Міністерство освіти і науки України

**Львівський національний університет імені Івана Франка**

### Факультет управління фінансами та бізнесу

Кафедра цифрової економіки та бізнес-аналітики

**Курсова робота**

на тему :

**“Інформаційна система для підприємства виготовлення меблів.”**

Студента 3 курсу, групи УФЕ-31с

напряму підготовки: Інформаційні технології в бізнесі

Гут Назарій.

Керівник :

Ярема О.Р. к.е.н, доц

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів : \_\_\_\_\_\_ Оцінка : ECTS \_\_\_

**Львів-2020**

**ЗМІСТ**

### Вступ..............................................................................................................3

### РОЗДІЛ 1. Обстеження і аналіз роботи підприємст меблів..............................................................................................................5

### 1.1Основні задачі інформаційної системи для підприємства виготовлення меблів......................................................................................5

### 1.2АНАЛІЗ підприємства виготовлення меблів..........................................6

### 1.3 Особливості підприємства виготовлення меблів...................................8

### РОЗДІЛ 2. ПОБУДОВА БАЗИ ДАНИХ ДЛЯ ПІДПРИЄМСТВА ВИГОТОВЛЕННЯ МЕБЕЛЬ........................................................................9

### 2.1 Опис моделі даних..................................................................................9

### 2.2Архітектура [СКБД](https://uk.wikipedia.org/wiki/Система_керування_базами_даних)..................................................................................13

### 2.2.1 Функціональні характеристики СУБД....................................14

### 2.2.2 Основні вимоги до СУБД........................................................17

### 2.3 Синтаксис операторів MySQL..............................................................19

### 2.3.1 Синтаксис операторів оператора CREATE TABLE...............19

### 2.3.2 Синтаксис оператора ALTER TABLE.....................................21

### 2.3.3 Синтаксис оператора CREATE INDEX..................................23

2.3.4 Типи CHAR та VARCHAR......................................................24

# ВИСНОВОК.........................................................................................26

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ...........................................27

ВСТУП

У цій курсовій роботі буде розроблятися Інформаційна система для підприємства виготовлення меблів.

У нас час кожне підпреємсто для ефективності та зручності, як для робітників та і клієнтів, повинне використовувати зручною інформаційною системою. Щоб підприємство було конкурентно спроможним та отримувало максимальний прибуток почали використовувати інформаційні системи, що надають змогу налагодити швидкий доступ до інформації для підвищення швидкості прийняття управлінських рішень, швидкого достопу клієнтів до продукції та налагодження звязків між компанією та їх клієнтами.

Інформаційні системи класифікують за функціональною ознакою та типом управління.

Структура інформаційної системи бувають як сукупність її функціональних підсистем, відповідно, інформаційна система класифікується за функціональною ознакою.

У практиці діяльності виробничих та комерційних об'єктів типовими видами діяльності, які визначають стандартну ознаку класифікації інформаційних систем, є виробнича, маркетингова, фінансова, кадрова діяльність. Виробнича діяльність пов'язана з безпосереднім випуском продукції і спрямована на створення і впровадження у виробництво науково-технічних новацій.

З розвитком інформаційних систем, інформація перетворилась в один з найважливіших ресурсів, якийзайняв місце поряд із трудовими, матеріальними та енергетичними ресурсами. Тому і виникла потреба у застосуванні у базах данних (БД). В БД використовуються класичні моделі даних: ієрархічна, мережева, реляційна та об’єктно-орієнтовна. Ієрархічна модель є зручною для роботи з ієрархічно впорядкованою інформацією та громіздка для інформації зі складними логічними зв’язками. Мережева модель — це представлення даних у вигляді довільного графа. Перевагою мережевої та ієрархічної моделей є їх ефективність реалізації по затратам пам’яті. Недоліком мережевої моделі даних є складність схеми БД. Реляційну модель запропонував американський математик Едгар Кодд, який довів, що будьяка сукупність даних предметної галузі може бути подана взаємопов‘язаними між собою математичними відношеннями. Об’єктноорієнтовні БД об’єднують в собі дві моделі даних, реляційну та мережеву, використовуються при створенні великих БД зі складними структурами даних.

**Мета і завдання дослідження**. *Метою курсової* робити є розробити Інформаційна система для підприємства виготовлення меблів.

Для досягнення мети в роботі поставлено й вирішено такі теоретичні та практичні *завдання*:

* визначити особливості підприємства виготовлення меблів;
* Аналіз роботи підприємства з виготовлення меблів;
* Обстеження роботи меблевої фірми підприємства з виготовлення меблів;
* Розробка бази даних;

*Об’єктом досліджен* є створення бази даних, що забезпечить швидкий доступ до інформації стосовно підприємства.

*Предметом* є теоретичні, методичні та практичні аспекти розробки БД,

Дляінформаційних система для підприємства виготовлення меблів.

**Практичне значення отриманих результатів**. База даних для роботи з інформацією підприємства.

**Використане програмне забезпечення**. Для створення програмного продукту використовувалось середовище розробки MySQL.

**РОЗДІЛ 1. Обстеження і аналіз роботи підприємства виготовлення меблів.**

**1.1 Основні задачі інформаційної системи для підприємства виготовлення меблів.**

Потрібно об’єднати інформацію підприємства в базі даних, яку зберігає підприємство, у базу даних, для пришвидчення й спрощення роботи з інформацію.

Ця інформаційна система повинна виконувати такі функції, як регістрація клієнта, обробка запиту клієнта, підписання договору з ним, створення нових матеріалів та створення калькуляції виробів, та підрахунок вартості виробу по калькуляції, та облік виробів.

Ця система повинна просто й доступно давати можливість вводити інформацію про клієнта з урахуванням того, чи він є фізичною або юридичною особою, вводити нові матеріали та вироби, укладати договіри та коректно вводити калькуляцію по виробам та договорам.

Таким чином кожена інформація, вказана у діаграмі потоків даних повинена бути присутня у програмі у вигляді окремої форми, яку можна буде вибрати із головного меню або із підлеглих меню, і яка давитиме можливість ввести дані, потрібні , щоб заповнити усі поля певної таблиці, а також повинна контролювати процес введення – тобто не дозволяти вводити запис, в якому деякі обов’язкові поля залишилися пустими і виводити спеціальне повідомлення про те що необхідно заповнити перше з пустих полей.

**1.2 АНАЛІЗ підприємства виготовлення меблів.**

Кожне проектування бере початок з дослідження даної предметної області, зовсім не потрібно створювати базу тільки для того, щоб створити базу даних. Результатом обстеження предметної області є створення звіту про предметну область. У ньому необхідно визначити коли, для кого і яка інформація потрібна. Слід зберігати ті дані, які необхідні для роботи тим або іншим працівникам. Також у звіті про предметну область визначається, які функції виконуються з цими даними.

Звіт має бути написаний у вільній формі. Його часто пишуть у формі сценарію діяльності, який містить наступні пункти:

1.Що відбувається у предметній області?

2.Яка реакція предметної області на ці події?

