ж результат продажів різних груп товарів. В будь-якому випадку до таких даних необхідно мати можливість отримати доступ тоді, коли в цьому виникне нагальна потреба.

Загалом доступ до даних передбачає, що дані можуть бути в потрібний момент часу створені або додані до вже існуючих. Протягом певного часу вони повинні надійно зберігатися з можливістю їх резервування. Крім того, необхідно мати можливість дані видаляти, оновлювати, сортувати і змінювати, здійснювати пошук в них згідно із заданими критеріями, відновлювати їх у випадку втрати і так далі.

Все це передбачає, з одного боку, що система керування даними повинна бути “багатогранною” – тобто за необхідності вона повинна реалізовувати ті або інші функції керування такими даними. З іншого боку, в експлуатації така система повинна буди достатньо простою і зручною.

Маючи таку “відправну точку”, в курсовому проекті розглядаються питання створення саме такої системи керування даними, але з обмеженими рамками курсового проектування функціональними можливостями.

Виконання курсового проекту передбачає виконання наступних попередніх дій: встановлення на персональному комп'ютері Microsoft SQL Server (існують безкоштовні версії) і Microsoft SQL Server Management Studio. Після встановлення вказаного програмного забезпечення виконання курсового проекту полягає в послідовному виконанні всіх пунктів завдання на курсовий проект із занесенням отриманих результатів в пояснювальну записку до курсового проекту.

Таке виконання передбачає створення SQL-бази даних, створення таблиці SQL-бази даних, наповнення таблиці SQL-бази даних даними певного типу згідно з варіантом, виведення вмісту SQL-бази даних і так далі. Воно також передбачає поступове підвищення складності виконуваних завдань: від найбільш простого (створення SQL-бази даних) до найбільш складного (створення курсорів).

2. ЗАВДАННЯ НА КУРСОВИЙ ПРОЕКТ

1. Визначити (згідно з варіантом) тип даних, який повинна утримувати SQL-база даних (див. табл. 2.1).

Таблиця 2.1. Тип даних, який повинна утримувати SQL-база даних

№ з/п	Тип даних
1	2
1	Інструменти, які продаються в музичному магазині
2	Автомобілі, які ремонтуються на станції технічного обслуговування
3	Об'єми споживання теплової енергії багатоквартирним будинком
4	Заробітна плата співробітників охоронної фірми
5	Читачі бібліотеки
6	Штрафи за порушення правил дорожнього руху
7	Асортимент магазину іграшок
8	Продажі на аукціоні
9	Експонати музею
10	Учасники конференції
11	Організація по обслуговуванню ліфтів
12	Книжковий фонд читальної зали
13	Продукція консервного заводу
14	Абоненти телефонної станції
15	Товари, які зберігаються на складі
16	Служба доставки піци
17	Репертуар кінотеатру
18	Аудиторний фонд університету
19	Замовлення в Internet-магазині
20	Успішність навчання студентів

2. Створити SQL-базу даних, використовуючи Microsoft SQL Server Management Studio (передбачається, що буде використовуватися тільки одна SQL-база даних).

3. Визначити, згідно з визначеним типом даних, стовпці (атрибути), які повинна утримувати таблиця SQL-бази даних (передбачається, що буде вико-

ристовуватися тільки одна таблиця). Кількість стовпців (атрибутів) повинна бути не менше 10.

4. Створити таблицю SQL-бази даних з визначеними стовпцями (атрибути), використовуючи Microsoft SQL Server Management Studio.

5. Заповнити, згідно з визначеним типом даних, таблицю SQL-бази даних, використовуючи SQL Server Management Studio. Кількість рядків (записів) повинна бути не менше 50.

6. Вивести уміст всієї таблиці SQL-бази даних, використовуючи Microsoft SQL Server Management Studio.

Результат виводу занести в пояснювальну записку.

7. Вивести уміст таблиці SQL-бази даних, застосовуючи від 3 до 5 різних умов відбору стовпців (атрибутів) і/або рядків (записів) і використовуючи Microsoft SQL Server Management Studio.

Результат виводу занести в пояснювальну записку.

8. Вивести уміст таблиці SQL-бази даних в окрему таблицю SQL-бази даних, застосовуючи від 3 до 5 різних умов відбору рядків (записів) і використовуючи Microsoft SQL Server Management Studio.

Результат виводу занести в пояснювальну записку.

9. Виконати від 3 до 5 різних розрахунків над рядками (записами) таблиці SQL-бази даних і вивести їх, використовуючи Microsoft SQL Server Management Studio.

Результат виводу занести в пояснювальну записку.

