**Тема\_роботи: База даних «Ресторан».**

**2024 рік**

**ЗАВДАННЯ**

  1. Аналіз предметної області

    2. Побудова ER-моделі

    3. Побудова реляційної схеми с ER-моделі

    4. Створення бази даних

    5. Створення та використання збережених тригерів

    6. Імпорт даних в створену таблицю

    7. Створення користувачів бази даних

    8. Створення представлень для запитів

9. Запити для визначених потреб користувачів

10. Створення та використання збережених процедур та генераторів

11. Оптимізація роботи бази даних

**Вступ**

**Теоретична частина**

1. Аналіз предметної області
   1. Опис предметної області
   2. Опис вхідних даних
   3. Опис вихідних даних
2. Проектування бази даних
   1. Інфологічна модель бази даних
      1. Опис сутностей
      2. Опис атрибутів
      3. Опис зв’язків
      4. ER-діаграма
   2. Нормалізація таблиць при проектуванні

**Практична частина**

1. Створення бази даних за допомогою MySQL
   1. Створення бази даних
   2. Створення таблиць бази даних з генераторами
   3. Створення діаграми бази даних
   4. Створення користувачів
   5. Заповнення бази даних даними
   6. Створення представлень
   7. Створення процедур та функцій
   8. Створення тригерів
   9. Написання DLM запитів
   10. Оптимізація та її результати

**Висновки**

**Список літератури**

**Додатки**

Додаток А. Створення таблиць бази даних

Додаток Б. Заповнення бази даних даними

Додаток В. Написання DLM запитів

ВСТУП

Будь-яка БД є частиною інформаційної системи і призначена для збору, зберігання і маніпулювання інформацією.

У сучасному світі навіть звичайний магазин не обходиться без бази даних. То що вже казати про ресторани. Однією із найбільш вагомих складових туристичної індустрії є ресторанний бізнес. Він є одним із найбільш розповсюджених видів малого бізнесу у світі. У той же час він характеризується високим ступенем конкурентності. Ресторанні заклади ведуть між собою постійну боротьбу за певні сегменти ринку, за пошук нових та за утримання постійних споживачів їхньої продукції та послуг. Доволі важливу роль у цій війні відіграють засоби, за допомогою яких заклади громадського харчування зберігають та обробляють власну інформацію.

Саме це і диктує потребу у надійній та зручній системі, яка дозволить закладам харчування зберігати та обробляти інформацію про основні аспекти своєї роботи. Серед них можна виділити інформацію про працівників закладу, продукцію, постачальників цієї продукції, страви, які подає заклад (в першу чергу для правильного вибору стратегії на ринку, адже це дозволяє аналізувати своє меню) та звісно замовлень.

Вигода цієї системи полягає в тому, що заклад зможе не лише оптимізувати свою роботу за рахунок того, що ця система значно спростить ведення документації та пошук необхідної інформації, але й дозволить самостійно аналізувати власний ринок. Така система дозволить швидко знаходити слабкі, або навпаки сильні сторони вашого бізнесу, за допомогою чого можна впроваджувати більш ефективні стратегії розвитку на ринку.

Метою даної роботи є проектування та реалізація бази даних для ресторану. Ця система повинна не лише зберігати основні дані для ведення документації, але й повинна дозволити користувачу аналізувати роботу свого закладу, що в майбутньому покращить якість послуг, які ним надаються.

Саме в цьому і заключається актуальність цієї роботи, адже в сфері харчування першочергову роль грає саме змога надати якомога кращі послуги, але при цьому не витратити на це занадто багато ресурсів, тобто реалізовувати актуальну та дієву стратегію, щоб заклад зміг розвиватися та приносити більше дивідендів. А саме в цьому я бачу головну задачу цієї системи.

**Теоретична частина**

1. Аналіз предметної області
   1. Опис предметної області

Предметна область - частина реального світу, що описує інформаційна система в залежності від її призначення.

В даний час усі організації повинні мати доступ до інформації. Це дуже важливо, тому бази даних, які задовольняють потреби зі зберіганням й управлінням даних, мають неабияку цінність, бо допомагають людям в цій справі.

При розробці реляційної бази даних для підтримки діяльності ресторану, було проведено дослідження предметної області.

Я зрозумів, що для реалізації даної теми в базі даних необхідно зберігати інформацію про персонал та їх посаду, про гостя та його замовлення, про інгредієнти та їх постачальників, а також про страви та напої.

* 1. Опис вхідних даних

Знаючи свою предметну область нам потрібно ввести дані, що призначені для введення, щоб наша база даних була потрібною. При розробці реляційної бази даних були виділені наступні вхідні дані:

* Інформація про посади
* Інформація про працівників
* Інформація про постачальників інгредієнтів
* Інформація про інгредієнти
* Інформація про страви
* Інформація, про склад страв
* Інформація про гостей
* Інформація про замовлення
* Інформація про страви у замовленнях
* Інформація про напої
* Інформація про напої у замовленнях
  1. Опис вихідних даних

Вихідні дані - повідомлення і результати, які видаються самою системою.

Беруться з постійних даних.

Вихідними даними для бази даних є вихідні запити (табл. 1.1 – 1.11):

Таблиця 1.1 – Довідник Посад (Position)

|  |  |  |
| --- | --- | --- |
| **Назва поля** | **Тип поля** | **Опис поля** |
| id\_position | INT | Id посади |
| Position | VARCHAR(25) | Назва посади |
| Salary | INT | Заробітня плата |
| Duties | TINYTEXT | Обов’язки |
| Requirements | TINYTEXT | Вимоги |

Таблиця 1.2 – Довідник Працівників (Employee)

|  |  |  |
| --- | --- | --- |
| **Назва поля** | **Тип поля** | **Опис поля** |
| id\_employee | INT | Id працівника |
| Last\_Name | VARCHAR(15) | Прізвище |
| First\_Name | VARCHAR(15) | Ім’я |
| Middle\_Name | VARCHAR(20) | По батькові |
| Gender | VARCHAR(6) | Стать |
| Phone | INT | Мобільний телефон |
| Age | INT | Вік |
| position\_id | INT | Id посади |

Таблиця 1.3 – Довідник Постачальників (Supplier)

|  |  |  |
| --- | --- | --- |
| **Назва поля** | **Тип поля** | **Опис поля** |
| id\_supplier | INT | Id постачальника |
| Firm | VARCHAR(25) | Назва фірми |
| Phone | INT | Мобільний телефон |
| Address | VARCHAR(30) | Фізична адреса |
| Type\_of\_ingredient | VARCHAR(25) | Тип інгредієнту |

Таблиця 1.4 – Довідник Інгредієнтів (Ingredient)

|  |  |  |
| --- | --- | --- |
| **Назва поля** | **Тип поля** | **Опис поля** |
| id\_ingredient | INT | Id інгредієнту |
| Ingredient\_Name | VARCHAR(25) | Назва інгредієнту |
| Weight | FLOAT | К-сть на складі у кг |
| Price\_for\_kg | FLOAT | Ціна за кг |
| supplier\_id | INT | Id постачальника |

Таблиця 1.5 – Довідник Страви (Meal)

|  |  |  |
| --- | --- | --- |
| **Назва поля** | **Тип поля** | **Опис поля** |
| id\_meal | INT | Id страви |
| Meal\_Name | VARCHAR(25) | Назва страви |
| Price | FLOAT | Ціна за порцію |
| Weight | FLOAT | Вага |

Таблиця 1.6 – Довідник Список інгредієнтів (list\_of\_ingredients)

|  |  |  |
| --- | --- | --- |
| **Назва поля** | **Тип поля** | **Опис поля** |
| id\_li | INT | Id списку |
| ingredient\_id | INT | Id інгредієнту |
| meal\_id | INT | Id страви |

Таблиця 1.7 – Довідник Гостей (Guest)

|  |  |  |
| --- | --- | --- |
| **Назва поля** | **Тип поля** | **Опис поля** |
| id\_guest | INT | Id гостя |
| Last\_Name | VARCHAR(15) | Прізвище |
| First\_Name | VARCHAR(15) | Ім’я |
| Middle\_Name | VARCHAR(15) | По батькові |
| Age | INT | Вік |

Таблиця 1.8 – Довідник Замовлень (Order)

|  |  |  |
| --- | --- | --- |
| **Назва поля** | **Тип поля** | **Опис поля** |
| id\_order | INT | Id замовлення |
| employee\_id | INT | Id офіціанта (працівника) |
| guest\_id | INT | Id гостя |
| Date\_of\_Order | DATE | Дата замовлення |
| Time\_of\_Order | TIME | Час замовлення |
| Payment\_Method | VARCHAR(4) | Спосіб оплати |
| Income | INT | Прибуток |

Таблиця 1.9 – Довідник Список страв (list\_of\_meal)

|  |  |  |
| --- | --- | --- |
| **Назва поля** | **Тип поля** | **Опис поля** |
| id\_lm | INT | Id списку |
| meal\_id | INT | Id страви |
| order\_id | INT | Id замовлення |

Таблиця 1.10 – Довідник Напої (Drink)

|  |  |  |
| --- | --- | --- |
| **Назва поля** | **Тип поля** | **Опис поля** |
| id\_drink | INT | Id напою |
| Drink\_Name | VARCHAR(25) | Назва напою |
| Price | FLOAT | Ціна |
| Size | FLOAT | Об’єм |
| Type | VARCHAR(13) | Алколь / не алкоголь |

Таблиця 1.11 – Довідник Список напоїв (list\_of\_drink)

|  |  |  |
| --- | --- | --- |
| **Назва поля** | **Тип поля** | **Опис поля** |
| id\_ld | INT | Id списку |
| drink\_id | INT | Id напою |
| order\_id | INT | Id замовлення |

1. Проектування бази даних
   1. Інфологічна модель бази даних

Проектувальники інфологічної моделі розглядають як модель сутність - зв'язок”.

