**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 підприємства виготовлення меблів.**

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

**Спеціальність:** 051 «Економіка»

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

**Освітній ступінь:**  бакалавр

|  |  |  |
| --- | --- | --- |
| **Науковий керівник:**  к.е.н., доц. Ярема О.Р.  (науковий ступінь, посада, прізвище, ініціали)  \_\_\_\_\_\_\_\_\_\_\_\_\_“\_\_\_” травня 2020 р.  (підпис) |  | **Виконавець:**  студент групи УФЕ-31 с  Гут Н.О  \_\_\_\_\_\_\_\_\_\_\_\_ “\_\_\_”травня 2020 р.  (підпис) |

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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис, ПІП членів комісії)

**ЛЬВІВ 2020**

**Зміст**

[**Вступ** 2](#_Toc43736568)

[**Розділ 1 Аналіз вимог** 5](#_Toc43736569)

[**1.1.** **Постановка завдання.** 5](#_Toc43736570)

[**1.2.** **Розробка моделі варіантів використання веб-сайту** 8](#_Toc43736571)

[**1.3.** **Аналіз засобів реалізації (техніко-економічне обґрунтування вибору);** 12](#_Toc43736572)

[**Розділ 2 Розробка бази даних** 14](#_Toc43736573)

[**2.1 Опис моделі даних** 15](#_Toc43736574)

[**2.2 Нормалізація реляційних відношень** 19](#_Toc43736575)

[**2.3 Визначення типів даних** 21](#_Toc43736576)

[**2.4 Обмеження цілісності даних** 23](#_Toc43736577)

[**2.5 Запити до таблиць бази даних** 23](#_Toc43736578)

[**Висновки** 26](#_Toc43736579)

[**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ** 27](#_Toc43736580)

[**ДОДАТКИ** 29](#_Toc43736581)

**Вступ**

**Актуальність теми.** Про гостроту проблеми, що розглядається говорить той факт, що більшу частину свого часу адміністратор фабрики витрачає на оформлення різної документації та звітів, а так само на пошук інформації про наявність на фабриці певних видів виробів, на це затрачається багато часу та зусиль. Внесення всіх цих даних в єдину базу дозволить систематизувати їх, а також спростити співробітникам доступ до необхідної для роботи інформації, що в наслідок тягне за собою заощадження коштів і часу, потрібних для вилучення необхідних відомостей, що повинно позитивно позначитися на ефективності роботи фабрики. Використання баз даних для спрощення роботи магазинів, фабрик, тур-фірм, готелів і тд. набуло широкого поширення в сучасному світі. Кінцевою метою розробки бази даних меблевої фабрики та автоматизованої системи для роботи з нею є підвищення якості обслуговування клієнтів і спрощення роботи для працівників.

Невід’ємною складовою життя людини є інформація. З розвитком суспільства збільшується обсяг інформації, при цьому інформація перетворилась в один з найбільш важливих ресурсів, який розглядається поряд із трудовими, матеріальними та енергетичними ресурсами. Фактично, інформація стала чинником, що визначає ефективність будь-якої сфери діяльності людини. Розвиток технологій набрав масштабних темпів. Колись для пошуку інформації потрібно було перечитати сотні, а то і тисячі книжок та листів в архівах чи бібліотеках. На даний час все шо потрібно для пошуку це ввести в браузері ключові слова і весь список потрібних для вас речей перед очима. Не виходячи з дому можна замовити їжу, книжку і багато іншого. І все це стало доступним після розробки і розвитку глобальної мережі. Кожен магазин старається вплинути на користувача через сайт. Адже вдома коли на вулиці погана погода, можна переглядати список товарів, фотографію ціну і відгуки людей, які замовили цей продукт і саме головне після замовлення цей товар доставлять під двері. Тому і виникла потреба у застосуванні найбільш перспективних комп’ютерних технологій для роботи з інформацією різного типу. За останні десятиліття бази даних стали основою інформаційних систем і докорінно змінили методи роботи багатьох організацій. Розвиток технології баз даних призвело до створення досить потужних і зручних в експлуатації програм. Вони являють собою зручний і надійний спосіб зберігання номенклатурних даних продукції, відомостей про клієнтські бази та іншої інформації, чиї обсяги перевищують можливості обліку за допомогою звичайних таблиць.

Сучасною формою інформаційних систем є банки даних, до складу яких входять обчислювальна система, одна чи декілька баз даних (БД), система управління базами даних та набір прикладних програм (ПП). В БД використовуються класичні моделі даних: ієрархічна, мережева, реляційна та об’єктно-орієнтовна. Ядром будь-якої бази даних є модель даних. Модель даних являє собою безліч структур даних, обмежень цілісності і маніпулювання даними. За допомогою моделі даних можуть бути представлені об'єкти предметної області та взаємозв'язки між ними.

**Мета і завдання дослідження**.

Метою даної роботи є обґрунтування теоретичних основ та реалізація бд орієнтованої системи покупки меблів онлайн. На початковому етапі практичної частини розроблялась база даних за допомогою мови запитів -MySQL.

Для вирішення даної мети були поставленні наступні завдання:

1. Проаналізувати предметну область інформаційної системи
2. Описати всі аспекти використання баз даних
3. Проаналізувати актуальність створення в подальшому веб-інтерфейсу для інформаційної системи.

Об’єктом дослідження виступає база даних, як основний елемент інформаційної системи меблевої фабрики.

Основним предметом дослідження є програмне середовище, яке застосовується для автоматизації меблевої фабрики.

**Практичне значення отриманих результатів**.

Базою даних користуватимуться власник і працівники меблевої фабрики.

**Використане програмне забезпечення**.

Для досягнення результату використовувалось програмне середовище IntellijIdea.

**Структура роботи**.

Курсова складається з двох розділів («Аналіз предметної області» та «Розробка бази даних»), а також з висновків, списку використаних джерел та додатків. Загальний обсяг роботи – 33 сторінок.

**Розділ 1 Аналіз вимог**

* 1. **Постановка завдання.**

Основним завданням даної роботи полягає в тому, щоб зробити аналіз актуальності створення інформаційної системи для меблевої фабрики в Україні та у разі позитивного результату створити інформаційну системи(веб-сайт) для даної фабрики.

В курсовому проекті за мету взято створення бази даних, яка має надати можливість зберігання великої кількості інформації, забезпечувати швидкий пошук інформації для клієнта. База даних повинна реалізовувати змогу обновлення та добавлення, або видалення даних які в ній зберігаються, а також зручне та надійне користування, та доступ усіх необхідних програм до неї.

При розробленні бази даних необхідно створити таблиці в SQL Workbanch, та заповнити їх необхідними полями(атрибутами).

Але до реалізації таблиць необхідно розробити модель в model.erwin, в якій буде зображено зв’язки між таблицями та основну схему побудови таблиць.

Після цього треба створити запити завдяки мові mysql, та перевірити їх роботу на видання різної інформації.

База даних (БД) — впорядкований набір логічно взаємопов'язаних даних, що використовується спільно, та призначений для задоволення інформаційних потреб користувачів. У технічному розумінні включно й система керування БД.

Головним завданням БД є гарантоване збереження значних обсягів інформації та надання доступу до неї користувачеві або ж прикладній програмі. Таким чином БД складається з двох частин : збереженої інформації та системи управління нею. З метою забезпечення ефективності доступу записи даних організовують як множину фактів (елемент даних).