10. Виконати від 3 до 5 різних розрахунків над рядками (записами) таблиці SQL-бази даних і вивести їх, використовуючи функції, які визначаються користувачем, і Microsoft SQL Server Management Studio.

Результат виводу занести в пояснювальну записку.

11. Виконати від 3 до 5 різних розрахунків над рядками (записами) таблиці SQL-бази даних і вивести їх, використовуючи збережені процедури, які визначаються користувачем, і Microsoft SQL Server Management Studio.

Результат виводу занести в пояснювальну записку.

12. Створити від 3 до 5 різних представлень і вивести їх, використовуючи Microsoft SQL Server Management Studio.

Результат виводу занести в пояснювальну записку.

13. Створити від 3 до 5 різних курсорів і вивести їх, використовуючи Microsoft SQL Server Management Studio.

Результат виводу занести в пояснювальну записку.

3. ВИМОГИ ДО СТРУКТУРИ І ПРАВИЛ ОФОРМЛЕННЯ КУРСОВОГО ПРОЕКТУ

Структура і правила оформлення курсового проекту з дисципліни “Організація баз даних” повинні відповідати [1д].

Курсовий проект повинен містити наступні структурні елементи:

- Титульний аркуш;
- Завдання на курсовий проект;
- Вступ;
- SQL-запити на мові Transact-SQL;
- Висновки;
- Бібліографічні посилання;
- Додатки (не обов’язково).

Курсовий проект повинен бути надрукованим на принтері шрифтом Times New Roman чорного кольору прямого накреслення через 1,25 міжрядкові інтервали кеглем 14 на одному боці аркушів білого паперу формату А4 (210 мм×297 мм).

Титульний аркуш є обов’язковим елементом курсового проекту, він повинен містити інформацію про наступне:

- де виконано курсовий проект;
- з якої дисципліни виконано курсовий проект;
- на яку тему виконано курсовий проект;
- варіант завдання на курсовий проект;
- хто виконав курсовий проект;
- хто перевірів курсовий проект;
- на яку оцінку було захищено курсовий проект;
- коли було захищено курсовий проект.

Титульний аркуш повинен мати приблизний вигляд, як наведено в Додатку А.

Завдання на курсовий проект є обов’язковим елементом курсового проекту, воно повинно містити інформацію про наступне:

- формулювання того, що треба виконати в курсовому проекті і на чому необхідно зосередити увагу;
- вихідні дані до курсового проекту (див. табл. 2.1).

Об’єм завдання на курсовий проект: 1 сторінка.

Вступ є обов’язковим елементом курсового проекту, він повинен містити

інформацію про наступне:

- сучасний стан організації SQL-баз даних;
 - необхідність використання SQL-баз даних в різних областях діяльності людини (в тому числі в різних технічних системах);
 - необхідність створення програмного забезпечення для SQL-баз даних;
 - основні вимоги до якості програмного забезпечення для SQL-баз даних.
- Об'єм вступу: 1 або 2 сторінки.

SQL-запити на мові Transact-SQL є обов'язковим елементом курсового проекту, вони повинні містити інформацію про наступне:

- послідовність дій при виконанні SQL-запитів на мові Transact-SQL;
- текст SQL-запитів на мові Transact-SQL;
- результати виконання SQL-запитів на мові Transact-SQL;
- пояснення результатів виконання SQL-запитів на мові Transact-SQL.

Об'єм SQL-запитів на мові Transact-SQL: від 20 до 40 сторінок.

Висновки є обов'язковим елементом курсового проекту, вони повинні містити інформацію про наступне:

- яке завдання на курсовий проект було поставлене;
- які основні результати були отримані під час виконання курсового проекту;
- чи є шляхи покращення отриманих під час виконання курсового проекту результатів.

Об'єм висновків: 1 сторінка.

Бібліографічні посилання є обов'язковим елементом курсового проекту (їх структура і правила оформлення повинні відповідати [2д]), вони повинні містити інформацію про наступне:

- бібліографічні посилання на основну літературу, яка використовувалася при виконанні курсового проекту;
- бібліографічні посилання на додаткову літературу, яка використовувалася при виконанні курсового проекту.

Об'єм бібліографічних посилань: 1 або 2 сторінки.

Додатки є необов'язковим елементом курсового проекту, при їх наявності вони можуть містити будь-яку інформацію, яка є необхідною для повного розуміння умісту курсового проекту.

Об'єм додатків (при їх наявності): від 1 до 10 сторінок.

4. ПРИКЛАД ВИКОНАННЯ КУРСОВОГО ПРОЕКТУ

1. Визначаємо тип даних, який повинна утримувати SQL-база даних.