Призначення інфографічної моделі полягає у наданні найбільш дійсних способів збору та представлення інформації, що зберігається у базі даних. Ось чому інфологічну модель даних пробують будувати по аналогії з дійсною мовою.

Початковим етапом в проектуванні баз даних - це інфологічна модель, яка допомагає в побудові таблиць та зв’язків між ними.

Існують різні способи опису інфологічної моделі, однак на даний момент найбільш популярним з підходів є той, що заснований на діаграмі "сутністьзв'язок". Отже, при створенні інфологічної моделі ми будемо використовувати ER – діаграму. Ця інфологічна модель - це модель предметної області, яка визначає сутності та зв'язки бази даних. Побудова такої моделі є індивідуальною і особливою, тому однієї методики створення інфологічної моделі немає, кожен робить так як хоче, так як саме він бачить цю модель предметної області.

Інфологічна модель – це опис бази даних, яка буде існувати в майбутньому, опис бд поданий за допомогою діаграм, таблиць, дійсної мови. Будування цієї 13 моделі є продовженням аналізу предметної області інфологічна модель є представленням БД з точки зору того, хто розробляє її.

Головними конструктивними елементами інфологічної моделі даних є: сутності, їх атрибутів та типи зв’язків.

* + - 1. Опис сутностей

Одним із елементом інфологічної моделі "сутність-зв'язок" є сутності.

Сутність - це те, про що накопичується інформація в інформаційній системі і що може бути однозначно унікальне ідентифіковане. При цьому ім'я сутності повинно відображати клас об'єкта або тип об’єкта.

У відповідності з описом предметної області були отримано такі сутності:

* «Посада» - інформація про посади
* «Працівник» - Інформація про працівників
* «Постачальник» - Інформація про постачальників інгредієнтів
* «Інгредієнт» - Інформація про інгредієнти
* «Страва» - Інформація про страви
* «Список інгредієнтів» - Інформація, про склад страв
* «Гість» - Інформація про гостей
* «Замовлення» - Інформація про замовлення
* «Список страв» - Інформація про страви у замовленнях
* «Напій» - Інформація про напої
* «Список напоїв» - Інформація про напої у замовленнях
  + 1. Опис атрибутів

Атрибут - це поіменована характеристика сутності, за допомогою якої моделюється її властивість. Атрибути називають ще інформаційними елементами. У відповідності з описом предметної області були виділені наступні атрибути у кожній сутності:

Посади

|  |
| --- |
| id\_position |
| Position |
| Salary |
| Duties |
| Requirements |

Працівника

|  |
| --- |
| id\_employee |
| Last\_Name |
| First\_Name |
| Middle\_Name |
| Gender |
| Phone |
| Age |
| position\_id |

Постачальника

|  |
| --- |
| id\_supplier |
| Firm |
| Phone |
| Address |
| Type\_of\_ingredients |

Інгредієнта

|  |
| --- |
| id\_ingredient |
| Ingredient\_Name |
| Weight |
| Price\_for\_kg |
| supplier\_id |

Страви

|  |
| --- |
| id\_meal |
| Meal\_Name |
| Price |
| Weight |

Списка інгредієнтів

|  |
| --- |
| id\_li |
| ingredient\_id |
| meal\_id |

Гостя

|  |
| --- |
| id\_guest |
| Last\_Name |
| First\_Name |
| Middle\_Name |
| Age |

Замовлення

|  |
| --- |
| id\_order |
| employee\_id |
| guest\_id |
| Date\_of\_Order |
| Time\_of\_Order |
| Payment\_Method |
| Income |

Списка страв

|  |
| --- |
| id\_lm |
| meal\_id |
| order\_id |

Напоя

|  |
| --- |
| id\_drink |
| Drink\_Name |
| Price |
| Size |
| Type |

Списка напоїв

|  |
| --- |
| id\_ld |
| drink\_id |
| order\_id |

* + 1. Опис зв’язків

Дві сутності можуть пов'язуватися через зв'язок екземплярів однієї сутності з екземплярами іншої сутності.

Основна вимога бази даних - вміти знаходити одну сутність за значеннями інших, для чого необхідно встановити зв’язок між ними. Так як часто в базах даних створюють більше 50 сутностей, то між цими сутністями може бути дуже багато зв’язків. Складність інфологічної моделей визначається наявністю великої кількості зв’язків. За допомогою зв'язків показують відношення між сутністю та атрибутами. Зв’язкам можна надати імена для зручності. Термін зв’язок використовується для позначення типу зв’язку і може відображати одноразовість чи багаторазовість.

За направленістю розрізняють такі зв'язки:

1) Однонаправлений зв'язок поділяється на: однозначний, якщо один екземпляр однієї сутності відповідає точно одному екземпляру іншої сутності, зворотного зв'язку не надається; багатозначний, якщо екземпляр сутності відповідає декільком екземплярам іншої сутності, зворотного зв'язку теж не надається.

2) Двонаправлений зв'язок поділяється на:

однозначний "один до одного" (1:1) одному екземпляру сутності однієї відповідає лише одному екземпляру другої сутності (навпаки теж працює);

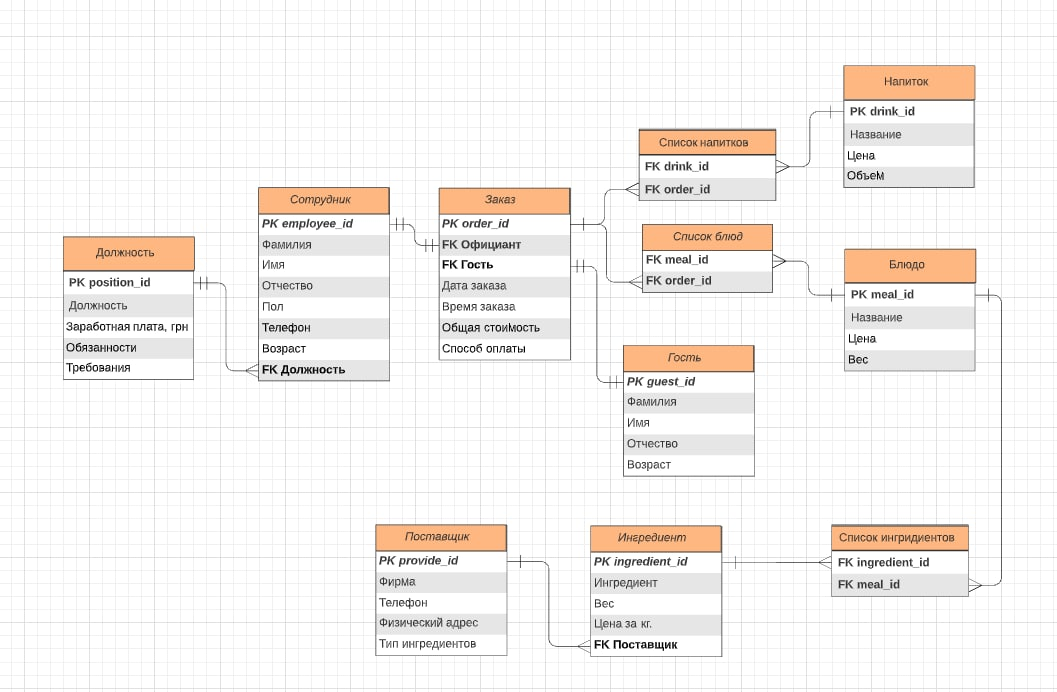
двозначні: "один до багатьох" (1: N) один екземпляр сутності відповідає декільком екземплярам іншої сутності, а кілька екземплярів іншої сутності відповідають лише одному екземпляру сутності "багато до багатьох" (М: N) певна кількість екземплярів однієї сутності відповідає певній кількості екземплярів іншої сутності (навпаки теж працює).

