

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 бізнес-процесів ресторану»

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

Спеціальність: _____ 051 «Економіка» _____

Спеціалізація: _____ «Інформаційні технології в бізнесі» _____

Освітній ступінь: _____ бакалавр _____

Науковий керівник:

к.ф.-м.н., доц. Депутат Б.Я. _____
(науковий ступінь, посада,
прізвище, ініціали)
_____ “___” травня 2022 р.
(підпис)

Виконавець:

студент(ка) групи УФЕ-31 с
Яринич Я.В.
_____ “___” травня 2022 р.
(підпис)

Загальна кількість балів _____

(підпис, ПІП членів комісії)

Львів 2022

ЗМІСТ

ЗМІСТ	2
ВСТУП.....	3
РОЗДІЛ 1 АНАЛІЗ ВИМОГ	5
1.1 Аналіз ринку ресторанів	5
1.2 Економічний зміст завдання	7
1.3 Економічно-технічний зміст завдання	8
РОЗДІЛ 2. РОЗРОБКА БАЗИ ДАНИХ.....	13
2.1 Опис моделі даних.....	13
2.2. Робота з MySQL Workbench.....	15
2.3. Розроблення структури бази даних.	15
2.4. Перелік таблиць бази даних	16
2.5 Перелік полів таблиць бази даних	17
2.6. Визначення типів даних таблиць бази даних	20
2.7 Реалізація SQL скрипту	21
РОЗДІЛ 3. РОЗРОБКА ВЕБ-ДОДАТКУ	27
3.1 Структура веб-сайту.....	27
3.2 Макет веб-сторінки	28
3.3 Програмування клієнтської частини.	34
ВИСНОВКИ.....	36
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	37
ДОДАТКИ.....	38

ВСТУП

Актуальність теми дослідження. Одним з найпопулярніших напрямків діяльності молодих підприємців є ресторанний бізнес. Більшість з них мріють про власні мережі закладів, а ті, хто вже давно у цій справі – максимізувати свій прибуток. Для цього потрібно розвиватися та діджиталізуватися. Для будь-яких рестораторів це новий етап ресторану – сфера інтернет замовлень, кур'єрських доставок, автоматизованої резервації. У сучасних реаліях бізнесу не варто обходитися без веб-сайту. Для цього я і розробив базу даних та веб-сайт для ресторану. Це новий етап еволюції малого, середнього та великого ресторанного бізнесу. Веб-сайт – онлайн картка закладу, він описує імідж ресторану, якість та комфорт. База даних допомагає працівникам, власникам частково автоматизувати деякі процеси, для зосередження уваги на інших. Наприклад, працівник, який відповідав за дзвінки – резервації зараз відповідає за процеси доставлення товару, аналіз ринку. На попередньому етапі це неможливо, так як час працівника займають постійні дзвінки про резервацію столика. Опитування рестораторів показує, що працівник працює значно ефективніше на цьому етапі бізнесу ніж на попередньому. Отже, можна зробити висновок всі потребують сайт та базу даних: кав'ярні третьої хвили чи великі заклади. У цій курсовій я виконав перший етап – створення початкового макету веб-сайту та бази даних.

Мета і завдання дослідження. Метою курсової роботи є розробка веб-сайту, на якому можна зарезервувати столик та дослідити цю інформаційну систему бізнес-процесів ресторану. Виходячи з цього, для досягнення мети в роботі поставлено й вирішено такі теоретичні та практичні завдання:

- охарактеризувати інформаційну систему для ресторану;
- проаналізувати предметну область;
- розробити архітектуру та спроектувати структуру бази даних;
- розробити інформаційну систему(веб-сайт), що дає змогу переглянути інформацію про ресторан з можливістю бронювання столика.

Об'єкт дослідження є особливість розробки інформаційної системи для ресторану за допомогою системи управління базами даних MySQL.

Предмет досліджень є теоретичні, методичні та практичні аспекти розробки програмного забезпечення за допомогою різних мовою проектування.

Використане програмне забезпечення. Для досягнення результату використовувалось середовище розробки Visual Studio Code, та мови програмування: HTML, CSS, JS, MySQL.

Структура роботи. Курсова робота складається з трьох розділів («Аналіз вимог», «Розробка бази даних» та «Розробка веб-додатку»), висновків, списку використаних джерел та додатків. Загальний обсяг роботи – 44 сторінки.

РОЗДІЛ 1 АНАЛІЗ ВИМОГ

1.1 Аналіз ринку ресторанів

Для кращого розуміння бізнес-процесів ресторанів необхідно їх проаналізувати, застосувавши такі інструменти як SWOT-аналіз та маркетинг-мікс.

SWOT-аналіз – це метод, який дозволяє оцінити переваги та недоліки закладу, можливості та загрози. Маркетинг-мікс 4P`s – це комплекс маркетингу, який містить всі потрібні параметри продукту, які може контролювати та розвивати маркетолог для результативного просування на ринку.

SWOT-аналіз Cloud no7 Panorana

1. Переваги:

Високий рівень просування у соціальних мережах: 16 тисяч підписників у мережі Instagram та 6 тисяч у мережі Facebook. Завдяки розташуванню, заклад має терасу з видом на центр міста, відповідно це спричинило великий попит саме на цей заклад. Також ресторан має таку послугу як організація різноманітних подій: весілля, корпоративи, дівич-вечори, дні народження. Влітку це відбувається на вишуканій терасі з видом на історичний центр міста, у міжсезоння все залежить від погоди, а взимку події відбуваються в основному залі, який гості бронюють повністю. Проаналізувавши відгуки гостей, близько 90% працівників є кваліфікованими та ввічливими, відповідно це покращує репутацію ресторану. У меню є дуже велике різноманіття страв, на будь-який смак та гаманець, пророблена презентація страв.

2. Недоліки:

Хоч заклад має високий рівень відгуків, а це близько 4.7/5 можливих балів, знайшов і конструктивні ревію про неякісне обслуговування закладу, зламані меблі, бруд. Також близько 10% гостей скаржаться на працівників закладу, яких на жаль, не звільняють. Ресторан змінює складові страви без попереджень та без

передруковувань меню закладу, багато страв немає в наявності. І останнє, це необґрунтовано високі ціни на звичайні страви.

3. Можливості:

На мою думку, слід змінити хостинг сайту, так як на теперішньому веб-сайт працює з довгою затримкою. Залучення маркетинг спеціалістів було би дуже хорошим плюсом для ресторану, адже вони розуміють суть таргетної реклами та мають досвід у розкручуванні як сайтів так і соціальні сторінки закладів. Познайти з новими потенційними бізнес партнерами завжди вважається плюсом, у нашому випадку з можливим інвестором, для покращення роботи інтернет-ресурсів та оновлення інтер'єру закладу. Зважаючи, на відгуки про погану якість продуктів слід знайти нових постачальників, щоб закрити ось це питання з якістю.

4. Загрози:

Як я вже писав раніше, заклад має високий рівень популярності через терасу, але це як плюс закладу так і його загроза. Через те, що ресторан знаходиться на 7 поверсі Торгового Центру, його дуже складно знайти, відповідно випадкових гостей закладу не буде. Тут можна переглянути пункт Можливості та найняти маркетологів. Через завищені ціни ніж в інших закладах, потенційні клієнти можуть вибрати інші, більш доступні заклади, також з терасами або навіть акваріумами. Ринок є ринок, і нові заклади з'являються кожен день, ріст конкурентів на ринку може суттєво погіршити потік клієнтів у заклад, а також через 10% працівників закладу, які є не ввічливі, постачальники не хочуть працювати з ними, і це є одною зі загроз – залишитися без якісних продуктів.

Маркетинг-мікс 4P's Ресторанія Бачевських

Product:

Продукт являє собою те, що компанія пропонує ринку і споживачеві. Ім'я бренду - Ресторанія Бачевських. Фірмовий стиль закладу є Галицький. Функціонал продукту займає важливу частину ресторану, а саме галицькі страви, настоянки та наливки власного виробництва. Зовнішній вигляд продукту гармонійно

презентується у вишуканому посуді, який відповідає тематиці ресторану, те саме з напоями. Доставка страв можлива та відбувається за допомогою кур'єра. Рівень сервісу високий, за даними близько 14032 відгуків вказують на дуже високий рівень сервісу, близько 900 – на середній.

Price:

Ціна визначається на основі собівартості продукту, цін конкурентів та бажаної величини прибутку. Роздрібна середня ціна закладу починається від 193 гривень. Сніданок коштує 160 гривень, а коктейль – 120 гривень. Також є наявність сезонних знижок та акцій.

Place:

Місце продажу забезпечує доступність продукту для цільового ринку і означає, що товар компанії повинен бути присутнім на ринку в потрібному місці в потрібний час. Це унікальний ресторан з галицькими мотивами, відповідно стратегії ринкової еспансії не підходять. Також заклад знаходиться у дуже потрібному місці – історичному центрі Львова.

Promotion:

Просування – це всі маркетингові комунікації, які дозволяють привернути увагу споживача до товару. Заклад має пошукову оптимізацію сайту, тобто при вводі основних тегів ресторації одразу вибиває їхній сайт. Реклама поки вимкнена. Часто заклад бере участь у різноманітних подіях.