Припустимо, що в SQL-базі даних будуть дані агентства, яке займається рекламною діяльністю. Робота такого агентства полягає в тому, що воно здає в оренду щити, які розташовані по всій території населеного пункту, в оренду. Умови здачі в оренду (термін здачі, уміст рекламного матеріалу, вартість здачі, місце розташування щита і так далі) визначаються договором між агентством і орендарем. Робота агентства контролюється керівництвом такого населеного пункту.

2. Створюємо SQL-базу даних, використовуючи Microsoft SQL Server Management Studio.

Натискаємо правою кнопкою миші на вузлі **Databases** і обираємо в контекстному меню **New Database....** Після цього з'явиться вікно **New Database**. В цьому вікні необхідно виконати наступні дії.

У полі **Database name** задати ідентифікатор нової SQL-бази даних (наприклад, це може бути **Agency**).

У випадаючому списку **Owner** обрати **<default>**.

У таблиці **Database files** у полях **Logical Name** задати логічні ідентифікатори файлів, в яких буде зберігатися SQL-база даних (наприклад, це може бути "Agency" і "Agency_log").

У таблиці **Database files** у полях **Initial Size (MB)** задати початковий розмір файлів, в яких буде зберігатися SQL-база даних (наприклад, це може бути 20 Мбайт і 5 Мбайт).

У таблиці **Database files** у полях **Autogrowth / Maxsize** задати крок збільшення розміру і максимальний розмір файлів, в яких буде зберігатися SQL-база даних (наприклад, це може бути 4 Мбайт /100 Мбайт і 1 Мбайт /100 Мбайт).

У таблиці **Database files** у полях **Path** задати каталоги, в яких будуть зберігатися файли SQL-бази даних (наприклад, це може бути "C:\Agency" і "C:\Agency").

У таблиці **Database files** у полях **FileName** задати імена файлів, в яких буде зберігатися SQL-база даних (наприклад, це може бути "Agency.mdf" і "Agency_log.ldf").

Якщо SQL-база даних створена неправильно, можна натиснути правою кнопкою миші на вузлі **Agency** створеної SQL-бази даних і обрати в контекст-

ному меню **Delete**. Далі з'явиться вікно **Delete Object**. В цьому вікні необхідно підтвердити видалення SQL-бази даних. Після цього треба повторити попередні пункти по створенню нової SQL-бази даних.

3. *Визначаємо, згідно з визначеним типом даних, стовпці (атрибути), які повинна утримувати таблиця SQL-бази даних.*

Нехай таблиця SQL-бази даних утримує стовпці (атрибути), які наведені в табл. 4.1.

Таблиця 4.1. **Стовпці (атрибути), які повинна утримувати таблиця SQL-бази даних рекламного агентства**

№ з/п	Стовпець	Ідентифікатор стовпця
1	2	3
1	Номер щита	Billboard_Number
2	Ширина щита, м	Billboard_Width
3	Висота щита, м	Billboard_Height
4	Площа щита, кв. м	Billboard_Area
5	Район міста, де розташований щит	City_District
6	Дата початку оренди, дд.мм.рррр	Rental_Start
7	Дата закінчення оренди, дд.мм.рррр	Rental_End
8	Тривалість оренди, дддд	Rental_Duration
9	Організація-орендатор	Renter_Organization
10	Керівник організації-орендатора, ПІБ	Renter_Director
11	Телефон керівника організації-орендатора	Renter_Director_Phone
12	Номер договору оренди	Rental_Agreement_Number
13	Вартість оренди 1 кв. м площі щита за день, грн.	Rental_Unit_Cost_Day
14	Вартість оренди щита за день, грн.	Rental_Cost_Day
15	Вартість загальна оренди щита, грн.	Rental_General_Cost
16	Коментар	Comment

4. *Створюємо таблицю SQL-бази даних з визначеними стовпцями (атрибутами), використовуючи Microsoft SQL Server Management Studio.*

Розкриваємо вузол **Agency** створеної SQL-бази даних. Натискаємо правою кнопкою миші на вузлі **Tables** і обираємо в контекстному меню **New** –

Table....

Після цього з'явиться вікно дизайнера таблиць. В цьому вікні необхідно виконати наступні дії.

У полі **Column Name** увести **Billboard_Number**, у полі **Data Type** увести **smallint**, скинути прапорець **Allow Nulls**.

У полі **Column Name** увести **Billboard_Width**, у полі **Data Type** увести **float**, скинути прапорець **Allow Nulls**.

У полі **Column Name** увести **Billboard_Height**, у полі **Data Type** увести **float**, скинути прапорець **Allow Nulls**.

