

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА**  
**ФАКУЛЬТЕТ УПРАВЛІННЯ ФІНАНСАМИ ТА БІЗНЕСУ**

**Кафедра цифрової економіки та бізнес-аналітики**

**КУРСОВА РОБОТА**  
**з навчальної дисципліни**  
**«Проектування та адміністрування БД і СД»**

на тему:

**Інформаційна система для будівельного супермаркету**

**Галузь знань:** \_\_\_\_\_ **05 «Соціальні та поведінкові науки»** \_\_\_\_\_

**Спеціальність:** \_\_\_\_\_ **051 «Економіка»** \_\_\_\_\_

**Спеціалізація:** \_\_\_\_\_ **«Інформаційні технології в бізнесі»** \_\_\_\_\_

**Освітній ступінь:** \_\_\_\_\_ **бакалавр** \_\_\_\_\_

**Науковий керівник:**

к.ф.-м.н., доц. Депутат Б.Я

(науковий ступінь, посада, прізвище, ініціали)

\_\_\_\_\_ “\_\_\_” травня 2020 р.

(підпис)

**Виконавець:**

студент групи УФЕ-31 с

Медвідь О.І.

\_\_\_\_\_ “\_\_\_” травня 2020 р.

(підпис)

**Загальна кількість балів** \_\_\_\_\_

\_\_\_\_\_  
(підпис, ПП членів комісії)

**ЛЬВІВ 2020**

## ЗМІСТ

<b>ВСТУП .....</b>	<b>3</b>
<b>РОЗДІЛ 1. СУТНІСТЬ БАЗ ДАНИХ І ДЖЕРЕЛА ЇХ ФОРМУВАННЯ .....</b>	<b>6</b>
<b>РОЗДІЛ 2. РОЗРОБКА РЕЛЯЦІЙНОЇ БАЗИ ДАНИХ .....</b>	<b>10</b>
<b>2.1 Проектування бази даних .....</b>	<b>10</b>
<b>2.2 Архітектура бази даних SQL Server .....</b>	<b>13</b>
<b>2.3 База даних MySQL .....</b>	<b>17</b>
<b>2.4 Таблиці баз даних .....</b>	<b>19</b>
<b>2.5 Перелік таблиць бази даних build_supermarket.....</b>	<b>21</b>
<b>2.6 Перелік полів, які ідентифікують кожну з таблиць створеної БД.....</b>	<b>21</b>
<b>2.7 Запити, протестовані на розробленій базі даних .....</b>	<b>23</b>
<b>ВИСНОВОК .....</b>	<b>25</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>26</b>
<b>ДОДАТОК .....</b>	<b>27</b>

## ВСТУП

Інформаційна система — це сукупність різноманітних взаємопов'язаних або взаємозалежних усебічних відомостей про стан об'єкта управління та процеси, що відбуваються на ньому і які виражені в показниках і інших інформаційних сукупностях, зібраних та оброблених за допомогою технічних (інформаційних і обчислювальних) засобів за визначеною методикою та заданих алгоритмах, і які відповідають вимогам керівної системи при її впливі на керовану.

Для розв'язання проблеми спільного використання інформаційної системи різними користувачами створюється інформаційне забезпечення.

Під інформаційним забезпеченням розуміють сукупність форм документів різних видів призначення, нормативної бази та реалізованих рішень щодо обсягів, розміщення і форм існування інформації, яка використовується в інформаційній системі під час її функціонування на об'єкті управління.

Основні вимоги до інформаційного забезпечення такі:

- інформаційне забезпечення має бути достатнім для використання всіх функцій інформаційної системи, які автоматизуються;
- для кодування інформації, що використовується як на об'єкті управління, так і на вищому рівні, необхідно використовувати погоджені класифікатори, які в них є
- інформаційне забезпечення даної інформаційної системи має бути поєднане з інформаційним забезпеченням інших систем, з якими воно взаємодіє;
- форми документів і відеокadrів, які вводяться системою, мають відповідати вимогам стандартів, технічним характеристикам терміналів, а також погоджені з замовником;
- сукупність інформаційних масивів організовується у вигляді бази даних.

Загальноприйнятої класифікації інформаційних систем поки що не існує.

Проте їх можна класифікувати за різними ознаками:

- за рівнем або сферою діяльності — державні; територіальні або регіональні; міжгалузеві; галузеві або відомчі; об'єднань, фірм, підприємств або установ; технологічних процесів тощо;

- за рівнем автоматизації процесів управління — автоматизованого збирання й обробки інформації; інформаційно-пошукові; інформаційно-довідкові; об'єктивного інформаційного забезпечення менеджменту (керівників і фахівців); системи забезпечення прийняття рішень або підтримки прийняття рішень; інтелектуальні системи;

- за ступенем централізації обробки інформації — централізовані, децентралізовані та змішані інформаційні системи, інформаційні системи колективного використання;

- за ступенем інтеграції функцій — багаторівневі з інтеграцією за рівнями управління (підприємство — об'єднання, об'єднання — галузь і т.ін.), багаторівневі з інтеграцією за рівнями планування; однорівневі інформаційні системи.