1.2 Економічний зміст завдання

Інформаційна система допомагає задовільнити певні потреби як замовника так і працівників закладу. Розробка веб-сайту задовільнить потребу у резервуванні столика без телефонування адміністратору ресторану, а база даних задовільнить працівника, тому що він бачитиме кількість замовлень на конкретний день, які страви є в наявності, які працівники обслуговують який стіл, у кого яка зарплатня та загальний

чек кожного замовлення. Також клієнт має змогу переглянути меню закладу, загальну інформацію, розташування та соціальні мережі працівників на сайті, це значно покращує імідж закладу. Це дуже важливо для сучасного ресторану, тому я розробив перший етап бази даних та веб-сайту для ресторану.

1.3 Економічно-технічний зміст завдання

Наявність інформаційної системи ресторану має декілька переваг:

- Детальне меню ресторану. Меню – дуже важлива частина будь-якого ресторану. З якісних інгредієнтів та вміння професіоналів приготувати неймовірну запашну італійську страву – неймовірний скіл.
- Веб-додаток допомагає збільшити кількість потенційних клієнтів і автоматизувати резервацію столика.
- Веб-додаток також показує високий рівень ресторану, так як хороший оптимізований сайт вимагає великих інвестицій.
- Веб-додаток – це один з декількох символів престижного закладу, де готують та обслуговують професіонали.

Аналізуючи ринок ресторанів та кафе, я дійшов висновку, що необхідно розробити user story та use case для інформаційної системи.

Як розробник веб-додатку мені необхідно добре розуміти потенційного користувача, тому на допомогу прийшли user story. User story – це коротке формулювання наміру, що описує щось, що система повинна робити для користувача. Це по суті, вимоги, але прості для сприйняття користувача (Рис.1.3, Рис.1.4).

На першому етапі розроблення інформаційної системи ці вимоги не є реалізовані, але на другому етапі 75% цих користувацьких вимог мають бути реалізовані.

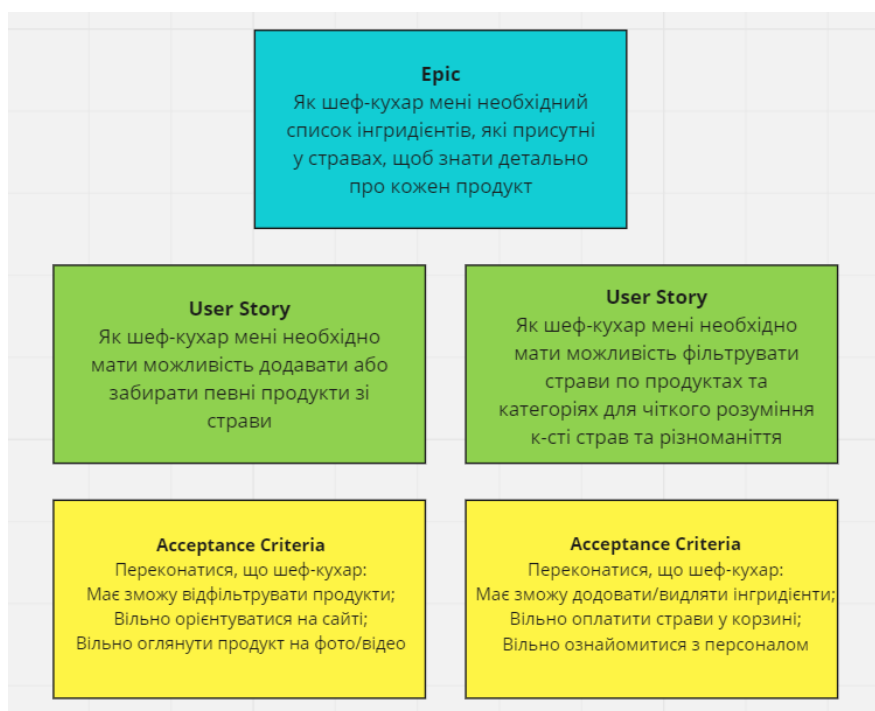


Рис.1.3. Користувацькі вимоги інформаційної системи.

Один з двох макетів User story для розуміння потреб користувача.

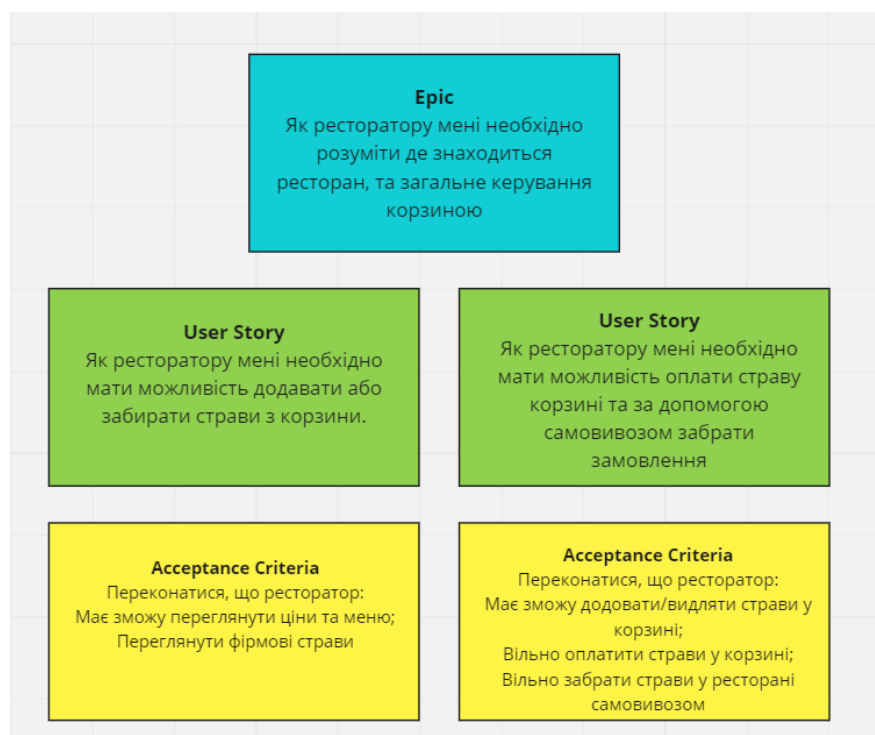


Рис.1.4. Користувацькі вимоги інформаційної системи.

Use case – це перелік дій, сценарій за яким користувач взаємодіє з додатком.

Сайт ресторану перш за все призначений для оприлюднення актуального меню ресторану, його працівників, графіку та соціальних мереж. Завдяки цьому користувач має змогу зайти на веб-сайт та переглянути всю актуальну інформацію про ресторан.

По-друге, сайт призначений для безпосереднього бронювання столика у ресторані, за допомогою розділу Reservation. Будь-який відвідувач сайту може стати клієнтом у ресторані, відповідно необхідно зробити максимально якісний, але в той же час простий сайт для користувача. На першому етапі веб-додаток, повторюсь, немає фільтрації пошуку чи підключення Google Pay, але це все можливо зробити у процесі росту закладу. Як висновок цього опитування зробив use case & user story для майбутніх етапів додатку.

Також потрібно пам'ятати за розробку бази даних для працівника закладу. Розроблена база даних допоможе адміністратору контролювати та тримати завжди біля себе певні пункти бізнес-процесів закладу, такі як: Інформація про зарплатню працівників, їхній досвід роботи, також відслідковування рентабельності та прохідності закладу: можливість переглядати кількість резервацій за день, середній чек за день, які працівники обслуговували який столик.

Цю use case діаграму я розробив для першого етапу інформаційної системи Рис.1.5. Мій веб-додаток знаходиться на першому етапі розробки, тому він поки не відповідає use case на всі 100%.



Рис.1.5. Use case діаграма інформаційної системи.

На сьогоднішній день, зважаючи на події у світі, замовник повинен мати можливість забронювати столик на сайті, безконтактно. Велика частина західних ресторанів активно реалізують цю потребу, але Центральна, Східна Європа поки ні. Так як у бізнесі завжди потрібно порівнювати себе з кращими, моя use case діаграма має пункт – «Резервація столика».

Також не менш важливою складовою є меню та загальна інформація ресторану. Потенційний клієнт має бажання та цікавість дізнатися, що смачного подають у цьому закладі, де він знаходиться і чи вартує це його уваги. Отож моя діаграма має другий пункт «Перегляд меню» та третій пункт «Ознайомлення з рестораном» З іншого боку розробка цієї система допоможе працівникам закладу, тому що адміністратор має

змогу переглянути прибуток та кількість клієнтів за день, загальну наповненість залу, зарплатню працівникам.

Зважаючи на те, що це лише перший етап розробки веб-сайту, додатково ще немає складових страв, звідки інгредієнти походять та фільтрація страв по продуктах. Це все буде розроблено на другому етапі.

Проробивши цю всю інформацію, я акцентуватиму увагу на якості верстки веб-сайту та технічної структури бази даних. Також варто якісно наповнити базу та зробити запити, по яких працівнику буде легше працювати з бізнес-процесами закладу.

РОЗДІЛ 2. РОЗРОБКА БАЗИ ДАНИХ.

2.1 Опис моделі даних

База даних (БД) – це організована структура, яка призначена для зберігання, зміни та обробки взаємозалежної інформації, переважно великих обсягів.

