

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
(повна назва інституту/факультету)

КАФЕДРА інформатики та програмної інженерії
(повна назва кафедри)

КУРСОВА РОБОТА

з дисципліни «Бази даних»
(назва дисципліни)

на тему: База даних ресторану

Студента (ки) 2 курсу ІП-14 групи
спеціальності 121 «Інженерія програмного
забезпечення»
Шляхтуна Д. М.
(прізвище та ініціали)

Керівник Марченко О. І.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала _____

Кількість балів: _____ Оцінка ECTS _____

Члени комісії

_____	_____
(підпис)	(вчене звання, науковий ступінь, прізвище та ініціали)
_____	_____
(підпис)	(вчене звання, науковий ступінь, прізвище та ініціали)
_____	_____
(підпис)	(вчене звання, науковий ступінь, прізвище та ініціали)

Київ – 2022 рік

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет Інформатики та обчислювальної техніки
(повна назва)

Кафедра Інформатики та програмної інженерії
(повна назва)

Дисципліна Бази даних

Курс 2 Група ІІІ-14 Семестр 3

**З А В Д А Н Н Я
НА КУРСОВУ РОБОТУ СТУДЕНТУ**

Шляхтуну Денису Михайловичу

(прізвище, ім'я, по батькові)

1. Тема роботи База даних ресторану

керівник роботи Марченко Олена Іванівна

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

2. Строк подання студентом роботи 14.01.2023

3. Вихідні дані до роботи завдання на розробку бази даних для ресторану

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) _____

1) Аналіз предметного середовища

2) Побудова ER-моделі

3) Побудова реляційної схеми з ER-моделі

4) Створення бази даних, у форматі обраної системи управління базою даних

5) Створення користувачів бази даних

6) Імпорт даних з використанням засобів СУБД в створену базу даних

7) Створення мовою SQL запитів

8) Оптимізація роботи запитів

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Дата видачі завдання 08.11.2022

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання курсового проекту	Строк виконання етапів проекту	Примітка
1	Аналіз предметного середовища	12.11.2022	
2	Побудова ER-моделі	19.11.2022	
3	Побудова реляційної схеми з ER-моделі	26.11.2022	
4	Створення бази даних, у форматі обраної системи управління базою даних	03.12.2022	
5	Створення користувачів бази даних	10.12.2022	
6	Імпорт даних з використанням засобів СУБД в створену базу даних	12.12.2022	
7	Створення мовою SQL запитів	17.12.2022	
8	Оптимізація роботи запитів	24.12.2022	
9	Оформлення пояснювальної записки	28.12.2022	
10	Захист курсової роботи	14.01.2022	

Студент

_____ Шляхтун Д. М.
(підпис) (прізвище та ініціали)

Керівник роботи

_____ Марченко О. І.
(підпис) (прізвище та ініціали)

ЗМІСТ

ВСТУП	6
1 ТЕОРЕТИЧНА ЧАСТИНА	7
1.1 Аналіз предметного середовища	7
1.1.1 Опис предметного середовища	7
1.1.2 Опис вхідних та вихідних даних	9
1.1.3 Опис основних процесів використання даних	10
1.1.4 Постановка завдання	10
1.2 Проектування бази даних	10
1.2.1 Опис сутностей предметного середовища	10
1.2.1.1 Атрибути сутностей	11
1.2.1.2 Зв'язки між сутностями	13
1.2.2 Побудова ER-моделі	14
1.2.3 Даталогічна модель бази даних	15
2 ПРАКТИЧНА ЧАСТИНА	19
2.1 Реалізація бази даних	19
2.1.1 Створення таблиць	19
2.1.2 Імпорт даних у таблиці	21
2.1.3 Створення користувачів	22
2.1.4 Створення збережених процедур та функцій	23
2.1.5 Створення тригерів	25
2.1.6 Створення представлень	26
2.1.7 DML-запити типу SELECT	26
2.1.8 Оптимізація бази даних	33
ВИСНОВКИ	34

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	35
ДОДАТКИ.....	36
Додаток А. Використані запити.....	36
Додаток Б. Заповнення таблиць реляційної бази даних	56

ВСТУП

Будь-яка база даних є частиною інформаційної системи і призначена для збору, зберігання і маніпулювання інформацією. База даних – це організована структура, призначена для зберігання, зміни і обробки взаємозалежної інформації, переважно великих обсягів. Проектування бази даних – складний трудомісткий процес відображення предметного середовища.

Метою курсової роботи є створення реляційної бази даних ресторану. Основною ціллю бази даних є збереження та автоматизація замовлень ресторану, призначена для використання персоналом для внесення, перегляду замовлень.

Протягом виконання роботи будуть виконані наступні завдання:

- аналіз предметного середовища, опис об'єкту дослідження, вхідних даних, визначення основних вимог до бази даних;
- опис сутностей, їх атрибутів та зв'язків;
- формування моделі даних, що зберігатимуться у базі даних;
- побудова логічної моделі даних у вигляді таблиць із зазначенням усіх атрибутів та ключів;
- створення бази даних, таблиць, збережених процедур, функцій, тригерів, представлень, користувачів бази даних;
- заповнення бази даних значеннями;
- створення запитів для вибірки даних з бази даних;
- оптимізація роботи запитів.

В якості СУБД для виконання поставлених цілей використана Microsoft SQL Server. Основні завдання, що покладаються на СУБД:

- організація багатокористувацької моделі;
- маніпуляція інформацією та об'єктами бази даних.

1 ТЕОРЕТИЧНА ЧАСТИНА

1.1 Аналіз предметного середовища

1.1.1 Опис предметного середовища

Вимагається розробити інформаційну систему для автоматизації замовлень ресторану. Система повинна передбачувати режими ведення меню, рецептів страв та замовлень.

Страва з меню може мати свій рецепт, що складається з 1 або більше інгредієнтів. Кожна страва належить до певного типу, наприклад «Закуски», «Салати», «Супи», «Гарячі страви», «Десерти», «Напої». Кожна страва з меню характеризується наступними параметрами:

- назва страви;
- тип страви;
- вартість страви.

Страви складаються з окремих інгредієнтів. Інгредієнти мають різні одиниці вимірювання, наприклад шт., кг, л. Кожен інгредієнт характеризується наступними параметрами:

- назва інгредієнту;
- одиниця вимірювання;
- кількість інгредієнту;
- вартість інгредієнту.

Замовлення містить наступні поля:

- номер столика;
- офіціант;
- сума замовлення;
- час замовлення.

Замовлення може містити більше, ніж 1 страву, тому кожне замовлення складається із окремих рядків, що мають наступні параметри:

- страва;
- кількість;
- вартість кожної страви;
- відповідальна особа.

Загальна вартість замовлення рахується як сума вартостей з кожного рядка замовлення, взята з меню.

Сутність співробітників має своє місце як офіціант у замовленні та як відповідальна особа у окремих рядках замовлення. Кожен співробітник має наступні параметри:

- ім'я;
- прізвище;
- посада.

Перелік посад включає, але не обмежується списком: офіціант, кухар, шеф-кухар, бармен.

Передбачити наступні обмеження на інформацію в системі:

- інгредієнти, які внесені у систему і використовуються в інших сутностях, не можуть бути видаленими;
- посади, одиниці вимірювання чітко визначені і не можуть відрізнитися від певного словника.

З даною інформаційною системою повинні працювати наступні групи користувачів:

- офіціанти;
- виконувачі замовлень (кухари, бармени тощо);
- шеф-кухарі;
- адміністрація ресторану.

При роботі з системою офіціант повинен мати можливість:

- переглядати меню;
- вносити нові замовлення у систему.

Виконувачі замовлень повинні мати наступні можливості:

- переглядати склад страв;
- переглядати окремі рядки замовлень;
- зазначати відповідальних осіб за окремий пункт замовлення.