В курсовій роботі були використані наступні типи зв'язків між таблицями:

|  |  |  |  |
| --- | --- | --- | --- |
| № зв’язку | Головна таблиця | Дочірня таблиця | Тип зв'язку |
| 1 | Position | Employee | 1:М |
| 2 | Employee | Order | 1:М |
| 3 | Guest | Order | 1:М |
| 4 | Supplier | Ingredient | 1:М |
| 5 | Ingredient | List\_of\_ingredients | 1:М |
| 6 | Meal | List\_of\_ingredients | 1:М |
| 7 | Meal | List\_of\_meal | 1:М |
| 8 | Drink | List\_of\_drink | 1:М |
| 9 | Order | List\_of\_drink | 1:М |
| 10 | Order | List\_of\_meal | 1:М |
| 11 | Clients | Contracts | 1:М |

* + 1. ER-діаграма

На рисунку представлена ER-діаграма бази даних, на якій відображені всі сутності, їх атрибути та зв’язки між сутностями:



* 1. Нормалізація таблиць при проектуванні

Нормалізація – це формальна процедура, у ході якої атрибути даних групуються в таблиці, а таблиці групуються в базу даних. Нормалізація бази даних – це одна з важливих вимог БД. Нормалізація потрібна для усунення надмірності даних, тобто видалення повторення даних в різних рядках однієї таблиці.

Перший крок до нормалізації бази даних - це позбавлення від повторень.

Другий крок до нормалізації бази даних – всі поля повинні бути неділимими. Тобто треба розділити усі поля, які можна, щоб в подальшому можна було легше шукати потрібну інформацію і працювати з БД.

Третій крок до нормалізації БД – не повинно бути полів, які можна знайти за допомогою інших полів.

Четвертий крок до нормалізації БД – не потрібні поля в БД, які позначають різні види одного і того ж.

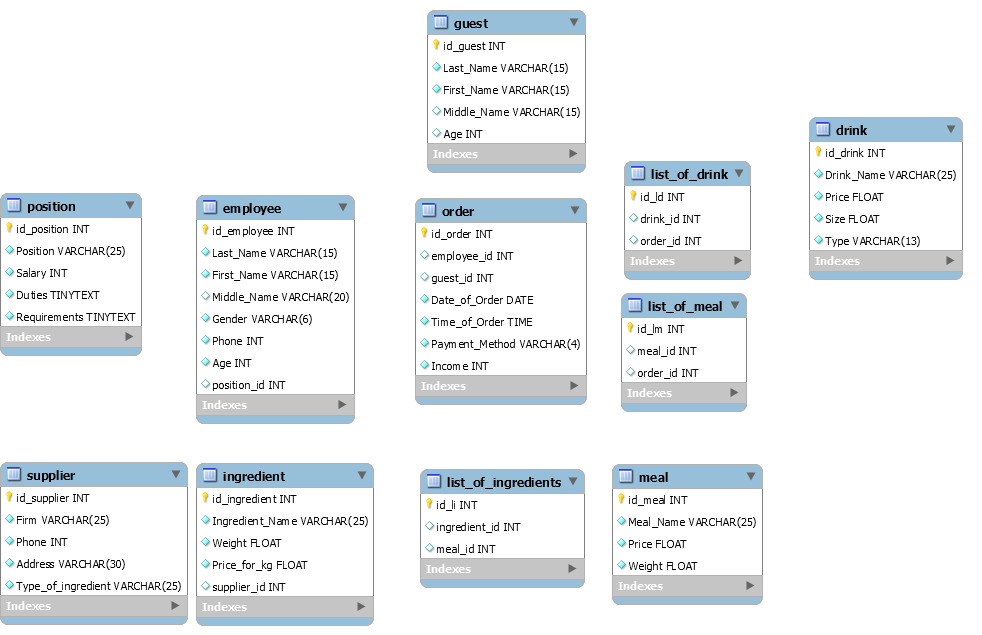
Кожна наступна нормальна форма має кращі властивості, ніж попередня форма. У теорії реляційних БД виділяють п’ять нормальних форми і нормальну форму Бойса-Кодда. Ми будемо розглядати тільки три нормальні форми.

§ Перша нормальна форма вимагає, щоб кожне поле таблиці бази даних було неділимим і не містить повторень.

§ Друга нормальна форма вимагає, щоб всі атрибути таблиці залежали від первинного ключа.

§ Третя нормальна форма вимагає, щоб в таблиці не малося транзитивних залежностей між атрибутами таблиці.

Вигляд після нормалізації таблиць:



**Практична частина**

1. Створення бази даних за допомогою MySQL
   1. Створення бази даних

CREATE DATABASE restaurant;

* 1. Створення таблиць баз даних с генераторами

В базі даних *restaurant* створюємо наступні таблиці:

* Посада (Position)
* Працівник (Employee)
* Постачальник (Supplier)
* Інгредієнт (Ingredient)
* Страва (Meal)
* Список інгредієнтів (list\_of\_ingredients)
* Гість (Guest)
* Замовлення (Order)
* Список страв (list\_of\_meal)
* Напій (Drink)
* Список напоїв (list\_of\_drink)

За допомогою команд:

-- Table `restaurant`.`drink`

CREATE TABLE IF NOT EXISTS `restaurant`.`drink` (

`id\_drink` INT NOT NULL AUTO\_INCREMENT,

`Drink\_Name` VARCHAR(25) NOT NULL,

`Price` FLOAT NOT NULL,

`Size` FLOAT NOT NULL,

`Type` VARCHAR(13) NOT NULL,

PRIMARY KEY (`id\_drink`));

-- Table `restaurant`.`position`

CREATE TABLE IF NOT EXISTS `restaurant`.`position` (

`id\_position` INT NOT NULL AUTO\_INCREMENT,

`Position` VARCHAR(25) NOT NULL,

`Salary` INT NOT NULL,

`Duties` TINYTEXT NOT NULL,

`Requirements` TINYTEXT NOT NULL,

PRIMARY KEY (`id\_position`));

-- Table `restaurant`.`employee`

CREATE TABLE IF NOT EXISTS `restaurant`.`employee` (

`id\_employee` INT NOT NULL AUTO\_INCREMENT,

`Last\_Name` VARCHAR(15) NOT NULL,

`First\_Name` VARCHAR(15) NOT NULL,

`Middle\_Name` VARCHAR(20) NULL DEFAULT NULL,

`Gender` VARCHAR(6) NOT NULL,

`Phone` INT NOT NULL,

`Age` INT NOT NULL,

`position\_id` INT NOT NULL,

PRIMARY KEY (`id\_employee`),

INDEX `id\_position\_idx` (`position\_id` ASC),

CONSTRAINT `position\_id`

FOREIGN KEY (`position\_id`)

REFERENCES `restaurant`.`position` (`id\_position`)

ON DELETE CASCADE

ON UPDATE CASCADE);

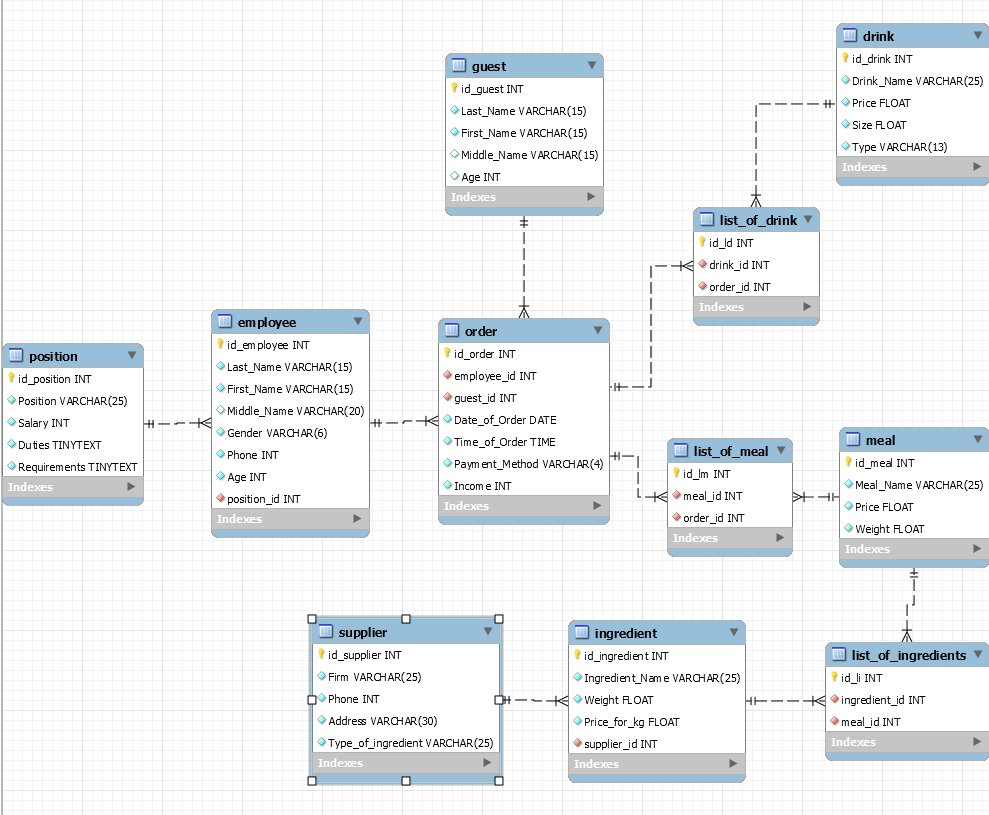
…



Всі інші таблиці можна переглянути в «Додатку А»

* 1. Створення діаграми бази даних

За допомогою вбудованих в MySQL WorkBench функцій створюємо діаграму бази даних *restaurant*



* 1. Створення користувачів

В базі даних створюємо наступних користувачів:

* Адміністратор (Admin) - має всі можливі права. Слідкує за цілісністю бази даних, за потреби може вносити перві правки.
* Менеджер (Manager) – може переглядати усі таблиці. Переглядає та слідкує за усією інформацією.
* Офіціант (Waiter) – може переглядати усі таблиці. Має права додавати значення в таблиці: Замовлення, Гість, Список страв, Список напоїв.
* Кухар (Cook) – може переглядати такі таблиці як: Страва, Інгредієнт, Список інгредієнтів та Постачальник.