Дані – це інформація, відомості, показники, необхідні для ознайомлення з ким-, чим-небудь, для характеристики когось, чогось або для прийняття певних висновків, рішень. В базах даних важливу роль відіграє інформація - Інформація — абстрактне поняття, що має різні значення залежно від контексту.

Кожна створена база даних має свою предметну область. Предметна область – це необхідний для розробки бази даних об’єкт, який має в собі дані, які будуть зберігатися в базі даних.

Модель даних — абстрактне представлення реального світу, що відображає тільки ті об'єкти, що безпосередньо стосуються програми. Це, як правило, визначає специфічну групу об'єктів, їх атрибутивне значення і відношення між ними.

Відомі два підходи до організації інформаційних масивів: файлова організація та організація у вигляді бази даних. Файлова організація передбачає спеціалізацію та збереження інформації, орієнтованої, як правило, на одну прикладну задачу, та забезпечується прикладним програмістом. Така організація дозволяє досягнути високої швидкості обробки інформації, але характеризується рядом недоліків. Характерна риса файлового підходу - вузька спеціалізація як обробних програм, так і файлів даних, що служить причиною великої надлишковості, тому що ті самі елементи даних зберігаються в різних системах. Оскільки керування здійснюється різними особами (групами осіб), відсутня можливість виявити порушення суперечливості збереженої інформації. Розроблені файли для спеціалізованих прикладних програм не можна використовувати для задоволення запитів користувачів, які перекривають дві і більше області. Крім того, файлова організація даних внаслідок відмінностей структури записів і форматів передання даних не забезпечує виконання багатьох інформаційних запитів навіть у тих випадках, коли всі необхідні елементи даних містяться в наявних файлах. Тому виникає необхідність відокремити дані від їхнього опису, визначити таку організацію збереження даних з обліком існуючих зв'язків між ними, яка б дозволила використовувати ці дані одночасно для багатьох застосувань. Вказані причини обумовили появу баз даних. База даних може бути визначена як структурна сукупність даних, що підтримуються в активному стані та відображає властивості об'єктів зовнішнього (реального) світу. В базі даних містяться не тільки дані, але й описи даних, і тому інформація про форму зберігання вже не схована в сполученні "файл-програма", вона явним чином декларується в базі.

База даних орієнтована на інтегровані запити, а не на одну програму, яку випадку файлового підходу, і використовується для інформаційних потреб багатьох користувачів. В зв'язку з цим бази даних дозволяють в значній мірі скоротити надлишковість інформації. Перехід від структури БД до потрібної структури в програмі користувача відбувається автоматично за допомогою систем управління базами даних (СУБД).

Системи управління базами даних – це програмні засоби, за допомогою яких можна створювати бази даних, заповнювати їх та працювати з ними. У світі існує багато різноманітних систем управління базами даних. Багато з них насправді є не закінченими продуктами, а спеціалізованими мовами програмування, за допомогою яких кожний, хто вивчить дану мову, може сам створювати такі структури, які йому потрібні, і вводити в них необхідні елементи управління. До таких мов відносяться Clipper, Paradox, FoxPro та інші.

* 1. **Розробка моделі варіантів використання веб-сайту**

Для того, щоб більш точно зрозуміти як повинна працювати система, все частіше використовується опис функціональності системи через варіанти використання (Use Case або прецеденти). Варіанти використання це – опис послідовності дій, які може здійснювати система у відповідь на зовнішні впливи користувачів або інших програмних систем. Варіанти використання відображають функціональність системи з точки зору отримання відчутного результату для користувача, тому вони точніше дозволяють ранжувати функції за значимістю одержуваного результату.

Варіанти використання призначені в першу чергу для визначення функціональних вимог до системи і керують усім процесом розробки. Всі основні види діяльності такі як аналіз, проектування, тестування виконуються на основі варіантів використання. Під час аналізу і проектування варіанти використання дозволяють зрозуміти як результати, які хоче отримати користувач впливають на архітектуру системи і як повинні поводитися компоненти системи, для того щоб реалізувати потрібну для користувача функціональність. Варіанти використання ілюструються за допомогою діаграми варіантів використання. Діаграма варіантів використання складається з акторів, для яких система виробляє дію і власне варіанти використання (Use Case), які описують те, що актор хоче отримати від системи. Актор позначається значком чоловічка, а варіанти використання - овалом. Додатково в діаграми можуть бути додані коментарі.

Відповіді на такі питання дозволять визначити акторів, що взаємодіють з системою:

* хто взаємодіє з системою або використовує систему;
* хто передає чи приймає інформацію в/з системи;
* хто є зовнішнім по відношенню до системи.

Кожен варіант використання показує, як конкретний актор використовує

систему і надалі розширюється діаграмами станів і послідовності дій.

Між акторами і варіантами використання можуть бути різні види взаємодії. Основні види взаємодії наступні: − Проста асоціація - відображається лінією між актором і варіантом використання (без стрілки). Відображає зв'язок актора і варіанта використання. − Спрямована асоціація - те ж що і проста асоціація, але показує, що варіант використання ініціалізуєтся актором. Позначається стрілкою. − Спадкування - показує, що нащадок успадкує атрибути і поведінку свого прямого предку. Може застосовуватися як для акторів, так для варіантів використання. − Розширення (extend) - показує, що варіант використання розширює базову послідовність дій і вставляє власну послідовність. При цьому на відміну від типу відносин "включення" розширена послідовність може здійснюватися в залежності від певних умов. − Включення (include) - показує, що варіант використання включається в базову послідовність і виконується завжди (на рисунку не показаний). Існують і інші види взаємодії, але вони менш важливі і рідше застосовуються. Для створення діаграми використання можна використовувати редактор діаграм UML StarUML, або інший. В проекті також необхідно деталізувати кожний варіант використання, тобто виконати його уточнення за допомогою специфікації варіанту використання. Основний потік описує «ідеальний» хід розвитку подій у варіанті використання. Альтернативні потоки можуть перехоплювати помилки, відгалуження і переривання основного потоку. Основний потік завжди починається з дій головного актора, спрямованих на ініціацію варіанту використання.

Вдалим способом початку потоку можна визначити таку форму запису:

Прецедент починається, коли <актор> <дія>.

Кожен етап потоку варіанту використання повинен бути виражений в такій формі: <Номер> <хтось> <робить деяку дію>. Використання пасивного стану для опису етапу є невірним: «Вводяться дані покупця».

Альтернативні потоки можуть бути замінені за допомогою ключового

слова «Якщо».

Основний потік:

1 Варіант використання починається коли Клієнт вибирає заявку зі списку.

2 Якщо Клієнт вибирає "видалити заявку".

2.1Система призначає заявці статус "Відхилена".

Логічна структура сайту може бути представлена у вигляді чотирьох основних моделей: лінійна модель, моделі "ґрати", "дерево" (ієрархія) і "павутина". Крім того, існують різні комбінації на базі основних моделей, що дозволяють реалізувати будь-яку логічну структуру сайту.

Строго лінійна структура сайту використовується досить рідко, оскільки дозволяє відвідувачам сайту переходити тільки від попередньої сторінки до наступної, і ніяк інакше. Таке подання інформації з однієї сторони полегшує процес взаємодії з відвідувачами: уже на етапі створення сайту можна досить точно прогнозувати поведінку відвідувачів. З іншої сторони подібна організація структури сайту не залишає відвідувачам волі вибору, що може їм не сподобатися.