Паралельно вивчається термінологія цієї предметної області. Також необхідно вказувати те, як часто використовується та певна інформація, виконується та або інша дія, який обсяг інформації, як вона розміщується і які перспективи розвитку даного підприємства, тому що все це може вплинути на вибір засобів розробки майбутньої бази даних.

Фірма-замовник займається виробництвом корпусних меблів. Велика роль приділяється спілкуванню з клієнтами з метою отримання замовлень й оформлення договору про те, що за певну оплату фірма в певний термін виготовить і установить визначений об’єкт. Замовленням може бути як окремий предмет (наприклад, шафа-купе або комп’ютерний стіл) так і певна сукупність предметів (наприклад, кухня, дитяча кімната, вітальня і т.і.).

Фірма є приватним підприємством і тому може прийняти не більше ніж двадцять замовлень на місяць.

Замовлення ― це інформація, яку одержують від клієнта, про виріб, який повинна зробити фірма. На основі цієї інформації отримують калькуляцію (документ, цо містить назви всіх деталей виробу, їхні габаритний розмір, матеріал), а також ціну виробу.

При цьому клієнт є фізичною чи юридичною особою, що може здійснити замовлення на продукцію, вироблену фірмою.

Основними функціями є облік укладених договорів, виробів,що виготовляє фірма, збереження інформації про калькуляції виробів і облік доступних матеріалів. Все це зберігається в паперовому вигляді в різних каталогах.

Обмін цією інформацією відбувається між клієнтом та працівником фірми.

Кожен новий клієнт повинен бути зареєстрований у журналі клієнтів. Реєстрація клієнта полягає в запиті інформації про нього, такої як Ф.І.П., адреса проживання, контактний телефон, інформація про те, чи є він фізичною або юридичною особою. Якщо клієнт – фізична особа, то інформація про нього доповнюється домашнім телефоном і серією та номером паспорта. Якщо ж клієнт – юридична особа (представник фірми, організації), то потрібною є інформація про назву фірми, представником якої він є, факс і дані про банківський рахунок фірми (що включає назву банку, ОКПО, МФО та розрахунковий рахунок). Під час спілкування з клієнтом може з’явитися додаткова інформація про нього описового характеру. Кількість клієнтів може досягати порядку сотні на місяць, але вона рідко буває більша. У середньому їх біля тисячі на рік.

Під час каристування клієнт може зробити запит про вартість, розцвічення, зовнішній вид виробів, зроблених фірмою раніше (така інформація зберігається фірмою у вигляді розрізнених неструктурованих файлів на комп’ютері), а також про колір і вартість доступних матеріалів для виготовлення нових виробів (з таблиці матеріалів, що зберігається у паперовому вигляді).

Прийнявши рішення, клієнт може укласти договір про те, що він робить замовлення у фірми. У цьому договорі повинна бути зазначена інформація про клієнта (Ф.І.П., якщо клієнт – юридична особа, то ще і назва фірми, представником якої він є, і номер її особового рахунку), назву виробу, дату укладення договору, термін для установки (тобто дату, до якої фірма зобов’язується виготовити і встановити виріб) і даті закінчення гарантійного терміну. До договору додається загальний вигляд виробу.

Коли клієнт прийде другий раз він може зробити ще одне замовлення, у такому випадку інформація про нього буде знайдена по його Ф.І.П. для фізичної особи або за назвою фірми, яку він представляє, і буде оформлений новий договір. Клієнт також може вимагати ремонту виробу за гарантією, у цьому випадку необхідна перевірка того, чи не закінчився гарантійний термін даного виробу. По закінченню цього терміну ремонт не надається.

Можуть з’явитися нові матеріали, які повинні будуть бути добавлені до списку вже існуючих й тих, що використовуються. В цьому списку усі доступні матеріали, у тому числі і фурнітура, розділені по групах й для них вказані найменування, вартість у гривнях, у доларах та їх зовнішній вигляд (у вигляді фотографії або зразка матеріалу).

На даний момент уся ця інформація у фірмі зберігається у каталогах, в паперових списках та у вигляді розрізнених проектних файлів на комп’ютері. Це значно сповільнює пошук потрібної інформації через відсутність її структурованості. Окрім того, інформація, що викладена на паперових носіях, може бути легко втрачена або пошкоджена, а також займає немало місця.

Метою проведення такої роботи є оптимізація діяльності цієї меблевої фірми шляхом об’єднання чисельної інформації, що вона зберігає, у базу даних, що значно прискорить й спростить роботу з нею.

Цей програмний продукт повинен буде виконувати такі функції, як регістрація клієнта, обробка запиту клієнта, підписання договору з ним, введення нових (або змінення вже існуючих) матеріалів та створення калькуляції виробів, а також підрахунок вартості вироба по калькуляції.

**1.3 Особливості підприємства виготовлення меблів.**

Одним з найприбутковіших і стабільних видів бізнесу є виробництво і збут меблів. Інвестуючи кошти в подібне підприємство, можна забезпечити дохід в розмірі 40% річних. Успіх значною мірою залежить від обраної ноші і якості стратегії просування. Тому потрібно з розумом підходити до складання бізнес-плану і орієнтуватися на останні тенденції в цій сфері.

Концепція меблевого підприємства.Cуть бізнесу по виробництву меблів полягає в організації комплексного підприємства по виготовленню обраного типу продукції і подальшою її реалізацією.Компанії, що працюють з меблями, в основному роблять ставку на спеціалізацію, тобто вузьконаправлене виробництво товару під конкретного цільового покупця. Зараз існує безліч фірм, які воліють виготовляти меблі для аптек, торгових залів супермаркетів, книжкових крамниць та ін. Однак підприємці пішли далі: з’явилися організації, які випускають виключно столи для комп’ютерів, стільці на колесах або офісні перегородки, стійки, стільці в бар.

Вибір спеціалізації виробництваЗа даними маркетингових досліджень, найбільш перспективним напрямком визнано виробництво офісних, кухонних та спеціалізованих меблів. Останні роки набирають популярність вбудовані меблі, які виготовляються індивідуально під кожного клієнта з урахуванням архітектурних особливостей квартири.Корпусні меблі користується стабільно високим попитом і охоплює велмку аудиторію споживачів — від приватних осіб до великих компаній. Причому прибуток підприємства може бути забезпечена як при продажу готових меблів, так і виготовленні меблів на замовлення. Безкаркасні меблі відрізняється зручністю і оригінальною формою.

Для їх виготовлення не потрібно купувати дорогі верстати та інструменти, що є перевагою для підприємців з невеликим стартовим капіталом.Дерев’яні вироби вважаються класикою меблевого мистецтва. Для їх виробництва необхідні навички тесляра і майстерність скульптора. Такі меблі не вимагають значних витрат при виготовленні своїми руками, отже, торгова націнка буде більшою.Можна виробляти меблі з цінних порід дерева — бука, дуба, махагон. Такі ексклюзивні предмети інтер’єру затребувані серед споживачів з достатком вище середнього, що гарантує високий прибуток.Вузькоспеціалізовані меблі: садові, дачні, промислові, для тварин і т. д. В кожній з цих ніш, як правило, існує невелика кількість професійних компаній. Вибираючи виробництво подібних предметів, бізнесмен забезпечує собі стабільний потік клієнтів, які цілком можуть перетворитися в постійних замовників.Меблі, різняться за матеріалом виготовлення: палет, штучний ротанг, на метало каркас, з фанери, м’які. Для виробництва можна вибрати ту, яка стане візитною карткою вашого підприємства.При виборі спеціалізації виробництва можна орієнтуватися на динаміку суміжних ринків.