За допомогою команд:

CREATE USER 'administrator'@'localhost' IDENTIFIED BY 'admin123';

GRANT ALL PRIVILEGES ON restaurant.\* TO 'administrator'@'localhost';

-- DROP USER 'administrator'@'localhost';

CREATE USER 'manager'@'localhost' IDENTIFIED BY 'manager123';

GRANT SELECT ON restaurant.\* TO 'manager'@'localhost';

-- DROP USER 'manager'@'localhost';

CREATE USER 'waiter'@'localhost' IDENTIFIED BY 'waiter123';

GRANT SELECT ON restaurant.\* TO 'waiter'@'localhost';

GRANT INSERT ON restaurant.order TO 'waiter'@'localhost';

GRANT INSERT ON restaurant.guest TO 'waiter'@'localhost';

GRANT INSERT ON restaurant.list\_of\_meal TO 'waiter'@'localhost';

GRANT INSERT ON restaurant.list\_of\_drink TO 'waiter'@'localhost';

-- DROP USER 'waiter'@'localhost';

CREATE USER 'cook'@'localhost' IDENTIFIED BY 'cook123';

GRANT SELECT ON restaurant.meal TO 'cook'@'localhost';

GRANT SELECT ON restaurant.ingredient TO 'cook'@'localhost';

GRANT SELECT ON restaurant.list\_of\_ingredients TO 'cook'@'localhost';

GRANT SELECT ON restaurant.supplier TO 'cook'@'localhost';

-- DROP USER 'cook'@'localhost';

* 1. Заповнення бази даних даними

Оскільки даних досить багато, то я наведу приклади заповнень кількох табличок, а код заповнення усіх табличок можна переглянути в «Додатку Б»

INSERT position (Position, Salary, Duties, Requirements)

VALUES

('Manager', 52000, 'https://jobs.ua/rus/job\_description/view/919', 'Well-groomed appearance, strict dress code, meeting deadlines'),

('Chef', 70000, 'https://jobs.ua/rus/job\_description/view/250', 'Prepare dishes of increased complexity, meet deadlines, have a diligent look'),

('Sous chef', 52000, 'https://kc-maestro.ru/raznoe/dolzhnostnye-instrukcii/dolzhnostnaya-instrukciya-su-shefa', 'Prepare dishes of increased complexity, replace the chef, meet deadlines, have a diligent look'),

('Cook', 42000, 'https://jobs.ua/rus/job\_description/view/89', 'Prepare meals, meet deadlines, look diligent'),

('Waiter', 30000, 'https://jobs.ua/rus/job\_description/view/708', 'Have a diligent look, observe etiquette, smooth, quickly serve customers'),

('Bartender', 42000, 'https://jobs.ua/rus/job\_description/view/98', 'Have a diligent look, observe etiquette, smooth, quickly serve customers'),

('Cleaner', 15000, 'https://jobs.ua/rus/job\_description/view/54', 'Do the job diligently, look diligent, do not interfere with other personnel');

INSERT employee (Last\_Name, First\_Name, Middle\_Name, Gender, Phone, Age, position\_id)

VALUES

('Goncharova', 'Victoria', 'Sergeevna', 'Female', 0666469693, 32, 1),

('Lavrov', 'Maxim', 'Leonidovich', 'Male', 0925651770, 25, 4),

('Barinov', 'Victor', 'Petrovich', 'Male', 0947019846, 48, 2),

('Anisimov', 'Konstantin', 'Timofeevich', 'Male', 0983384418, 26, 6),

('Soloviev', 'Lev', 'Semyonovich', 'Male', 0997702941, 32, 3),

('Kenensarova', 'Ainura', 'Zhannatbekovna', 'Female', 0664689850, 35, 7),

('Yurchenko', 'Fyodor', 'Mikhailovich', 'Male', 0663878369, 32, 4),

('Benoit', 'Louis', NULL, 'Male', 0990059878, 29, 4),

('Beletska', 'Eva', 'Albertovna', 'Female', 0991559878, 27, 5),

('Noname', 'Ilya', 'Vladimirovich', 'Male', 0990059354, 35, 5),

('Anisimova', 'Anastasia', 'Stepanovna', 'Female', 0660059878, 29, 5),

('Krylov', 'Denis', 'Andreevich', 'Male', 0950059878, 25, 5),

('Chuganin', 'Arseniy', 'Andreevich', 'Male', 050168878, 33, 4),

('Krylova', 'Ekaterina', 'Viktorovna', 'Female', 0663577057, 28, 4),

('Rokossovsky', 'Alexander', 'Sergeevich', 'Male', 050268978, 33, 4),

('Chicherina', 'Violetta', NULL, 'Female', 0992855566, 32, 5),

('Bubnova', 'Alexandra', NULL, 'Female', 0666469703, 24, 5);

* 1. Створення представлень

Створюємо представлення аби полегшити подальшу роботу.

CREATE VIEW notEnoughIngredients AS

SELECT meal\_id, meal.Meal\_Name, ingredient.Ingredient\_Name FROM list\_of\_ingredients

INNER JOIN Meal ON list\_of\_ingredients.meal\_id = meal.id\_meal

INNER JOIN Ingredient ON list\_of\_ingredients.ingredient\_id = ingredient.id\_ingredient

WHERE ingredient.weight = 0;

SELECT \* FROM notEnoughIngredients;

CREATE VIEW showMealWithIngredient AS

SELECT meal\_id, meal.Meal\_Name, ingredient.Ingredient\_Name FROM list\_of\_ingredients

INNER JOIN Meal ON list\_of\_ingredients.meal\_id = meal.id\_meal

INNER JOIN Ingredient ON list\_of\_ingredients.ingredient\_id = ingredient.id\_ingredient

WHERE ingredient.ingredient\_name = 'Onion';

SELECT \* FROM showMealWithIngredient;

* 1. Створення процедур та функцій

Створюємо процедури для отримання потрібних даних з БД та функції для отримання для використання в тригерах.

DELIMITER |

DROP FUNCTION IF EXISTS func1 |

CREATE FUNCTION func1 (guest\_id INT)

RETURNS VARCHAR(50)

DETERMINISTIC

BEGIN

DECLARE adult\_age INT DEFAULT 18;

DECLARE GuestAge INT;

SELECT Age INTO GuestAge FROM guest WHERE guest.id\_guest = guest\_id;

IF GuestAge >= adult\_age THEN

RETURN 'Guest can order alcohol';

ELSE

RETURN 'Guest can not order alcohol';

END IF;

END;

SELECT func1(1);

DELIMITER |

DROP FUNCTION IF EXISTS func2 |

CREATE FUNCTION func2 (drink\_id INT)

RETURNS VARCHAR(60)

DETERMINISTIC

BEGIN

DECLARE Type\_of\_drink VARCHAR(15);

SELECT drink.Type INTO Type\_of\_drink FROM drink WHERE drink.id\_drink = drink\_id;

IF Type\_of\_drink = 'alcoholic' THEN

RETURN 'This drink is alcoholic and cannot be sold to minors';

ELSE

RETURN 'This drink is non-alcoholic and can be sold to minors';

END IF;

END;

SELECT func2(1);

# Процедура

DROP PROCEDURE IF EXISTS Employee\_by\_Last\_name

DELIMITER |

CREATE PROCEDURE Employee\_by\_Last\_name (IN Last\_Name VARCHAR(15))

BEGIN

SELECT \* FROM employee

WHERE employee.Last\_Name = Last\_Name;

END;

CALL Employee\_by\_Last\_name('Lavrov');

DROP PROCEDURE IF EXISTS Ingredient\_by\_ingredient\_name

DELIMITER |

CREATE PROCEDURE Ingredient\_by\_ingredient\_name (IN Ingredient\_Name VARCHAR(15))

BEGIN

SELECT \* FROM Ingredient

WHERE ingredient.Ingredient\_Name = Ingredient\_Name;

END;

CALL Ingredient\_by\_ingredient\_name('Onion');

* 1. Створення тригерів

Для уникнення можливих проблем створюємо наступний тригер:

* При INSERT в таблицю Посада (Position) виникає похибка, оскільки усі можливі посади вже й так додані.