У полі **Column Name** увести **Billboard_Area**, у полі **Data Type** увести **float**, встановити прапорець **Allow Nulls**.

У полі **Column Name** увести **City_District**, у полі **Data Type** увести **nchar(25)**, скинути прапорець **Allow Nulls**.

У полі **Column Name** увести **Rental_Start**, у полі **Data Type** увести **date**, скинути прапорець **Allow Nulls**.

У полі **Column Name** увести **Rental_End**, у полі **Data Type** увести **date**, скинути прапорець **Allow Nulls**.

У полі **Column Name** увести **Rental_Duration**, у полі **Data Type** увести **int**, встановити прапорець **Allow Nulls**.

У полі **Column Name** увести **Renter_Organization**, у полі **Data Type** увести **nchar(50)**, скинути прапорець **Allow Nulls**.

У полі **Column Name** увести **Renter_Director**, у полі **Data Type** увести **nchar(80)**, скинути прапорець **Allow Nulls**.

У полі **Column Name** увести **Renter_Director_Phone**, у полі **Data Type** увести **nchar(12)**, скинути прапорець **Allow Nulls**.

У полі **Column Name** увести **Rental_Agreement_Number**, у полі **Data Type** увести **smallint**, скинути прапорець **Allow Nulls**.

У полі **Column Name** увести **Rental_Unit_Cost_Day**, у полі **Data Type** увести **money**, скинути прапорець **Allow Nulls**.

У полі **Column Name** увести **Rental_Cost_Day**, у полі **Data Type** увести **money**, встановити прапорець **Allow Nulls**.

У полі **Column Name** увести **Rental_General_Cost**, у полі **Data Type** увести **money**, встановити прапорець **Allow Nulls**.

У полі **Column Name** увести **Comment**, у полі **Data Type** увести **nchar(150)**, встановити прапорець **Allow Nulls**.

На рядку (запису), поле **Column Name** якого містить **Billboard_Number**,

натиснути правою кнопкою миші і обрати в контекстному меню **Set Primary Key**.

Закрити вікно дизайнера таблиць, при цьому в процесі закриття з'явиться вікно **Choose Name**. В цьому вікні задати ідентифікатор таблиці (наприклад, це може бути **Billboards**) і натиснути кнопку **OK**.

Якщо якийсь із стовпців створений неправильно, можна розкрити вузол **Agency** створеної SQL-бази даних, розкрити вузол **Tables**, натиснути правою кнопкою миші на вузлі **dbo.Billboards** створеної таблиці і обрати в контекстному меню **Design**. Після цього з'явиться вікно дизайнера таблиць.

В цьому вікні можна обрати потрібний запис, натиснути правою кнопкою миші на ньому і обрати в контекстному меню **Delete Column** – таким чином, потрібний запис у вікні дизайнера таблиць буде видалено. Крім того, в цьому вікні можна обрати певний запис, натиснути правою кнопкою миші на ньому і обрати в контекстному меню **Insert Column** – таким чином, потрібний запис у вікні дизайнера таблиць буде вставлено.

Такі дії вставки і видалення стовпців таблиці (кожного разу з новими даними) можна повторювати багаторазово.

Для оновлення вмісту вікна дизайнера таблиць необхідно час від часу натискати кнопку **Refresh**.

5. Заповнюємо, згідно з визначеним типом даних, таблицю SQL-бази даних, використовуючи Microsoft SQL Server Management Studio.

Натискаємо правою кнопкою миші на вузлі **dbo.Billboards** створеної таблиці і обираємо в контекстному меню **Edit Top 200 Rows**. Після цього з'явиться вікно редактора таблиць. В цьому вікні необхідно заповнити записи довільними даними згідно з табл. 4.1.

6. Виводимо вміст всієї таблиці SQL-бази даних, використовуючи Microsoft SQL Server Management Studio.

Натискаємо правою кнопкою миші на вузлі **Agency** створеної SQL-бази даних і обираємо в контекстному меню **New Query**. Після цього з'явиться вікно дизайнера запитів. В цьому вікні необхідно виконати наступний SQL-запит:

SELECT * FROM Billboards ;

Результати такого SQL-запиту наведені на рис. 4.1, рис. 4.2 і рис. 4.3.