Тому на практиці лінійна структура сайту застосовується з деякими видозмінами, такими як пропуск певних сторінок, бічні відгалуження. Однак надмірне захоплення відгалуженнями перетворює лінійну структуру в ієрархічну. Застосовується лінійна структура сайту для різних онлайн-презентацій, опису покрокових процесів. Основна перевага застосування такої моделі організації сайту полягає в передбачуваності поведінки відвідувача на сайті, завдяки чому можливо вгадати дії відвідувачів ще на етапі створення сайту. Лінійна структура сайту не підходить для сайтів з великим обсягом інформації. Для більш зручного подання великих обсягів інформації використовуються інші моделі логічної структури, наприклад, «ґрати».

Логічна структура сайту "ґрати" нерідко зустрічається при організації інтернет-магазинів. Ґрати являють собою двонаправлену лінійну структуру, у який присутні й горизонтальні, і вертикальні взаємозв'язки між сторінками.

Наприклад, якщо в каталозі інтернет магазина товари об'єднані в категорії за видом (штани, куртки, сукні) і в категорії за ціною (дешеві, середні за ціною, дорогі), то така структура дозволяє відвідувачам переглядати товари за обома ознаками (вид і ціна). Як наслідок, відвідувачі почувають себе вдоволеними завдяки наданій волі вибору.

Логічна структура сайту "дерево" - це модель організації сайту що зустрічається найбільш часто. "Дерево" - ієрархічна структура - дозволяє відвідувачам сайту за своїм бажанням управляти глибиною відвідування сайту. Відвідувачі можуть заходити тільки на сторінки верхніх рівнів ієрархії або ж «спуститися» до нижніх рівнів. При цьому воля вибору відвідувачів залежить від «ширини дерева».

Якщо відвідувачам для досягнення кінцевої мети необхідно зробити занадто багато клацань миші, то ієрархічна структура сайтів може виявитися в цьому випадку занадто вузькою. Відвідувачів дратує нескінченне «клацання», що не приносить очікуваних результатів.

У той же час, занадто широке «дерево», ґрунтуючись на дуже великій кількості варіантів вибору, може змусити відвідувачів витратити багато часу на вивчення наданих варіантів. Це також приводить до роздратування відвідувачів і вони залишають сайт.

Якщо логічна структура сайту заплутана й незрозуміла, те таку модель організації сайту прийнято називати "павутиною". Така структура сайту може з'явитися внаслідок неправильного проектування сайту або ж навмисно, для забезпечення більш виразного подання інформації. Приміром, статті про технічні вироби можуть посилатися на схеми цих виробів, норми й вимоги до виробництва виробів, ряд інших документів. Такий підхід дозволяє відвідувачам одержати максимально повну інформацію.

Якщо модель організації сайту обрана вірно, а логічна структура сайту відповідає потребам відвідувачів, то юзабіліті і ефективність такого сайту будуть високими. Тому до розробки структури сайту варто підійти найретельнішим чином.

З вище сказаної інформації даного параграфу можна зробити висновок, що при створенні веб-сайту для меблевої фабрики буде раціонально використовувати Логічна структура сайту "дерево".

* 1. **Аналіз засобів реалізації (техніко-економічне обґрунтування вибору);**

Було вирішено обрати наступні технології для створення бази даних для меблевої фабрики:

* мова запитів Sql.

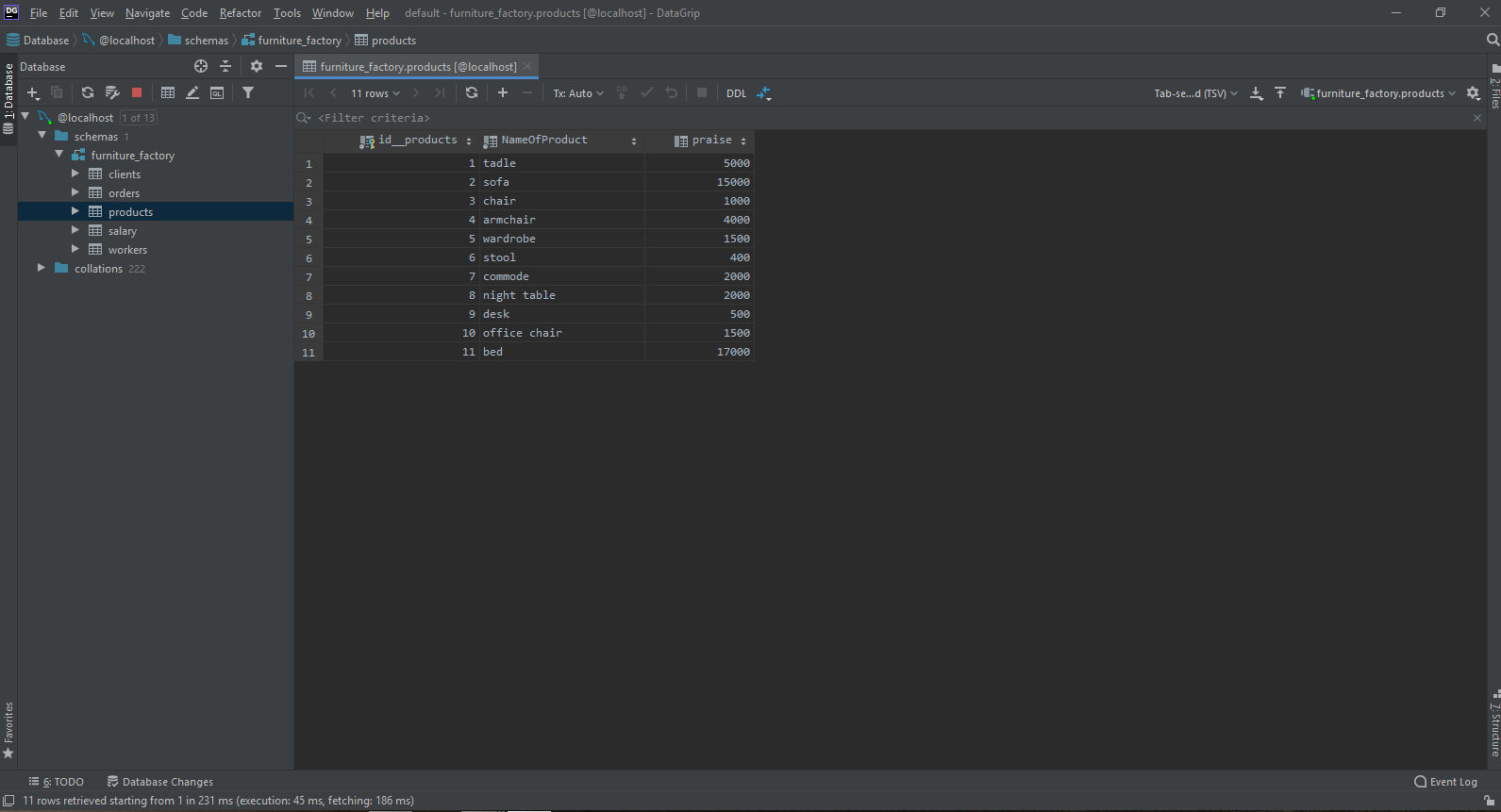
Чому саме MySql? Відповідь очевидна.