Шеф-кухар повинен мати наступні можливості:

- додавати нові інгредієнти, страви у меню та рецепти;
- редагувати рецепти.

Адміністрація повинна мати наступні можливості:

- встановлювати ціни на інгредієнти і страви;
- отримувати дані про загальний обіг коштів за замовлення;
- отримувати дані про продуктивність кожного члена персоналу;
- дізнаватися орієнтовний рівень витрат інгредієнтів за певний проміжок часу.

1.1.2 Опис вхідних та вихідних даних

Відповідно до опису предметного середовища, для роботи бази даних вимагається наявність наступної інформації:

- меню (перелік страв);
- склад страв;
- інгредієнти;
- замовлення;
- працівники;

Вихідними даними є повідомлення та результати, що видаються Системою відповідно до вимог предметного середовища.

1.1.3 Опис основних процесів використання даних

Серед процесів використання даних бази ресторану можна виділити наступні:

- перегляд меню;
- створення нових замовлень;
- перегляд склад страв;
- встановлення цін на інгредієнти і страви;
- отримання даних за обіг коштів;
- отримання рівня витрат інгредієнтів.

1.1.4 Постановка завдання

Завдання полягає у створення реляційної бази даних ресторану, що зберігатиме дані, виконуватиме функції і буде мати обмеження, що прописані в описі предметного середовища. Зокрема повинні бути виконані наступні дії:

- побудова ER-діаграми;
- побудова реляційної схеми;
- створення бази даних у форматі обраної СУБД;
- створення користувачів;
- імпорт даних у створену базу даних;
- написання запитів для потреб користувачів;
- оптимізація роботи запитів.

1.2 Проектування бази даних

1.2.1 Опис сутностей предметного середовища

Сутність – це клас однотипних об’єктів, інформація про які використовується в моделі. Перелік сутностей, що можна виділити відповідно до опису предметного середовища:

- меню;
- склад страви;
- група страви/напій;

- інгредієнт;
- одиниці вимірювання;
- замовлення;
- рядок замовлення;
- столик;
- співробітник;
- посада.

1.2.1.1 Атрибути сутностей

Атрибут сутності – це поіменована характеристика, яка є властивістю сутностей цієї множини. Надалі будуть продемонстровані атрибути до кожної сутності у вигляді таблиць. Атрибути формуються на основі вимог предметного середовища.

Таблиця 1.1

Меню
Унікальний ID
Назва
Тип
Вартість

Таблиця 1.2

Склад страви
Унікальний ID
Страва
Інгредієнт
Кількість інгредієнту

Таблиця 1.3

Група страви/напій

Тип страви/напою

Таблиця 1.4

Інгредієнт
Унікальний ID
Назва інгредієнту
Кількість
Одиниці вимірювання
Вартість

Таблиця 1.5

Одиниці вимірювання
Одиниця вимірювання

Таблиця 1.6

Замовлення
Унікальний ID
Офіціант
Номер столику
Дата замовлення
Час замовлення
Сума замовлення

Таблиця 1.7

Рядок замовлення
Унікальний ID

Замовлення
Страва/напій
Кількість
Вартість
Відповідальна особа

Таблиця 1.8

Столик
Унікальний ID
Номер столику

Таблиця 1.9

Співробітник
Унікальний ID
Ім'я
Прізвище
Посада

Таблиця 1.10

Посада
Назва посади

1.2.1.2 Зв'язки між сутностями

Зв'язок встановлюється між двома сутностями. Розрізняють 3 види зв'язків:

- один-до-одного - кожен момент часу одному екземпляру сутності А відповідає не більше одного примірника сутності В і навпаки.

- один-до-багатьох - одному екземпляру сутності А відповідає 0, 1 або більше примірників сутності В, але кожен екземпляр сутності В пов'язаний не більше ніж з 1 екземпляром сутності А.
- багато-до-багатьох - передбачає, що в кожен момент часу одному екземпляру сутності А відповідає 0, 1 або більше екземплярів сутності В і навпаки.

Таблиця 1.11 – Зв'язки між сутностями

№	Перша сутність	Друга сутність	Тип зв'язку
1	Меню	Склад страви	один до багатьох
2	Меню	Рядок замовлення	один до багатьох
3	Інгредієнт	Склад страви	один до багатьох
4	Одиниці вимірювання	Інгредієнт	один до багатьох
5	Група страви/напій	Меню	один до багатьох
6	Замовлення	Рядок замовлення	один до багатьох
7	Столик	Замовлення	один до багатьох
8	Співробітник	Замовлення	один до багатьох
9	Співробітник	Рядок замовлення	один до багатьох
10	Посада	Співробітник	один до багатьох

1.2.2 Побудова ER-моделі

На основі виявлених сутностей, їхніх атрибутів і зв'язків між ними побудована ER-модель:

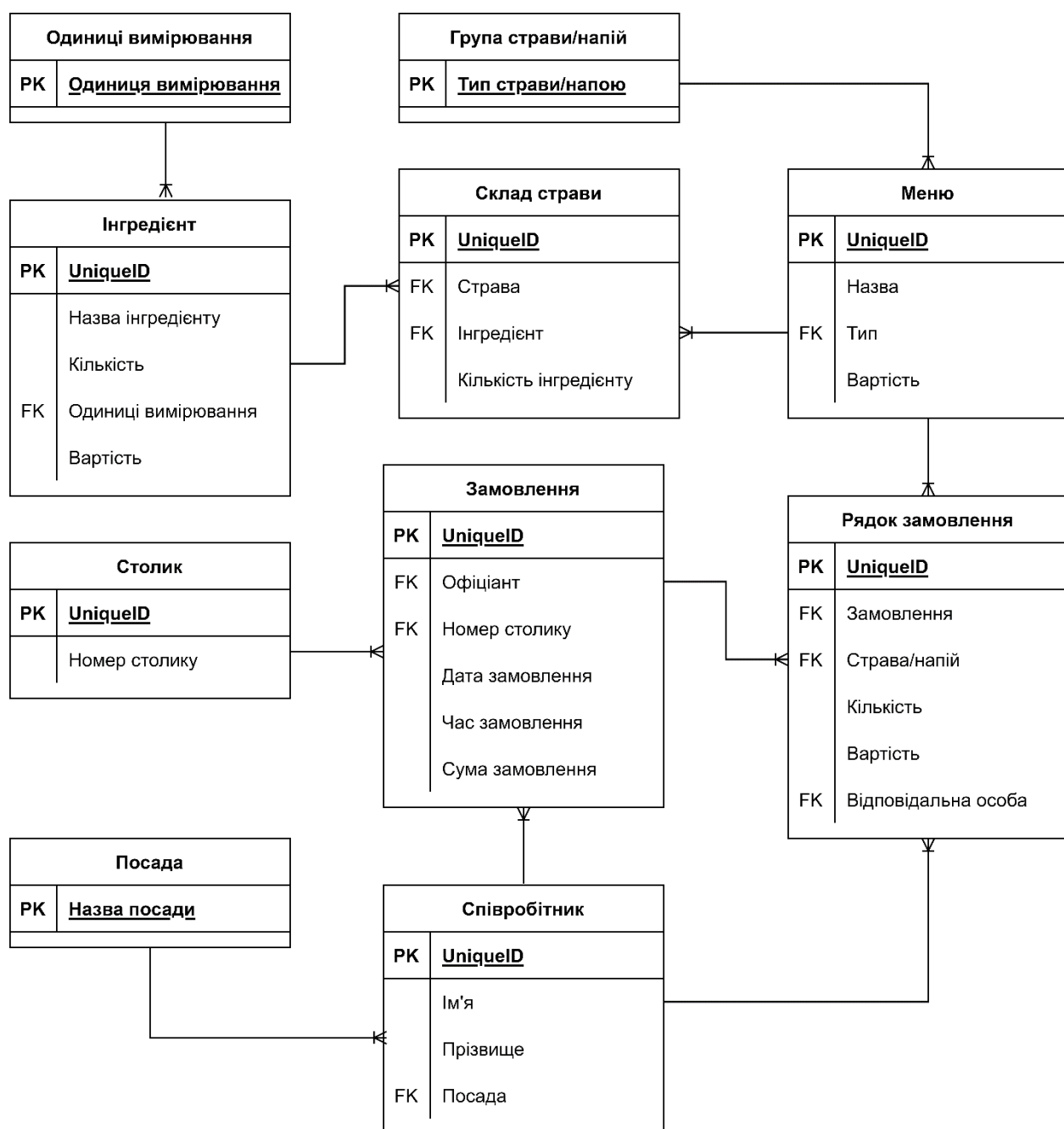


Рисунок 1.1 – ER-діаграма

1.2.3 Даталогічна модель бази даних

Даталогічне проектування – затвердження рішення про те, як логічна модель буде реалізована у базі даних, що створюється з використанням обраної СУБД.