	Billboard_Number	Billboard_Width	Billboard_Height	Billboard_Area	City_District	Rental_Start
1	1	2,4	2	NULL	Дніпровський	2014-05-11
2	2	2,8	2,2	NULL	Подільський	2015-12-16
3	3	2,6	2,4	NULL	Дарницький	2007-06-15
4	4	2,8	2	NULL	Подільський	2010-02-04
5	5	2,4	2,8	NULL	Голосіївський	2017-11-28
6	6	3	2,2	NULL	Оболонський	2005-03-27
7	7	2,6	2,4	NULL	Святошинський	2011-08-30
8	8	3,2	2	NULL	Деснянський	2011-01-03
9	9	3	2,8	NULL	Печерський	2019-08-23
10	10	2,8	2,6	NULL	Солом'янський	2016-08-01
11	11	3,2	2,2	NULL	Солом'янський	2012-11-09
12	12	3,2	1,8	NULL	Дарницький	2008-02-10
13	13	2,4	2	NULL	Подільський	2006-07-29
14	14	3	1,8	NULL	Святошинський	2019-09-02
15	15	2,6	1,8	NULL	Голосіївський	2008-09-24
16	16	3	1,8	NULL	Солом'янський	2005-04-08
17	17	3	1,8	NULL	Деснянський	2014-04-14
18	18	3,2	2,2	NULL	Деснянський	2018-04-13
19	19	3	2	NULL	Шевченківськ...	2007-12-17
20	20	2,4	2,6	NULL	Оболонський	2012-02-20
21	21	2,8	2,8	NULL	Дарницький	2016-07-25
22	22	2,8	2,6	NULL	Печерський	2013-05-19
23	23	2,6	2,2	NULL	Шевченківськ...	2013-10-26
24	24	3,2	2,4	NULL	Шевченківськ...	2017-10-18
25	25	3	2,4	NULL	Дніпровський	2015-12-05
26	26	2,4	2,4	NULL	Святошинський	2006-03-12
27	27	3	2,6	NULL	Печерський	2010-01-07
28	28	2,6	2,8	NULL	Дніпровський	2019-06-21
29	29	3	2,6	NULL	Оболонський	2009-10-06
30	30	3	2,8	NULL	Голосіївський	2009-06-22

Рисунок 4.1. Результати SQL-запиту (частина 1)

Rental_End	Rental_Duration	Renter_Organization	Renter_Director	Renter_Director_Phone
2023-04-30	NULL	АТ "Тонас"	Боролюк І.К.	380987687933
2023-12-10	NULL	ТОВ "М'ясні вироби"	Маркін К.Д.	380962958949
2022-09-19	NULL	АТ "Дача, город"	Терновський А.А.	380953773260
2022-11-22	NULL	ТОВ "Спортивне взуття"	Федорчук З.А.	380670921575
2022-02-21	NULL	ТОВ "Все для спорту"	Житній О.О.	380685039068
2024-01-20	NULL	ТОВ "Якісні годинники"	Царик Г.О.	380631053903
2024-06-11	NULL	ТОВ "Смачні обіди"	Заєць Б.М.	380977878142
2024-01-01	NULL	АТ "Автозапчастини"	Котов А.М.	380929606148
2022-07-12	NULL	АТ "Агробуд"	Ульянченко Д.О.	380937433893
2024-03-07	NULL	АТ "Дім меблів"	Нікітін В.Ж.	380947547009
2023-10-29	NULL	ТОВ "Керамограніт"	Петренко Я.Ш.	380390276214
2024-12-17	NULL	ТОВ "Будівельник"	Сидоренко Ю.С.	380688599091
2024-06-18	NULL	ТОВ "Радіоресурс"	Іжакевич К.Г.	380672281452
2023-04-02	NULL	ТОВ "Доставка піци"	Естов П.Т.	380929941409
2022-08-23	NULL	ТОВ "Промзбут"	Харченко Р.В.	380996031204
2023-10-24	NULL	АТ "Космос"	Щербина В.П.	380945826713
2022-05-05	NULL	АТ "Посуд для дому"	Власенко Г.І.	380683663876
2022-02-13	NULL	АТ "Ремонтник"	Юревич С.У.	380910185044
2022-08-25	NULL	ТОВ "Технічні прилади"	Яровий О.Л.	380913449451
2023-11-14	NULL	ТОВ "Мобільні телефо...	Горбатов В.В.	380989638851
2024-05-06	NULL	ТОВ "Все для саду"	Літовченко Н.Ф.	380634478939
2024-11-27	NULL	АТ "Обладнання для к...	Шишов О.А.	380975833882
2022-03-26	NULL	АТ "Сигналізація, охор...	Данілов М.Ю.	380680167687
2024-07-28	NULL	АТ "Продукти харчува...	Чернов Ф.Ю.	380933110899
2023-01-08	NULL	ТОВ "Вантажні переве...	Іволгін Л.Д.	380958014642
2023-09-03	NULL	АТ "Спецмонтаж"	Йосін А.Ф.	380392120244
2023-03-15	NULL	АТ "Рибалка"	Єремєєв К.Г.	380990269333
2022-07-09	NULL	ТОВ "Послуги страхув...	Омельченко Я.П.	380507203011
2024-09-04	NULL	АТ "Шпалери"	Андрійчук Д.М.	380504355503
2023-05-16	NULL	АТ "Сучасні комп'ютери"	Радостін С.Б.	380982691305