Інформаційна система, що характеризує внутрішній стан і процеси, які відбуваються на об'єкті управління, та його зовнішнє середовище, а також є моделлю цього об'єкта — це система автоматизованого збирання й обробки інформації. Вона є основою для організації системи об'єктивного інформаційного забезпечення менеджменту.

**Мета і завдання дослідження :** розробити інформаційну систему взаємодії клієнта із будівельним супермаркетом.

Для досягнення мети була розроблена база даних і проведена вся необхідна з нею робота.

**Об'єктом дослідження** є інформаційна система будівельного супермаркету.

**Предметом дослідження** є база даних, розроблена з метою створення інформаційної системи будівельного супермаркету.

**Практичне значення отриманих результатів** : отримана інформаційна система може використовуватись для наповнення даними веб-сайту будівельного супермаркету.

**Використане програмне забезпечення** : для розробки бази даних використовувалось програмне забезпечення MySQL і мова SQL.

## РОЗДІЛ 1. СУТНІСТЬ БАЗ ДАНИХ І ДЖЕРЕЛА ЇХ ФОРМУВАННЯ

Система об'єктивного інформаційного забезпечення менеджменту, як і будь-яка інша інформаційна система, включає як складову частину інформаційну базу.

Інформаційна база, яка є сукупністю певним чином організованої, збереженої та контрольованої інформації, зафіксованої на різних носіях, і що відображає стан і процеси, які відбуваються на об'єкті управління та його зовнішньому середовищі, включає дві частини: позамашинну інформаційну базу і машинну інформаційну базу.

Позамашинна інформаційна база — це перша (вхідна) частина інформаційної бази системи, яка являє собою сукупність організованої, збереженої та контрольованої достовірної і точної інформації (вона зафіксована на різних документах-носіях, що безпосередньо сприймаються людиною) і яка відображає стан і процеси, що відбуваються на об'єкті управління та в зовнішньому середовищі, що впливає на цей об'єкт. Така сукупність інформації призначена для формування машинної інформаційної бази.

Машинна інформаційна база — це друга частина інформаційної бази системи, що являє собою сукупність інформаційних масивів, сформованих на основі даних позамашинної інформаційної бази, які зберігаються на машинних (магнітних та ін.) носіях та в пам'яті ЕОМ.

Масив — це ідентифікована сукупність примірників (однорідних записів) логічно пов'язаних між собою даних, які містяться в зовнішній пам'яті (на магнітних та інших носіях чи в пам'яті ЕОМ) і є доступними для програми.

У системі об'єктивного інформаційного забезпечення менеджменту машинна інформаційна база, до складу якої входить повний набір первинних та інших масивів, являє собою інформаційну модель об'єкта управління.

Відомо, що донедавна при організації обробки інформації на ЕОМ

застосовувався підхід, за якого на основі інформації однієї і тієї самої предметної області формувалися масиви лінійної структури.

Недоліком таких масивів, які мають лінійну структуру, є те, що інформація одного й того самого об'єкта управління розосереджується поміж багатьма різними масивами (нормативними, плановими та ін.), що неминуче призводить до дублювання деяких реквізитів, ускладнення під час спільної їх обробки тощо, а головне — не дає змоги реалізувати принцип незалежності даних від прикладних програм користувача. Лінійні масиви, сформовані традиційним способом, ефективні, як правило, з позиції одного застосування.

База даних — це відповідним чином поійменована, структурована сукупність взаємопов'язаних даних, що характеризують окрему предметну область і перебувають під управлінням СУБД. При цьому дані зберігаються на машинних (магнітних та ін.) носіях, не залежать від прикладних програм і можуть використовуватися багатьма користувачами. Структура бази даних відповідає інформаційній моделі предметної області за станом на кожний даний момент.

Під предметною областю в даному разі розуміють інформаційний об'єкт з однорідною інформацією, яка моделюється за допомогою бази даних і використовується для розв'язування різних взаємопов'язаних задач, що належать переважно до цього об'єкта.

Система управління базою даних (СУБД) забезпечує автоматичне виконання основних функцій бази даних і включає комплекс програмних і певних засобів загального та спеціального призначення, які необхідні для створення та управління базою даних, підтримки її в актуальному стані, підтримки цілісності й захисту даних, маніпулювання даними й організації доступу до них різних користувачів чи прикладних програм в умовах чинної технології обробки даних.

В основі організації бази даних є модель логічного рівня, яка підтримується засобами конкретної СУБД і визначає правила, згідно з якими

структуруються дані. Це зовнішній рівень моделювання. За допомогою зазначеної моделі подається велика кількість даних і описуються взаємозв'язки між ними. Найпоширенішими є такі моделі даних: ієрархічна, сіткова, реляційна.