Переваги  даної мови[:](http://www.znannya.org/?view=concept:302)

* [простота у встановленні та використанні;](http://www.znannya.org/?view=concept:302" \o "<strong id=)
* [підтримується необмежена кількість користувачів, що одночасно працюють із БД;](http://www.znannya.org/?view=concept:302" \o "<strong id=)
* [кількість рядків у таблицях може досягати 50 млн.;](http://www.znannya.org/?view=concept:302" \o "<strong id=)
* [висока швидкість виконання команд;](http://www.znannya.org/?view=concept:302" \o "<strong id=)
* [наявність простої і ефективної системи безпеки.](http://www.znannya.org/?view=concept:302" \o "<strong id=)
* Програмне середовище DataGrip

Переваги даної програми:

* + Зручний інтерфейс прграми;
  + Можливість створювати таблиці та наповняти їх без використання жодної лінійки коду.
  + Можливість візуально оглядати таблиці та переглядати звязки між ними;
  + Можливість одночасно працювати з декількома базами даних та поєднювати їх між собою.
  + Можливість імортувати та експортувати файли.
  + Можливість імпортувати системні настройки та налаштовувати систему індивідуально під власні потреби.
  + Дана програма надає логічні підказки щодо синтаксису MySQl та будування структури бази даних.
  + В даній програмі з легкістю можна зберегти схему чи таблицю в потрібному форматі.



*Рис.1.1 програмний інтерфейс DataGrip*

У зв’язку з вище перерахованими перевагами було прийнято рішення обирати дані технології.

**Розділ 2 Розробка бази даних**

Проектована база даних повинна володіти певними властивостями.

* + Цілісність бази даних досягається внаслідок введення обмежень, пов'язані з нормалізацією БД. У кожен момент часу існування БД відомості, що містяться в ній, повинні бути несуперечливі.
  + Відновлюваня - можливість відновлення БД після збою системи або окремих видів поломок системи.
  + Безпека - припускає захист даних від навмисного і ненавмисного доступу, модифікації або руйнування.
  + Ефективність. Властивість ефективності зазвичай оцінюється двома параметрами: мінімальний час реакції на запит користувача і мінімальні потреби в пам'яті, а також поєднанням цих параметрів.
* свою чергу, система управління базами даних, за допомогою якої користувачі можуть визначати, створювати і підтримувати базу даних, а також здійснювати до неї контрольований доступ, ділиться на два класи - система загального призначення та спеціалізована система. СУБД загального призначення не орієнтовані на якусь конкретну предметну область або на інформаційні потреби конкретної групи користувачів. Кожна система такого роду реалізується як програмний продукт, здатний функціонувати на деякій моделі ЕОМ у певній операційній обстановці. СУБД загального призначення володіє засобами налаштування на роботу з конкретною БД в умовах конкретного застосування. У деяких ситуаціях СУБД загального призначення не дозволяють добитися необхідних проектних та експлуатаційних характеристик (продуктивність, зайнятий обсяг пам'яті та інше).

**2.1 Опис моделі даних**

База даних (БД) - упорядкований набір логічно взаємопов'язаних даних, що використовується спільно, та призначений для задоволення інформаційних потреб користувачів. У технічному розумінні включно й система управління БД. Система управління базами даних (СУБД) - це комплекс програмних і мовних засобів, необхідних для створення баз даних, підтримання їх в актуальному стані та організації пошуку в них необхідної інформації.

Централізований характер управління даними в базі даних передбачає необхідність існування деякої особи (групи осіб), на яку покладаються функції адміністрування даними, що зберігаються в базі.

Головним завданням БД є гарантоване збереження значних обсягів інформації та надання доступу до неї користувачеві або ж прикладній програмі. Таким чином БД складається з двох частин: збереженої інформації та системи управління нею. З метою забезпечення ефективності доступу записи даних організовують як множину фактів (елемент даних).

Реляційна модель орієнтована на організацію даних у вигляді двовимірних таблиць. Кожна реляційна таблиця являє собою двовимірний масив і має наступні властивості:

* кожен елемент таблиці - один елемент даних;
* всі осередки в стовпчику таблиці однорідні, тобто всі елементи в стовпчику мають однаковий тип (числовий, символьний тощо);
* кожен стовпчик має унікальне ім'я;
* однакові рядки в таблиці відсутні;
* порядок проходження рядків і стовпчиків може бути довільним.

Об'єктна СУБД ідеально підходить для інтерпретації складних даних, на відміну від реляційних СУБД, де додавання нового типу даних досягається ціною втрати продуктивності або за рахунок різкого збільшення термінів і вартості розробки додатків. Об'єктна база, на відміну від реляційної, не вимагає модифікації ядра при додаванні нового типу даних. Новий клас і його екземпляри просто надходять у зовнішні структури бази даних. Система управління ними залишається без змін.

Процес розробки бази даних являє собою складний процес проектування відображення опису предметної області у схему внутрішньої моделі даних. Перебіг цього процесу є послідовністю більш простих процесів проектування менш складних відображень. Ця послідовність у процесі проектування весь час уточнюється, вдосконалюється таким чином, щоб були визначені об'єкти, їх властивості та зв'язки, які будуть потрібні майбутнім користувачам системи.

Для реалізації серверної частини було обрано СУБД MySQL. Для управління базами даних використовується програма DataGrip.

DataGrip підтримує безліч різних СУБД:

* PostgreSQL
* [MySQL](https://www.jetbrains.com/ru-ru/datagrip/features/mysql.html)
* Oracle Database
* SQL Server
* Azure
* Amazon Redshift
* SQLite
* DB2
* H2
* Sybase
* Exasol
* Apache Derby
* MariaDB
* HyperSQL
* Snowflake
* Cassandra
* ClickHouse
* Greenplum
* Apache Hive
* Vertica
* MongoDB

У процесі розробки проекту у базі даних «furniture\_factory» створено п'ять таблиць:

* workers – таблиця, яка містить дані про працівників меблевої фабрики.
* id\_worker – унікальний номер
* name – містить ім’я працівника
* phone – містить контактний телефон працівника
* age – поле, у якому вказано вік працівника
* position – поле, у якому вказана позиція працівника.
* salary – таблиця, яка містить дані про заробітню плату працівників меблевої фабрики.
* id– унікальний номер
* id\_worker – містить айді працівнка – вторинний ключ
* size\_salary – поле, у якому вказано розмір заробітної плати працівника
* products – таблиця, яка містить дані про продукцію меблевої фабрики
* id\_products – унікальний номер виду продукції – вторинний ключ
* nameOfProduct – назва продукції
* praise – ціна продукції
* clients – таблиця, яка містить дані про клієнтів меблевої фабрики
* id\_client – унікальний ідентифікаційний номер клієнта – вторинний ключ
* name – П. І. клієнта
* phone – поле, де вказано контактний номер клієнта
* orders – проміжна таблиця, яка містить дані про замовлення клієнтів меблевої фабрики
* id\_ order – унікальний номер замовлення
* id\_product
* id\_client