Рисунок 4.2. Результати SQL-запиту (частина 2)

Rental_Agreement_Number	Rental_Unit_Cost_Day	Rental_Cost_Day	Rental_General_Cost	Comment
19	25,00	NULL	NULL	NULL
18	35,00	NULL	NULL	NULL
4	35,00	NULL	NULL	NULL
12	30,00	NULL	NULL	NULL
10	40,00	NULL	NULL	NULL
3	30,00	NULL	NULL	NULL
9	30,00	NULL	NULL	NULL
17	25,00	NULL	NULL	NULL
20	40,00	NULL	NULL	NULL
2	30,00	NULL	NULL	NULL
16	20,00	NULL	NULL	NULL
25	20,00	NULL	NULL	NULL
30	40,00	NULL	NULL	NULL
6	25,00	NULL	NULL	NULL
15	25,00	NULL	NULL	NULL
21	40,00	NULL	NULL	NULL
8	40,00	NULL	NULL	NULL
28	20,00	NULL	NULL	NULL
11	25,00	NULL	NULL	NULL
29	35,00	NULL	NULL	NULL
24	30,00	NULL	NULL	NULL
5	35,00	NULL	NULL	NULL
27	20,00	NULL	NULL	NULL
26	20,00	NULL	NULL	NULL
1	40,00	NULL	NULL	NULL
13	35,00	NULL	NULL	NULL
22	25,00	NULL	NULL	NULL
7	30,00	NULL	NULL	NULL
23	20,00	NULL	NULL	NULL
14	35,00	NULL	NULL	NULL

Рисунок 4.3. Результати SQL-запиту (частина 3)

7. Виводимо уміст таблиці SQL-бази даних, застосовуючи 3 різні умови відбору стовпців (атрибутів) і/або рядків (записів) і використовуючи Microsoft SQL Server Management Studio.

Виконаємо 1-й SQL-запит:

**SELECT City_District, Renter_Organization, Renter_Director,
Renter_Director_Phone, Rental_Agreement_Number FROM Billboards ;**

Таким чином, будуть виведені всі рядки (записи), але тільки з даними про район міста, де розташований щит, організацію-орендатора, керівника організації-орендатора, телефон керівника організації-орендатора і номер договору оренди.

Виконаємо 2-й SQL-запит:

**SELECT * FROM Billboards WHERE (10 <= Billboard_Number) AND
(Billboard_Number <= 20) ;**

Таким чином, будуть виведені всі рядки (записи), для яких номер щита

одночасно більше 10 (включно) і менше 20 (включно).

Виконаємо 3-й SQL-запит:

SELECT TOP(5) * FROM Billboards ORDER BY Rental_Start ASC ;

Таким чином, будуть виведені 5 перших рядків (записів), відсортовані по зростанню дати початку оренди.

8. Виводимо уміст таблиці SQL-бази даних в окрему таблицю SQL-бази даних, застосовуючи 3 різні умови відбору рядків (записів) і використовуючи Microsoft SQL Server Management Studio.

Виконаємо 1-й SQL-запит:

**IF OBJECT_ID('BillboardsWidths') IS NOT NULL DROP
TABLE BillboardsWidths ;
GO
SELECT * INTO BillboardsWidths FROM Billboards WHERE
Billboard_Width = 2.8 ;**

Таким чином, будуть виведені всі рядки (записи), для яких ширина щита дорівнює 2,8 м, в таблицю **BillboardsWidths**.

Виконаємо 2-й SQL-запит:

**IF OBJECT_ID('RentalAgreementNumber') IS NOT NULL DROP
TABLE RentalAgreementNumber ;
GO
SELECT * INTO RentalAgreementNumber FROM Billboards WHERE
(Rental_Agreement_Number <= 6) OR
(22 <= Rental_Agreement_Number) ;**

Таким чином, будуть виведені всі рядки (записи), для яких номер договору оренди або менше 16 (включно), або більше 22 (включно), в таблицю **RentalAgreementNumber**.