Реляційна база даних — це тип бази даних, що зберігає інформацію в електронних таблицях і здійснює пошук даних в одній таблиці на підставі визначених ключових полів іншої таблиці.

Принципи нормалізації:

- в кожній таблиці БД не повинно бути повторюваних полів;
- в кожній таблиці повинен бути унікальний ідентифікатор (первинний ключ);
- кожному значенню первинного ключа повинна відповідати достатня інформація про тип суті або про об'єкт таблиці (наприклад, інформація про успішність, про групу або студентах);
- зміна значень в полях таблиці не повинна впливати на інформацію в інших полях (крім змін у полях ключа).

Основна відмінність між СУБД і РСУБД є організація зберігання даних. Якщо у РСУБД дані зберігаються у таблицях, у нереляційних СУБД немає зв'язків між таблицями (дані можуть зберігатися навіть у файлах або у ієрархічних структурах). Моделювання таблиць та проектування бази даних здійснюється за допомогою безкоштовного інструменту MySQL Workbench.

Перед тим як зберігати якісь дані у СУБД потрібно описати їх модель. Тобто визначити самі дані та описати їх взаємозв'язки з іншими даними у таблиці. Тобто перефразувавши, — “вказати на клітинку”, куди дані мають потрапити, за допомогою запитів SQL, визначених індексів, різного роду обмежень, а по правильному то набір атрибутів.

Набір атрибутів, кожен з яких описує певне поле таблиці, описують структуру однієї з таблиць в базі даних і БД в цілому. Забезпечують управління даними всередині баз даних.

У моєму прикладі, модель замовлення таблиці order може виглядати так:

- Дані замовника,
- Дані резервації,
- Час бронювання,
- Дата бронювання.

Таблиця – це об'єкт бази даних, який використовується для зберігання відповідних записів у вигляді рядків та стовпців.

Поле таблиці – це характеристика, що використовується для зберігання певного типу даних у таблиці, як цифри типу, символи, дати тощо. Поле таблиці формує модель таблиці.

Нормалізація бази даних – це техніка проектування, яка використовується для проектування бази даних.

Нормалізація SQL допомагає згрупувати пов'язані дані в одну єдину таблицю. Будь-які атрибутивні дані або опосередковано пов'язані дані розміщується у різних таблицях, і ці таблиці пов'язані логічним взаємозв'язком між таблицями.

Є декілька основних типів звичайних форм:

Перша нормальна форма (1NF) – це сутність, яка немає жодних повторювальних стовпців або груп даних. У 1NF кожен стовпець унікальний. Друга нормальна форма (2NF) – це сутність, яка дорівнює 1NF та один з її атрибутів визначається як первинний ключ, а решта атрибутів залежать від первинного ключа. Третя нормальна форма (3NF) – це сутність, яка вже знаходиться у другій нормальній формі, а стовпці таблиці/сутності залежать від первинного ключа.

Існують два типи ключів: Первинний та зовнішній або як ще його називають - вторинний. Первинний ключ — це одне або кілька полів (стовпців), комбінація значень яких однозначно визначає кожний запис у таблиці. Первинний ключ не

допускає значень Null і завжди повинен мати унікальний індекс. Первинний ключ використовується для зв'язування таблиці з зовнішніми ключами в інших таблицях. Зовнішній (вторинний) ключ — це одне або кілька полів (стовпців) у таблиці, що містять посилання на поле або поля первинного ключа в іншій таблиці. Зовнішній ключ визначає спосіб об'єднання таблиць.

З двох логічно пов'язаних таблиць одну називають таблицею первинного ключа або головною таблицею, а іншу таблицею вторинного ключа або підпорядкованою таблицею. СУБД дозволяють зіставити споріднені записи з обох таблиць і спільно вивести їх у формі, звіті або запиті.

2.2. Робота з MySQL Workbench

MySQL Workbench – це уніфікований візуальний інструмент для архітектури бази даних. Забезпечує моделювання та розроблення бази даних, та комплексні інструменти адміністрування для налаштування сервера, адміністрування користувача, резервного копіювання та багато іншого.

На ринку існує низка систем управління реляційними базами даних. На мою думку, MySQL Workbench є найкращим СКБД через те, що MySQL підтримує декілька способів зберігання даних, кожен з яких має свої специфікації, тоді як інші системи, такі як SQL-сервер, підтримують лише один механізм зберігання даних. Також з переваг включають простоту та високу продуктивність.

2.3. Розроблення структури бази даних.

Для розроблення своєї бази даних я використав саме реляційну модель та через ряд певних переваг:

- Спрощені схеми даних для користувачів,

- Оптимізація доступу до бази даних,
- Покращення цілісності та захисту даних,
- Забезпечення методологічного підходу.

Діаграма концептуальної реляційної моделі даних представлена на рисунку 2.3.

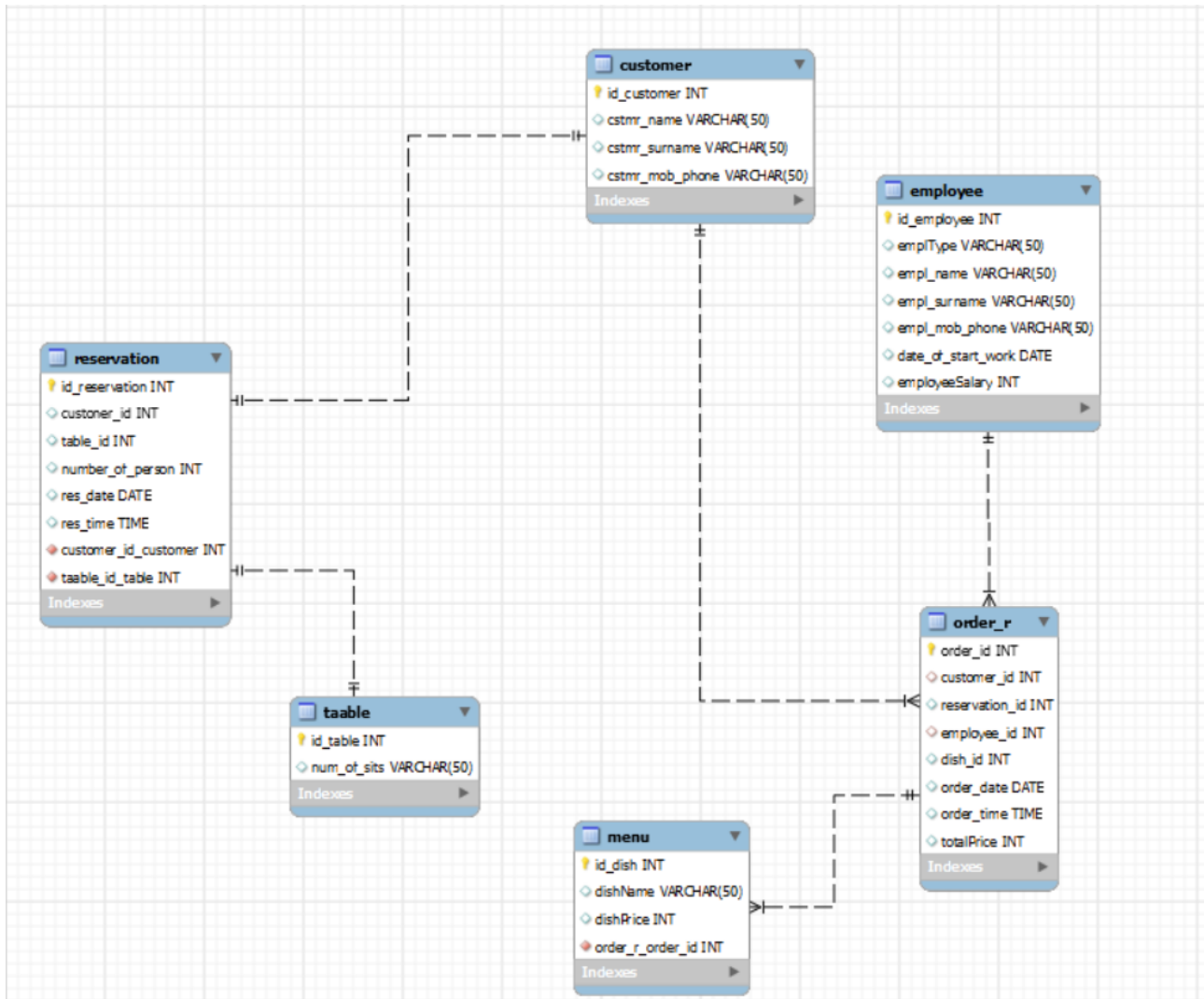


Рис.2.3. Концептуальна модель бази даних.