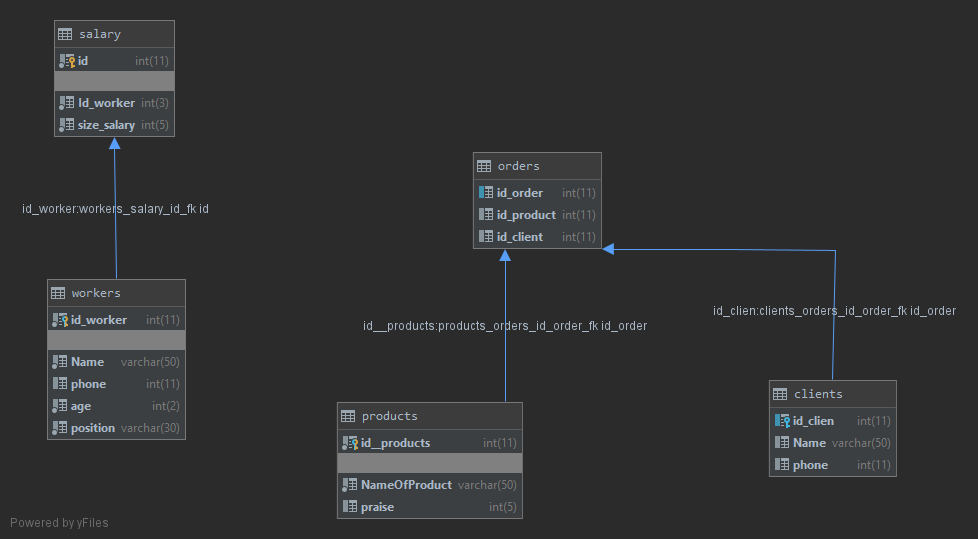
Для відображення візуального процесу предметної області існує модель

сутність-зв'язок. Це модель даних, яка дозволяє описувати концептуальні схеми за допомогою узагальнених конструкцій блоків. ER-модель — це мета-модель даних, тобто засіб опису моделей даних. Існує ряд моделей для представлення знань, але одним з найзручніших інструментів уніфікованого представлення даних, незалежного від програмного забезпечення, що його реалізує, є модель «сутність-зв'язок». Важливим є той факт, що з моделі «сутність-зв'язок» можуть бути породжені всі існуючі моделі даних (ієрархічна, мережева, реляційна, об'єктна), тому вона є найзагальнішою.

Розробка бази даних за методом сутність-зв'язок складається з наступних етапів:

1. визначення сутностей;
2. визначення зв'язків;
3. визначення атрибутів;
4. визначення ключів сутностей;
5. визначення ступеня зв'язку;
6. визначення класу приналежності.

Для кожної таблиці унікальний номер – це первинний ключ. На рисунку 2.1 зображена діаграма сутностей та зв’язків.



*Рис. 2.1 Діаграма сутностей та зв’язків*

**2.2 Нормалізація реляційних відношень**

Нормалізація - це розбивка таблиці на дві або більш, які характеризуються кращими властивостями при доповненні, зміні і вилученні даних. Процес нормалізації є одним з найважливіших, що визначає процедури приведення моделі бази даних до виду, формує ефективну базу даних, що реалізується в коректному поданні і високошвидкісний обробці даних. Щоб виконати нормалізацію моделі бази даних, розробнику необхідно представити модель у вигляді сутності з набором атрибутів, які се будуть характеризувати, визначаючи інформаційне опис предметної області. У цій суті будуть виникати поєднання атрибутів, які представляються функціональними залежностями. Під функціональною залежністю розуміють відношення з безліччю атрибутів, де одна сукупність атрибутів кожним своїм значенням однозначно функціонально залежить від єдиного значення сукупності інших атрибутів. У теорії реляційних баз даних прийнято виділяти таку послідовність нормальних форм:

- перша нормальна форма, 1NF (First Normal Form);

- друга нормальна форма, 2NF (Second Normal Form); 13

- третя нормальна форма, ЗNF (Third Normal Form);

- нормальна форма Бойса – Кодда, ВСNF (Brice – Codd Normal Form);

- четверта нормальна форма, 4NF (Fourth Normal Form);

- п'ята нормальна форма, 5NF (Fifth Normal Form).

База меблевої фабрики розроблена і проаналізована на 3 нормальні форми:

* Перша нормальна форма. Відношення відповідає 1NF тоді, коли на перетині кожного стовпця і кожного рядка знаходяться тільки елементарні (неподільні) значення атрибутів і не містяться групи, що повторюються.
* Друга нормальна форма. Відношення знаходиться в 2NF, якщо виконуються обмеження 1NF і кожен описовий атрибут функціонально повно залежить від первинного ключа (у тому числі і складеного).
* Третя нормальна форма. Відношення знаходиться у ЗNF, якщо виконуються обмеження 2NF і всі описові атрибути відношення взаємно незалежні і повністю залежать від первинного ключа, тобто кожний описовий атрибут не транзитивно залежить від ключа.

**2.3 Визначення типів даних**

База даних – це структурований обсяг інформації, що містить таблиці, які у свою чергу складаються із записів, а записи – з полів. Поле має два атрибути – ім'я і тип даних. Нижче наведено типи даних, які використовуються у створеній базі даних – «furtine\_factory»:

- INT – тип даних, що представляє ціле число, діапазон від -2 147 483 648 до 2 147 483 647;

- VARCHAR – набір символьних даних, може зберігати не більше 255 символів.

*Таблиця 2.1*

*Типи даних таблиці «clients»*

|  |  |  |  |
| --- | --- | --- | --- |
| № | Ім’я атрибуту | Тип даних | Розмір поля |
| 1. | id\_client (PK FK) | INT | 11 |
| 2. | name | VARCHAR | 50 |
| 3. | phone | VARCHAR | 11 |

*Таблиця 2.2*

*Типи даних таблиці «workers»*

|  |  |  |  |
| --- | --- | --- | --- |
| № | Ім’я атрибуту | Тип даних | Розмір поля |
| 1. | id\_worker(PK,FK) | INT | 11 |
| 2. | name | VARCHAR | 50 |
| 3. | phone | INT | 11 |
| 4. | age | INT | 2 |
| 5. | position | VARCHAR | 30 |

*Таблиця 2.3*

*Типи даних таблиці «salary»*

|  |  |  |  |
| --- | --- | --- | --- |
| № | Ім’я атрибуту | Тип даних | Розмір поля |
| 1. | id (PK) | INT | 11 |
| 2. | Id\_worker | INT | 3 |
| 3. | size\_salary | INT | 5 |

*Таблиця 2.4*

*Типи даних таблиці «products»*

|  |  |  |  |
| --- | --- | --- | --- |
| № | Ім’я атрибуту | Тип даних | Розмір поля |
| 1. | Id\_products(PK,FK) | INT | 11 |
| 2. | NameOfProducts | VARCHAR | 50 |
| 3. | products | INT | 5 |

*Таблиця 2.5*

*Типи даних таблиці «orders»*

|  |  |  |  |
| --- | --- | --- | --- |
| № | Ім’я атрибуту | Тип даних | Розмір поля |
| 1. | id\_order (PK) | INT | 11 |
| 2. | id\_product | INT | 11 |
| 3. | id\_client | INT | 11 |

**2.4 Обмеження цілісності даних**

Обмеження цілісності — це правила, які обмежують усі можливі стани бази даних, а також переходи з одного стану в інший. Таким чином, обмеження цілісності визначають множину «допустимих» станів і переходів між ними. База даних перебуває в цілісному стані, якщо вона відповідає всім визначеним для неї вимогам цілісності.

У навчальній базі “t&q\_base” застоcовано наступні обмеження щодо первинних ключів:

* not null;
* primary key;
* auto\_increment;

Not null – обмеження, яке застосовується до всіх полів таблиць бази даних.