Виконаємо 3-й SQL-запит:

IF OBJECT_ID('RenterDirectorPhone') IS NOT NULL DROP

```

TABLE RenterDirectorPhone ;
GO
SELECT * INTO RenterDirectorPhone FROM Billboards WHERE
SUBSTRING(Renter_Director_Phone, 3, 3) = '067' ;

```

Таким чином, будуть виведені всі рядки (записи), для яких код мобільного оператора телефону керівника організації-орендатора дорівнює 067, в таблицю **RenterDirectorPhone**.

9. Виконуємо 2 різні розрахунки над рядками (записами) таблиці SQL-бази даних і виводимо їх, використовуючи Microsoft SQL Server Management Studio.

Виконаємо 1-й SQL-запит:

```

UPDATE Billboards
SET Billboard_Area = Billboard_Width * Billboard_Height ;

```

Таким чином, будуть визначені площі щитів згідно з формулою

$$\text{Billboard_Area} = \text{Billboard_Width} * \text{Billboard_Height} \quad (4.1)$$

Виконаємо 2-й SQL-запит:

```

UPDATE Billboards
SET Rental_Cost_Day = Rental_Unit_Cost_Day * Billboard_Area ;

```

Таким чином, будуть визначені вартості оренди щитів за день згідно з формулою

$$\text{Rental_Cost_Day} = \text{Rental_Unit_Cost_Day} * \text{Billboard_Area} \quad (4.2)$$

10. Виконуємо 2 різні розрахунки над рядками (записами) таблиці SQL-бази даних і виводимо їх, використовуючи функції, які визначаються користувачем, і Microsoft SQL Server Management Studio.

Виконаємо 1-й SQL-запит:

```

CREATE FUNCTION F1
(
@_Rental_Start_date,

```

```

@_Rental_End_date
)
RETURNS int
AS
BEGIN
DECLARE @_Rental_Duration_ int ;
SET @_Rental_Duration_ = DATEDIFF (day, @_Rental_End_,
@_Rental_Start_) ;
RETURN (@_Rental_Duration_) ;
END ;
GO
UPDATE Billboards
SET Rental_Duration = dbo.F1(Billboards.Rental_End,
Billboards.Rental_Start) ;

```

Таким чином, будуть визначені тривалості оренди згідно з формулою

$$\text{Rental_Duration} = \text{Rental_End} - \text{Rental_Start} \quad (4.3)$$

Виконаємо 2-й SQL-запит:

```

CREATE FUNCTION F2
(
@_Rental_Cost_Day_money,
@_Rental_Duration_int
)
RETURNS money
AS
BEGIN
DECLARE @_Rental_General_Cost_money;
SET @_Rental_General_Cost_ = @_Rental_Cost_Day_ *
@_Rental_Duration_ ;
RETURN (@_Rental_General_Cost_) ;
END ;
GO
UPDATE Billboards
SET Rental_General_Cost = dbo.F2(Billboards.Rental_Cost_Day,

```

Billboards.Rental_Duration) ;

Таким чином, будуть визначені вартості загальні оренд щитів згідно з формулою

$$\text{Rental_General_Cost} = \text{Rental_Cost_Day} * \text{Rental_Duration} \quad (4.4)$$

11. Виконуємо 2 різні розрахунки над рядками (записами) таблиці SQL-бази даних і виводимо їх, використовуючи збережені процедури, які визначаються користувачем, і Microsoft SQL Server Management Studio.

Виконаємо 1-й SQL-запит:

```
CREATE PROCEDURE P1
@_Billboard_Area_Average_ float OUTPUT
AS
BEGIN
SELECT @_Billboard_Area_Average_ =
AVG(Billboard_Area) FROM Billboards;
END ;
GO
DECLARE @_Result_Avg_ float ;
EXECUTE dbo.P1 @_Result_Avg_ OUTPUT ;
SELECT @_Result_Avg_ AS RESULT ;
```

Таким чином, буде визначено і виведено середнє значення площі всіх щитів.

Виконаємо 2-й SQL-запит:

```
CREATE PROCEDURE P2
@_Rental_Duration_Min_ int OUTPUT,
@_Rental_Duration_Max_ int OUTPUT
AS
BEGIN
SELECT @_Rental_Duration_Min_ =
MIN(Rental_Duration) FROM Billboards;
SELECT @_Rental_Duration_Max_ =
```

```

MAX(Rental_Duration) FROM Billboards;
END ;
GO
DECLARE @_Result_Min_ float, @_Result_Max_ float ;
EXECUTE dbo.P2 @_Result_Min_ OUTPUT,
@_Result_Max_ OUTPUT ;
SELECT @_Result_Min_ AS RESULT ;
GO
SELECT @_Result_Max_ AS RESULT ;

```

Таким чином, будуть визначені і виведені мінімальне і максимальне значення тривалості оренди.