**РОЗДІЛ 2. ПОБУДОВА БАЗИ ДАНИХ ДЛЯ ПІДПРИЄМСТВА ВИГОТОВЛЕННЯ МЕБЕЛЬ.**

**2.1 Опис моделі даних**

База даних – сукупність даних, що організовані у відповідності до

концептуальної структури, яка описує характеристики цих даних та відношення між ними, при чому така сукупність даних, яка підтримує одну або більше областей використання. У джерелах наводяться класифікації за декількомаознаками.

За ступенем розподіленості існують централізовані та розподілені БД.

Централізовані – повністю підтримуються на одній ЕОМ. Розподіленими є ті,компоненти яких розміщуються в різних вузлах комп’ютерної мережі у відповідності до певного критерію. Очевидно, що розподілені БД є більш надійними, оскільки при відмові одного з вузлів у мережі інші залишаються дієздатними, а при умові, що кожний вузол буде мати резервні копії даних з інших вузлів, система буде мати високий рівень відмовостійкості. Проте, такі БД потребують значно більших ресурсів для підтримки роботи, а в контексті діяльності кафедри, потрібно враховувати обмеженість ресурсів, як людських, так і матеріальних. Також при взаємодії компонентів через мережу можна очікувати меншу швидкодію, порівняно з централізованою системою. Саме тому для вирішення даної задачі можна зупинитися на централізованому типі БД, що значно спростить підтримку та використання. Така БД зможе бути використана або на сервері кафедри, або безпосередньо встановлена на ЕОМ людини, відповідальної за укладання документів, що генеруються автоматизованою системою, яка розробляється. А надійність бази даних можна забезпечити регулярним створенням резервних копій – дампів.

За середовищем постійного збереження виокремлюють БД, що

зберігаються у вторинній, тобто постійній пам’яті комп’ютера (як правило

жорсткий диск), в оперативній пам’яті, та третинній – змінних носіях

інформації. В даному випадку, оскільки для підтримки документації важливим фактором є надійність збереження даних, оптимальним варіантом слід вважати традиційну модель зберігання – у вторинній пам’яті. Вона програє у швидкодії оперативній пам’яті, проте менш вибаглива до характеристик ЕОМ, на якій буде працювати БД.

Ще однією принциповою класифікацією баз даних є класифікація за моделлю даних. Модель даних визначає логічну структуру БД та визначає

основні принципи відносно того, як дані можуть зберігатися, організовуватись, та яким чином з ними можна працювати .

У ієрархічній моделі БД дані подаються у вигляді деревовидної структури. Вони зберігаються як записи, що поєднуються між собою за допомогою посилань. Запис – це набір полів, кожне з яких містить одне значення.Тип сутності запису визначає, які поля запис має.

Запис у ієрархічній моделі БД відповідає рядку (кортежу) у реляційні

моделі, а тип сутності відповідає таблиці.

Ієрархічна модель строго визначає, що кожний запис-нащадок, має тільки один запис-предок, в той час як один предок може мати декілька нащадків. Щоб отримати дані з ієрархічної моделі усе дерево необхідно обійти починаючи від кореневого вузла. Ця модель визнана першою моделлю БД та створена IBM ще у 1960-х, тож багато спеціалістів вважає її морально застарілою. У реляційній моделі використовується підхід оперування даними використовуючи структуру та мову, сумісну з логікою предикатів першого

порядку, вперше описаною Едгаром Коддом, в якій усі дані представлені в

термінах кортежів, згрупованих у відношення. БД, що організовані в рамках

даної моделі, називаються реляційними базами даних – Реляційна модель

Метою реляційної моделі є надання декларативного методу визначення

даних та запитів: користувачі чітко задають, яку інформацію містить БД, та яку вони хочуть отримати, залишаючи при цьому усі деталі реалізації на програмне забезпечення СУБД.

Більшість баз даних використовують мову SQL. Вона є стандартизованою,

але різні реалізації в рамках певних СУБД можуть дещо відхилятися від цього стандарту з метою надання більших можливостей чи поліпшення швидкодії. Слід зазначити, що дана модель організації даних у БД є наразі найбільш поширеною, та має безліч реалізацій, як з відкритим програмним кодом, так і комерційних, коштовних систем.

У об’єктно-орієнтованій моделі інформація представляється у вигляді

об’єктів, так само, як і в об’єктно-орієнтованих мовах програмування. Це

дозволяє значно спростити налагодження взаємодії між БД та ОО-мовою

програмування, оскільки не потрібно використовувати ORM-фреймворки, чи

самостійно писати об’єкти для доступу к даним БД.

Для використання в рамках даної роботи було прийнято рішення обрати

реляційну базу даних, оскільки даний формат має велику розповсюдженість,

відповідно наявна велика кількість джерел інформації стосовно використання

таких БД, а також наявна велика кількість СУБД під цей тип, тобто є можливістьобрати оптимальний варіант для використання в системі, що розробляється.

Сучасні системи управління базами даних

На сьогоднішній день існує досить багато систем управління базами даних.

Було розглянуто деякі з них.

PostgreSQL. Безкоштовна об'єктно-реляційна система управління базами

даних. Вона є однією з найбільш просунутих СУБД, що в першу чергу

орієнтується на повну відповідність стандартам та можливості розширення,

тобто повністю намагається відповідати SQL–стандартам ANSI/ISO. З-поміж

інших цю СУБД також виокремлює той факт, що вона має об’єктноорієнтований функціонал та підтримує відповідні концепти.

Система PostgreSQL заснована на ядрі, створеному безліччю розробників.

У подібних випадках розумно зосередитися на оснащенні системи новими можливостями, але не займатися оптимальним їх втіленням, оскільки у випадку виникнення необхідності завжди можна буде повернутися до оптимізаціївідповідних ділянок коду.

Також варто зазначити що ця система не достатньо оптимізована для

вирішення повсякденних не дуже важких задач. Її використання більше передбачається у випадках, коли важливими виявляються підвищена надійність та підтримки об’єктних підходів до БД. У випадку, коли більш важливим є виконання простих операцій зчитування-запису, PostgreSQL показує не найкращі результати.