Ієрархічна модель даних будується на основі принципу підпорядкованості елементів даних і є деревоподібною структурою, що складається з вузлів (сегментів) і дуг (гілок). Дерево в ієрархічній структурі впорядковане за чинними правилами розміщення його сегментів і гілок: на горішньому рівні перебуває один сегмент — кореневий (вихідний); сегмент другого рівня — породжений — залежить від першого, вихідного; доступ до кожного породженого (крім кореневого) відбувається через його вихідний сегмент; кожен сегмент може мати по кілька примірників конкретних значень елементів даних, а кожен елемент породженого сегмента пов'язаний з примірником вихідного і створює один логічний запис; примірник породженого сегмента не може існувати самостійно, тобто без кореневого сегмента; при видаленні примірника кореневого сегмента також видаляються всі підпорядковані і взаємопов'язані з ним примірники породжених сегментів.

Сіткова модель даних являє собою орієнтований граф з поименованими вершинами та дугами. Вершини графа — записи, що є поименованою сукупністю логічних взаємопов'язаних елементів даних або агрегатів даних. Під агрегатом даних розуміють поименовану сукупність елементів даних, які є всередині запису. Для кожного типу записів може бути кілька примірників конкретних значень його інформаційних елементів. Два записи, взаємопов'язані дугою, створюють набір даних. Запис, з якого виходить дуга, називається власником набору, а запис, до якого вона спрямована, — членом набору.

Реляційна модель даних являє собою набір двовимірних пласких таблиць, що складаються з рядків і стовпців. Первинний документ або лінійний масив являє собою пласку двовимірну таблицю. Така таблиця називається відношенням, кожен стовпець — атрибутом, сукупність значень одного типу



(стовпця) — доменом, а рядка — кортежем. Отже, стовпці таблиці є традиційними елементами даних, а рядки — записами. Таблиці (відношення) мають імена. Імена присвоюються також і стовпцям таблиці. Кожний кортеж (запис) відношення має ключ. Ключі бувають прості та складні. Простий ключ — це ключ, який складається з одного атомарного атрибута, значення якого унікальне (не повторюється). Складний ключ складається з двох і більше атрибутів. Для зв'язків відношень одного з одним у базі даних є зовнішні ключі. Атрибут або комбінація атрибута відношення є зовнішнім ключем, якщо він не є основним (первинним) ключем цього відношення, але є первинним ключем для іншого відношення.

Внутрішній рівень пов'язаний з фізичним розміщенням даних у пам'яті ЕОМ. На цьому рівні формується фізична модель бази даних, яка містить структури зберігання даних у пам'яті ЕОМ і включає опис форматів записів, їхнє логічне чи фізичне впорядкування, розміщення за типами пристроїв, а також характеристики і шляхи доступу до даних. Запит оформляється за певною формою та охоплює назву даних, період часу, за який потрібні дані, а також структуру та зміст відео- або документограм.

Від параметрів фізичної моделі залежать такі характеристики бази даних: обсяг пам'яті та час реакції системи. Фізичні параметри бази даних можна змінювати в процесі її експлуатації (не змінюючи при цьому опису інших рівнів) з метою підвищення ефективності функціонування системи.

## РОЗДІЛ 2. РОЗРОБКА РЕЛЯЦІЙНОЇ БАЗИ ДАНИХ

### 2.1 Проектування бази даних

Створення бази даних слід починати з її проектування (розробки). У результаті проектування має бути визначена структура бази, тобто склад таблиць, їхня структура та логічні зв'язки. Структура реляційної таблиці визначається складом стовпців, їхньою послідовністю, типом даних кожного стовпця та їхнім розміром, а також ключем таблиці. Процес проектування можна здійснювати двома підходами. За першого підходу спочатку визначають основні задачі, для розв'язання яких створюється база, та потреби цих задач у даних. За другого підходу визначають предметну область (сферу), здійснюють аналіз її даних і встановлюють типові об'єкти предметної області. Найбільш раціональним підходом проектування бази даних є поєднання обох підходів.

Зазвичай з базами даних працюють дві категорії користувачів. Перша категорія - проектувальники. Процес проектування бази даних поділяється на етапи, кожний з яких передбачає виконання певних дій. Перший етап-розробка інформаційно-логічної моделі даних предметної області, який базується на описі предметної області, отриманому в результаті її обстеження. На цьому етапі спочатку визначають склад і структуру даних предметної області, які мають міститись у базі даних та забезпечувати виконання запитів, задач і застосувань користувача. Ці дані мають форму реквізитів, що містяться в різних документах - джерелах завантаження бази даних. Аналіз виявлених даних дозволить визначити функціональні залежності реквізитів, які використовують для виділення інформаційних об'єктів, що відповідають вимогам нормалізації даних. Подальше визначення структурних зв'язків між об'єктами дозволяє побудувати інформаційно-логічну модель.

Другий етап - визначення логічної структури бази даних. Для реляційної бази даних цей етап є значною мірою формальним, оскільки інформаційно-логічна модель відображається в структуру реляційної бази даних адекватно.