Таблиця 1.12 – сутність меню

Назва поля	Тип поля	Розмір поля	Ключ
ID	INT	визначається БД	РК

Food	NVARCHAR	100	
Food_type	NVARCHAR	30	FK
Price	MONEY	визначається БД	

Таблиця 1.13 – сутність склад страви

Назва поля	Тип поля	Розмір поля	Ключ
ID	INT	визначається БД	PK
Food_ID	INT	визначається БД	FK
Ingredient_ID	INT	визначається БД	FK
Ingredient_amount	FLOAT	визначається БД	

Таблиця 1.14 – сутність група страви/напій

Назва поля	Тип поля	Розмір поля	Ключ
Type	NVARCHAR	30	PK

Таблиця 1.15 – сутність інгредієнт

Назва поля	Тип поля	Розмір поля	Ключ
ID	INT	визначається БД	PK
Ingredient	NVARCHAR	50	
Amount	FLOAT	визначається БД	
Unit_of_measurement	NVARCHAR	20	FK
Price	MONEY	визначається БД	

Таблиця 1.16 – сутність одиниці вимірювання

Назва поля	Тип поля	Розмір поля	Ключ
Unit_of_measurement	NVARCHAR	20	PK

Таблиця 1.17 – сутність замовлення

Назва поля	Тип поля	Розмір поля	Ключ
ID	INT	визначається БД	РК
Waiter_ID	INT	визначається БД	FK
Table_ID	INT	визначається БД	FK
Order_date	DATE	визначається БД	
Order_time	TIME	визначається БД	
Price	MONEY	визначається БД	

Таблиця 1.18 – сутність рядок замовлення

Назва поля	Тип поля	Розмір поля	Ключ
ID	INT	визначається БД	РК
Order_ID	INT	визначається БД	FK
Food_ID	INT	визначається БД	FK
Amount	INT	визначається БД	
Price	MONEY	визначається БД	
Worker_ID	INT	визначається БД	FK

Таблиця 1.19 – сутність столик

Назва поля	Тип поля	Розмір поля	Ключ
ID	INT	визначається БД	РК
Number	INT	визначається БД	

Таблиця 1.20 – сутність співробітник

Назва поля	Тип поля	Розмір поля	Ключ
ID	INT	визначається БД	РК
Name	NVARCHAR	30	

Surname	NVARCHAR	50	
Position	NVARCHAR	20	FK

Таблиця 1.21 – сутність посада

Назва поля	Тип поля	Розмір поля	Ключ
Position	NVARCHAR	20	PK

2 ПРАКТИЧНА ЧАСТИНА

2.1 Реалізація бази даних

Створення бази даних здійснювалося за допомогою СУБД Microsoft SQL Server 2019 і операції виконувалися через застосунок Microsoft SQL Server Management Studio (SSMS). Створення бази даних під назвою “Restaurant” було здійснено засобами цього застосунку.

2.1.1 Створення таблиць

В базі даних створюємо таблиці відповідно до даталогічної моделі бази даних. Перелік усіх таблиць:

- Food_Type
- Menu
- Unit_of_measurement
- Ingredient
- Recipe
- Position
- Worker
- Table_number
- New_order
- Order_row

Приклад створення таблиці засобами мови Transact SQL:

CREATE TABLE New_order

```
(
    ID INT IDENTITY (1, 1) PRIMARY KEY,
    Waiter_ID INT REFERENCES Worker(ID),
    Table_ID INT REFERENCES Table_number(ID),
    Order_date DATE DEFAULT GETDATE(),
    Order_time TIME DEFAULT CONVERT(TIME, GETDATE()),
    Price MONEY,
```

);

Серед ключових слів мови Transact SQL можна виділити:

- **CREATE TABLE** – створення нової таблиці, після ключових слів вказується назва таблиці і у дужках перелік полів.
- **INT, DATE, TIME, MONEY** – типи даних полів.
- **IDENTITY(1, 1)** – генератор послідовних значень чисел.
- **DEFAULT** – значення за замовченням, якщо не було вказане інакше при введенні.
- **PRIMARY KEY** – первинний ключ.
- **REFERENCES...** – зовнішній ключ.

Повні тексти команд представлені у Додатку А.

Засобами SSMS у розділі діаграм бази даних Restaurant створимо нову діаграму:

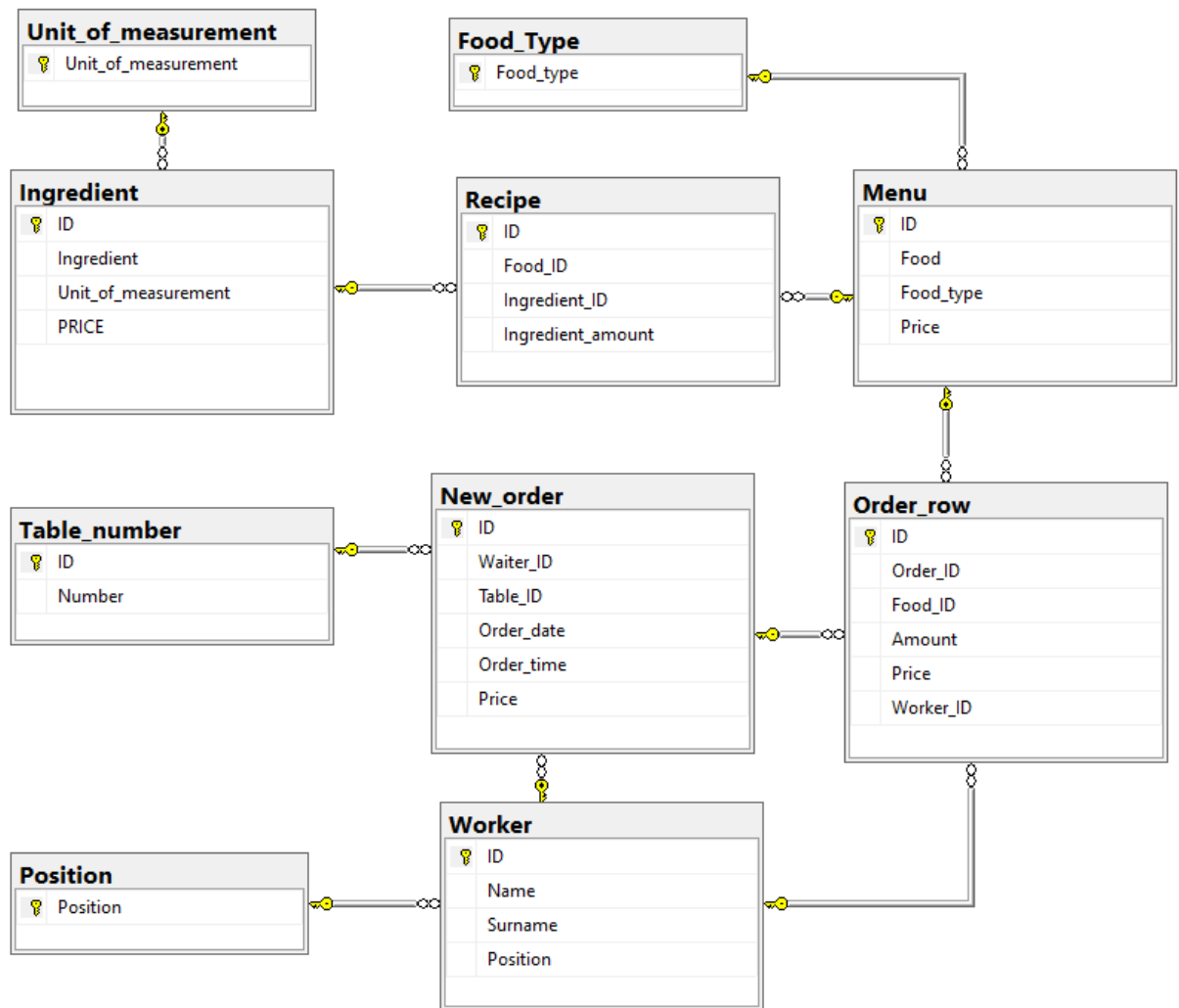


Рисунок 2.1 – діаграма бази даних Restaurant

2.1.2 Імпорт даних у таблиці

Імпорт даних було здійснено за допомогою SQL Server Import and Export Wizard, що викликається у середовищі SSMS при натисканні правою кнопкою на базу даних Restaurant, вибором пункту Tasks і Import data. Дані, якими заповнювалася база даних, представлені у додатку Б.

ID	Food	Food_type	Price
1	Тартар з яловичини	Закуска	200,00
2	Асорті сала	Закуска	300,00
3	Кобб з куркою	Салат	10,00
4	Мільфей з лососем	Салат	200,00
5	Грибна юшка	Суп	100,00
6	Рибний суп	Суп	200,00
7	Борщ з яловичиною	Суп	200,00
8	Томлена яловичина	М'ясо та птиця	200,00
9	Січений стейк	М'ясо та птиця	200,00
10	Куряче філе в карі соусі	М'ясо та птиця	300,00
11	Морозиво	Десерт	100,00
12	Мармуровий чізкейк	Десерт	200,00
13	Апельсиновий фреш	Безалкогольний напій	100,00
14	Чай смородиновий	Безалкогольний напій	200,00
15	Американо	Безалкогольний напій	200,00

Рисунок 2.2 – приклад заповнення таблиці Menu після імпортування

2.1.3 Створення користувачів

На основі вимог програмного середовища було створено наступні ролі із відповідними правами:

1. waiter
 - a. SELECT: Menu, New_order, Order_row, Order_info, Order_row_info (останні два є представленнями, що будуть розглянуті пізніше)
 - b. INSERT: New_order, Order_row
 - c. EXECUTE
2. worker
 - a. SELECT: New_order, Order_row, Menu, Recipe, Ingredient, Worker, Order_info, Order_row_info, CheckRecipe (функція)
 - b. UPDATE: Order_row
 - c. EXECUTE
3. chef
 - a. такі ж, що і у worker
 - b. UPDATE: Menu, Recipe, Ingredient
 - c. INSERT: Menu, Recipe, Ingredient

- d. DELETE: Recipe
- 4. administration
 - a. SELECT
 - b. UPDATE
 - c. INSERT
 - d. DELETE
 - e. EXECUTE
 - f. DENY DELETE ON Ingredient

Приклад створення ролі, логіна та юзера:

```
CREATE ROLE administration
GRANT SELECT, UPDATE, INSERT, DELETE, EXECUTE TO administration
DENY DELETE ON Ingredient TO administration
CREATE LOGIN admin_login WITH PASSWORD = '123', DEFAULT_DATABASE =
Restaurant
CREATE USER user_admin FOR LOGIN admin_login
ALTER ROLE administration ADD MEMBER user_admin
```

2.1.4 Створення збережених процедур та функцій

Для виконання дій із системою, передбачених програмним середовищем, були реалізовані функції та процедури – збережені послідовності запитів, які можуть приймати певні параметри.

Процедури:

- SelectMenu – процедура вибірки меню.
- NewOrder @waiter_id INT, @table_id INT – створення нового замовлення.
- AddToOrder @order_id INT, @food_id INT, @amount INT – додавання до існуючого замовлення страв.
- UpdateOrderRowWorker @row_id INT, @worker_id INT – встановлення відповідального працівника за рядок замовлення.

- UpdateIngredientPrice @ingredient_id INT, @price MONEY – оновлення ціни на вказаний інгредієнт.
- UpdateMenuPrice @food_id INT, @price MONEY – оновлення ціни на вказану страву з меню.
- GetIncomePerDay – вибірка доходу ресторану по датам.

Приклад створення процедури:

```
CREATE PROCEDURE NewOrder
```

```
@waiter_id INT, @table_id INT
```

```
AS
```

```
BEGIN
```

```
INSERT New_order(Waiter_ID, Table_ID) VALUES(@waiter_id, @table_id)
```

```
DECLARE @order_id INT
```

```
SET @order_id = (SELECT MAX(ID) FROM New_order)
```

```
DECLARE @waiter_surname NVARCHAR(50)
```

```
SET @waiter_surname = (SELECT Surname FROM Worker WHERE ID =
```

```
@waiter_id)
```

```
PRINT N'Замовлення №' + CONVERT(VARCHAR, @order_id)
```

```
+ N', офіціант ' + @waiter_surname
```

```
END
```

```
GO
```

Функції:

- CheckRecipe (@food_id INT) – повертає таблицю складу вказаної страви.
- GetIncomePerWorker () – повертає таблицю доходу на кожного працівника по датам.
- GetIngredientSpendPerDay (@date DATE) – повертає таблицю витрат інгредієнтів по вказаній даті.
- GetSpendPerIngredient (@ingredient_id INT) – повертає таблицю витрат вказаного інгредієнту по датам.

Приклад створення функції:

```
CREATE FUNCTION CheckRecipe (@food_id INT)
RETURNS TABLE AS
RETURN (SELECT Ingredient.Ingredient, Recipe.Ingredient_amount,
Ingredient.Unit_of_measurement FROM Ingredient JOIN Recipe ON
Recipe.Ingredient_ID = Ingredient.ID WHERE Recipe.Food_ID = @food_id)
GO
```

2.1.5 Створення тригерів

Для підтримання роботи бази даних були розроблені тригери.

Тригер Ingredient_price_update – таблиця Ingredient після UPDATE, виводить повідомлення про перевищення цін інгредієнтів по рецепту за ціни меню.

Аналогічно, тригер Menu_price_update – таблиця Menu після UPDATE, виводить повідомлення, якщо ціна страви в меню менша за ціну інгредієнтів з урахування складу страви.

Тригер Order_row_insert – таблиця Order_row після INSERT, оновлює ціну замовлення з урахуванням доданих рядків замовлення. Всередині тригеру використовується курсор для проходження кожного запису з таблиці inserted, тобто усіх значень, що вставили за допомогою оператора INSERT у таблицю Order_row.

Приклад створення тригеру з використанням курсору:

```
CREATE TRIGGER Order_row_insert ON Order_row
WITH EXECUTE AS OWNER
AFTER INSERT
AS
BEGIN
    DECLARE curs CURSOR FOR SELECT Order_ID FROM inserted FOR READ
    ONLY
    DECLARE @id INT
    OPEN curs
```

```

    FETCH NEXT FROM curs INTO @id
    UPDATE New_order SET Price = (SELECT Sum(Price*Amount) FROM
Order_row WHERE Order_ID = @id) WHERE ID = @id
    WHILE @@FETCH_STATUS = 0
    BEGIN
        FETCH NEXT FROM curs INTO @id
        UPDATE New_order SET Price = (SELECT Sum(Price*Amount) FROM
Order_row WHERE Order_ID = @id) WHERE ID = @id
    END
    CLOSE curs
    DEALLOCATE curs
END

```

2.1.6 Створення представлень

Для виведення загальної інформації про замовлення було створено 2 представлення.