DELIMITER |

CREATE TRIGGER position\_update

BEFORE INSERT ON restaurant.position

FOR EACH ROW BEGIN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = "You can not add new position";

END;

* 1. Написання DLM запитів

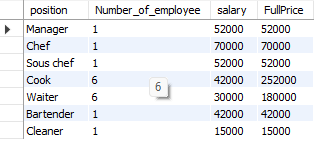
Створюємо 20 DML запитів в рамках бази даних restaurant за допомогою коду (Весь код можна подивитись у «Додатку В»):

-- 1 Скільки грошей необхідно, щоб віддати зарплатню усім працівникам

SELECT position, (SELECT COUNT(\*) FROM employee WHERE position\_id = position.id\_position) AS Number\_of\_employee,

salary, salary \* (SELECT COUNT(\*) FROM employee WHERE position\_id = position.id\_position) AS FullPrice

FROM position;



-- 2 Повна інформація про працівника

SELECT id\_employee, CONCAT(Last\_Name, ' ', First\_Name) AS Full\_Name,

(SELECT position FROM position WHERE position\_id = id\_position) AS Position,

(SELECT salary FROM position WHERE position\_id = id\_position) AS Salary

FROM employee

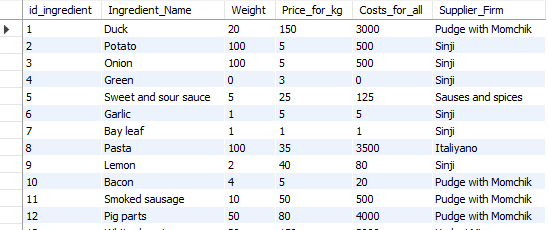
ORDER BY Salary

DESC;

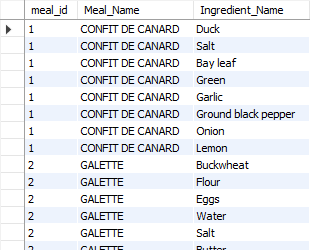


-- 3 Інформація про інгредієнт, а також про фірму-постачальника



-- 4 Інформація про інгредієнт, постачальника, а також про те, скільки було витрачено на них грошей

-- 5 Страва та інгредієнти для неї

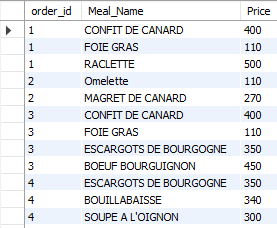


-- 6 Список страв у замовленні

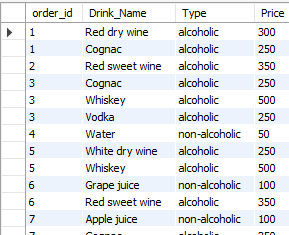
SELECT order\_id, meal.Meal\_Name, meal.Price FROM list\_of\_meal

INNER JOIN Meal ON list\_of\_meal.meal\_id = meal.id\_meal

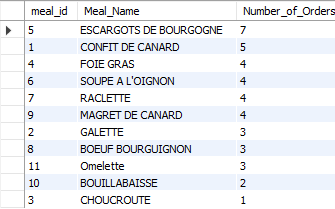
ORDER BY order\_id;



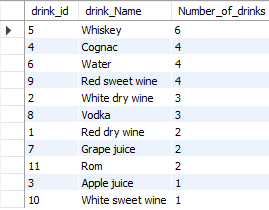
-- 7 Список напоїв у замовленні



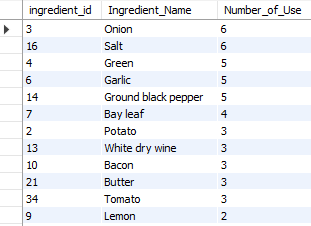
-- 8 Топ страв, по к-сті замовлень



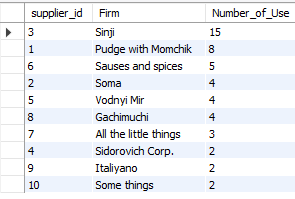
-- 9 Топ напоїв по к-сті замовлень



-- 10 Топ інгредієнтів по к-сті «використань»



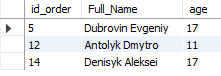
-- 11 Топ фірм по к-сті інгредієнтів, що вони поставляють



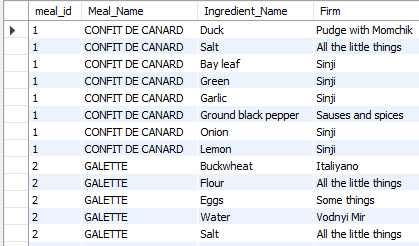
-- 12 Повна інформація про замовлення



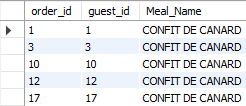
-- 13 Список тих, хто не може замовляти алкогольні напої



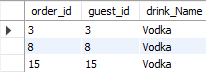
-- 14 Список фірм постачальників, та для яких страв їх інгредієнти потрібні



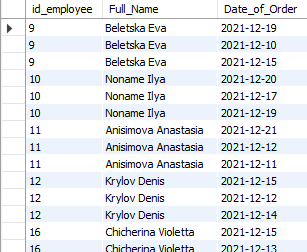
-- 15 ID клієнтів, котрі замовили страву «Назва страви»



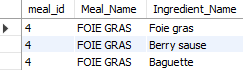
-- 16 ID клієнтів, котрі замовили напій «Назва напою»



-- 17 Інформація про офіціантів, та дні коли вони приймали замовлення



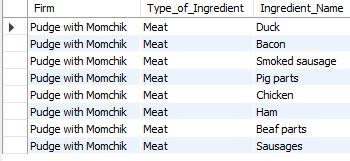
-- 18 Інгредієнти для конкретної страви



-- 19 Порахувати підвищення певному робітнику



-- 20 Список інгредієнтів разом з фірмою-постачальником, котрий сортується за типом інгредієнта



ВИСНОВКИ

Метою курсової роботи було закріплення студентом навичок з проектування, реалізації реляційних баз даних та їх використання для практичних задач.

Тож для виконання курсової роботи були проведені необхідні дослідження, що безпосередньо стосуються предметної області, а саме ресторану. Спочатку я провів повний аналіз аналіз нашої предметної області, а вже тоді почав її описувати.

Після цього я побудував концептуальну модель для якої я використовував мову ER-опису предметної області, яка базується на концепції, що інформаційна модель предметної області може бути описана із застосування таких понять, як сутність, атрибут, зв’язок. Після побудови ER діаграми, я провів нормалізацію таблиць нашої бази даних.

І тільки після цих дій я перейшов до практичної частини і почав створювати саму базу даних, її таблиць. Потім я створив діаграму, аби перевірити цілісність бази даних і коректність.

Далі я створив користувачів, заповнив базу даних даними, створив представлення, процедури, функції, тригери та 20 DML запитів. І в кінці оптимізував їх.

В даній курсовій роботі була розроблена база даних «База даних ресторану» в системі управління базами даних MySQL WorkBench.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

* Лекційний матеріал з предмету «Бази даних»
* METANIT <https://metanit.com/sql/mysql/1.1.php>
* Офіційна документація MySQL <https://dev.mysql.com/doc/>

**Додатки**

ДОДАТОК А

Скрипти для створення таблиць

CREATE SCHEMA IF NOT EXISTS `restaurant` DEFAULT CHARACTER SET utf8 ;

USE `restaurant` ;

-- -----------------------------------------------------

-- Table `restaurant`.`drink`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `restaurant`.`drink` (

`id\_drink` INT NOT NULL AUTO\_INCREMENT,

`Drink\_Name` VARCHAR(25) NOT NULL,

`Price` FLOAT NOT NULL,

`Size` FLOAT NOT NULL,

`Type` VARCHAR(13) NOT NULL,

PRIMARY KEY (`id\_drink`));

-- -----------------------------------------------------

-- Table `restaurant`.`position`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `restaurant`.`position` (

`id\_position` INT NOT NULL AUTO\_INCREMENT,

`Position` VARCHAR(25) NOT NULL,

`Salary` INT NOT NULL,

`Duties` TINYTEXT NOT NULL,

`Requirements` TINYTEXT NOT NULL,

PRIMARY KEY (`id\_position`));

-- -----------------------------------------------------

-- Table `restaurant`.`employee`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `restaurant`.`employee` (

`id\_employee` INT NOT NULL AUTO\_INCREMENT,

`Last\_Name` VARCHAR(15) NOT NULL,

`First\_Name` VARCHAR(15) NOT NULL,

`Middle\_Name` VARCHAR(20) NULL DEFAULT NULL,

`Gender` VARCHAR(6) NOT NULL,

`Phone` INT NOT NULL,

`Age` INT NOT NULL,

`position\_id` INT NOT NULL,

PRIMARY KEY (`id\_employee`),

INDEX `id\_position\_idx` (`position\_id` ASC),

CONSTRAINT `position\_id`

FOREIGN KEY (`position\_id`)

REFERENCES `restaurant`.`position` (`id\_position`)

ON DELETE CASCADE

ON UPDATE CASCADE);

-- -----------------------------------------------------

-- Table `restaurant`.`guest`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `restaurant`.`guest` (

`id\_guest` INT NOT NULL AUTO\_INCREMENT,

`Last\_Name` VARCHAR(15) NOT NULL,

`First\_Name` VARCHAR(15) NOT NULL,

`Middle\_Name` VARCHAR(15) NULL DEFAULT NULL,

`Age` INT NULL DEFAULT NULL,

PRIMARY KEY (`id\_guest`));