Наступний етап - конструювання таблиць бази даних, який здійснюється засобами СУБД, та узгодження їх із замовником. Структура таблиць бази даних задається за допомогою засобів опису (конструювання) таблиць у СУБД із цілковитою відповідністю інформаційним об'єктам. Крім таблиць, проектувальники розробляють й інші об'єкти бази даних, які призначені, з одного боку, для автоматизації роботи з базою, а з іншого - для обмеження функціональних можливостей роботи з базою (безпека бази даних). Після формування структури таблиць база даних може наповнюватись даними з документів-джерел. Проектувальники, як правило, не наповнюють базу конкретними даними (оскільки замовник може вважати дані конфіденційними і не надавати стороннім особам). Винятком є експериментальне наповнення модельними даними на етапі відлагодження об'єктів бази.

Друга категорія - користувачі, які працюють із базами даних. Вони отримують вхідну базу даних від проектувальників, наповнюють її та обслуговують. Користувачі не мають засобів доступу до управління структурою бази, вони мають доступ тільки до даних, при цьому тільки до тих, робота з якими передбачена на їхньому конкретному робочому місці.

Проектування бази даних починається з вивчення технічного завдання на проектування бази даних, яке повинен надати замовник. Отже, бажано, щоб замовник володів відповідною термінологією і знав, принаймні в загальних рисах, технічні можливості основних СУБД. На жаль, на практиці ці побажання виконуються не завжди. Тому зазвичай розробники використовують такі підходи: демонструють замовникові роботу аналогічної бази даних, після чого узгоджують специфікацію відмінностей; якщо аналога немає, з'ясовують коло задач і вимог замовника, після чого допомагають йому підготувати технічне

завдання. Під час підготовки технічного завдання складають: перелік вхідних даних, з якими працює замовник; перелік вихідних даних, потрібних замовникові для управління структурою свого підприємства; перелік вихідних даних, які не є необхідними для замовника, але які він повинен надати іншим організаціям (у вищеперераховані структури, в органи статистики, інші адміністративні і контрольні організації).

Визначивши основну частину даних, які замовник використовує, розпочинають розробку структури бази, тобто структури її основних таблиць.

1. Робота починається з визначення генерального переліку полів, який може нараховувати десятки і сотні позицій.
2. Відповідно до типу даних, що розміщуються в кожному полі, визначають тип кожного поля.
3. Розподіляють поля генерального списку по базових таблицях. На першому етапі розподіл здійснюють за функціональною ознакою. Мета - забезпечити одноразове введення даних в одну таблицю по можливості в рамках одного підрозділу, або (ще краще) - на одному робочому місці. На другому етапі розподілу полів здійснюють нормалізацію даних з метою вилучення повторів даних у таблицях бази даних.
4. Для кожної таблиці визначають ключове поле. Ключовим вибирають поле, дані в якому повторюватись не можуть. Наприклад, для таблиці даних про студентів таким полем може бути індивідуальний шифр студента. Для таблиць, у яких міститься розклад занять, такого поля можна і не знайти, але його можна створити штучно комбінуванням полів "Час заняття" і "Номер аудиторії". Ця комбінація унікальна, оскільки в певній аудиторії в певний час назагал не проводять двох різних занять. Якщо ж у таблиці взагалі немає полів, які можна було б використовувати як ключові, завжди можна ввести додаткове поле типу

лічильник - воно за визначенням не може містити дані, що повторюються.

5. На наступному етапі визначають зв'язки між таблицями (схему даних). Тобто на стороні "один" має бути ключове поле, яке не повторюється, значення на стороні "багато" можуть повторюватися.
6. "Паперовий" етап роботи над технічними пропозиціями закінчується розробкою схеми даних. Цю схему слід узгодити із замовником, після чого розпочати безпосереднє створення бази даних. Слід пам'ятати, що в ході розробки проекту замовникові неодмінно будуть надходити нові ідеї. Можливість гнучкого використання його побажань суттєво залежить від кваліфікації розробника бази даних. Якщо схема даних складена правильно, підключити до бази нові таблиці неважко. Якщо структура бази нераціональна, розробник може наштовхнутись на суттєві труднощі і дійти суперечності із замовником. Суперечка виконавця із замовником завжди свідчить про недостатню кваліфікацію виконавця. На цьому етапі завершується попереднє проектування бази даних, і на наступному етапі починається її безпосередня розробка (впровадження).

## 2.2 Архітектура бази даних SQL Server

Сьогодні розроблено велику кількість систем управління базами даних, які поділяються на персональні і серверні. Попри те, що вони можуть по-різному працювати з об'єктами й надавати користувачеві різні функції і засоби, більшість СУБД спираються на єдиний установлений комплекс основних понять. Це дає нам змогу розглянути одну систему і узагальнити її поняття, прийоми і методи на весь клас СУБД. Таким навчальним об'єктом пропонуємо СУБД SQL Server.

SQL Server - це серверна реляційна СУБД, яка забезпечує однотипний метод доступу клієнтів до бази даних і розподіл "обов'язків" між клієнтом і

сервером. Основними компонентами SQL Server є реляційна база даних, мова Transact-SQL та графічні інтерфейси користувача (GUT).

Основу бази даних SQL Server утворює сервер, або ядро бази даних. Ядро бази даних відповідає за обробку запитів, які надходять від клієнтів, і передачу відповідних результатів клієнтським компонентам.