Oracle Database або Oracle RDBMS. Об'єктно-реляційна система управління базами даних компанії Oracle Corporation. Ця СУБД забезпечує ефективне, надійне і безпечне управління даними таких критично важливих для бізнесу додатків, як онлайнові середовища, виконує масштабну обробкутранзакцій (OLTP), сховища даних з високою інтенсивністю потоку запитів, а також ресурсоємні інтернет-додатки. Редакція Oracle Database Enterprise Edition надає інструментальні засоби і функції, що забезпечують відповідність вимогам сучасних корпоративних додатків в області доступності та масштабованості. Ця редакція містить всі компоненти Oracle Database, а також допускає розширення за допомогою придбання додаткових модулів та програм. Система Oracle Database дозволяє звертатися до даних з будь-якогододатку, розробленого із застосуванням технологій Microsoft. NET, Visual Studio та веб-додатків. Основною умовою є лише наявність справних бібліотек, що дають змогу підключатися до серверу бази даних Oracle. Це комерційна СУБД, але є безкоштовна версія, яку можна без проблем скачати прямо з офіційного сайту компанії. Втім, варто зазначити, що безкоштовна версія має зменшений функціонал. Microsoft SQL Server. Система Microsoft SQL Server відштовхується відконцепції платформи даних Майкрософт: вона спрощує управління будь-якими даними в будь-якому місці і в будь-який момент часу.

Вона дозволяє зберігати в базах даних інформацію, отриману з структурованих, напівструктурованих і неструктурованих джерел, таких як зображення та музика. У SQL Server є великий набір інтегрованих служб, які розширюють можливості використання даних: можна складати запити, виконувати пошук, проводити синхронізацію, робити звіти, аналізувати дані. Всі дані зберігаються на основних серверах, що входять до складу центру обробки даних. До них здійснюється доступ знастільних комп'ютерів і мобільних пристроїв. Таким чином, можна повністю контролювати дані незалежно від того, де вони збережені.

Система MS SQL Server дозволяє звертатися до даних з будь-якого додатку, розробленого із застосуванням технологій Microsoft. NET та Visual Studio, а також в межах сервісно-орієнтованої архітектури і бізнес-процесів – через Microsoft BizTalk Server. SQL Server дозволяє створити надійну,продуктивну, інтелектуальну платформу, що відповідає всім вимогам по роботі з даними . Ця система є комерційною, тобто потребує значних витрат на встановлення та налагодження роботи.

Система MySQL. Безкоштовна система управління базами даних. MySQL є власністю компанії Oracle Corporation, що отримала її разом з поглиненою Sun Microsystems, що здійснює розробку і підтримку програми. Розповсюджується під GNU General Public License або під власною комерційною ліцензією. Крім цього розробники створюють функціональність за замовленням ліцензійних користувачів.

MySQL є рішенням для малих і середніх додатків. Входить до складу серверів WAMP, LAMP і в портативні збірки серверів Denver, XAMPP. Зазвичай MySQL використовується як сервер, до якого звертаються локальні або віддалені клієнти, проте в дистрибутив входить бібліотека внутрішнього сервера, що дозволяє включати MySQL в автономні програми .

Як вже було зазначено раніше, СУБД, що обиралася, повинна задовольняти вимогам щодо простоти використання, безкоштовності та

наявності достатньої кількості довідкового матеріалу. Тому, вибір було зробленона користь MySQL.

По-перше, дана СУБД є безкоштовним open-source продуктом, тобто не викличе жодного фінансового навантаження при використанні. По-друге, супутній продукт MySQL Workbench дозволяє просто взаємодіяти зі встановленою БД, надаючи можливість через зручний інтерфейс користувача виконувати будь-які маніпуляції з даними. І нарешті, ця СУБД є однією з найбільш розповсюджених, займаючи друге місце за популярністю у світі, поступаючись лише корпоративній СУБД від Oracle

Саме тому в мережі наявна велика кількість матеріалів, як довідкових, включно з офіційнимсайтом, так і навчальних, де розглядається широкий спектр проблем та шляхи їхвирішення.

Отже, для більш чіткого обґрунтування вибору СУБД, що буде використовуватись, наводиться перелік переваг MySQL

* Відкритий вихідний код та безкоштовність.
* Продумана та швидкодіюча.
* Займає небагато дискового простору.
* Може бути легко встановлена на багатьох операційних системах,
* таких як Windows, Unix-like і інших.
* Враховуючи її розповсюдження в Мережі можна знайти безліч
* матеріалів, в яких розглядається вирішення певних питань та
* проблем, що виникають при роботі з нею.
* Добре підходить для невеликих застосунків.

**2.2Архітектура** [**СКБД**](https://uk.wikipedia.org/wiki/Система_керування_базами_даних)

Системи управління базами даних (СКБД) - це програмні комплекси, Призначені для роботи зі спеціально організованими файлами (масивами даних,довготривало збереженими в зовнішній пам'яті обчислювальних систем), які називаються базами даних.

Основними функціями СУБД є управління файлами БД («внутрішня» функція) і обробка прикладних програм (запитів) користувачів БД («зовнішня» функція). До забезпечує функцій СУБД відносяться: підтримка цілісності БД в процесі її експлуатації, захист БД від несанкціонованого доступу, управління обміном даними між БД і зовнішнім середовищем (в т.ч. управління розподіленою обробкою даних). Крім того, сучасні СУБД часто оснащуються додатковими інструментальними засобами розробки додатків (прикладних задач обробки даних).

СУБД як програмний продукт характеризується великою трудомісткістю виготовлення, високою наукоемкостью, тривалими термінами розробки, значною вартістю. В даний час на ринку інформаційних технологій пропонуються десятки різних СУБД, доведених до «комерційного» (тобто рекомендованого до промислової експлуатації) зразка. відповідно, перед споживачем, для тих, до створення інформаційної системи, постає проблема усвідомленого вибору необхідної йому моделі, версії і конфігурації СУБД.

Сучасні БД служать для збереження і обробки інформації, що стосується самих різних аспектів діяльності як окремих підрозділів (що складаються може бути всього з кількох людей), так і великих організацій і підприємств (де активними користувачами можуть бути сотні і тисячі чоловік). При виборі типу і конфігурації СУБД прийнято виділяти такі класи інформаційних систем, що функціонують на основі технології БД ( систем баз даних): - системи оперативної обробки транзакцій (OLTP). Характеризуються тим, що до БД надходить інтенсивний потік запитів на оновлення даних, в той час як потік запитів на додавання великих обсягів даних невеликий. Типовим прикладом є системи резервування квитків на пасажирському транспорті; - системи підтримки прийняття рішень Характеризуються використанням БД великого об'єму, в яких накопичується інформація за тривалий період часу і здійснюються в основному операції пошуку і зчитування даних, що використовуються додатками для формування звітів. Типовим прикладом можуть служити системи верхнього рівня управління підприємством; - системи пакетної обробки (PP). характеризуються здатністю тривалий час працювати в автономному режимі з мінімальною участю оператора. Як правило, БД тут використовується обмеженим числом прикладних програм. Типовий приклад - системи управління технологічними процесами.

**2.2.1 Функціональні характеристики СУБД**

Основні можливості СУБД визначаються набором таких її функціональнотехнологічних характеристик, як підтримувана модель даних, рівень вхідного мови, масштабованість і переносимість, архітектура розподіленої обробки. Модель даних. Визначає способи та форми подання даних на зовнішньому (призначеному для користувача) рівні.