-- -----------------------------------------------------

-- Table `restaurant`.`supplier`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `restaurant`.`supplier` (

`id\_supplier` INT NOT NULL AUTO\_INCREMENT,

`Firm` VARCHAR(25) NOT NULL,

`Phone` INT NOT NULL,

`Address` VARCHAR(30) NOT NULL,

`Type\_of\_ingredient` VARCHAR(25) NOT NULL,

PRIMARY KEY (`id\_supplier`));

-- -----------------------------------------------------

-- Table `restaurant`.`ingredient`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `restaurant`.`ingredient` (

`id\_ingredient` INT NOT NULL AUTO\_INCREMENT,

`Ingredient\_Name` VARCHAR(25) NOT NULL,

`Weight` FLOAT NOT NULL,

`Price\_for\_kg` FLOAT NOT NULL,

`supplier\_id` INT NOT NULL,

PRIMARY KEY (`id\_ingredient`),

INDEX `id\_supplier\_idx` (`supplier\_id` ASC),

CONSTRAINT `supplier\_id`

FOREIGN KEY (`supplier\_id`)

REFERENCES `restaurant`.`supplier` (`id\_supplier`)

ON DELETE CASCADE

ON UPDATE CASCADE);

-- -----------------------------------------------------

-- Table `restaurant`.`order`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `restaurant`.`order` (

`id\_order` INT NOT NULL AUTO\_INCREMENT,

`employee\_id` INT NOT NULL,

`guest\_id` INT NOT NULL,

`Date\_of\_Order` DATE NOT NULL,

`Time\_of\_Order` TIME NOT NULL,

`Payment\_Method` VARCHAR(4) NOT NULL,

`Income` INT NOT NULL,

PRIMARY KEY (`id\_order`),

INDEX `id\_employee\_idx` (`employee\_id` ASC),

INDEX `id\_guest\_idx` (`guest\_id` ASC),

CONSTRAINT `employee\_id`

FOREIGN KEY (`employee\_id`)

REFERENCES `restaurant`.`employee` (`id\_employee`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `guest\_id`

FOREIGN KEY (`guest\_id`)

REFERENCES `restaurant`.`guest` (`id\_guest`)

ON DELETE CASCADE

ON UPDATE CASCADE);

-- -----------------------------------------------------

-- Table `restaurant`.`list\_of\_drink`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `restaurant`.`list\_of\_drink` (

`id\_ld` INT NOT NULL AUTO\_INCREMENT,

`drink\_id` INT NOT NULL,

`order\_id` INT NOT NULL,

INDEX `id\_drink\_idx` (`drink\_id` ASC),

INDEX `id\_order\_idx` (`order\_id` ASC),

PRIMARY KEY (`id\_ld`),

CONSTRAINT `drink\_id`

FOREIGN KEY (`drink\_id`)

REFERENCES `restaurant`.`drink` (`id\_drink`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `order\_id2`

FOREIGN KEY (`order\_id`)

REFERENCES `restaurant`.`order` (`id\_order`)

ON DELETE CASCADE

ON UPDATE CASCADE);

-- -----------------------------------------------------

-- Table `restaurant`.`meal`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `restaurant`.`meal` (

`id\_meal` INT NOT NULL AUTO\_INCREMENT,

`Meal\_Name` VARCHAR(25) NOT NULL,

`Price` FLOAT NOT NULL,

`Weight` FLOAT NOT NULL,

PRIMARY KEY (`id\_meal`));

-- -----------------------------------------------------

-- Table `restaurant`.`list\_of\_ingredients`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `restaurant`.`list\_of\_ingredients` (

`id\_li` INT NOT NULL AUTO\_INCREMENT,

`ingredient\_id` INT NOT NULL,

`meal\_id` INT NOT NULL,

INDEX `id\_ingredient\_idx` (`ingredient\_id` ASC),

INDEX `id\_meal\_idx` (`meal\_id` ASC),

PRIMARY KEY (`id\_li`),

CONSTRAINT `ingredient\_id`

FOREIGN KEY (`ingredient\_id`)

REFERENCES `restaurant`.`ingredient` (`id\_ingredient`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `meal\_id1`

FOREIGN KEY (`meal\_id`)

REFERENCES `restaurant`.`meal` (`id\_meal`)

ON DELETE CASCADE

ON UPDATE CASCADE);

-- -----------------------------------------------------

-- Table `restaurant`.`list\_of\_meal`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `restaurant`.`list\_of\_meal` (

`id\_lm` INT NOT NULL AUTO\_INCREMENT,

`meal\_id` INT NOT NULL,

`order\_id` INT NOT NULL,

INDEX `id\_meal\_idx` (`meal\_id` ASC),

INDEX `id\_order\_idx` (`order\_id` ASC),

PRIMARY KEY (`id\_lm`),

CONSTRAINT `meal\_id2`

FOREIGN KEY (`meal\_id`)

REFERENCES `restaurant`.`meal` (`id\_meal`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `order\_id1`

FOREIGN KEY (`order\_id`)

REFERENCES `restaurant`.`order` (`id\_order`)

ON DELETE CASCADE

ON UPDATE CASCADE);

Додаток Б

Скрипти на додавання тестових даних до таблиць:

USE restaurant;

-- після цієї команди всі подальші запроси будуть виконуватися в контексті бази restaurant

INSERT position (Position, Salary, Duties, Requirements)

VALUES

('Manager', 52000, 'https://jobs.ua/rus/job\_description/view/919', 'Well-groomed appearance, strict dress code, meeting deadlines'),

('Chef', 70000, 'https://jobs.ua/rus/job\_description/view/250', 'Prepare dishes of increased complexity, meet deadlines, have a diligent look'),

('Sous chef', 52000, 'https://kc-maestro.ru/raznoe/dolzhnostnye-instrukcii/dolzhnostnaya-instrukciya-su-shefa', 'Prepare dishes of increased complexity, replace the chef, meet deadlines, have a diligent look'),

('Cook', 42000, 'https://jobs.ua/rus/job\_description/view/89', 'Prepare meals, meet deadlines, look diligent'),

('Waiter', 30000, 'https://jobs.ua/rus/job\_description/view/708', 'Have a diligent look, observe etiquette, smooth, quickly serve customers'),

('Bartender', 42000, 'https://jobs.ua/rus/job\_description/view/98', 'Have a diligent look, observe etiquette, smooth, quickly serve customers'),

('Cleaner', 15000, 'https://jobs.ua/rus/job\_description/view/54', 'Do the job diligently, look diligent, do not interfere with other personnel');

INSERT employee (Last\_Name, First\_Name, Middle\_Name, Gender, Phone, Age, position\_id)

VALUES

('Goncharova', 'Victoria', 'Sergeevna', 'Female', 0666469693, 32, 1), -- 1

('Lavrov', 'Maxim', 'Leonidovich', 'Male', 0925651770, 25, 4), -- 2

('Barinov', 'Victor', 'Petrovich', 'Male', 0947019846, 48, 2), -- 3

('Anisimov', 'Konstantin', 'Timofeevich', 'Male', 0983384418, 26, 6), -- 4

('Soloviev', 'Lev', 'Semyonovich', 'Male', 0997702941, 32, 3), -- 5

('Kenensarova', 'Ainura', 'Zhannatbekovna', 'Female', 0664689850, 35, 7), -- 6

('Yurchenko', 'Fyodor', 'Mikhailovich', 'Male', 0663878369, 32, 4), -- 7

('Benoit', 'Louis', NULL, 'Male', 0990059878, 29, 4), -- 8

('Beletska', 'Eva', 'Albertovna', 'Female', 0991559878, 27, 5), -- 9

('Noname', 'Ilya', 'Vladimirovich', 'Male', 0990059354, 35, 5), -- 10

('Anisimova', 'Anastasia', 'Stepanovna', 'Female', 0660059878, 29, 5), -- 11

('Krylov', 'Denis', 'Andreevich', 'Male', 0950059878, 25, 5), -- 12

('Chuganin', 'Arseniy', 'Andreevich', 'Male', 050168878, 33, 4), -- 13

('Krylova', 'Ekaterina', 'Viktorovna', 'Female', 0663577057, 28, 4), -- 14

('Rokossovsky', 'Alexander', 'Sergeevich', 'Male', 050268978, 33, 4), -- 15

('Chicherina', 'Violetta', NULL, 'Female', 0992855566, 32, 5), -- 16

('Bubnova', 'Alexandra', NULL, 'Female', 0666469703, 24, 5); -- 17

INSERT supplier (Firm, Phone, Address, Type\_of\_ingredient)