Під час установлення SQL Server створюється кілька баз даних (чотири системні - master, model, msdb, tempdb і дві користувацькі - pubs і north-wind), кожна з яких зберігається в одному або кількох файлах. У базі даних master зберігається вся інформація відносно конфігурації і функціонування

SQL Server. Вона містить відомості про всі облікові записи користувачів, про інші бази даних, а також про параметри сервера. База даних model є шаблоном для баз даних, що створюються, і завжди має бути в системі. Під час створення нової користувацької бази даних SQL Server створює копію бази даних model. У базі даних msdb зберігається інформація про планування задач (jobs) і подій (alerts), а також про організацію роботи операторів, які отримують повідомлення. База даних tempdb, яку формує SQL Server, призначена для збереження тимчасових таблиць. База даних tempdb є глобальним ресурсом, який автоматично доступний усім користувачам і створюється кожного разу під час запуску SQL Server. Усі тимчасові таблиці tempdb автоматично вилучаються, коли користувач відключається від сервера. База даних pubs входить до комплекту постачання SQL Server і являє собою навчальну базу даних. База даних northwind також входить до комплекту постачання SQL Server, є такою ж навчальною базою даних, як і pubs, але вона імітує базу даних Microsoft Access.

Кожна база даних SQL Server складається з кількох об'єктів, які використовуються для збереження, організації та обробки даних. Об'єктами бази даних SQL Server є таблиці, індекси, представлення, обмеження, правила, значення по замовчуванню, тригери, процедури і типи даних.

Таблиця є основним об'єктом, який зберігає всі записи, що належать до бази даних (файли таблиць мають розширення .mdf). У SQL Server є таблиці двох типів - системні і користувацькі. У системних таблицях зберігається інформація про SQL Server і його об'єкти, а в користувацьких - інформація з первинних документів. Імена всіх системних таблиць починаються з префікса sys. До складу таблиць входить файл транзакцій (Mf), який утворюється автоматично під час створення бази даних і призначається для забезпечення цілісності і відновлення бази даних у разі виникнення помилок (якщо в базу даних вносять зміни, то в журналі транзакцій зберігаються нові і старі значення рядків таблиці).

Індекси - це файли з розширенням idx, які використовують для скорочення часу виконання операцій пошуку і вибірки даних із таблиць (перелік індексних файлів визначає користувач). Індексні файли утворюються з номерів записів, розміщених у послідовності, яку вони б мали, якби були відсортовані за певними полями.

Представлення (Views) - це SQL-інструкція Select, на основі якої здійснюється вибірка даних з однієї або кількох таблиць і утворюється вихідна таблиця. Найчастіше представлення використовується для вибірки даних деякої множини рядків або стовпців із таблиць, об'єднання стовпців різних таблиць, обчислення підсумкових значень на основі даних одного або кількох стовпців.

Обмеження гарантують цілісність даних для таблиць, їх зазвичай додають користувачі до таблиці після її створення, і вони можуть бути визначені на рівні стовпців або на рівні таблиці. SQL Server підтримує обмеження цілісності п'яти типів.

- Primary Key (обмеження за первинним ключем) - гарантує, що для всіх рядків таблиці буде унікальний ключ, не рівний NULL. Використання обмеження первинного ключа, крім іншого, створює унікальний індекс по таблиці.

- Foreign Key (обмеження за зовнішнім ключем) - зв'язує один або кілька стовпців у таблиці з первинним ключем і гарантує, що між двома таблицями існують вказані відношення.
- Unique (обмеження унікальності) - запобігає появі в будь-якому стовпці значень, що повторюються.
- Check (обмеження за значенням) - забезпечує контроль значень, які можуть бути введені в стовпець таблиці. Наприклад, можна задати обмеження перевірки введення даних у стовпець таблиці у межах від 1 до 100.
- Not Null (обмеження на невизначене значення) - використовується для гарантії того, що стовпець не буде мати значення Null.

Правила - це, по суті, ті ж обмеження, але більш детальні. Правила будуються на основі списку значень і логічних виразів (у практичній роботі їх використовують рідко).

Значення по замовчуванню (defaults) - установлює значення, яке автоматично записується в комірку стовпця таблиці, якщо в це поле не було введено будь-яке значення.

Тригер - це процедура, яка здійснюється автоматично під час зміни таблиці з використанням інструкції Update, Insert, Delete.

Процедура - це кілька послідовних інструкцій Transact-SQLt які під час її створення компілюються у спеціальний формат, із подальшим використанням для реалізації різноманітних функцій адміністрування бази даних, обробки даних тощо.

Типи даних - визначають тип інформації, яку можна зберігати в певному стовпці таблиці. Типи даних поділяються на числові цілі, числові з плаваючою крапкою (дробові), текстові, дати і часу, спеціальні.



SQL Server використовує діалект мови структурованих запитів SQL - Transact-SQLy за допомогою якого описують дані БД і зв'язки між ними, виконують операції над даними (додавання таблиць, внесення змін у таблиці, формування запитів для вибірки даних з бази даних тощо).