Слід підкреслити, що сучасні СУБД орієнтовані в основному на обробку так званої фактографічної інформації, подається впорядкованою сукупністю фактів (значень у вигляді чисел, рядків символів, логічних констант і т.п.). Ці значення, як правило, об'єднуються в порівняно невеликі послідовності - записи, які відповідають окремим об'єктам (явищ) прикладної предметної області.

Сукупності однорідних записів об'єднуються в таблиці (така форма подання інформації, як відомо, найбільш широко застосовується як в «ручних», так і автоматизованих інформаційних системах). Таким чином, модель даних задає правила структуризації таблиць, їх логічного зв'язування в БД, а також правила виконання операцій над елементами «логічної» структури даних. Незважаючи на те, що існує безліч моделей представлення табличних даних, в даний час домінуюче становище на ринку комерційних СУБД займають реляційні системи, тобто СУБД, що підтримують реляційну модель даних. (Ця модель являє БД як сукупність жорстко не пов'язаних між собою таблиць, а операції тут мають простий і точний зміст, наприклад: «додати рядок в таблицю», «видалити з таблиці рядки, що відповідають деякому логічному умові», «об'єднати (злити) дві однорідні таблиці в одну »і т.п.). Крім простоти маніпулювання з даними реляційні системи мають ряд інших важливих переваг над нереляціоннимі. У подальшому обговоренні в силу зазначених обставин ми зосередимося виключно на реляційних СУБД.

Слід підкреслити, що сучасні реляційні СУБД надають користувачеві можливість працювати не тільки з фактографічної інформацією, але і з інформацією, представленою в інших формах (великі текстові документи, графіки, малюнки і т.д.). У цьому випадку окремі поля реляційних таблиць замість традиційних значень містять посилання на нетипові для СУБД інформаційні об'єкти, а сама СУБД оснащена засобами доступу до цих об'єктів (і, можливо, обмеженими засобами їх спеціалізованої обробки).

Підтримка моделі даних полягає в реалізації засобами СУБД мови обробки даних, що відповідає цій моделі. У загальному випадку СУБД можуть оснащуватися декількома вхідними мовами, відмінними рівнем деталізації команд на обробку даних. Мови дуже високого рівня реалізують обмежений набір операцій реляційної моделі і орієнтовані на кінцевого користувача. Вони дозволяють з найменшими затратами праці формулювати прості запити на обробку даних (набір яких, втім, може виявитися достатнім для вирішення всіх необхідних користувачеві завдань). Мови, що реалізують операції реляційної моделі в повному обсязі, також характеризуються малим ступенем деталізації виконуваних команд і відносяться до мов обробки даних високого рівня.

В даний час в індустрії систем обробки даних склався стандарт реляційного мови високого рівня, званий непроцедурного SQL, і підтримуваний більшістю сучасних СУБД (як правило, з деякими відхиленнями від стандарту). Непроцедурного SQL є більш, ніж реляційно повним мовою, оскільки включає ряд додаткових операцій над даними (арифметичні операції, операції порівняння подстрок символів та ін.). Проте, специфіка деяких додатків вимагає використання команд обробки даних низького рівня. З цією метою СУБД оснащуються процедурними засобами обробки даних. Зокрема, в стандарт SQL входить процедурне розширення мови, яке підтримується багатьма сучасними СУБД (правда, зі значними відхиленнями від стандарту). Інший підхід полягає в тому, що в якості вхідного використовується звичайну мову програмування, розширений конструкціями непроцедурного SQL.

Масштабованість і переносимість. У міру зростання інформаційних потреб і розвитку парку обчислювальної техніки перед організацією, експлуатуючої деяку СУБД, неминуче постає питання про модернізацію застосовуваної технології обробки даних. З цієї точки зору перевагою володіють СУБД, характеризуються широким діапазоном масштабованості, тобто системи, застосування яких з технічних і економічних міркувань доцільно і можливо як в умовах скромних можливостей наявного парку ВТ, так і для побудови великомасштабних систем баз даних. Слід мати на увазі, що СУБД як програмний засіб функціонує під управлінням певної операційної системи (як кажуть, - на певній платформі), і виробники СУБД розробляють свій продукт під різні платформи. Таким чином, при переході на нову платформу споживач повинен враховувати можливості перенесення експлуатованої СУБД, наявної БД і прикладних програм на цю платформу. Залежно від функціональних можливостей СУБД тут можуть виникнути такі ситуації: - існуюча СУБД піддається перенастроюванні, а додатки і БД залишаються без змін (найкращий варіант); - встановлюється нова версія СУБД без (явної) модернізації БД і додатків; - встановлюється нова версія СУБД з перезавантаженням БД, але без модифікації додатків; - встановлюється нова версія СУБД, перезавантажується БД і перепрограмовуються додатки Архітектура розподіленої СУБД.

Сучасні СУБД забезпечують можливість одночасної роботи з БД багатьох користувачів (в рамках локальної або територіально розподіленої обчислювальної мережі). При цьому є два різних підходи до організації такої роботи. В архітектурі файл сервер запити користувача обробляються локально на його робочої станції засобами локально встановленої на цій станції копії СУБД, а дані, необхідні для виконання запиту, можуть перебувати на інший робочої станції мережі. При цьому кожна локальна копія СУБД виконує роль так званого сервера БД, відстежуючи можливі одночасні звернення до одного фрагменту БД з різних додатків і встановлюючи порядок блокування і розблокування доступу до цих фрагментах. Подібна технологія застосовується зазвичай для невеликих локальних мереж і при низької інтенсивності обміну даними між робочими станціями. Архітектурі клієнт / сервер виділяється спеціальна серверна частина СУБД, яка зазвичай разом з основною БД розгортається на потужної обчислювальної машині, і клієнтська частина, що розгортається на робочих станціях.

Обробка запиту користувача практично повністю здійснюється сервером БД, а клієнтська частина відіграє роль інтерфейсу (сполучної ланки) між додатком і сервером. Така архітектура забезпечує високу надійність і ефективність системи, так як обмін даними по мережі мінімізується, а управління доступом здійснюється централізовано. У той же час СУБД, що підтримують дану архітектуру, вимагають для своєї роботи значних обчислювальних ресурсів.

**2.2.2 Основні вимоги до СУБД**

Якість СУБД багато в чому визначаться тим, якою мірою вона дозволяє задовольнити основні вимоги, що пред'являються до функціонування СУБД. Серед таких вимог (критеріїв) можна назвати: продуктивність, ресурси, цілісність БД, незалежність даних, безпеку, простоту використання, прозорість.

Дамо коротку характеристику кожного з цих критеріїв. Продуктивність кількісно можна оцінити як час реакції системи на запит користувача. (Природно, при цьому необхідно враховувати такі фактори, як інтенсивність одночасного звернення до фрагментів БД, обсяг БД, пропускна здатність каналів обміну і ін.). За інших рівних умов продуктивність забезпечується здатністю СУБД оптимізувати процес пошуку і оновлення даних в файлах БД, знаходити раціональний план обчислення (тобто порядок виконання елементарних операцій) запиту (системи одночасно обслуговуються запитів).