Представлення Order_info поєднує інформацію з таблиць New_order, Worker, Table_number для заміщення зовнішніх ключів таблиці New_order на тексти з інших таблиць, наприклад прізвища офіціантів замість їх ID.

Аналогічно, представлення Order_row_info поєднує таблиці Order_row, Order_info, Menu, Worker для відображення у кожного рядка замовлення прізвища офіціантів та відповідальних осіб, назв страв.

2.1.7 DML-запити типу SELECT

При реалізації бази даних були спроектовані запити вибірки, які виконують наступні задачі:

- 1) вибірка повної інформації про окреме замовлення;

ID	Waiter	Table number	Order_date	Order_time	Price
1	Шевченко	2	2022-12-18	19:00:00.0000000	1400,00
2	Шевченко	7	2022-12-18	19:10:00.0000000	1200,00
3	Шевченко	5	2022-12-18	19:20:00.0000000	600,00
4	Нестайко	12	2022-12-18	19:30:00.0000000	400,00
5	Нестайко	7	2022-12-18	19:40:00.0000000	500,00
6	Нестайко	5	2022-12-18	19:50:00.0000000	400,00
7	Шевченко	3	2022-12-18	20:00:00.0000000	1800,00
8	Шевченко	12	2022-12-18	20:10:00.0000000	200,00
9	Нестайко	15	2022-12-18	20:20:00.0000000	1200,00
10	Шевченко	27	2022-12-18	20:30:00.0000000	700,00
11	Нестайко	5	2022-12-18	20:40:00.0000000	400,00
12	Нестайко	7	2022-12-18	20:50:00.0000000	900,00
13	Нестайко	5	2022-12-19	19:00:00.0000000	600,00
14	Шевченко	3	2022-12-19	19:10:00.0000000	1100,00

Рисунок 2.3

2) вибірка повної інформації про кожен рядок замовлення:

ID	Order ID	Waiter	Table number	Order_date	Order_time	Food	Food_type	Price	Amount	Worker
1	1	Шевченко	2	2022-12-18	19:00:00.0000000	Тартар з яловичини	Закуска	200,00	1	Іванов
2	1	Шевченко	2	2022-12-18	19:00:00.0000000	Кобб з куркою	Салат	100,00	2	Бандера
3	1	Шевченко	2	2022-12-18	19:00:00.0000000	Асорті сала	Закуска	300,00	3	Терентій
4	1	Шевченко	2	2022-12-18	19:00:00.0000000	Грибна юшка	Суп	100,00	1	Іванов
5	2	Шевченко	7	2022-12-18	19:10:00.0000000	Рибний суп	Суп	200,00	3	Іванов
6	2	Шевченко	7	2022-12-18	19:10:00.0000000	Мільфей з лососем	Салат	200,00	2	Іванов
7	2	Шевченко	7	2022-12-18	19:10:00.0000000	Томлена яловичина	М'ясо та птиця	200,00	1	Іванов
8	3	Шевченко	5	2022-12-18	19:20:00.0000000	Борщ з яловичиною	Суп	200,00	2	Бандера
9	3	Шевченко	5	2022-12-18	19:20:00.0000000	Січений стейк	М'ясо та птиця	200,00	1	Бандера
10	4	Нестайко	12	2022-12-18	19:30:00.0000000	Американо	Безалкогольний напій	200,00	2	Бандера
11	5	Нестайко	7	2022-12-18	19:40:00.0000000	Апельсиновий фреш	Безалкогольний напій	100,00	1	Терентій
12	5	Нестайко	7	2022-12-18	19:40:00.0000000	Чай смородиновий	Безалкогольний напій	200,00	1	Бандера
13	5	Нестайко	7	2022-12-18	19:40:00.0000000	Мармуровий чізкейк	Десерт	200,00	1	Терентій

Рисунок 2.4

3) вибірка рецепту до кожної страви:

Food	Ingredient	Ingredient_amount	Unit_of_measurement
Тартар з яловичини	Яловичина	0,35	кг
Тартар з яловичини	Яйце	1	шт.
Асорті сала	Сало	0,2	кг
Кобб з куркою	Філе курки	0,3	кг
Кобб з куркою	Яйце	1	шт.

Рисунок 2.5

4) вибірка доходу, який припадає на кожного працівника по датам:

Order_date	Surname	Name	Income	Position
2022-12-18	Бандера	Степан	3000,00	Працівник
2022-12-18	Іванов	Богдан	2400,00	Працівник
2022-12-18	Нестайко	Петро	3800,00	Офіціант
2022-12-18	Терентій	Всеволод	4300,00	Працівник
2022-12-18	Шевченко	Іван	5900,00	Офіціант
2022-12-19	Бандера	Степан	4300,00	Працівник
2022-12-19	Іванов	Богдан	4100,00	Працівник
2022-12-19	Нестайко	Петро	5700,00	Офіціант
2022-12-19	Терентій	Всеволод	3300,00	Працівник
2022-12-19	Шевченко	Іван	6000,00	Офіціант
2022-12-20	Бандера	Степан	1600,00	Працівник
2022-12-20	Іванов	Богдан	1600,00	Працівник
2022-12-20	Нестайко	Петро	4000,00	Офіціант
2022-12-20	Терентій	Всеволод	1900,00	Працівник
2022-12-20	Шевченко	Іван	1100,00	Офіціант
2023-01-13	Шевченко	Іван	7700,00	Офіціант

Рисунок 2.6

5) вибірка витрат кожного інгредієнту по датам:

Order_date	Ingredient	Amount	Unit_of_measurement
2022-12-18	Картопля	0,8	кг
2022-12-18	Сало	0,6	кг
2022-12-18	Філе курки	1,5	кг
2022-12-18	Яйце	8	шт.
2022-12-18	Яловичина	1,05	кг
2022-12-19	Картопля	1,8	кг
2022-12-19	Сало	1,6	кг
2022-12-20	Картопля	0,5	кг
2023-01-13	Картопля	0,9	кг
2023-01-13	Філе курки	20,4	кг
2023-01-13	Яйце	68	шт.

Рисунок 2.7

6) вибірка цін з меню і собівартості страв по їхньому складу:

Food	Menu price	Recipe price
Кобб з куркою	10,00	55
Грибна юшка	100,00	120
Борщ з яловичиною	200,00	120
Рибний суп	200,00	120
Тартар з яловичини	200,00	709,65
Асорті сала	300,00	40

Рисунок 2.8

7) вибірка, скільки яких страв приготував кожен працівник по датам:

Order_date	Surname	Food	Amount
2022-12-18	Бандера	Американо	1
2022-12-18	Бандера	Борщ з яловичиною	1
2022-12-18	Бандера	Кобб з куркою	1
2022-12-18	Бандера	Куряче філе в карі соусі	1
2022-12-18	Бандера	Січений стейк	2
2022-12-18	Бандера	Тартар з яловичини	1
2022-12-18	Бандера	Чай смородиновий	2
2022-12-18	Іванов	Американо	1
2022-12-18	Іванов	Грибна юшка	1
2022-12-18	Іванов	Мільфей з лососем	1
2022-12-18	Іванов	Морозиво	1
2022-12-18	Іванов	Рибний суп	1
2022-12-18	Іванов	Тартар з яловичини	1

Рисунок 2.9

8) вибірка кількості замовлень на кожен столик:

Number of table	Orders
5	9
7	7
12	6
3	5
2	2
8	2
15	2
27	1

Рисунок 2.10

9) вибірка кількості кожної страви, що було замовлено:

Food	General amount
Американо	11
Апельсиновий фреш	1
Асорті сала	11
Борщ з яловичиною	15
Грибна юшка	15
Кобб з куркою	73
Куряче філе в карі соусі	5
Мармуровий чізкейк	7
Мільфей з лососем	25