**2.5 Запити до таблиць бази даних**

Навчальна інформаційна система реалізує такі запити в базі даних:

# 1)Показати всіх працівників фабрики;

SELECT \* from workers;

#2)Показати всіх працівників фабрики, які молодші за 30років;

select \* from workers where age < 30;

#3)Показати ім'я і телефон директора та власника фабрики

select Name, phone, Position from workers where Position ='owner' or Position='director';

#4) Показати перших 5 клієнтів по алфафіну

select \* from clients order by Name asc limit 5;

#5)Показати клієнта чий номер закінчується на 35

select \*from clients where Phone like '%35';

#6)Показати імя номер телефону працівників: Vasil Borey, Natalia Steslavska

select Name, phone from workers

where Name like 'Vasil Borey'

or Name like 'Valentyn Perol';

#7)показати ті товари і ціни на них, які коштуюте дороще 5000;

select NameOfProduct, praise from products where praise > 5000;

#8)Показати ті товари які коштують дороще 1200 і дешевше 2000

select NameOfProduct, praise from products where praise between 1200 and 2000;

#9) Порахувати яка буде загальна сума всіх товарів які є на фабриці (по 1шт)

select Sum(praise) from products ;

#10) Показати найдорощий товар фабрики

select NameOfProduct, Max(praise) from products;

#11) Показати всіх працівників які мають з/п більше 23000

select Name, Position from workers join salary s on workers.id\_worker = s.id

where `size\_salary`>23000

group by Name, Position;

#12) Показати 5 наймолодших працівників фабрики

select \* from workers

group by Name

order by min(Age)

limit 5;

#13) Показати інформацію про замовлення на суму більшу рівну 5000

select \* from orders join products p on orders.id\_order = p.id\_\_products

join clients c on orders.id\_order = c.id\_clien

where praise >= 5000

group by id\_clien ;

#14) Показати всі товари фабрики від найдорощого до найдешевшого

select \* from products

group by id\_\_products

order by min(praise) desc;

#15) збільшити з/п на 15% працівникам з айдішками 2,4,5

update salary

set `size\_salary` = `size\_salary`\*0.15+`size\_salary` where id like '2' or id like '4' or id like '5';

#16) збільшити ціну на стіл на 15%

update products

set praise = praise+ praise\*0.05 where NameOfProduct like 'table';

**Висновки**

У процесі виконання дослідження мета була досягнута, а завдання вирішені, а саме успішно виконано теоретичні і практичне завдання, (побудова бд для інформаційної системи для меблевої фабрики.)

У роботі проаналізовано всі аспекти і можливості використання MySQL. Для закріплення набутих знань розроблено базу даних меблевої фабрики, а також прийнято рішеня, що буде доцільно в майбутньому створити веб-додаток, який дасть змогу замовити меблі онлайн.

На першому етапі теоретичного завдання була проаналізована предметна область, визначені функціонал та методи користування сайтом.

* другому розділі реалізована база даних двома методами для проектування БД: метод Сутність - зв'язок і метод нормалізації відношень. А саме: детальний аналіз предметної області, ER-модель, декомпозиція та нормалізація відношень методом Сутність-зв'язок. Також створено універсальне відношення, визначені первинний і вторинний ключі. База даних має 5 таблиць («products», «workers», «salary», «clients», «orders»). За методом нормалізації визначено, що база даних «furniture\_factory» має три нормальні форми (1НФ, 2НФ, 3НФ).

Отож, враховуючи усе вище сказане, можна стверджувати про важливу роль інформаційних систем та веб сайтів, зокрема побудована інформаційна система позитивно впливає на розвиток бізнесу, а саме клієнт має змогу онлайн ознайомитись із запропонованими послугами, їх цінами, та працівниками, які надають ті чи інші послуги.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Мартин Грабер :SQL; Видавництво «Лори», 2016. -643 с. ;
2. Шнайдер Р. Microsoft SQL Server 6.5. Проектирование высокопроизводительных баз данных, – М.: Лори, 2010. – 361 c. 37. Яргер Р.Дж., Риз Дж., Кинг Т. MySQL и mSQL: Базы данных для небольших предприятий и Интернета; СПб: СимволПлюс, 2013. – 560 c.
3. Бази даних та інформаційні системи — Вікі ЦДПУ [Електронний ресурс]. – Режим доступу: <https://cutt.ly/CyTXnR1>.
4. БАЗЫ ДАННЫХ - УРОК 3. РЕЛЯЦИОННЫЕ БАЗЫ ДАННЫХ

[Електронний ресурс]. – Режим доступу: <https://site-do.ru/db/db3.php>.

1. ОСНОВИ БАЗ ДАНИХ [Електронний ресурс]. – Режим доступу:

<http://www.kievoit.ippo.kubg.edu.ua/kievoit/2013/118/118.html>.

1. ЧТО ТАКОЕ СУБД - RU-CENTER [Електронний ресурс]. – Режим доступу: <https://www.nic.ru/help/chto-takoe-subd_8580.html>.
2. РОЗДІЛ 1 . МОДЕЛЮВАННЯ РЕЛЯЦІЙНОЇ СТРУКТУРИ БД

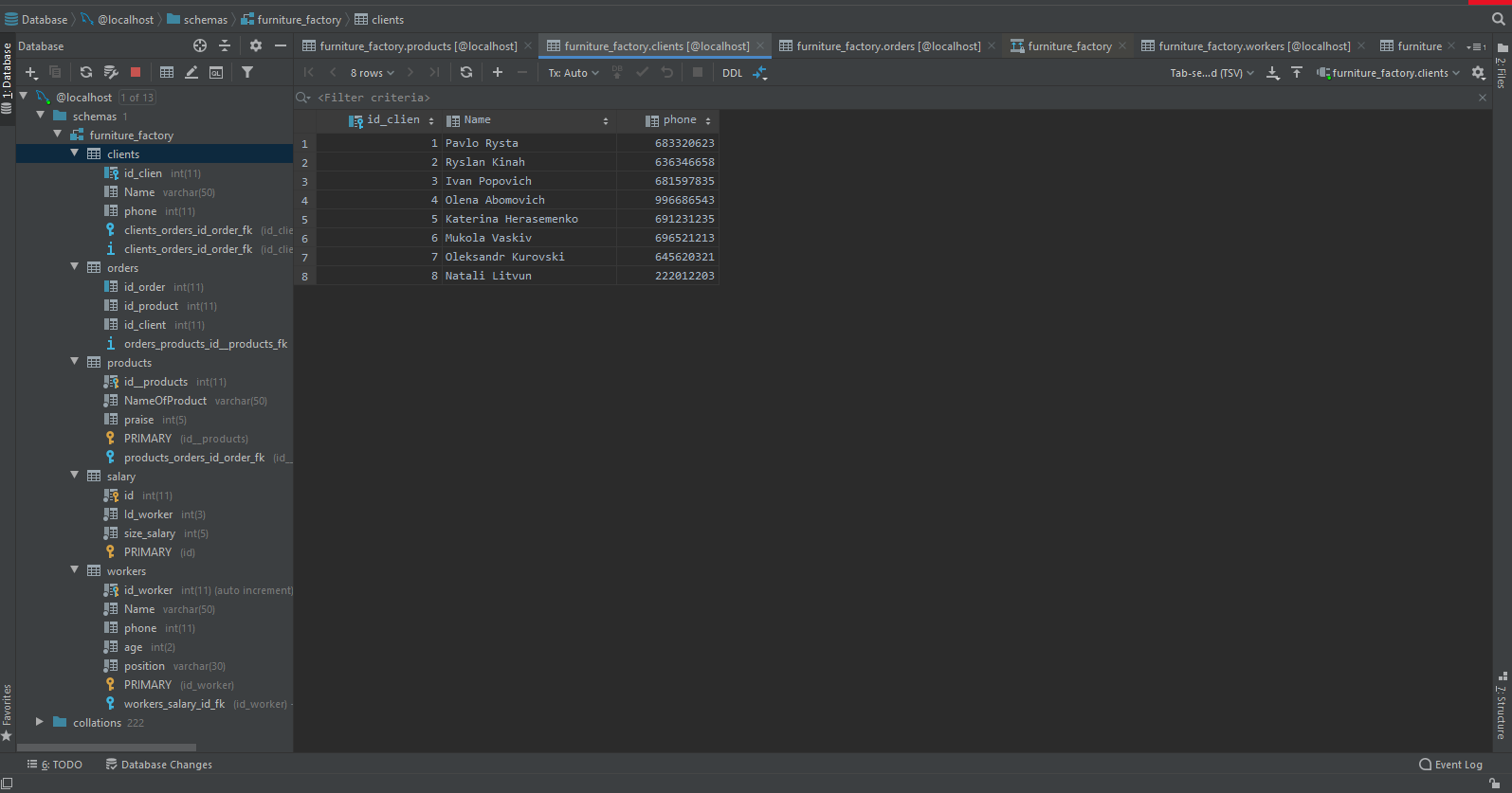
[Електронний ресурс]. – Режим доступу: <https://studfile.net/preview/5391756/>.