Для ефективної роботи СУБД, як правило, потрібні значні обчислювальні ресурси. В цьому відношенні особливо критичними виявляються обсяг оперативної пам'яті, пропускна здатність каналів введення / виведення і каналів обміну, продуктивність процесора, обсяг зовнішньої пам'яті.

Зрозуміло, що вимоги до ресурсів безпосередньо залежать від масштабності СБД і характеру розв'язуваних у ній завдань. У той же час СУБД повинні забезпечувати економне використання наявних ресурсів. Як показує практика їх експлуатації, така вимога далеко не завжди виконується. З метою раціонального розподілу ресурсів (що може здійснюватися і «вручну» шляхом конфігурації апаратних і програмних засобів) СУБД повинна бути оснащена засобами моніторингу та статистики.

Цілісність БД є основною умовою безвідмовного функціонування СБД. Практично всі сучасні СУБД оснащені потужним арсеналом засобів підтримки цілісності і парирування аварійних ситуацій. Тут головна роль відводиться засобам автоматичного копіювання і відновлення БД. Крім того, вхідні мови СУБД можуть містити команди звернення до процедурам обробки виняткових ситуацій, що дає користувачеві додаткові можливості забезпечення цілісності.

Незалежність даних розуміється як можливість внесення змін в структуру зберігається БД (що часто буває необхідно для підвищення ефективності її функціонування) без змін у «логічної» структури даних і прикладних програмах (запитах).

На практиці неможливо досягти абсолютної незалежності даних; СУБД забезпечують лише ту чи іншу ступінь незалежності (за рахунок наявності розвинених автоматично настроюються внутрішніх інтерфейсів «Користувач - логічна структура» і «логічна структура - структура зберігання»). Безпека БД розуміється як її захищеність від несанкціонованого використання. Кожен користувач (Прикладна програма) в СБД має певні права доступу до тих чи інших фрагментів БД.

Для того, щоб здійснити установку цих прав і проводити їх контроль в процесі функціонування СУБД повинна бути оснащена широким арсеналом коштів створення підтримки системи захисту. Слід розуміти, що підвищення вимог до рівня безпеки призводить до додаткових витрат ресурсів і зниження продуктивності.

Простота використання - неформальний критерій, виявляється у вимогах до кваліфікації обслуговуючого персоналу, зручність роботи користувачів різних категорій (прикладнихпрограмістів, адміністратора БД, операторів підготовки даних, кінцевих користувачів - споживачів вихідний інформації). Простота використання визначається такими факторами, як наявність в СУБД дружніх користувальницьких інтерфейсів, якість і методичний рівень супровідної документації, наявність спектра вхідних мов різного рівня, прозорість вихідного мови (мови повідомлень).

Вимога прозорості даних складається в наданні користувачам можливості звернення до розподілених в мережі даними без зміни технології своєї роботи. СУБД повинна по можливості «Приховувати» від користувача додаткові технологічні деталі операцій звернення до віддалених даних. Неважко помітити, що перераховані вимоги є часто суперечливими, і для вирішення питання вибору типу і конфігурації СУБД слід враховувати ступінь важливості кожного з вимог в конкретній обстановці. До того ж наведений тут перелік є далеко не повним. Зокрема, через специфіку питань ми не в повній мірі проаналізували вимоги до адміністрування БД, розробці додатків, адміністрування додатків і ін.

**2.3 Синтаксис операторів MySQL**

**2.3.1 Синтаксис операторів оператора CREATE TABLE**

Оператор CREATE TABLE створює таблицю з заданим ім'ям в поточній базі даних. Правила для допустимих імен таблиці наведені в розділі section 6.1.2 Імена баз даних, таблиць, стовпців, індекси псевдоніми. Якщо немає активної поточної бази даних або зазначена таблиця вже існує, то виникає помилка виконання команди.

У версії MySQL 3.22 і більш пізніх ім'я таблиці може бути зазначено як db\_name.tbl\_name. Ця форма запису працює незалежно від того, чи є зазначена база даних поточної.

У версії MySQL 3.23 при створенні таблиці можна використовувати ключове слово TEMPORARY. Тимчасова таблиця автоматично видаляється після закінчення з'єднання, а її ім'я дійсно тільки протягом даного з'єднання. Це означає, що в двох різних з'єднаннях можуть використовуватися тимчасові таблиці з однаковими іменами без конфлікту один з одним або з існуючою таблицею з тим же ім'ям (існуюча таблиця прихована, поки не видалена тимчасова таблиця). У версії MySQL 4.0.2 для створення тимчасових таблиць необхідно мати привілеї CREATE TEMPORARY TABLES.

У версії MySQL 3.23 і більш пізніх можна використовувати ключові слова IF NOT EXISTS для того, щоб не виникала помилка, якщо зазначена таблиця вже існує. Слід враховувати, що при цьому не перевіряється ідентичність структур цих таблиць.

Якщо не вказується ні NULL, ні NOT NULL, то стовпець інтерпретується так, як ніби вказано NULL.

Цілочисельний стовпець може мати додатковий атрибут AUTO\_INCREMENT. При записи величини NULL (рекомендується) або 0 в стовпець AUTO\_INCREMENT даний стовпець встановлюється в значення value + 1, де value являє собою найбільшу для цього стовпця значення в таблиці на момент запису. Послідовність AUTO\_INCREMENT починається з 1. See section 8.4.3.126 mysql\_insert\_id (). Якщо видалити рядок, що містить максимальну величину для стовпця AUTO\_INCREMENT, то в таблицях типу ISAM або BDB ця величина буде відновлена, а в таблицях типу MyISAM або InnoDB - немає. Якщо видалити всі рядки в таблиці командою DELETE FROM table\_name (без вираження WHERE) в режимі AUTOCOMMIT, то для таблиць всіх типів послідовність почнеться заново.

Примітка: в таблиці може бути тільки один стовпець AUTO\_INCREMENT, і він повинен бути індексований. Крім того, версія MySQL 3.23 буде правильно працювати тільки з позитивними величинами стовпчика AUTO\_INCREMENT. У разі внесення негативного числа воно інтерпретується як дуже велике позитивне число. Це робиться, щоб уникнути проблем з точністю, коли числа `` загортаються '' від позитивного до негативного і, крім того, для гарантії, що помилково не буде отриманий стовпець AUTO\_INCREMENT зі значенням 0. У таблицях MyISAM і BDB можна вказати вторинний стовпець AUTO\_INCREMENT з многостолбцовим ключем.

Величини NULL для стовпця типу TIMESTAMP обробляються інакше, ніж для стовпців інших типів. У стовпці TIMESTAMP не можна зберігати літерал NULL; при установці даного стовпця в NULL він буде встановлений в поточне значення дати і часу.

Оскільки стовпці TIMESTAMP поводяться подібним чином, то атрибути NULL і NOT NULL незастосовні в звичайному режимі і ігноруються при їх завданні. З іншого боку, щоб полегшити клієнтам MySQL використання стовпців TIMESTAMP, сервер повідомляє, що таким стовпцях можуть бути призначені значення NULL (що відповідає дійсності), хоча реально TIMESTAMP ніколи не буде містити значення NULL. Це можна побачити, застосувавши DESCRIBE tbl\_name для отримання опису даної таблиці.