## **2.3 База даних MySQL**

Для запису, вибірки та обробки даних, що зберігаються в комп'ютерній базі даних, необхідна система управління базою даних, якою і є програмне забезпечення (ПЗ) MySQL.

База даних MySQL - реляційна база даних. У реляційній базі даних дані зберігаються не всі скопом, а в окремих таблицях, завдяки чому досягається вигравш у швидкості та гнучкості. Таблиці зв'язуються між собою за допомогою відносин, завдяки чому забезпечується можливість об'єднувати при виконанні запиту дані з декількох таблиць. SQL як частина системи MySQL можна охарактеризувати як мова структурованих запитів плюс найбільш поширений стандартний мову, що використовується для доступу до баз даних.

СКБД MySQL є системою клієнт-сервер, яка містить багато-поточний SQL-сервер, що забезпечує підтримку різних обчислювальних машин баз даних, а також кілька різних клієнтських програм і бібліотек, засоби адміністрування і широкий спектр програмних інтерфейсів (API).

Сервер MySQL постійно працює на комп'ютері. Клієнтські програми (наприклад, скрипти PHP) посилають серверу MySQL SQL-запити через механізм сокетів (тобто за допомогою мережових засобів), сервер їх обробляє і запам'ятовує результат. Тобто скрипт (клієнт) вказує, яку інформацію він хоче отримати від сервера баз даних. Потім сервер баз даних посилає відповідь (результат) клієнтові (скрипту).

### Структура MySQL трирівнева:

- бази даних;
- таблиці;
- записи.

Логічно таблиця являє собою сукупність записів. А записи - це сукупність полів різного типу. Ім'я бази даних MySQL унікально в межах системи, а таблиці - в межах бази даних, поля – в межах таблиці. Один сервер MySQL може підтримувати відразу декілька баз даних, доступ до яких може розмежовуватися логіном і паролем. Знаючи ці логін і пароль, можна працювати з конкретною базою даних. Наприклад, можна створити або видалити в ній таблицю, додати записи і т. д. Зазвичай ім'я-ідентифікатор та пароль призначаються хостинг провайдерами, які і забезпечують підтримку MySQL для своїх користувачів.

ПЗ MySQL, є додатком з відкритим кодом. Кожен користувач може вивчити вихідний код і змінити його у відповідності зі своїми потребами.

### Основні переваги пакету MySQL :

- багатопоточність. Підтримка декількох одночасних запитів;
- оптимізація зв'язків з приєднанням багатьох даних за один прохід;
- записи фіксованої і змінної довжини;
- ODBC драйвер в комплекті з вихідним текстом;
- гнучка система привілеїв і паролів;
- до 16 ключів в таблиці. Кожен ключ може мати до 15 полів;
- заснована на потоках, швидка система пам'яті;

– всі операції роботи з рядками не звертають уваги на регістр символів в оброблюваних рядках.

## **2.4 Таблиці баз даних**

Як уже зазначалось, у реляційних базах даних, таких, як SQL Server, дані зберігаються в таблицях, які являють собою двовимірні масиви (рядки і стовпці таблиці). У процесі створення таблиць використовується така одиниця виміру як екстент (вісім сторінок, кожна з яких має розмір 8 Кбайт). Коли таблиця займає весь простір виділених екстентів, до них автоматично додаються нові екстенти. Так продовжується доти, доки не буде досягнута межа, що встановлена для загального розміру бази даних. Для будь-якої бази даних можна створити до двох мільярдів таблиць. Таблиці бази даних поділяють на постійні і тимчасові (локальні і глобальні). Локальна тимчасова таблиця доступна тільки для того сеансу, в якому вона була створена (сеанс - звернення до бази даних). Локальна тимчасова таблиця визначається іменем, в якому на місці першого символу стоїть знак фунта (#). По закінченні сеансу локальна тимчасова таблиця автоматично вилучається. В імені локальної тимчасової таблиці може бути не більше ніж 116 символів (включаючи знак #). Щоб створити глобальну тимчасову таблицю, потрібно на місці двох перших символів у її імені поставити знак фунта. Глобальна тимчасова таблиця автоматично вилучається по закінченні останнього сеансу роботи. Тимчасові таблиці, як правило, використовують для збереження даних, Над якими потрібно спочатку виконати деякі операції, а потім зберегти в постійних таблицях. Кожна таблиця бази даних повинна мати ключ, визначений користувачем у момент створення таблиці, за яким здійснюється автоматичне сортування даних.

Одним з основних моментів створення таблиць є визначення типів даних для їх стовпців. Тип даних стовпця таблиці визначає тип інформації, яку можна зберігати в цьому стовпці. Як уже зазначалося, типи даних поділяються на

числові цілі, числові з плаваючою крапкою (дробові), текстові, дати і часу, спеціальні.

Числові цілі - це один із кількох типів: Int - додатні або від'ємні числа в діапазоні від -231 до +231; Smallint - числа в проміжку від -32768 до +32767; Tinyint - числа в інтервалі від 0 до 255.