1. НОРМАЛІЗАЦІЯ БАЗ ДАНИХ — ВІКІПЕДІЯ [Електронний ресурс].– Режим доступу: <https://cutt.ly/tyTCqT8>.
2. Переваги MySQl [Електронний ресурс]. – Режим доступу: <https://studfile.net/preview/5607354/page:3/> .
3. Типи даних MySQL [Електронний ресурс]. – Режим доступу: <https://uk.wikibooks.org/wiki/SQL/%D0%A2%D0%B8%D0%BF%D0%B8_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85_MySQL>.
4. SQL Tuturial [Електронний ресурс] – Режим доступу до ресурсу: <https://www.w3schools.com/sql/default.asp>.
5. Реляційна база даних [Електронний ресурс]. - <https://uk.wikipedia.org/wiki>.
6. Реляційна модель бази даних. Історія розвитку реляційної моделі. Структура реляційних даних. Відношення в базі та їх властивості. Типи даних. [Електронний ресурс]. – Режим доступу: <https://studfile.net/preview/5454386/page:7/>.
7. Етапи проектування баз данних [Електронний ресурс]. – Режим доступу: <https://studfile.net/preview/5194057/page:2/>.
8. Проблеми проектування бд. Метод нормальних форм [Електронний ресурс]. – Режим доступу: <http://dn.khnu.km.ua/dn/k_default.aspx?M=k0537&T=04&lng=1&st=0>.