Рисунок 2.11

10) вибірка кількості страв по типам:

Food_type	Amount in menu
Безалкогольний напій	3
Десерт	2
Закуска	2
М'ясо та птиця	3
Салат	2
Суп	3

Рисунок 2.12

11) вибірка, скільки раз офіціант працював з яким столиком:

Surname	Number of table	Number of orders
Нестайко	5	6
Нестайко	12	4
Нестайко	7	4
Шевченко	3	4
Шевченко	7	3
Шевченко	5	3
Шевченко	12	2
Шевченко	8	2
Нестайко	15	1
Шевченко	15	1
Шевченко	27	1

Рисунок 2.13

12) вибірка кількості страв по датам замовлень:

Food	Order_date	Amount
Американо	2022-12-18	5
Апельсиновий фреш	2022-12-18	1
Асорті сала	2022-12-18	3
Борщ з яловичиною	2022-12-18	4
Грибна юшка	2022-12-18	1
Кобб з куркою	2022-12-18	5
Куряче філе в карі соусі	2022-12-18	3
Мармуровий чізкейк	2022-12-18	2
Мільфей з лососем	2022-12-18	8
Морозиво	2022-12-18	2
Рибний суп	2022-12-18	3

Рисунок 2.14

13) вибірка складності рецептів по кількості інгредієнтів для кожної страви:

Food	Food_type	Number of different ingredients
Американо	Безалкогольний напій	0
Апельсиновий фреш	Безалкогольний напій	0
Чай смородиновий	Безалкогольний напій	0
Мармуровий чізкейк	Десерт	0
Морозиво	Десерт	0
Асорті сала	Закуска	1
Тартар з яловичини	Закуска	2
Куряче філе в карі соусі	М'ясо та птиця	0
Січений стейк	М'ясо та птиця	0

Рисунок 2.15

14) вибірка, скільки раз кожна посада прийняла замовлень по датам:

Position	Order_date	Number of orders
Кухар	2022-12-18	20
Офіціант	2022-12-18	12
Шеф-кухар	2022-12-18	8
Кухар	2022-12-19	24
Офіціант	2022-12-19	12
Шеф-кухар	2022-12-19	11
Кухар	2022-12-20	12
Офіціант	2022-12-20	9
Шеф-кухар	2022-12-20	5
Офіціант	2023-01-13	1

Рисунок 2.16

15) вибірка офіціантів, що працювала в заданому діапазоні дат:

Surname	Order_date
Нестайко	2022-12-18
Нестайко	2022-12-19
Нестайко	2022-12-20
Шевченко	2022-12-18
Шевченко	2022-12-19
Шевченко	2022-12-20

Рисунок 2.17

16) вибірка кількості типів замовлених страв по датам:

Food_type	Order_date	Amount
Безалкогольний напій	2022-12-18	9
Десерт	2022-12-18	4
Закуска	2022-12-18	6
М'ясо та птиця	2022-12-18	10
Салат	2022-12-18	13
Суп	2022-12-18	8
Безалкогольний напій	2022-12-19	8
Десерт	2022-12-19	10
Закуска	2022-12-19	8
М'ясо та птиця	2022-12-19	6
Салат	2022-12-19	10
Суп	2022-12-19	18
Безалкогольний напій	2022-12-20	3

Рисунок 2.18

17) вибірка кількості страв, зроблених кожним працівником:

Surname	Food	Count
Бандера	Американо	3
Іванов	Американо	2
Терентій	Апельсиновий фреш	1
Іванов	Асорті сала	3
Терентій	Асорті сала	2
Бандера	Борщ з яловичиною	4
Терентій	Борщ з яловичиною	3
Бандера	Грибна юшка	2
Іванов	Грибна юшка	2
Терентій	Грибна юшка	1
Бандера	Кобб з куркою	1
Терентій	Кобб з куркою	1
Бандера	Куряче філе в карі соусі	1

Рисунок 2.19

18) вибірка кількості замовлень, що прийняв кожен офіціант по датам:

Surname	Order_date	Order count
Нестайко	2022-12-18	6
Шевченко	2022-12-18	6
Нестайко	2022-12-19	5
Шевченко	2022-12-19	7
Нестайко	2022-12-20	6
Шевченко	2022-12-20	3
Шевченко	2023-01-13	1

Рисунок 2.20

19) вибірка кількості страв на день на кожен столик:

Number of table	Order_date	Number of dishes per day
2	2022-12-18	7
3	2022-12-18	9
5	2022-12-18	7
7	2022-12-18	15
12	2022-12-18	3
15	2022-12-18	6
27	2022-12-18	3
2	2022-12-19	11
3	2022-12-19	8
5	2022-12-19	10
7	2022-12-19	17
8	2022-12-19	2

Рисунок 2.21

20) вибірка кількості страв, які приносить офіціант за замовлення:

ID	Surname	Order_date	Order_time	Number of dishes
1	Шевченко	2022-12-18	19:00:00.0000000	7
2	Шевченко	2022-12-18	19:10:00.0000000	6
3	Шевченко	2022-12-18	19:20:00.0000000	3
4	Нестайко	2022-12-18	19:30:00.0000000	2
5	Нестайко	2022-12-18	19:40:00.0000000	3
6	Нестайко	2022-12-18	19:50:00.0000000	2
7	Шевченко	2022-12-18	20:00:00.0000000	9
8	Шевченко	2022-12-18	20:10:00.0000000	1
9	Нестайко	2022-12-18	20:20:00.0000000	6
10	Шевченко	2022-12-18	20:30:00.0000000	3
11	Нестайко	2022-12-18	20:40:00.0000000	2
12	Нестайко	2022-12-18	20:50:00.0000000	6
13	Нестайко	2022-12-19	19:00:00.0000000	2

Рисунок 2.22

2.1.8 Оптимізація бази даних

Для оптимізації бази даних можна створити індекси для таблиць, у яких багато записів, на прикладі спроектованої бази даних – таблиці Order_row і New_order:

```
CREATE INDEX order_row_index ON Order_row(ID)
CREATE INDEX new_order_index ON New_order(ID)
```

ВИСНОВКИ

Метою курсової роботи була розробка бази даних ресторану. Для виконання роботи було проведено дослідження предметної області.

У предметному середовищі виділено основні сутності та їхні атрибути, описано їхні обмеження. Було побудовано ER-діаграму та даталогічну модель бази даних.

На основі побудованої моделі було реалізовано реляційну базу даних. Створено таблиці, які наповнювалися за допомогою інструментів імпорту даних. Визначено користувачів та їхні права, які дії їм дозволено виконувати з базою даних. Для виконання дій з базою даних були створені збережені процедури та функції. Були створені представлення для виведення узагальнених даних таблиць. Створені тригери на оновлення і вставлення даних.

В даній курсовій роботі була розроблена база даних ресторану в системі управління базами даних Microsoft SQL Server.