2.4. Перелік таблиць бази даних

Таблиці даних для інформаційної системи ресторану:

- reservation – інформація про резервація клієнтом столика
- taable – інформація про кількість місць за столиком

- customer – інформація про замовника
- employee – інформація про працівника, який безпосередньо обслуговує замовника
- order_r – інформація про замовлення разом
- menu – інформація про те, що є в наявності

2.5 Перелік полів таблиць бази даних

Поля, які характеризують властивості таблиці «customer» Рис.2.4:

id_customer – ідентифікатор замовника

cstmr_name – ім'я користувача

cstmr_surname – прізвище користувача

cstmr_mob_phone – номер телефону користувача

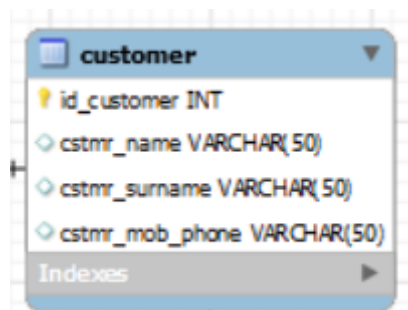


Рис.2.4. Таблиця «customer»

Поля, які характеризують властивості таблиці «table» Рис.2.5:

id_table – ідентифікатор столу

num_of_sits – кількість можливих місць у кожного столу

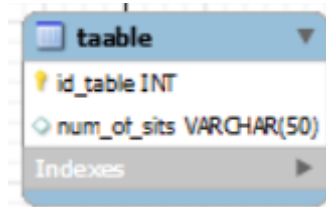


Рис.2.5. Таблиця «taable»

Поля, які характеризують властивості таблиці «menu» Рис.2.6:

id_dish – ідентифікатор страви, яка є в наявності

dishName – назва страви

dishPrice – ціна страви

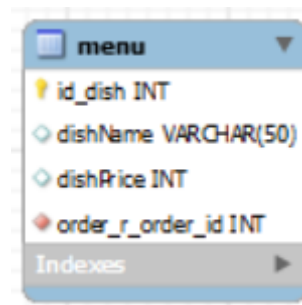


Рис.2.6. Таблиця «menu»

Поля, які характеризують властивості таблиці «employee» Рис 2.7:

id_employee – ідентифікатор працівника, який безпосередньо обслуговує клієнта

empl_name – ім`я працівника

empl_surname – прізвище працівника

empl_mob_phone – номер телефону працівника

date_of_start_work – досвід роботи офіціантом/барменом

employeeSalary – зарплатня працівника.

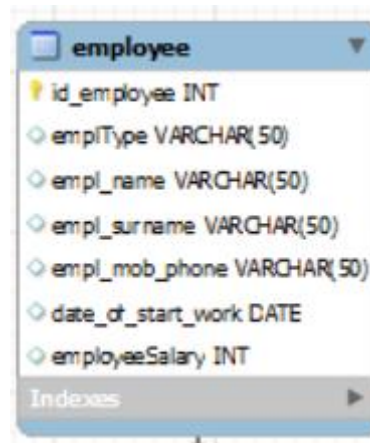


Рис.2.7. Таблица «employee»

Поля, які характеризують властивості таблиці «reservation» Рис.2.8:

id_reservation – ідентифікатор резервації столика

customer_id – ідентифікатор користувача

table_id – ідентифікатор столу, який замовив користувач

number_of_person – кількість людей за столиком

res_date – дата резервації

res_time – час резервації

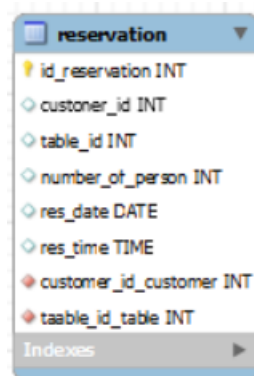


Рис.2.8. Таблица «reservation»

Поля, які характеризують властивості таблиці «order_r» Рис.2.9:

order_id – ідентифікатор замовлення

customer_id – ідентифікатор замовника

reservation_id – ідентифікатор резервації замовника

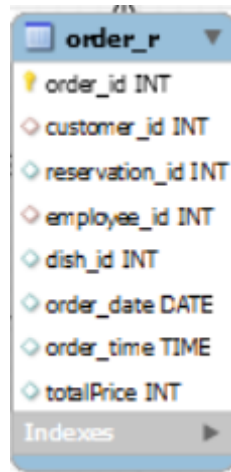
employee_id – ідентифікатор працівника

dish_id – ідентифікатор страви, яку замовив клієнт

order_date – дата фактичного замовлення

order_time – час фактичного замовлення

totalPrice – ціна за страву, яку замовив клієнт.



order_r	
order_id	INT
customer_id	INT
reservation_id	INT
employee_id	INT
dish_id	INT
order_date	DATE
order_time	TIME
totalPrice	INT
Indexes	

Рис.2.9. Таблиця «order_r»

2.6. Визначення типів даних таблиць бази даних

Перед створенням таблиць у базі даних, потрібно визначити, що зберігатиметься у комірках, тобто визначити тип даних.

У цій інформаційній системі я використав основні типи даних SQL:

- INT

Використання: використовується для зберігання позитивних і від’ємних цілих даних.

Досягти: від -2,147,483,648 до 2,147,483,647

- VARCHAR

Використання: використовується для зберігання рядкових даних динамічного розміру.

Охоплення: від 0 до 255 символів (версія 4.1), від 0 до 65 535 (версія 5.0.3)

- **TIME**

Використання: використовується для економії часу.
Досягніть: -838: 59: 59 до +838: 59: 59 (HH: MM: SS) **Розмір:** 3 байти.

- **DATE**

Використання: використовується для зберігання даних про дату.
Діапазон: від 1000-01-01 до 9999-12-31 (YYYY-MM-DD) **Розмір:** 3 байти.

У створенні бази даних для інформаційної системи ресторану я використав усі 4 типи даних. INT використав для ідентифікатора таблиць, кількості людей за столиком та зарплатні працівників. DATE застосував для дати початку роботи працівника та на який день замовник бронює стіл. TIME використав для резервації столика на конкретну годину. VARCHAR для решти полів.

2.7 Реалізація SQL скрипту

Лістинг 2.1.1

Перший запит показує всі замовлення у діапазоні цін від 20 до 30 євро. Вибравши поля таблиця order_r і застосувавши умову, виводить усі замовлення рис. 2.7.1

```
1.SELECT order_id, customer_id, employee_id, reservation_id
   FROM order_r WHERE totalPrice BETWEEN 20 AND 30 ORDER BY
   totalPrice;
```

	order_id	customer_id	employee_id	reservation_id
►	201	1	100	10001
	203	3	200	10003
	205	5	300	10005
	209	9	500	10009
	202	2	100	10002

Рис.2.7.1. Результати першого запиту.

Другий запит показує нам максимальну ціну на страву зі всього меню, Ціна якого становить 50 євро рис. 2.7.2.

```
2.SELECT MAX(dishPrice) AS MAX_SUM FROM menu JOIN order_r
ON id_dish=dish_id;
```

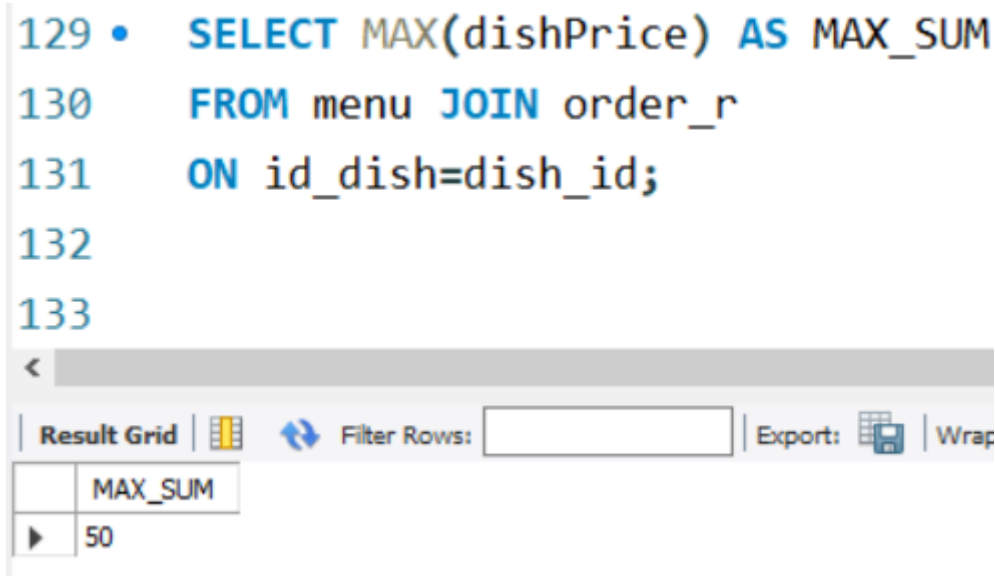


Рис.2.7.2 Результат другого запиту.