12. Створюємо 2 різних представлення і виводимо їх, використовуючи Microsoft SQL Server Management Studio.

Виконаємо 1-й SQL-запит:

```

CREATE VIEW V1
AS
SELECT * FROM Billboards WHERE City_District='Печерський' ;
GO
SELECT * FROM V1 ;

```

Таким чином, будуть виведені всі рядки (записи), які відповідають розташуванню щитів в Печерському районі міста.

Виконаємо 2-й SQL-запит:

```

CREATE VIEW V2
AS
SELECT * FROM Billboards WHERE
SUBSTRING(Renter_Organization, 1, 2) = 'AT' ;
GO
SELECT * FROM V2

```

Таким чином, будуть виведені всі рядки (записи), які відповідають організаціям-орендаторам, що представляють собою акціонерні товариства.

13. Створюємо 2 різних курсори і виводимо їх, використовуючи Microsoft SQL Server Management Studio.

Виконаємо 1-й SQL-запит:

```
DECLARE @_Billboard_Number_ smallint, @_Billboard_Width_ float,  
@_Billboard_Height_ float, @_Billboard_Area_ float ;  
DECLARE C1 CURSOR FOR  
SELECT Billboard_Number, Billboard_Width, Billboard_Height,  
Billboard_Area FROM Billboards WHERE YEAR(Rental_Start)>=2018 ;  
OPEN C1 ;  
FETCH NEXT FROM C1 INTO @_Billboard_Number_,  
@_Billboard_Width_, @_Billboard_Height_, @_Billboard_Area_ ;  
WHILE @@FETCH_STATUS = 0  
BEGIN  
SELECT @_Billboard_Number_ AS BN, @_Billboard_Width_ AS BW,  
@_Billboard_Height_ AS BH, @_Billboard_Area_ AS BA ;  
FETCH NEXT FROM C1 INTO @_Billboard_Number_,  
@_Billboard_Width_, @_Billboard_Height_, @_Billboard_Area_ ;  
END ;  
CLOSE C1 ;  
DEALLOCATE C1 ;
```

Таким чином, будуть виведені всі рядки (записи), для яких рік початку оренди більше або дорівнює 2018.

Виконаємо 2-й SQL-запит:

```
DECLARE @_Renter_Organization_ nchar(50),  
@_Renter_Director_ nchar(80) ;  
DECLARE C2 CURSOR FOR SELECT Renter_Organization,  
Renter_Director FROM Billboards WHERE  
SUBSTRING(Renter_Director, 1, 1) = 'A' OR  
SUBSTRING(Renter_Director, 1, 1) = 'B' OR  
SUBSTRING(Renter_Director, 1, 1) = 'B' OR  
SUBSTRING(Renter_Director, 1, 1) = 'T' ;  
OPEN C2 ;  
FETCH NEXT FROM C2 INTO @_Renter_Organization_,
```

```
@_Renter_Director_ ;  
WHILE @@FETCH_STATUS = 0  
BEGIN  
SELECT @_Renter_Organization_ AS RO, @_Renter_Director_ AS RD ;  
FETCH NEXT FROM C2 INTO @_Renter_Organization_,  
@_Renter_Director_ ;  
END ;  
CLOSE C2 ;  
DEALLOCATE C2 ;
```

Таким чином, будуть виведені всі рядки (записи), для яких прізвище керівника організації-орендатора починається з символів “А”, “Б”, “В” або “Г”.

БІБЛІОГРАФІЧНІ ПОСИЛАННЯ

Основна література

1. Гайдаржи В.І. Бази даних в інформаційних системах / В.І. Гайдаржи, І.В. Ізварін. – К.: Університет “Україна”, 2018. – 418 с. – ISBN 978–966–388–569–8.
2. Пасічник В.В. Сховища даних. Підручник / В.В. Пасічник, Н.Б. Шаховська. – Львів: Магнолія, 2021. – 496 с. – ISBN 978–966–2025–18–7.

Додаткова література

1. ДСТУ 3008:2015 Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлення.
2. ДСТУ 8302:2015 Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання.

ДОДАТОК А.
ЗОВНІШНІЙ ВИГЛЯД ТИТУЛЬНОГО АРКУШУ

Міністерство освіти і науки України
Таврійський національний університет імені В.І. Вернадського
Навчально-науковий інститут муніципального управління та міського господарства
Кафедра інженерних систем та технологій

КУРСОВИЙ ПРОЕКТ

з дисципліни “Організація баз даних”
на тему “Організація доступу до даних за допомогою SQL-баз даних”

Варіант № ____

Виконав: _____

Перевірив: _____

Оцінка: _____

Дата: _____

м. Київ – 2024