Створення бази даних для ресторану є актуальною та корисною темою для автоматизації дій та для звітування роботи ресторану.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Пасічник В.В., Резниченко В.А. Організація баз даних та знань - К: Видавнича група BHV, 2006. 384 с.
2. Г.А.Гайна. Основи проектування баз даних: Навчальний посібник. – К.; Кондор, 2008. 200 с.
3. Документація Microsoft SQL Server URL: <https://learn.microsoft.com/en-us/sql/>
4. ДСТУ 3008:2015. Звіти у сфері науки і техніки. Структура та правила оформлювання. Національний стандарт України – Київ: ДП «УкрНДНЦ», 2016. 15 с.
5. Лішук К. І. Конспекти і презентації лекцій по дисципліні «Бази даних», 2022

ДОДАТКИ

Додаток А. Використані запити

Створення таблиць

```
USE Restaurant;
```

```
--таблиця група страви/напій
```

```
CREATE TABLE Food_Type
```

```
(
```

```
    Food_type NVARCHAR(30) PRIMARY KEY,
```

```
);
```

```
--таблиця меню
```

```
CREATE TABLE Menu
```

```
(
```

```
    ID INT IDENTITY (1, 1) PRIMARY KEY,
```

```
    Food NVARCHAR(100) NOT NULL,
```

```
    Food_type NVARCHAR(30) REFERENCES Food_Type(Food_type),
```

```
    Price MONEY
```

```
);
```

```
--таблиця одиниці вимірювання
```

```
CREATE TABLE Unit_of_measurement
```

```
(
```

```
    Unit_of_measurement NVARCHAR(20) PRIMARY KEY
```

```
);
```

```
--таблиця інгредієнт
```

```
CREATE TABLE Ingredient
```

```
(
```

```

    ID INT IDENTITY (1, 1) PRIMARY KEY,
    Ingredient NVARCHAR(50) NOT NULL,
    Unit_of_measurement NVARCHAR(20) REFERENCES
Unit_of_measurement(Unit_of_measurement),
    PRICE Money
);

```

--таблиця склад страви

```

CREATE TABLE Recipe
(
    ID INT IDENTITY (1, 1) PRIMARY KEY,
    Food_ID INT REFERENCES Menu(ID) NOT NULL,
    Ingredient_ID INT REFERENCES Ingredient(ID) NOT NULL,
    Ingredient_amount FLOAT DEFAULT 0,
);

```

--таблиця посада

```

CREATE TABLE Position
(
    Position NVARCHAR(20) PRIMARY KEY
);

```

--таблиця співробітник

```

CREATE TABLE Worker
(
    ID INT IDENTITY (1, 1) PRIMARY KEY,
    Name NVARCHAR(30),
    Surname NVARCHAR(50) NOT NULL,
    Position NVARCHAR(20) REFERENCES Position(Position)
);

```

```
);
```

```
--таблиця столик
```

```
CREATE TABLE Table_number  
(  
    ID INT IDENTITY (1, 1) PRIMARY KEY,  
    Number INT NOT NULL,  
);
```

```
--таблиця замовлення
```

```
CREATE TABLE New_order  
(  
    ID INT IDENTITY (1, 1) PRIMARY KEY,  
    Waiter_ID INT REFERENCES Worker(ID),  
    Table_ID INT REFERENCES Table_number(ID),  
    Order_date DATE DEFAULT GETDATE(),  
    Order_time TIME DEFAULT CONVERT(TIME, GETDATE()),  
    Price MONEY,  
);
```

```
--таблиця рядок замовлення
```

```
CREATE TABLE Order_row  
(  
    ID INT IDENTITY (1, 1) PRIMARY KEY,  
    Order_ID INT REFERENCES New_order(ID),  
    Food_ID INT REFERENCES Menu(ID),  
    Amount INT DEFAULT 1,  
    Price MONEY,  
    Worker_ID INT REFERENCES Worker(ID)
```

```
);
```

Створення ролей і користувачів

```
USE Restaurant;
```

```
--офіціант
```

```
CREATE ROLE waiter
```

```
GRANT SELECT ON Menu TO waiter
```

```
GRANT SELECT ON New_order TO waiter
```

```
GRANT SELECT ON Order_row TO waiter
```

```
GRANT INSERT ON New_order TO waiter
```

```
GRANT INSERT ON Order_row TO waiter
```

```
GRANT EXECUTE TO waiter
```

```
GRANT SELECT ON ORDER_INFO TO waiter
```

```
GRANT SELECT ON ORDER_ROW_INFO TO waiter
```

```
CREATE LOGIN Shevchenko_login WITH PASSWORD = '123',
```

```
DEFAULT_DATABASE = Restaurant
```

```
CREATE USER Shevchenko FOR LOGIN Shevchenko_login
```

```
ALTER ROLE waiter ADD MEMBER Shevchenko
```

```
/*дії для офіціанта:
```

```
* EXEC SelectMenu
```

```
* EXEC NewOrder @waiter_id, @table_id
```

```
* EXEC AddToOrder @order_id, @food_id, @amount
```

```
* вибірка представлень New_order_info і Order_row_info
```

```
*/
```

```
--працівник
```

```

CREATE ROLE worker
GRANT SELECT ON New_order TO worker
GRANT SELECT ON Order_row TO worker
GRANT UPDATE ON Order_row TO worker
GRANT SELECT ON Menu TO worker
GRANT SELECT ON Recipe TO worker
GRANT SELECT ON Ingredient TO worker
GRANT SELECT ON Worker TO worker
GRANT EXECUTE TO worker
GRANT SELECT ON ORDER_INFO TO worker
GRANT SELECT ON ORDER_ROW_INFO TO worker
GRANT SELECT ON CheckRecipe TO worker

CREATE LOGIN Bandera_login WITH PASSWORD = '123', DEFAULT_DATABASE
= Restaurant
CREATE USER Bandera FOR LOGIN Bandera_login

ALTER ROLE worker ADD MEMBER Bandera

/*дії для працівника:
* EXEC SelectMenu
* SELECT * FROM CheckRecipe(@food_id)
* EXEC UpdateOrderRowWorker @row_id, @worder_id
* вибірка представлень New_order_info і Order_row_info
*/

--шеф-кухар
CREATE ROLE chef
ALTER ROLE worker ADD MEMBER chef

```



```
GRANT SELECT, UPDATE, INSERT ON Menu TO chef
GRANT SELECT, UPDATE, DELETE, INSERT ON Recipe TO chef
GRANT SELECT, UPDATE, INSERT ON Ingredient TO chef
```

```
CREATE LOGIN Ivanov_login WITH PASSWORD = '123', DEFAULT_DATABASE
= Restaurant
```

```
CREATE USER Ivanov FOR LOGIN Ivanov_login
```

```
ALTER ROLE chef ADD MEMBER Ivanov
```

```
/*дії для шеф-кухаря:
```

```
* такі ж, що для працівника
```

```
* вставлення даних у таблиці Menu, Recipe, Ingredient
```

```
* редагування цих даних
```

```
*/
```

```
CREATE ROLE administration
```

```
GRANT SELECT, UPDATE, INSERT, DELETE, EXECUTE TO administration
```

```
DENY DELETE ON Ingredient TO administration
```

```
CREATE LOGIN admin_login WITH PASSWORD = '123', DEFAULT_DATABASE =
Restaurant
```

```
CREATE USER user_admin FOR LOGIN admin_login
```

```
ALTER ROLE administration ADD MEMBER user_admin
```

```
/*дії адміністрації
```

```
* EXEC UpdateIngredientPrice @ingredient_id, @price
```

```
* EXEC UpdateMenuPrice @food_id, @price
```

```

* EXEC GetIncomePerDay
* SELECT * FROM GetIncomePerWorker()
* SELECT * FROM GetIngredientSpendPerDay(@date)
* SELECT * FROM GetSpendPerIngredient (2)
* вибірки різного роду
*/