VALUES

('Pudge with Momchik', 0992855566, 'Radunskaya 42/10', 'Meat'), -- 1

('Soma', 0664567056, 'Malo-solenaya 23', 'Fish and seafood'), -- 2

('Sinji', 0505515562, 'Ovoshnaya 21', 'Vegetables/Fruits'), -- 3

('Sidorovich Corp.', 0990990999, 'Stalkerskaya 2', 'Rare Ingredients'), -- 4

('Vodnyi Mir', 0963492674, 'Vodnaya 4', 'Drinks'), -- 5

('Sauses and spices', 0679692425, 'Rovnoe 11', 'Sauces/Spices'), -- 6

('All the little things', 0679692425, 'Rovnoe 11', 'Sugar/Salt/Flour'), -- 7

('Gachimuchi', 0902134545, 'Mlechnaya 20', 'Milk Products'), -- 8

('Italiyano', 0505502523, 'Italiyanskaya 21', 'Cereals/Pasta'), -- 9

('Some things', 0501123562, 'Monolitnaya 8', 'Others'); -- 10

INSERT ingredient (Ingredient\_Name, Weight, Price\_for\_kg, supplier\_id)

VALUES

('Duck', 20, 150, 1), -- 1

('Potato', 100, 5, 3), -- 2

('Onion', 100, 5, 3), -- 3

('Green', 0, 3, 3), -- 4

('Sweet and sour sauce', 5, 25, 6), -- 5

('Garlic', 1, 5, 3), -- 6

('Bay leaf', 1, 1, 3), -- 7

('Pasta', 100, 35, 9), -- 8

('Lemon', 2, 40, 3), -- 9

('Bacon', 4, 5, 1), -- 10

('Smoked sausage', 10, 50, 1), -- 11

('Pig parts', 50, 80, 1), -- 12

('White dry wine', 20, 150, 5), -- 13

('Ground black pepper', 50, 80, 6), -- 14

('Sugar', 50, 30, 7), -- 15

('Salt', 50, 20, 7), -- 16

('Flour', 50, 25, 7), -- 17

('Foie gras', 3, 2340, 4), -- 18

('Berry sause', 5, 25, 6), -- 19

('Grape snail', 0, 65, 4), -- 20

('Butter', 30, 40, 8), -- 21

('Bouquet garni', 2, 19, 6), -- 22

('Chicken', 20, 65, 1), -- 23

('Baguette', 5, 35, 10), -- 24

('Hard cheese', 10, 65, 8), -- 25

('Cheese raclette', 10, 105, 8), -- 26

('Pickles', 10, 40, 3), -- 27

('Cherry tomatoes', 10, 75, 3), -- 28

('Canned mushrooms', 10, 55, 3), -- 29

('Ham', 10, 75, 1), -- 30

('Beaf parts', 50, 105, 1), -- 31

('Red dry wine', 20, 155, 5), -- 32

('Carrot', 50, 30, 3), -- 33

('Tomato', 50, 40, 3), -- 34

('Celery', 10, 45, 3), -- 35

('Sauerkraut', 10, 30, 3), -- 36

('Mushroom', 50, 25, 3), -- 37

('Cognac', 20, 150, 5), -- 38

('Carp', 20, 90, 2), -- 39

('Mussels', 20, 110, 2), -- 40

('Shrimp', 10, 200, 2), -- 41

('Squid', 15, 150, 2), -- 42

('Buckwheat', 20, 50, 9), -- 43

('Apple', 20, 50, 3), -- 44

('Water', 100, 20, 5), -- 45

('Eggs', 2, 10, 10), -- 46

('Sausages', 30, 10, 1), -- 47

('Spice set', 1, 20, 6), -- 48

('Milk', 10, 45, 8); -- 49

INSERT meal (Meal\_Name, Price, Weight)

VALUES

('CONFIT DE CANARD', 400, 0.3), -- 1

('GALETTE', 250, 0.2), -- 2

('CHOUCROUTE', 500, 0.4), -- 3

('FOIE GRAS', 110, 0.02), -- 4

('ESCARGOTS DE BOURGOGNE', 350, 0.1), -- 5

("SOUPE A L'OIGNON", 300, 0.4), -- 6

('RACLETTE', 500, 0.45), -- 7

('BOEUF BOURGUIGNON', 450, 0.35), -- 8

('MAGRET DE CANARD', 270, 0.4), -- 9

('BOUILLABAISSE', 340, 0.35), -- 10

('Omelette', 110, 0.3); -- 11

INSERT list\_of\_Ingredients (ingredient\_id, meal\_id)

VALUES (1,1),(16,1),(7,1),(4,1),(6,1),(14,1),(3,1),(9,1),

(43,2),(17,2),(46,2),(45,2),(16,2),(21,2),(25,2),(30,2),

(2,3),(10,3),(36,3),(11,3),(12,3),(47,3),(13,3),(6,3),(4,3),(16,3),(14,3),(7,3),

(18,4),(19,4),(24,4),

(20,5),(21,5),(6,5),(4,5),(3,5),(14,5),(16,5),(13,5),(22,5),

(3,6),(21,6),(23,6),(4,6),(24,6),(25,6),

(2,7),(26,7),(27,7),(3,7),(28,7),(29,7),(30,7),

(10,8),(31,8),(32,8),(3,8),(33,8),(35,8),(6,8),(34,8),(4,8),(7,8),(23,8),(37,8),

(1,9),(16,9),(14,9),(38,9),(48,9),

(39,10),(40,10),(41,10),(42,10),(6,10),(14,10),(2,10),(34,10),(9,10),(13,10),(7,10),(3,10),(48,10),(16,10),

(46,11),(49,11),(34,11),(10,11);

INSERT guest (Last\_Name, First\_Name, Middle\_Name, Age)

VALUES

('Antolyk', 'Danylo', 'Igorovich', 18), -- 1

('Bahmutchenko', 'Egor', NULL, 19), -- 2

('Frankov', 'Aleksey', 'Romanov', 20), -- 3

('Nagorskiy', 'Maksym', 'Leonidovich', 18), -- 4

('Dubrovin', 'Evgeniy', 'Romanov', 17), -- 5

('Tryhub', 'Diana', 'Konstantinovna', 18), -- 6

('Kotenko', 'Dmytryo', 'Oleksandrovich', 24), -- 7

('Makarov', 'Ilya', 'Sergeyovich', 45), -- 8

('Herrington', 'Billy', NULL, 48), -- 9

('Darkholme', 'Van', NULL, 39), -- 10

('Hrishu', 'Dmytro', 'Pavlovich', NULL), -- 11

('Antolyk', 'Dmytro', 'Igorovich', 11), -- 12

('Antolyk', 'Ihor', 'Mykhaolovich', 44), -- 13

('Denisyk', 'Aleksei', 'Pervyi', 17), -- 14

('Savitskiy', 'Yaroslav', NULL, 18), -- 15

('Didur', 'Mykhail', 'Anatolievich', 27), -- 16

('Nakonechna', 'Iryna', NULL, 18), -- 17

('Uzumaki', 'Naruto', NULL, 19); -- 18

INSERT restaurant.order (employee\_id, guest\_id, Date\_of\_Order,Time\_of\_Order, Payment\_Method, Income)

VALUES -- 9 10 11 12 16 17

(10, 1, '2021-12-20','15:25:34', 'cash', 1560), -- 1

(9, 2, '2021-12-19','16:35:34', 'card', 730), -- 2

(11, 3, '2021-12-21','17:15:34', 'cash', 2310), -- 3

(12, 4, '2021-12-15','18:45:34', 'card', 1040), -- 4

(16, 5, '2021-12-15','19:25:34', 'cash', 1350), -- 5

(17, 6, '2021-12-20','20:35:34', 'card', 1220), -- 6

(11, 7, '2021-12-12','12:25:34', 'cash', 1580), -- 7

(10, 8, '2021-12-17','13:55:34', 'card', 1200), -- 8

(9, 9, '2021-12-10','14:45:34', 'cash', 460), -- 9

(17, 10, '2021-12-19','15:45:34', 'card', 1250), -- 10

(16, 11, '2021-12-13','16:15:34', 'card', 2550), -- 11

(12, 12, '2021-12-12','17:25:34', 'cash', 860), -- 12

(10, 13, '2021-12-19','18:45:34', 'card', 850), -- 13

(11, 14, '2021-12-11','19:35:34', 'card', 620), -- 14

(9, 15, '2021-12-15','15:25:34', 'cash', 1440), -- 15

(12, 16, '2021-12-14','14:15:34', 'card', 360), -- 16

(16, 17, '2021-12-13','17:15:34', 'cash', 2150), -- 17

(17, 18, '2021-12-16','19:55:34', 'card', 350); -- 18

INSERT list\_of\_meal (meal\_id, order\_id)

VALUES

(1,1),(7,1),(4,1), -- 1

(9,2),(11,2), -- 2

(4,3),(5,3),(1,3),(8,3), -- 3

(6,4),(5,4),(10,4), -- 4

(2,5),(5,5), -- 5

(7,6),(9,6), -- 6

(3,7),(9,7),(4,7), -- 7

(7,8),(8,8), -- 8

(11,9), -- 9

(6,10),(1,10),(2,10), -- 10

(5,11),(5,11),(5,11), -- 11

(11,12),(1,12), -- 12

(2,13),(6,13), -- 13

(9,14), -- 14

(10,15),(8,15), -- 15

(4,16), -- 16

(1,17),(7,17),(5,17), -- 17

(6,18); -- 18

INSERT drink (Drink\_Name, Price, Size, Type)