Текстові дані дають змогу вводити в стовпець таблиці букви, числа і спеціальні символи, наприклад, ?, ,! тощо: Char (n) - число в круглих дужках визначає кількість символів, наприклад, якщо для стовпця таблиці визначено тип Char (12), тоді для кожного значення стовпця можна вводити 12 символів. Якщо буде введено меншу кількість символів, тоді SQL Server додасть пропуски після останнього символу; VARCHAR - цей тип дає змогу зберігати дані змінної довжини, тобто пропуски не додаються. Наприклад, якщо для стовпця таблиці визначено тип VARCHAR (22), тоді для кожного значення стовпця можна вводити до 22 символів. Якщо буде введено меншу кількість символів, тоді SQL Server не буде додавати пропуски після останнього символу.

DateTime - цей тип даних дає можливість зберігати дату і час у діапазоні від 01.01.1753 до 31.12.9999 року, від'ємне число говорить про те, що подія сталася до 1 січня 1900 року, а додатне - після цієї дати. Значення дати відображається у стандартному форматі - MMMDD YYYY hh:miAM/PM. Наприклад, Sep 15 2003 10:23 PM. Якщо ввести час без дати, тоді буде прийнята стандартна дата, 1 січня 1900 року. Якщо ввести дату без часу, тоді буде прийнято стандартний час, 12:00 AM. Дату можна вводити кількома способами: якщо ввести тільки два розряди року, то для чисел, менших ніж 50, як століття вибереться 20, а для чисел, більших за 50, - 19. Наприклад, якщо ввести рік 25, тоді він буде збережений як 2025. Якщо ж ввести рік 93, тоді він буде збережений як 1993. Значення дати можна вводити з використанням похилої риски, дефіса і крапки: 6/21 /04; 06/21 /04; 6-21 -04; 6.21.04; 06.21.04.

Для стовпця таблиці можна визначати не тільки тип даних, а ще одну додаткову характеристику - Null або Not Null. Якщо для стовпця таблиці визначено атрибут Null, тоді цей атрибут дозволяє не вводити даних у цей стовпець. Якщо ж для стовпця визначено атрибут Not Null, SQL Server не дозволяє залишати стовпець без введених даних (по замовчуванню стовпцю назначається атрибут Not Null).

## **2.5 Перелік таблиць бази даних build\_supermarket**

1. shop – таблиця, яка містить дані про магазин будівельних товарів
2. client – таблиця, яка містить дані про клієнта магазину
3. assortment – таблиця, яка містить дані про товари супермаркету
4. bill – таблиця, яка містить дані чеку про покупку певного товару

## **2.6 Перелік полів, які ідентифікують кожну з таблиць створеної БД**

Таблиця shop :

- shop\_id – ідентифікатор магазину
- name – назва магазину
- address – адреса магазину
- assortment\_amount – кількість товарів, що є в продажі

Таблиця client :

- client\_id – ідентифікатор клієнта
- last\_name – прізвище та ім'я клієнта
- phone\_number – номер телефону клієнта
- bonus\_card – номер бонусної картки клієнта
- shop\_id – ідентифікатор магазину, в якому була здійснена покупка

Таблиця assortment :

- article\_id – ідентифікатор товару
- name – найменування товару
- appointment – призначення товару
- quality – якість товару (у відсотковому відношенні)
- country\_producer – країна-виробник товару
- price – ціна товару
- shop\_id – ідентифікатор магазину, в якому продається даний товар

Таблиця bill :

- bill\_id – ідентифікатор чеку
- article\_id – ідентифікатор товару, який є в чеку
- sum\_to\_pay – сума до сплати
- payment\_way – спосіб оплати (картка або готівка)
- client\_id – ідентифікатор клієнта, який здійснив цю покупку

Даний перелік таблиць формує єдину базу даних будівельного супермаркету. Її ER-діаграму бачимо на Рис. 2.1.

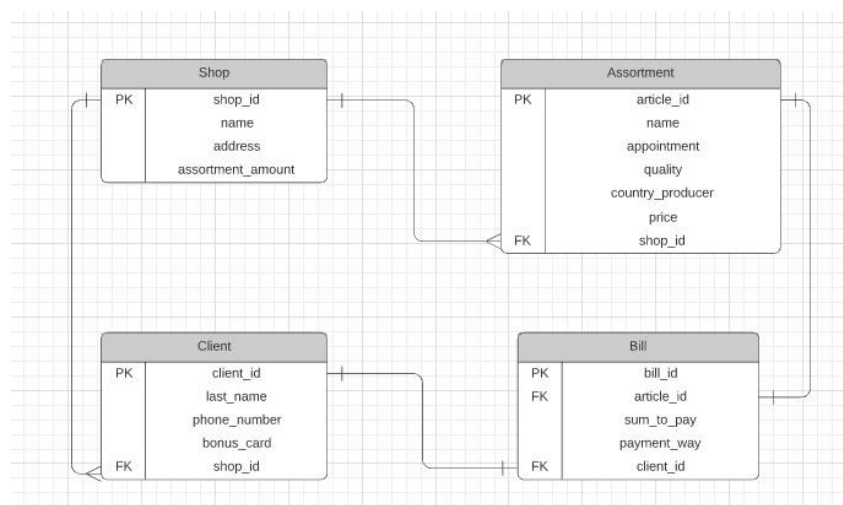


Рис. 2.1

РК – первинний ключ таблиці, FK – вторинний (зовнішній) ключ таблиці

Між таблицями використовуються такі типи зв'язків :