Третій запит показує максимальну зарплатню зі всіх працівників закладу, які безпосередньо працюють з клієнтами, тобто офіціантами або барменом рис. 2.7.3.

```
3.SELECT empl_name, empl_surname, employeeSalary FROM
employee WHERE employeeSalary = (SELECT
MAX(employeeSalary) FROM employee);
```

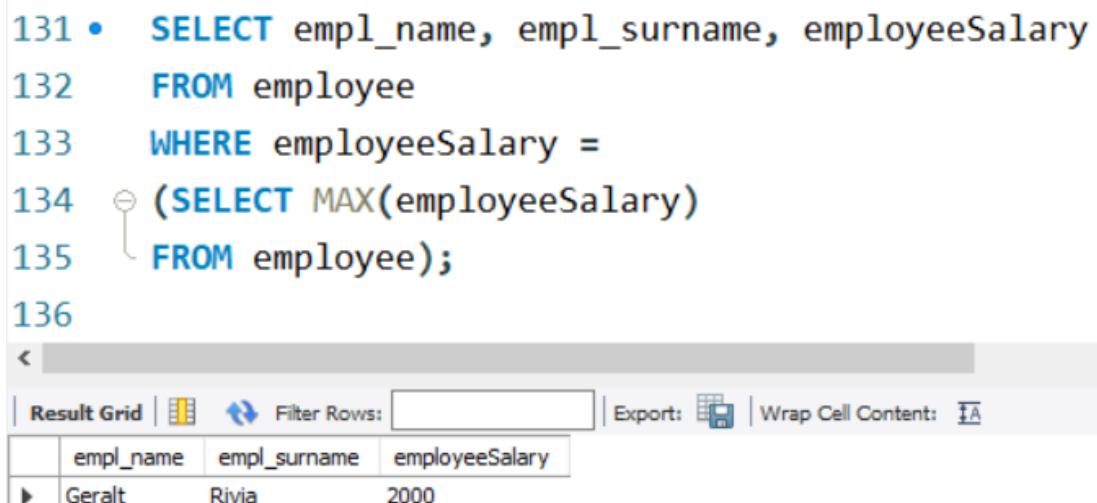
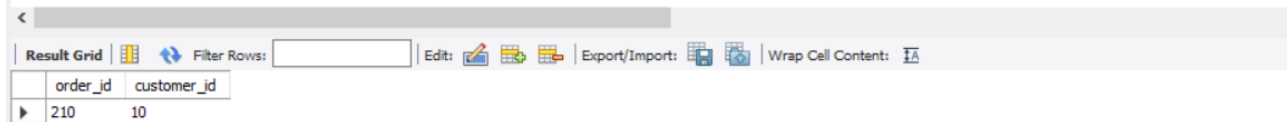


Рис.2.7.3. Результат третього запиту.

Четвертий запит виводить ідентифікатор замовлення, замовник якого повинен оплатити більше ніж 40 євро рис. 2.7.4.

```
4.SELECT order_id, customer_id FROM order_r WHERE totalPrice
  > 40 ORDER BY totalPrice;
```

141 • **SELECT** order_id, customer_id **FROM** order_r **WHERE** totalPrice > 40 **ORDER BY** totalPrice;
 142
 143



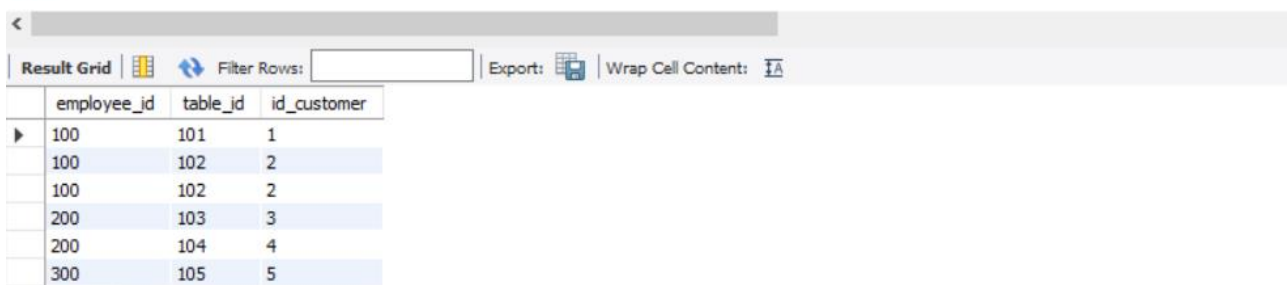
order_id	customer_id
210	10

Рис.2.7.4. Результат четвертого запиту

П'ятий запит показує який офіціант або бармен відповідає за який стіл. Тобто офіціант з ідентифікатором 100 відповідає за стіл з ідентифікатором 101 та 102, у результаті вибило 101 стіл двічі, через те, що клієнт з ідентифікатором 2 приходив на одне і теж місце двічі.

```
5.SELECT      order_r.employee_id,      reservation.table_id,
  customer.id_customer FROM order_r, reservation, customer
WHERE customer.id_customer = reservation.customer_id AND
customer.id_customer = order_r.customer_id GROUP BY
order_id;
```

143 • **SELECT** order_r.employee_id, reservation.table_id, customer.id_customer
 144 **FROM** order_r, reservation, customer
 145 **WHERE** customer.id_customer = reservation.customer_id **AND** customer.id_cus
 146



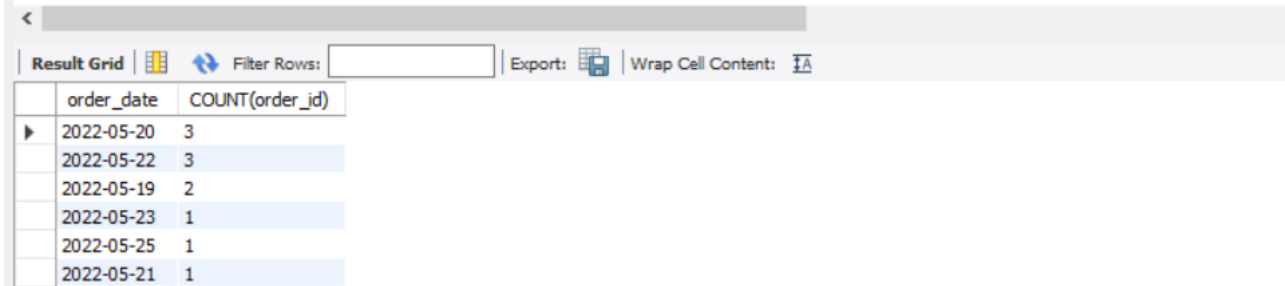
employee_id	table_id	id_customer
100	101	1
100	102	2
100	102	2
200	103	3
200	104	4
300	105	5

Рис.2.7.5. Результат п'ятого запиту.

Шостий запит створений саме для працівників закладу, а саме для адміністратора, щоб він міг аналізувати прибуткові дні рис.2.7.6.

```
6.SELECT order_date, COUNT(order_id) FROM order_r GROUP BY
order_date;
```

145 • **SELECT** order_date, COUNT(order_id) FROM order_r GROUP BY order_date;
146



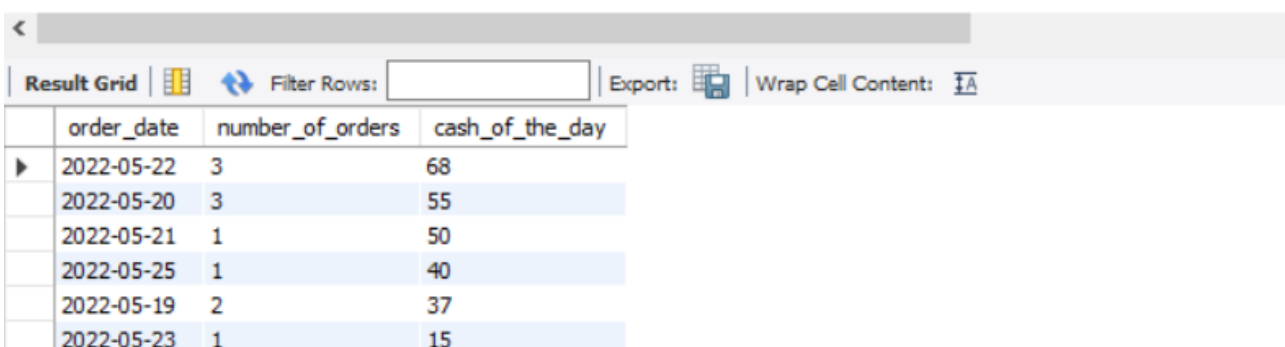
order_date	COUNT(order_id)
2022-05-20	3
2022-05-22	3
2022-05-19	2
2022-05-23	1
2022-05-25	1
2022-05-21	1

Рис.2.7.6. Результат шостого запиту.

Сьомий запит розроблений спеціально для працівників закладу, які відповідають за аналіз та розвиток ресторану. Цей селект закриває проблему необраховування прибутку по кількості замовлень та загальної суми. Дуже зручно та просто використовується рис.2.7.7.

```
7.SELECT order_date, COUNT(order_id ) AS number_of_orders,
SUM(totalPrice) AS cash_of_the_day FROM order_r GROUP BY
order_date ORDER BY cash_of_The_day DESC;
```

147 • **SELECT** order_date, COUNT(order_id) AS number_of_orders,
148
149
150



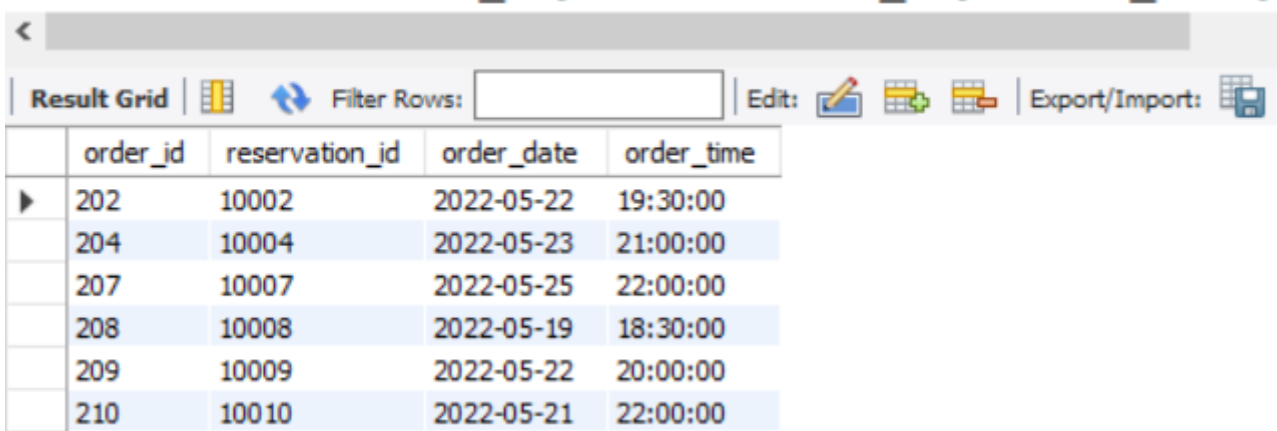
order_date	number_of_orders	cash_of_the_day
2022-05-22	3	68
2022-05-20	3	55
2022-05-21	1	50
2022-05-25	1	40
2022-05-19	2	37
2022-05-23	1	15

Рис.2.7.7. Результат сьомого запиту.