Слід враховувати, що установка стовпчика TIMESTAMP в 0 не рівнозначна установці його в NULL, оскільки 0 для TIMESTAMP є допустимою величиною. Величина DEFAULT повинна бути константою, вона не може бути функцією або виразом. Якщо для даного стовпця не ставить ніякої величини DEFAULT, то MySQL автоматично призначає її. Якщо стовпець може приймати NULL як допустиму величину, то за замовчуванням присвоюється значення NULL.

Якщо стовпець оголошений як NOT NULL, то значення за замовчуванням залежить від типу стовпця:

* Для числових типів, за винятком оголошених з атрибутом AUTO\_INCREMENT, значення за замовчуванням дорівнює 0.

Для стовпця AUTO\_INCREMENT значенням за замовчуванням є таке значення в послідовності для цього стовпця.

* Для типів дати і часу, відмінних від TIMESTAMP, значення за замовчуванням дорівнює відповідній нульовий величиною для даного типу. Для першого стовпця TIMESTAMP в таблиці значення за замовчуванням є поточне значення дати і часу. Типи даних дати і часу.

Для строкових типів, крім ENUM, значенням за замовчуванням є порожній рядок. Для ENUM значення за замовчуванням дорівнює першій перераховується величиною (якщо явно не задано інше значення за замовчуванням за допомогою директиви DEFAULT).

Значення за замовчуванням повинні бути константами. Це означає, наприклад, що не можна встановити для стовпця дати як значення за замовчуванням величину функції, такий як NOW () або CURRENT\_DATE.

**2.3.2 Синтаксис оператора ALTER TABLE**

Оператор ALTER TABLE забезпечує можливість змінювати структуру існуючої таблиці. Наприклад, можна додавати або видаляти стовпці, створювати або знищувати індекси або перейменовувати стовпці або саму таблицю. Можна також змінювати коментар для таблиці і її тип. Синтаксис оператора CREATE TABLE.

Якщо оператор ALTER TABLE використовується для зміни визначення типу стовпця, але DESCRIBE tbl\_name показує, що стовпець не змінився, то, можливо, MySQL ігнорує дану модифікацію по одній з причин, Мовчазні зміни визначень стовпців. Наприклад, при спробі змінити стовпець VARCHAR на CHAR MySQL буде продовжувати використовувати VARCHAR, якщо дана таблиця містить інші стовпці з змінною довжиною.

Оператор ALTER TABLE під час роботи створює тимчасову копію вихідної таблиці. Необхідну зміна виконується на копії, потім вихідна таблиця віддаляється, а нова перейменовується. Так робиться для того, щоб в нову таблицю автоматично потрапляли всі оновлення крім невдалих. Під час виконання ALTER TABLE вихідна таблиця доступна для читання іншими клієнтами. Операції оновлення та записи в цій таблиці припиняються, поки не буде готова нова таблиця.

Слід зазначити, що при використанні будь-якої іншої опції для ALTER TABLE крім RENAME, MySQL завжди буде створювати тимчасову таблицю, навіть якщо дані, строго кажучи, і не потребують копіюванні (наприклад, при зміні імені стовпця). Ми плануємо виправити це в майбутньому, однак, оскільки ALTER TABLE виконується не так часто, ми (розробники MySQL) не вважаємо цю задачу першочерговим. Для таблиць MyISAM можна збільшити швидкість відтворення індексного частини (що є найбільш повільної частиною в процесі відновлення таблиці) шляхом установки змінної myisam\_sort\_buffer\_size досить великого значення.

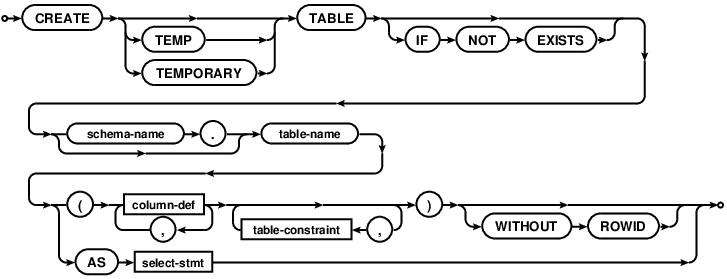
* Для використання оператора ALTER TABLE необхідні привілеї ALTER, INSERT і CREATE для даної таблиці.
* Опція IGNORE є розширенням MySQL по відношенню до ANSI SQL92. Вона управляє роботою ALTER TABLE при наявності дублікатів унікальних ключів в новій таблиці. Якщо опція IGNORE не задана, то для даної копії процес переривається і відбувається відкат назад. Якщо IGNORE вказується, тоді для рядків з дублікатами унікальних ключів тільки перший рядок використовується, а решта будуть видалені.
* Можна запустити кілька виразів ADD, ALTER, DROP і CHANGE в одній команді ALTER TABLE. Це є розширенням MySQL по відношенню до ANSI SQL92, де допускається тільки один вислів із згаданих в одній команді ALTER TABLE.

При зміні типу стовпця, але не його імені синтаксис вираження CHANGE все одно вимагає вказівки обох імен стовпчика, навіть якщо вони однакові. наприклад:

mysql> ALTER TABLE t1 CHANGE b b BIGINT NOT NULL;

Однак починаючи з версії MySQL 3.22.16a можна також використовувати вираз MODIFY для зміни типу стовпця без перейменування його:

mysql> ALTER TABLE t1 MODIFY b BIGINT NOT NULL;



* При використанні CHANGE або MODIFY для того, щоб зменшити довжину стовпчика, по частині якого побудований індекс (наприклад, індекс по першим 10 символам стовпчика VARCHAR), не можна зробити стовпець коротше, ніж число проіндексованих символів.
* при зміні типу стовпця з використанням CHANGE або MODIFY MySQL намагається перетворити дані в новий тип якомога коректніше.
* У версії MySQL 3.22 і більш пізніх можна використовувати FIRST або ADD ... AFTER ім'я\_стовпця для додавання стовпця на задану позицію всередині табличній рядки. За замовчуванням стовпець додається в кінці. Починаючи з версії MySQL 4.0.1, можна також використовувати ключові слова FIRST і AFTER в опціях CHANGE або MODIFY.

**2.3.3 Синтаксис оператора CREATE INDEX**

CREATE [UNIQUE|FULLTEXT] INDEX index\_name

ON tbl\_name (col\_name[(length)],... )

Команда CREATE INDEX в версіях MySQL до 3.22 не виконує ніяких дій. У версії 3.22 і більш пізніх CREATE INDEX відповідає команді ALTER TABLE в частині створення індексів. Синтаксис оператора ALTER TABLE.

Зазвичай все індекси створюються в таблиці під час створення самої таблиці командою CREATE TABLE. Синтаксис оператора CREATE TABLE. CREATE INDEX дає можливість додати індекси до існуючих таблиць.

Список стовпців в формі (col1, col2, ...) створює індекс для кількох стовпців. Величини індексів формуються шляхом конкатенації величин зазначених стовпців.