1. «один до одного» (1-1) :



2. «один до багатьох» (1-N) :



3. «багато до одного» (N-1) :



## 2.7 Запити, протестовані на розробленій базі даних

```
SELECT bill_id FROM bill WHERE sum_to_pay>1000;
```

Запит, який виводить усі ідентифікатори чеків, у яких сума до сплати становить більше 1000 одиниць.

```
SELECT SUM(price) FROM assortment WHERE country_producer='Germany';
```

Запит, який виводить загальну ціну товарів, виготовлених у Німеччині.

```
SELECT * FROM client ORDER BY shop_id;
```

Запит, який сортує дані таблиці «клієнт» за параметром ідентифікатора магазину.

```
UPDATE shop SET assortment_amount=12000 WHERE shop_id=3;
```

Запит, який змінює кількість товарів у таблиці «магазин» відповідного до заданого параметра shop\_id.

```
DELETE FROM bill WHERE article_id>5;
```

Запит, який видаляє з таблиці «чек» рядки, в яких ідентифікатор товару є більшим за 5.

```
SELECT name FROM assortment INNER JOIN bill using(article_id) WHERE  
payment_way="in cash";
```

Запит, який об'єднавши дві таблиці «асортимент» і «чек», виводить назву товару, за який розрахунок був проведений готівкою.

Запити на виведення усіх таблиць містяться в Додатку.



## **ВИСНОВОК**

Технічні характеристики сучасних засобів керування базами даних постійно зростають. Для забезпечення високих показників роботи необхідно правильно обрати СКБД відповідно задачі. Задачею цього курсового проекту було створення бази даних для будівельного супермаркету. Для створення бази даних було обрано СКБД MySQL. В базі даних були реалізовані запити на додавання даних, оновлення даних, видалення та вибір інформації з таблиць. Всі запити працюють надійно та видають чітку інформацію.

## **СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

<https://library.if.ua/book/119/7982.html>

<http://aphd.ua/publication-146/>

[http://ukrhosting.ua/bazi\\_daniv\\_mysql-p-263950.html](http://ukrhosting.ua/bazi_daniv_mysql-p-263950.html)

<https://studfile.net/preview/5454020/page:19/>

<https://studopedia.info/9-20422.html>

[https://pidru4niki.com/10611207/bankivska\\_sprava/arhitektura\\_baz\\_daniv](https://pidru4niki.com/10611207/bankivska_sprava/arhitektura_baz_daniv)

<https://works.doklad.ru/view/JfSC0RtpoAY.html>

## ДОДАТОК

```
mysql> SELECT * FROM shop;
```

shop_id	name	address	assortment_amount
1	ForReallyMens	Lviv, Shevchenka str.	10000
2	SamSobiHospodar	Lviv, Vyhovskogo str.	15000
3	VseDlyaRemontu	Lviv, Shota Rustaveli str.	5000

```
3 rows in set (0.63 sec)
```

Рис. 2.2 Таблица shop

```
mysql> SELECT * FROM client;
```

client_id	last_name	phone_number	bonus_card	shop_id
1	Karpenko Volodymyr	56789340	10021	1
2	Vasyliiev Dmytro	54732894	56032	3
3	Panchyn Vladyslav	56782329	23202	2
4	Gladyshev Ivan	55327894	23256	2
5	Lybkiv Gennadiy	54548591	56021	3
6	Kulykovskiy Stepan	56782392	10034	1

```
6 rows in set (0.06 sec)
```

Рис. 2.3 Таблица client

```
mysql> SELECT * FROM assortment;
```

article_id	name	appointment	quality	country_producer	price	shop_id
1	glue	for instruments	100%	Ukraine	150	1
2	screwdriver	for screws	70%	Poland	95	2
3	perforator	for walls	100%	Germany	2500	2
4	switch	for light	60%	China	50	3
5	nails	for wood	90%	Ukraine	70	3
6	hammer	for big nails	100%	Poland	120	1
7	shovel	for gardening	70%	Ukraine	155	2
8	ax	for woods	100%	Poland	700	1
9	lazer level	for levelling	100%	Germany	4000	3
10	drill	for beton	90%	Ukraine	250	1

10 rows in set (0.06 sec)

Рис. 2.4 Таблица assortment

```
mysql> SELECT * FROM bill;
```

bill_id	article_id	sum_to_pay	payment_way	client_id
1	3	2500	by card	4
2	1	150	by card	1
3	9	4000	in cash	6
4	4	50	in cash	2
5	10	250	in cash	3

5 rows in set (0.16 sec)

Рис. 2.5 Таблица bill