Восьмий запит показує заповненість залу шляхом оброблена кількості замовлень по часу. З 10 замовлень 6 відбуваються у вечірній час. Можна зробити висновок, що на перспективу можна додати у ресторан живу італійську музику для збільшення кількості гостей до 10 столиків рис.2.7.8.

```
8.SELECT order_id, reservation_id, order_date, order_time
FROM order_r WHERE order_time > '18:00';
```

149 • **SELECT** order_id, reservation_id, order_date,



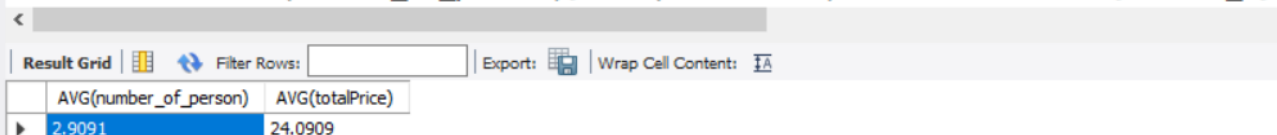
	order_id	reservation_id	order_date	order_time
▶	202	10002	2022-05-22	19:30:00
	204	10004	2022-05-23	21:00:00
	207	10007	2022-05-25	22:00:00
	208	10008	2022-05-19	18:30:00
	209	10009	2022-05-22	20:00:00
	210	10010	2022-05-21	22:00:00

Рис.2.7.8. Результат восьмого запиту.

Дев`ятий селект реалізовує середню кількість людей за одним столиком та середню суму замовлення. Це дуже допомагає для оформлення ресторану, так як вигідніше поставити 10 столів по 2 – 3 місця ніж навпаки рис.2.7.9.

```
9.SELECT AVG(number_of_person), AVG(totalPrice) FROM
reservation, order_r;
```

149 • **SELECT** AVG(number_of_person), AVG(totalPrice) FROM reservation, order_r;



	AVG(number_of_person)	AVG(totalPrice)
▶	2.9091	24.0909

Рис.2.7.9. Результат дев`ятого запиту.

Десятий запит показує скільки разів клієнт відвідав заклад, зарезервувавши його на сайті. У перспективі можна розробити систему лояльності клієнтів, де якщо клієнт відвідав більше 5 разів – знижка 20% на наступне замовлення рис.2.7.10.

```
10.      SELECT  customer_id,  cstmr_name,  cstmr_surname,
COUNT(order_id) AS order_count FROM order_r CROSS JOIN
customer ON order_r.customer_id = customer.id_customer
GROUP BY customer_id ORDER BY order_count DESC;
```

```
151 • SELECT customer_id, cstmr_name, cstmr_surname,
152
153
```



The screenshot shows a database interface with a 'Result Grid' tab. Above the grid is a 'Filter Rows' input field and an 'Export' button. The grid displays the results of the SQL query, ordered by 'order_count' in descending order. The columns are 'customer_id', 'cstmr_name', 'cstmr_surname', and 'order_count'.

	customer_id	cstmr_name	cstmr_surname	order_count
▶	2	Sadie	Ward	2
	1	Anna	Lee	1
	3	Riley	Hall	1
	4	Charlotte	Robinson	1
	5	Robert	King	1
	6	Saitama	Punch	1

Рис.2.7.10. Результат десятого запиту.

РОЗДІЛ 3. РОЗРОБКА ВЕБ-ДОДАТКУ

3.1 Структура веб-сайту

В якості середовища для написання коду було обрано програмний продукт Visual Studio Code, який використовується створення, редагування сучасних вебзастосунків і програм для хмарних систем. Середовище підтримує необхідні мови та розмітки програмування – HTML, CSS, JavaScript.

Програмна реалізація веб-сайту почалася з верстки сайту.

Верстка виконувалася стандартною мовою розмітки документів для перегляду веб-сторінок у браузері (HTML). Для стилізації розмітки сайту використовували каскадні таблиці стилів - CSS. Для створення сценаріїв веб-сторінок, що надає можливість на боці клієнта взаємодіяти з користувачем використали JavaScript.

Сайт створений у вигляді односторінкового сайту так званий landing page. Landing page (з англ. «посадкова сторінка») – це цільова сторінка, яка спонукає відвідувача виконати певну дію. У нашому випадку ця дія – резервація столика.

Структура шапки веб-сторінки ресторану(рис.3.1.1)



Рис.3.1.1. Шапка сайту ресторану.

3.2 Макет веб-сторінки

Потенційний відвідувач сайту (клієнт) може взаємодіяти з:

- Шапкою сайту;
- Розділом “Menu”;
- Розділом “Reservation”;
- Розділом “Our Team”;
- Розділом “Contact”;
- Підвалом сайту.

Взаємодія користувача зі шапкою відбувається за допомогою натискання на кнопки розділів сайту, після цього якір прогорне клієнта на потрібний розділ. Якір використовується в HTML для створення посилань всередині одного документу (рис.3.1.2).

Для цього нам потрібно в тег на початку кожного розділу вписати id. Після цього прописати в тег `<a>` id розділу, на який потрібно переходити.



Рис.3.1.2. Кнопки, які переходять на розділи.



Рис.3.1.3. Перший розділ сайту.

Блочний елемент `<p></p>` містить у собі текст About us та `id="aboutUs"`. Це `id` вписуємо у `href` тегу `<a>` (рис.3.1.4, рис. 3.1.5).

```
<ul id="menuUl">
  <li><a href="#" class="active">HOME</a></li>
  <li><a href="#aboutUs">ABOUT US</a></li>
  <li><a href="#restaurantMenu">MENU</a></li>
  <li><a href="#reservation">RESERVATION</a></li>
  <li><a href="#sliderGallery">GALLERY</a></li>
  <li><a href="#chefs">OUR TEAM</a></li>
  <li><a href="#contact">CONTACT</a></li>
</ul>
```

Рис.3.1.4. Ненумерований список шапки сайту.

```
<section class="aboutUs">
  <p class="discoverText">Discover</p>
  <p class="aboutUsTitle" id="aboutUs">ABOUT US</p>
```

Рис.3.1.5 `id="aboutUs"` у тезі `<p></p>`.

У розділі Menu користувач має змогу натиснути на картинки, перейти на іншу сторінку та побачити меню ресторану (рис.3.1.6, рис.3.1.7).

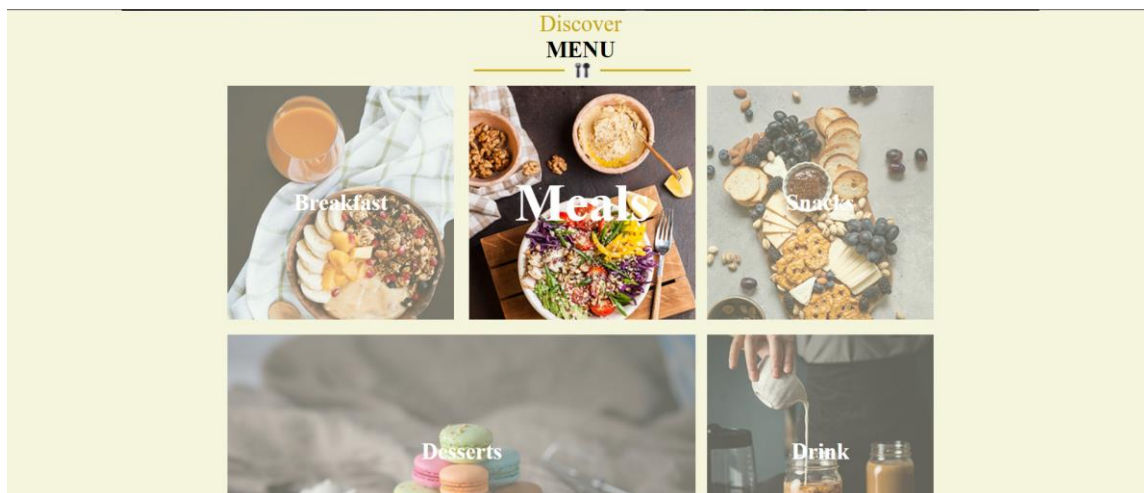


Рис.3.1.6. Кнопки для відкриття меню.