VALUES

('Red dry wine', 300, 0.7, 'alcoholic'), -- 1

('White dry wine', 250, 0.5, 'alcoholic'), -- 2

('Apple juice', 100, 0.5, 'non-alcoholic'), -- 3

('Cognac', 250, 0.1, 'alcoholic'), -- 4

('Whiskey', 500, 0.25, 'alcoholic'), -- 5

('Water', 50, 0.5, 'non-alcoholic'), -- 6

('Grape juice', 100, 0.5, 'non-alcoholic'), -- 7

('Vodka', 250, 0.1, 'alcoholic'), -- 8

('Red sweet wine', 350, 0.5, 'alcoholic'), -- 9

('White sweet wine', 300, 0.5, 'alcoholic'), -- 10

('Rom', 350, 0.25, 'alcoholic'); -- 11

INSERT List\_of\_Drink (drink\_id, order\_id)

VALUES

(1,1),(4,1), -- 1

(9,2), -- 2

(4,3),(5,3),(8,3), -- 3

(6,4), -- 4

(2,5),(5,5), -- 5

(7,6),(9,6), -- 6

(3,7),(9,7),(4,7), -- 7

(8,8), -- 8

(11,9), -- 9

(6,10),(2,10), -- 10

(5,11),(5,11),(5,11), -- 11

(11,12), -- 12

(2,13),(6,13), -- 13

(9,14), -- 14

(10,15),(8,15), -- 15

(4,16), -- 16

(1,17),(7,17),(5,17), -- 17

(6,18); -- 18

Додаток В

1. Скільки грошей необхідно, щоб віддати зарплатню усім працівникам;

SELECT position, (SELECT COUNT(\*) FROM employee WHERE position\_id = position.id\_position) AS Number\_of\_employee,

salary, salary \* (SELECT COUNT(\*) FROM employee WHERE position\_id = position.id\_position) AS FullPrice

FROM position;

1. Повна інформація про працівника;

SELECT id\_employee, CONCAT(Last\_Name, ' ', First\_Name) AS Full\_Name,

(SELECT position FROM position WHERE position\_id = id\_position) AS Position,

(SELECT salary FROM position WHERE position\_id = id\_position) AS Salary

FROM employee

ORDER BY Salary

DESC;

1. Інформація про інгредієнт, а також про фірму-постачальника;

SELECT id\_ingredient, Ingredient\_Name,

(SELECT Firm FROM supplier WHERE supplier\_id = id\_supplier) AS Supplier\_Firm

FROM Ingredient

ORDER BY Supplier\_firm;

1. Інформація про інгредієнт, постачальника, а також про те, скільки було витрачено на них грошей;

SELECT id\_ingredient, Ingredient\_Name, Weight, Price\_for\_kg, Price\_for\_kg \* weight AS Costs\_for\_all,

(SELECT Firm FROM supplier WHERE supplier\_id = id\_supplier) AS Supplier\_Firm

FROM Ingredient;

1. Страва та інгредієнти для неї;

SELECT meal\_id, meal.Meal\_Name, ingredient.Ingredient\_Name FROM list\_of\_ingredients

INNER JOIN Meal ON list\_of\_ingredients.meal\_id = meal.id\_meal

INNER JOIN Ingredient ON list\_of\_ingredients.ingredient\_id = ingredient.id\_ingredient;

1. Список страв у замовленні;

SELECT order\_id, meal.Meal\_Name, meal.Price FROM list\_of\_meal

INNER JOIN Meal ON list\_of\_meal.meal\_id = meal.id\_meal

ORDER BY order\_id;

1. Список напоїв у замовленні;

SELECT order\_id, drink.Drink\_Name, drink.Type, drink.Price FROM list\_of\_drink

INNER JOIN Drink ON list\_of\_drink.drink\_id = drink.id\_drink

ORDER BY order\_id;

1. Топ страв, по к-сті замовлень;

SELECT meal\_id, meal.Meal\_Name, COUNT(\*) Number\_of\_Orders FROM list\_of\_meal

JOIN Meal ON list\_of\_meal.meal\_id = meal.id\_meal

GROUP BY meal\_id

ORDER BY Number\_of\_Orders

DESC;

1. Топ напоїв по к-сті замовлень;

SELECT drink\_id, drink.drink\_Name, COUNT(\*) Number\_of\_drinks FROM list\_of\_drink

JOIN drink ON list\_of\_drink.drink\_id = drink.id\_drink

GROUP BY drink\_id

ORDER BY Number\_of\_drinks

DESC;

1. Топ інгредієнтів по к-сті “використань”;

SELECT ingredient\_id, ingredient.Ingredient\_Name, COUNT(\*) Number\_of\_Use FROM list\_of\_ingredients

JOIN Ingredient ON list\_of\_ingredients.ingredient\_id = ingredient.id\_ingredient

GROUP BY ingredient\_id

ORDER BY Number\_of\_Use

DESC;

1. Топ фірм по к-сті інгредієнтів, що вони поставляють;

SELECT supplier\_id, supplier.Firm, COUNT(\*) Number\_of\_Use FROM Ingredient

JOIN Supplier ON Ingredient.supplier\_id = supplier.id\_supplier

GROUP BY supplier\_id

ORDER BY Number\_of\_Use

DESC;

1. Повна інформація про замовлення;

SELECT id\_order,

(SELECT CONCAT(Last\_Name, ' ', First\_Name) FROM employee WHERE employee\_id = id\_employee) AS Waiter,

(SELECT CONCAT(Last\_Name, ' ', First\_Name) FROM guest WHERE guest\_id = id\_guest) AS Guest,

Date\_of\_order,Time\_of\_order, Payment\_Method, Income

FROM restaurant.order;

1. Список тих, хто не може замовляти алкогольні напої;

SELECT id\_order, CONCAT(guest.Last\_Name, ' ', guest.First\_Name) AS Full\_Name, guest.age FROM restaurant.order

INNER JOIN Guest ON restaurant.order.guest\_id = guest.id\_guest

WHERE age < 18;

1. Список фірм постачальників, та для яких страв їх інгредієнти потрібні;

SELECT meal\_id, meal.Meal\_Name, ingredient.Ingredient\_Name, supplier.Firm FROM list\_of\_ingredients

INNER JOIN Meal ON list\_of\_ingredients.meal\_id = meal.id\_meal

INNER JOIN Ingredient ON list\_of\_ingredients.ingredient\_id = ingredient.id\_ingredient

INNER JOIN Supplier ON ingredient.supplier\_id = supplier.id\_supplier

ORDER BY meal\_id;

1. ID клієнтів, котрі замовили страву «Назва страви»;

SELECT order\_id, restaurant.order.guest\_id, meal.Meal\_Name FROM list\_of\_meal

INNER JOIN restaurant.order ON list\_of\_meal.order\_id = restaurant.order.id\_order

INNER JOIN Meal ON list\_of\_meal.meal\_id = meal.id\_meal

WHERE Meal\_Name = 'CONFIT DE CANARD'

ORDER BY order\_id;

1. ID клієнтів, котрі замовили напій «Назва напою»;

SELECT order\_id, restaurant.order.guest\_id, drink.drink\_Name FROM list\_of\_drink

INNER JOIN restaurant.order ON list\_of\_drink.order\_id = restaurant.order.id\_order

INNER JOIN drink ON list\_of\_drink.drink\_id = drink.id\_drink

WHERE drink\_Name = 'Vodka'

ORDER BY order\_id;

1. Інформація про офіціантів, та дні коли вони приймали замовлення;

SELECT id\_employee, CONCAT(Last\_Name, ' ', First\_Name) AS Full\_Name, restaurant.order.Date\_of\_Order FROM Employee

INNER JOIN restaurant.order ON restaurant.order.employee\_id = employee.id\_employee

ORDER BY id\_employee;

1. Інгредієнти для конкретної страви;

SELECT meal\_id, meal.Meal\_Name, ingredient.Ingredient\_Name FROM list\_of\_ingredients

INNER JOIN Meal ON list\_of\_ingredients.meal\_id = meal.id\_meal

INNER JOIN Ingredient ON list\_of\_ingredients.ingredient\_id = ingredient.id\_ingredient

WHERE Meal\_Name = 'FOIE GRAS';

1. Порахувати підвищення певному робітнику;

SELECT id\_employee, CONCAT(Last\_Name, ' ', First\_Name) AS Full\_Name,

(SELECT position FROM position WHERE position\_id = id\_position) AS Position,

(SELECT salary FROM position WHERE position\_id = id\_position) AS Salary,

(SELECT salary \* 1.15 FROM position WHERE position\_id = id\_position) AS Increased\_Salary

FROM employee

WHERE Last\_Name = 'Lavrov'

ORDER BY Salary

DESC;

1. Список інгредієнтів разом з фірмою-постачальником, котрий сортується за типом інгредієнта.

SELECT Firm, Type\_of\_Ingredient, Ingredient.Ingredient\_Name FROM supplier

INNER JOIN Ingredient ON supplier.id\_supplier = ingredient.supplier\_id

WHERE Type\_of\_Ingredient = 'Meat';