**ДОДАТКИ**

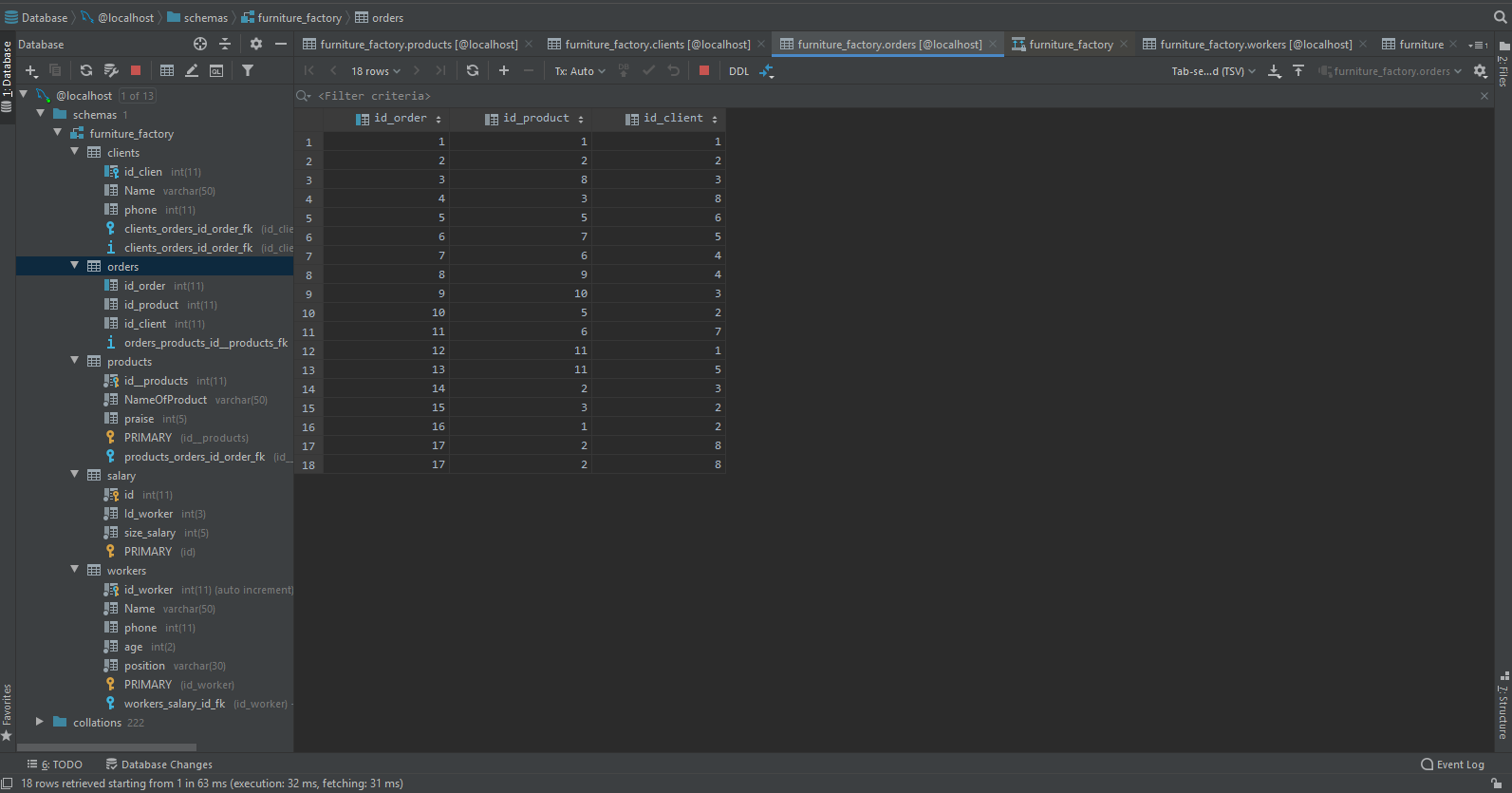
Додаток А

*Таблиця clients*



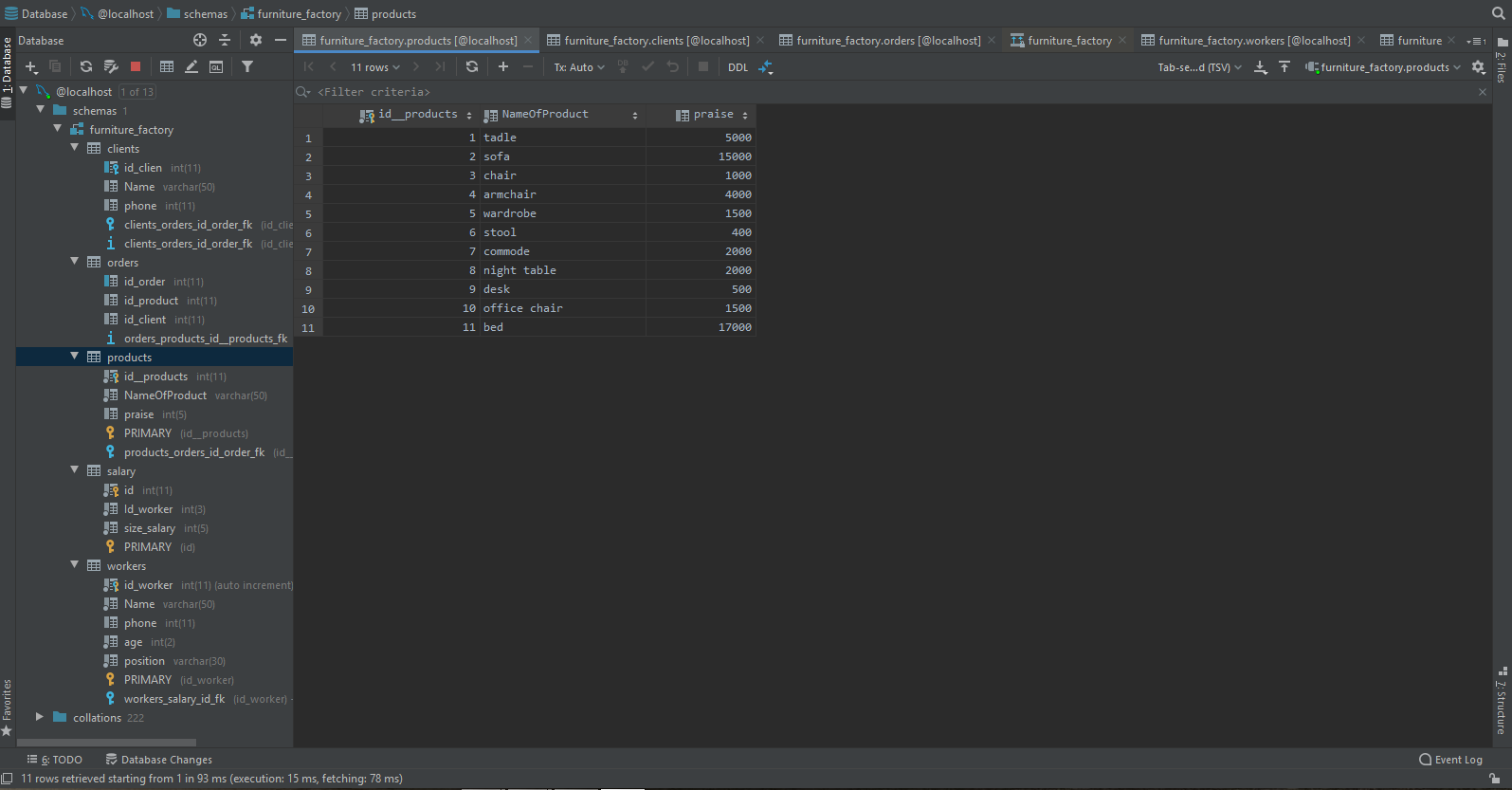
Додаток Б

*Таблиця orders*



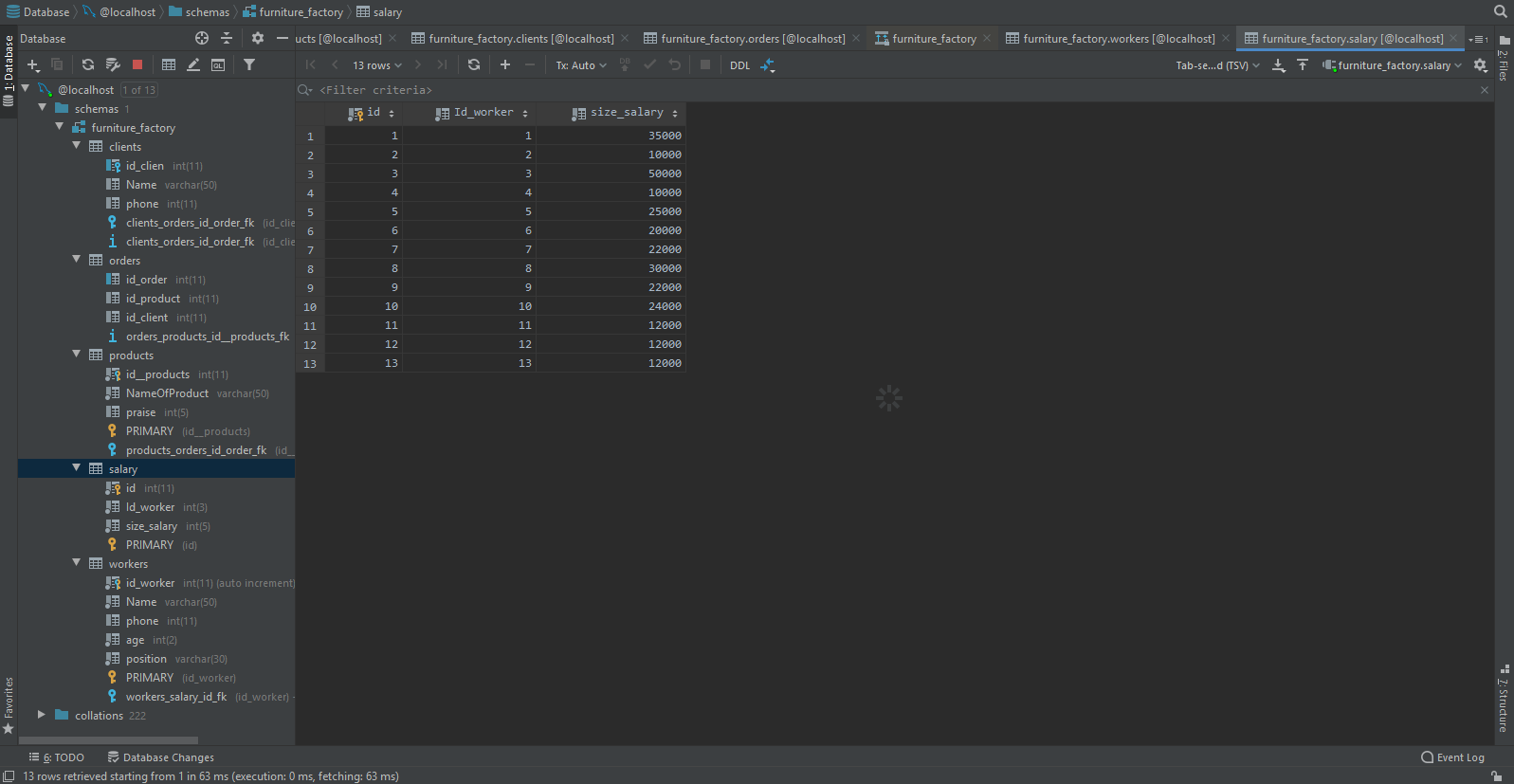
Додаток В

*Таблиця products*



Додаток Г

*Таблиця clients*



Додаток Д

*Таблиця workers*