HOME	ABOUT US	MENU	RESERVATION	GALLERY	OUR TEAM	CONTACT
BREAKFAST						
Meetball Sandwich.....10€						
Pizza Sandwich.....12€						
Italian Sausage Sandwich.....15€						
Gnoci.....20€						
Submarine Sandwich.....22€						
Chicken Sandwich.....25€						
MEALS						
Spaghetti with meat sause.....20€						
Ravioli with meat balls.....22€						
Lasagne.....15€						
Ribollita.....25€						
Polenta.....22€						
Risotto.....40€						
SNACKS						
Pizza Margherita.....15€						
Vegeterian Combo.....20€						
Garlic lovers.....22€						

Рис.3.1.7. Сторінка menu.html.

У розділі Reservation відвідувач сайту має змогу забронювати стіл у ресторані. (рис.3.1.8).

У ній користувач вводить дані:

- ПІБ;
- Електронна пошта;
- Кількість людей за столиком;
- Час та дата резервації;
- Номер телефону.

Make a Reservation
Book Now

Lorem ipsum dolor sit amet consectetur adipiscing elit. Quae, fugit.

Your full name ?

Your email address ?

How many people ?

What time ?

What is the date ?

Your phone number ?

SUBMIT

Рис.3.1.8. Резерваційна форма бронювання столу.

Після натискання кнопки Submit дані відправляються у консоль розробника, де зберігатимуться поки адміністратор не перегляне їх (рис.3.1.9)

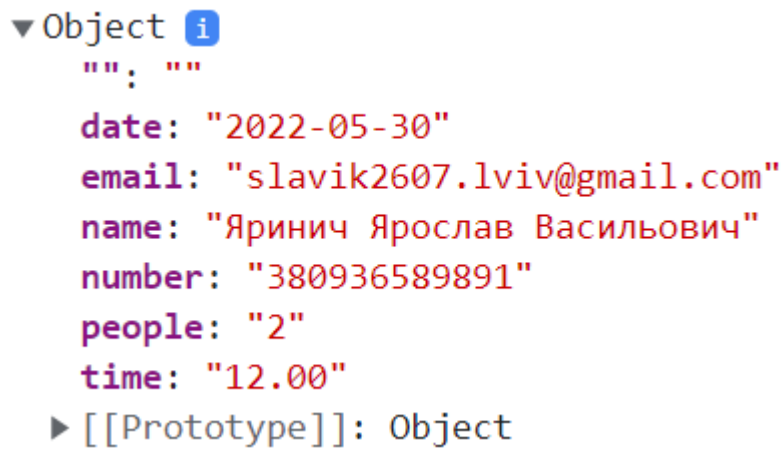


Рис. 3.1.9. Дані про клієнта.

Взаємодія з користувачем реалізована на мові JavaScript.

Лістинг 3.2.1

```

const form = document.getElementById('form');

function retrieveFormValue(event) {

    event.preventDefault();

    const fields = document.querySelectorAll('input, select,
textarea')

    const values = {}

    fields.forEach(field => {

        const {name, value} = field;

        values[name] = value;

    })

    console.log(values);

};

form.addEventListener('submit', retrieveFormValue);
  
```


У розділі Our Team клієнт має змогу переглянути всіх кухарів, які працюють у закладі. Також можуть перейти на їхні соціальні мережі за допомогою тегу `<a>` (рис.3.2.0).



Рис.3.2.0. Перехід на соц.мережі працівників.

Для покращення якості сайту, розробник використав плавне збільшення картинки та hover ефект з градієнтом. Hover – це псевдоклас, який працює, коли користувач наводить мишкою по елементу, але не обов'язково активує його. (рис.3.2.1)

```
.twitter:hover{
  background: #009FFF;
  background: -webkit-gradient(linear, left top, right top, from(#ec2F4B), to(#009FFF));
  background: linear-gradient(to right, #ec2F4B, #009FFF);
}
```

Рис.3.2.1. Hover ефект для іконки twitter.

У розділі Contact користувач має змогу ввести свої дані та написати запитання у полі Message. Також дізнатися інформацію про роботу закладу, номер телефону, електронну пошту, адресу (рис.3.2.2).



Рис.3.2.2. Розділ Contact з формою та даними про ресторан.

Користувач має змогу користуватися мапою на сайті, для легшого знаходження закладу (рис.3.2.3).

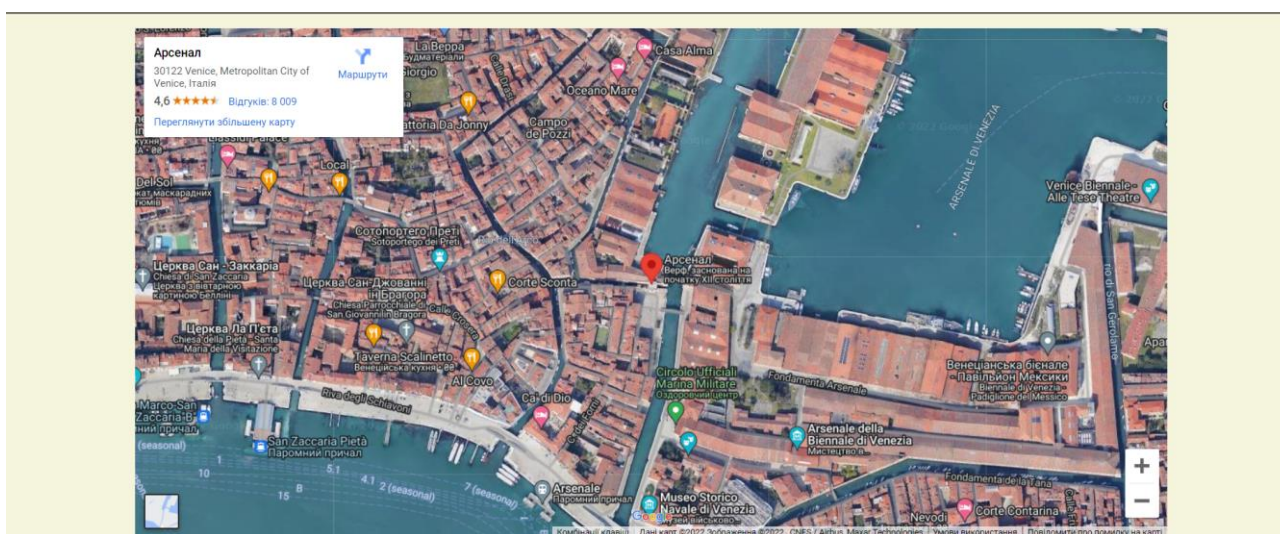


Рис.3.2.3. Мапа, на якій позначено заклад.

У підвалі сайту розміщується посилання на всі соціальні мережі закладу (рис.3.2.4).



Рис.3.2.4. Підвал сайту.

Також використано hover ефект (рис.3.2.5).

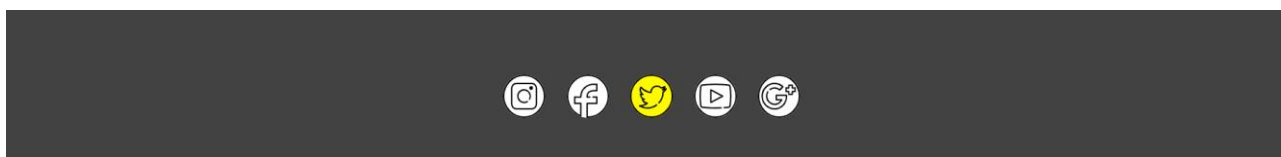


Рис. 3.2.5. Підвал сайту з hover ефектом.

3.3 Програмування клієнтської частини.

Програмування клієнтської частини є дуже важливою складовою будь-якої інформаційної системи. Front-end – це частина веб-сайту, яку користувач бачить та може взаємодіяти і контактувати напрямку. У моїй роботі більш ніж 800 рядків коду, розглянемо лише шапку сайту.

Лістинг 3.3.1

```
<header>

    <div class="header-container">

        <nav id="navig">

            <ul id="menuUl">

                <li><a href="#"
class="active">HOME</a></li>

                <li><a href="#aboutUs">ABOUT US</a></li>

                <li><a href="#restaurantMenu">MENU</a></li>

                <li><a
href="#reservation">RESERVATION</a></li>

                <li><a href="#sliderGallery">GALLERY</a></li>

                <li><a href="#chefs">OUR TEAM</a></li>
```

```
        <li><a href="#contact">CONTACT</a></li>

    </ul>

</nav>

<div class="mainTitle">

    <h1 class="welcome">Welcome</h1>

    <h2 class="welcomeG">To the Gambero</h2>

</div>

</div>

</header>
```

Потрібно також звернути увагу на те, що програмування Front-end частини настільки ж важливо як і програмування Back-end частини. Найкраща система працює, при комбінації цих двох частин.

ВИСНОВКИ

Під час курсової роботи проаналізовано можливості використання HTML, CSS, JS та роботу з базами даних. Для закріплення набутих знань розроблено базу даних ресторану та веб-сайт, який дає змогу резервувати стіл, переглянути інформацію про працівників, детальну інформацію про графік роботи ресторану, місцезнаходження та меню.

На першому етапі виконання було проаналізовано область ресторанного бізнесу, його бізнес-процеси, поставлена мета і завдання відповідно до змісту курсової роботи, а саме інформаційної системи ресторану.