Для стовпців типів CHAR і VARCHAR за допомогою параметра col\_name (length) можуть створюватися індекси, для яких використовується тільки частина стовпчика (для стовпців BLOB і TEXT потрібно вказувати довжину). Команда, наведена нижче, створює індекс, використовуючи перші 10 символів стовпця name:

mysql> CREATE INDEX part\_of\_name ON customer (name(10));

Поскольку большинство имен обычно имеют отличия друг от друга в первых 10 символах, данный индекс не должен быть намного медленнее, чем созданный из столбца name целиком. Кроме того, используя неполные столбцы для индексов, можно сделать файл индексов намного меньше, а это позволяет сэкономить место на диске и к тому же повысить скорость операций INSERT!

слід враховувати, що у версії MySQL 3.23.2 і більш пізніх для таблиць типу MyISAM можна додавати індекс тільки для стовпців, які можуть приймати значення NULL або для стовпців BLOB / TEXT.

**2.3.4 Типи CHAR та VARCHAR**

HARІ VARCHARтипи схожі, але відрізняються тим , як вони зберігаються і витягуються. Вони також відрізняються максимальною довжиною та тим, чи зберігаються проміжні місця.

CHARІ VARCHARтипи оголошені з довжиною , яка вказує максимальну кількість символів , які ви хочете зберегти. Наприклад, CHAR(30)може містити до 30 символів.

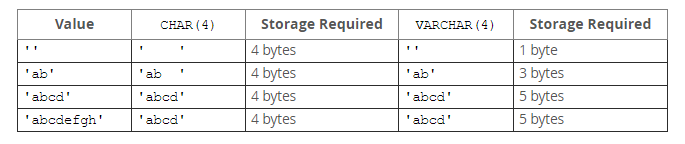
Довжина CHARстовпця фіксується на довжину, яку ви оголошуєте під час створення таблиці. Довжина може бути будь-яким значенням від 0 до 255. Коли CHAR значення зберігаються, вони розміщені праворуч з пробілами до заданої довжини. Після отримання CHARзначень пробіли видаляються, якщо [PAD\_CHAR\_TO\_FULL\_LENGTH](https://dev.mysql.com/doc/refman/8.0/en/sql-mode.html#sqlmode_pad_char_to_full_length)не включений режим SQL.

ачення VARCHARстовпців - це рядки змінної довжини. Довжину можна вказати як значення від 0 до 65,535. Ефективна максимальна довжина a VARCHARзалежить від максимального розміру рядка (65,535 байт, який поділяється між усіма стовпцями) та використовуваного набору символів. На відміну від CHAR, VARCHARзначення зберігаються у вигляді 1-байтової або 2-байтної префікса плюс дані. Префікс довжини вказує кількість байтів у значенні.

У стовпці використовується один байт довжини, якщо для значень потрібно не більше 255 байт, два байти довжини, якщо для значень може знадобитися більше 255 байт. Якщо жорсткий режим SQL не ввімкнено, і ви присвоюєте значення a CHARабо VARCHARстовпцю, що перевищує максимальну довжину стовпця, це значення обрізається таким чином, щоб відповідати, і створюється попередження. Для врізання символів, що не мають простору, ви можете викликати помилку (а не попередження) і придушити введення значення, використовуючи суворий режим SQL. Для VARCHARстовпців проміжки, що перевищують довжину стовпця, усілять перед вставкою і генерується попередження, незалежно від використовуваного режиму SQL. Для CHARстовпців обрізання зайвих пробілів із вставлених значень виконується безшумно, незалежно від режиму SQL.VARCHARЗначення не заповнюються, коли вони зберігаються.

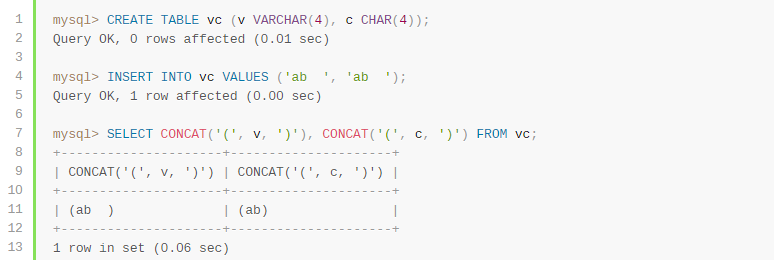
Проміжки проміжку зберігаються, коли значення зберігаються та отримуються відповідно до стандартних SQL.

Наступна таблиця ілюструє відмінності між CHARі VARCHAR, показуючи результат зберігання різних значеньрядків, у CHAR(4) та VARCHAR (4) стовпці (якщо припустити, що стовпець використовує однобайтовий набір символів, такий як latin1).



Значення, показані як збережені в останньому рядку таблиці, застосовуються лише тоді, коли не використовується суворий режим SQL ; якщо включений суворий режим, значення, що перевищують довжину стовпця, не зберігаються , а результати помилки.

InnoDBкодує поля фіксованої довжини, що перевищують 768 байт у довжину, як поля змінної довжини, які можна зберігати поза сторінкою. Наприклад, CHAR(255)стовпець може перевищувати 768 байт, якщо максимальна довжина байтів набору символів перевищує 3, як це стосується utf8mb4.



Значення в CHAR, VARCHARта та TEXTстовпцях сортуються та порівнюються відповідно до порівняння набору символів, призначеного стовпцю.

Щоб визначити атрибут pad для зіставлення, використовуйте INFORMATION\_SCHEMA COLLATIONSтаблицю, в якій PAD\_ATTRIBUTEстовпчик.

# **ВИСНОВОК**

У даній роботі було розроблена база даниз для підприємства виготовлення меблів. Була розглянута предметна область з її основними задачами та визначена її проблематика, були проведені її обстеження та аналіз, що сформували задачу роботи.

База даних для підприємства виготовлення меблів надає безліч переваг а саме

· Підвищилася продуктивність праці і стійкість нашого бізнесу;  
· Оптимізована технологія виробництва м'яких меблів і, як результат, підвищилася якість кінцевого продукту, продукція стала більш конкурентоспроможною;  
· Значно скоротився шлюб кінцевого продукту завдяки впровадженим точкам технологічного контролю на виробництві;  
· Час обробки замовлень скоротилася в кілька разів, причому цей час має тенденцію до подальшого скорочення в міру накопичення бази виробів;  
· Досягнута значна економія матеріалів;  
· Налагодилася злагоджена робота всіх підрозділів;  
· У керівництва підприємства з'явилася ефективна система звітності, що дозволяє своєчасно визначати стан справ в цеху і вчасно відстежувати позитивні і негативні зміни для прийняття обгрунтованих управлінських рішень;

# **СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. MySQL 4 - Строковые функции //http://www.codenet.ru/db/mysql/mystring4.php

2. <http://www.mysql.ru/news/>

3. Навчальний методичний посібник «Теорія систем і системний аналіз», Томськ: Томський міжвузівський центр дистанційної освіти, 2003, 34 с.

4. авчальний посібник «Моделі та проектування баз даних», В.Д. Сибільов, Томськ, 2002.