На другому етапі було спроектовано і реалізовано базу даних, визначили тип даних, створили таблиці, визначили первинні та вторинні ключі, розробили запити. Для чіткішого та кращого розуміння бази даних побудовано ER-діаграму.

На третьому етапі було описано функціонал веб-додатку, а також показано зовнішній вигляд сайту. Наведені частини коду програмування для кращого розуміння веб-додатку. Верстка виконувалася за допомогою HTML, для стилів використали каскадні таблиці стилів – CSS та для реалізації лічильника страв, слайдера, резерваційного поля – JavaScript.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Що таке MySQL [Електронний ресурс]. – Режим доступу: <https://www.ukraine.com.ua/uk/blog/programming/sql-baza-dannih-dlya-chego-prednaznachena-baza-dannih.html>
2. JavaScript tutorial [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/en-US/docs/Learn/JavaScript>
3. HTML посилання, якоря [Електронний ресурс]. – Режим доступу: <https://webcomplex.com.ua/osnovy-html4/urok-3-html-ssylki-yakorya-i-yakornye-ssylki.html>
4. Що таке landing page [Електронний ресурс]. – Режим доступу: <https://ag.marketing/blog/shcho-take-lending/>
5. SQL Joins [Електронний ресурс]. – Режим доступу: https://www.w3schools.com/sql/sql_join.asp
6. MySQL Workbench [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/MySQL_Workbench
7. SQL Tutorial [Електронний ресурс]. – Режим доступу: <https://www.javatpoint.com/sql-tutorial>
8. SQL Select [Електронний ресурс]. – Режим доступу: <https://www.programiz.com/sql/select>
9. The modern JavaScript tutorial [Електронний ресурс]. – Режим доступу: <https://javascript.info/>
10. CSS Animation [Електронний ресурс]. – Режим доступу: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Animations/Using_CSS_animations
11. Use case diagram tutorial [Електронний ресурс]. – Режим доступу: <https://creatly.com/blog/diagrams/use-case-diagram-tutorial/>
12. An introduction to the world of user stories [Електронний ресурс]. – Режим доступу: <https://www.simplilearn.com/tutorials/agile-scrum-tutorial/user-stories>

ДОДАТКИ

Додаток А

Створення та заповнення бази даних.

Лістинг 1.1.1

```
CREATE TABLE customer(  
    id_customer INT,  
    cstmr_name VARCHAR(50),  
    cstmr_surname VARCHAR(50),  
    cstmr_mob_phone VARCHAR(50),  
    PRIMARY KEY(id_customer));  
  
CREATE TABLE taable(  
    id_table INT,  
    num_of_sits VARCHAR(50),  
    PRIMARY KEY(id_table));  
  
CREATE TABLE menu(  
    id_dish INT,  
    dishName VARCHAR(50),  
    dishPrice INT,  
    PRIMARY KEY(id_dish));  
  
ALTER TABLE menu MODIFY dishName VARCHAR(50);  
  
CREATE TABLE employee(  
    id_employee INT,
```

```
emplType VARCHAR(50),  
empl_name VARCHAR(50),  
empl_surname VARCHAR(50),  
empl_mob_phone VARCHAR(50),  
date_of_start_work DATE,  
employeeSalary INT,  
PRIMARY KEY(id_employee));
```

```
CREATE TABLE reservation(  
id_reservation INT,  
custoner_id INT,  
table_id INT,  
number_of_person INT,  
res_date DATE,  
res_time TIME,  
PRIMARY KEY(id_reservation),  
FOREIGN KEY(custoner_id) REFERENCES customer(id_customer),  
FOREIGN KEY(table_id) REFERENCES taable(id_table));
```

```
CREATE TABLE order_r(  
order_id INT,  
customer_id INT,  
reservation_id INT,
```

```

employee_id INT,

dish_id INT,

order_date DATE,

order_time TIME,

totalPrice INT,

PRIMARY KEY(order_id),

FOREIGN KEY(customer_id) REFERENCES customer(id_customer),

FOREIGN          KEY(reservation_id)          REFERENCES
reservation(id_reservation),

FOREIGN KEY(employee_id) REFERENCES employee(id_employee),

FOREIGN KEY(dish_id) REFERENCES menu(id_dish));

```

Також кожному ідентифікатору видали відповідний до них ключ.

Після цього за допомогою INSERT INTO заповнюємо таблицю:

Лістинг 2.1.2

```

INSERT INTO customer(id_customer, cstmr_name, cstmr_surname,
cstmr_mob_phone) values

('1', 'Anna', 'Lee', '+390936588111'),

('2', 'Sadie', 'Ward', '+390666511116'),

('3', 'Riley', 'Hall', '+390442382341'),

('4', 'Charlotte', 'Robinson', '+390991238766'),

('5', 'Robert', 'King', '+390674419196'),

('6', 'Saitama', 'Punch', '+390448722091'),

('7', 'Forte', 'Air', '+390676589890'),

('8', 'Foltest', 'Merigold', '+390622992648'),

```



```
('9', 'Leo', 'Bonhart', '+390936511116'),  
('10', 'Yarpen', 'Zigrin', '+390665433891');  
INSERT INTO taable(id_table, num_of_sits) values  
('101', '4'),  
('102', '4'),  
('103', '4'),  
('104', '2'),  
('105', '2'),  
('106', '2'),  
('107', '6'),  
('108', '6'),  
('109', '8'),  
('110', '2');  
INSERT INTO menu(id_dish, dishName, dishPrice) values  
('0001', 'Gnocchi', '20'),  
('0002', 'Lasagne', '15'),  
('0003', 'Gelato', '10'),  
('0004', 'Ribollita', '25'),  
('0005', 'Polenta', '22'),  
('0006', 'Tortelli and Ravioli', '30'),  
('0007', 'Pizza Margherita', '15'),  
('0008', 'Risotto', '40'),  
('0009', 'Maraaschino', '28'),
```

```
('0010', 'Limoncello', '50');
```

```
INSERT INTO employee(id_employee, emplType, empl_name,
empl_surname,      empl_mob_phone,      date_of_start_work,
employeeSalary) values
```

```
('0100', 'waiter', 'Nick', 'Fury', '+390936588771', '2022-01-
01', '1250');
```

```
INSERT INTO employee(id_employee, emplType, empl_name,
empl_surname,      empl_mob_phone,      date_of_start_work,
employeeSalary) values
```

```
('0200', 'waiter', 'Steven', 'Rogers', '+390641123133',
'2021-09-01', '1750'),
```

```
('0300', 'waiter', 'Clinton', 'Barton', '+390090033001',
'2022-03-01', '800'),
```

```
('0400', 'waiter', 'Carol', 'Danvers', '+390325655552',
'2022-04-15', '450'),
```

```
('0500', 'bartender', 'Geralt', 'Rivia', '+390123456781',
'2021-07-30', '2000');
```

```
INSERT INTO reservation(id_reservation, custoner_id,
table_id, number_of_person, res_date, res_time) values
```

```
('10001', '1', '101', '4', '2022-05-20', '15:30'),
```

```
('10002', '2', '102', '2', '2022-05-22', '19:30'),
```

```
('10003', '3', '103', '2', '2022-05-19', '17:00'),
```

```
('10004', '4', '104', '2', '2022-05-23', '21:00'),
```

```
('10005', '5', '105', '2', '2022-05-20', '18:00'),
```

```
('10006', '6', '106', '2', '2022-05-20', '16:00'),
```

```
('10007', '7', '107', '4', '2022-05-25', '22:00'),
```

```

('10008', '8', '108', '4', '2022-05-19', '18:30'),
('10009', '9', '109', '6', '2022-05-22', '20:00'),
('10010', '10', '110', '2', '2022-05-21', '22:00');

INSERT INTO reservation(id_reservation, custoner_id,
table_id, number_of_person, res_date, res_time) values
('10011', '2', '105', '2', '2022-05-20', '13:30');

INSERT INTO order_r(order_id, customer_id, reservation_id,
employee_id, dish_id, order_date, order_time, totalPrice)
values
('201', '1', '10001', '0100', '0001', '2022-05-20', '15:30',
'20'),
('202', '2', '10002', '0100', '0006', '2022-05-22', '19:30',
'30'),
('203', '3', '10003', '0200', '0005', '2022-05-19', '17:00',
'22'),
('204', '4', '10004', '0200', '0002', '2022-05-23', '21:00',
'15'),
('205', '5', '10005', '0300', '0004', '2022-05-20', '18:00',
'25'),
('206', '6', '10006', '0300', '0003', '2022-05-20', '16:00',
'10'),
('207', '7', '10007', '0400', '0008', '2022-05-25', '22:00',
'40'),
('208', '8', '10008', '0400', '0007', '2022-05-19', '18:30',
'15'),

```

```
('209', '9', '10009', '0500', '0009', '2022-05-22', '20:00',  
'28'),  
  
('210', '10', '10010', '0500', '0010', '2022-05-21', '22:00',  
'50');
```

```
INSERT INTO order_r(order_id, customer_id, reservation_id,  
employee_id, dish_id, order_date, order_time, totalPrice)  
values
```

```
('211', '2', '10011', '0100', '0003', '2022-05-22', '13:30',  
'10');
```

За допомогою SQL запитів розробник має змогу дістати будь-яку інформацію, яка йому необхідна.