```

Створення збережених процедур та функцій

```
USE Restaurant;
```

```
GO
```

```
-- перегляд меню
```

```
CREATE PROCEDURE SelectMenu
```

```
AS
```

```
SELECT * FROM Menu
```

```
GO
```

```
-- створення нових замовлень
```

```
CREATE PROCEDURE NewOrder
```

```
    @waiter_id INT, @table_id INT
```

```
AS
```

```
BEGIN
```

```
    INSERT New_order(Waiter_ID, Table_ID) VALUES(@waiter_id,
@table_id)
```

```
    DECLARE @order_id INT
```

```
    SET @order_id = (SELECT MAX(ID) FROM New_order)
```

```
    DECLARE @waiter_surname NVARCHAR(50)
```

```
    SET @waiter_surname = (SELECT Surname FROM Worker WHERE ID =
@waiter_id)
```

```
    PRINT N'Замовлення №' + CONVERT(VARCHAR, @order_id)
```

```
+ N', офіціант ' + @waiter_surname
```

```
END
```

```
GO
```

```
EXEC NewOrder 1, 1
```

```
GO
```

```
CREATE PROCEDURE AddToOrder
```

```
    @order_id INT, @food_id INT, @amount INT
```

```
AS
```

```
BEGIN
```

```
    INSERT Order_row(Order_ID, Food_ID, Amount, Price)
```

```
VALUES(@order_id, @food_id, @amount, (SELECT Price FROM Menu WHERE  
@food_id = Menu.ID))
```

```
    DECLARE @food NVARCHAR(100)
```

```
    SET @food = (SELECT Food FROM Menu WHERE ID = @food_id)
```

```
    PRINT N'До замовлення №' + CONVERT(VARCHAR, @order_id)
```

```
        + N' успішно додано "' + @food
```

```
        + N'" у кількості: ' + CONVERT(VARCHAR, @amount)
```

```
END
```

```
GO
```

```
EXEC AddToOrder 37, 3, 2
```

```
GO
```

```
-- перегляд склад страв
```

```
CREATE FUNCTION CheckRecipe (@food_id INT)
```

```
RETURNS TABLE AS
```

```
RETURN (SELECT Ingredient.Ingredient, Recipe.Ingredient_amount,
Ingredient.Unit_of_measurement FROM Ingredient JOIN Recipe ON
Recipe.Ingredient_ID = Ingredient.ID WHERE Recipe.Food_ID =
@food_id)
```

```
GO
```

```
SELECT * FROM CheckRecipe(3)
```

```
GO
```

```
-- встановлення відповідального за рядок замовлення
```

```
CREATE PROCEDURE UpdateOrderRowWorker
```

```
    @row_id INT, @worker_id INT
```

```
AS
```

```
UPDATE Order_row SET Worker_ID = @worker_id WHERE ID = @row_id
```

```
GO
```

```
-- встановлення цін на інгредієнти і страви
```

```
CREATE PROCEDURE UpdateIngredientPrice
```

```
    @ingredient_id INT, @price MONEY
```

```
AS
```

```
UPDATE Ingredient SET PRICE = @price WHERE ID = @ingredient_id
```

```
GO
```

```
EXEC UpdateIngredientPrice 3, 1200
```

```
GO
```

```
CREATE PROCEDURE UpdateMenuPrice
```

```
    @food_id INT, @price MONEY
```

```
AS
```

```
UPDATE Menu SET Price = @price WHERE ID = @food_id
GO
```

```
EXEC UpdateMenuPrice 3, 10
GO
```

```
-- отримання даних за обіг коштів
CREATE PROCEDURE GetIncomePerDay AS
BEGIN
    SELECT Order_date, SUM(Price) AS 'Income' FROM New_order
GROUP BY Order_date
END
GO
```

```
EXEC GetIncomePerDay
GO
```

```
--продуктивність кожного працівника
CREATE FUNCTION GetIncomePerWorker () RETURNS TABLE AS
RETURN
    SELECT New_order.Order_date, Worker.Surname, Worker.Name,
SUM(Order_row.Price) AS 'Income', N'Працівник' AS 'Position' FROM
Order_row JOIN New_order ON Order_row.Order_ID = New_order.ID JOIN
Worker ON Worker.ID = Order_row.Worker_ID GROUP BY
New_order.Order_date, Worker.Surname, Worker.Name
UNION
    SELECT New_order.Order_date, Worker.Surname, Worker.Name,
SUM(New_order.Price) AS 'Income', N'Офіціант' AS 'Position' FROM
```

```
New_order JOIN Worker ON Worker.ID = New_order.Waiter_ID GROUP BY
New_order.Order_date, Worker.Surname, Worker.Name
```

```
GO
```

```
SELECT * FROM GetIncomePerWorker()
```

```
GO
```

```
-- отримання рівня витрат інгредієнтів
```

```
CREATE FUNCTION GetIngredientSpendPerDay (@date DATE)
```

```
RETURNS TABLE
```

```
AS
```

```
RETURN
```

```
    (SELECT Ingredient.Ingredient,
        SUM(Order_Row.Amount * Recipe.Ingredient_amount) AS Amount,
        Ingredient.Unit_of_measurement
    FROM New_Order JOIN Order_Row ON New_Order.ID =
    Order_Row.Order_ID
        JOIN Menu ON Menu.ID = Order_Row.Food_ID
        JOIN Recipe ON Menu.ID = Recipe.Food_ID
        JOIN Ingredient ON Ingredient.ID = Recipe.Ingredient_ID WHERE
    New_order.Order_date = @date
        GROUP BY New_order.Order_date, Ingredient.Ingredient,
        Ingredient.Unit_of_measurement)
```

```
GO
```

```
SELECT * FROM GetIngredientSpendPerDay('2022-12-18')
```

```
GO
```

```
CREATE FUNCTION GetSpendPerIngredient (@ingredient_id INT)
```

RETURNS TABLE

AS

RETURN

```
(SELECT New_order.Order_date, Ingredient.Ingredient,
SUM(Order_Row.Amount * Recipe.Ingredient_amount) AS Amount,
Ingredient.Unit_of_measurement
FROM New_Order JOIN Order_Row ON New_Order.ID =
Order_Row.Order_ID
JOIN Menu ON Menu.ID = Order_Row.Food_ID
JOIN Recipe ON Menu.ID = Recipe.Food_ID
JOIN Ingredient ON Ingredient.ID = Recipe.Ingredient_ID WHERE
Ingredient.ID = @ingredient_id
GROUP BY New_order.Order_date, Ingredient.Ingredient,
Ingredient.Unit_of_measurement)
GO
```

```
SELECT * FROM GetSpendPerIngredient (1)
```

GO

Створення тригерів

```
USE Restaurant;
```

GO

```
CREATE TRIGGER Ingredient_price_update ON Ingredient
```

```
WITH EXECUTE AS OWNER
```

```
AFTER UPDATE
```

```
AS
```

```
BEGIN
```

```
DECLARE @temptable TABLE (foodPrice MONEY, ingredientPrice
Money)
```

```

INSERT INTO @temptable SELECT Menu.Price, SUM
(Ingredient.PRICE * Recipe.Ingredient_amount)
FROM Ingredient JOIN Recipe ON Ingredient_ID = Ingredient.ID
JOIN Menu ON Food_ID = Menu.ID GROUP BY Recipe.Food_ID, Menu.Price

```

```

DECLARE @count1 INT
SET @count1 = (SELECT COUNT(*) FROM @temptable WHERE
foodPrice < ingredientPrice)
IF (@count1 > 0)
BEGIN
    PRINT 'WARNING! The price of ingredients exceeds prices
of food in menu! ' + CONVERT(VARCHAR, @count1) + ' positions in
the menu need corrections'
END
END
GO

```

```

CREATE TRIGGER Menu_price_update ON Menu
WITH EXECUTE AS OWNER
AFTER UPDATE
AS
BEGIN
    DECLARE @temptable TABLE (foodPrice MONEY, ingredientPrice
Money)
    INSERT INTO @temptable SELECT Menu.Price, SUM
(Ingredient.PRICE * Recipe.Ingredient_amount)
FROM Ingredient JOIN Recipe ON Ingredient_ID = Ingredient.ID
JOIN Menu ON Food_ID = Menu.ID WHERE Food_ID IN (SELECT ID FROM
inserted) GROUP BY Recipe.Food_ID, Menu.Price

```



```

        IF ((SELECT COUNT(*) FROM @temptable WHERE foodPrice <
ingredientPrice) > 0)
            BEGIN
                PRINT 'WARNING! The price of ingredients exceeds the
price of food in menu!'
            END
        END
    END
GO

CREATE TRIGGER Order_row_insert ON Order_row
WITH EXECUTE AS OWNER
AFTER INSERT
AS
BEGIN
    DECLARE curs CURSOR FOR SELECT Order_ID FROM inserted FOR
READ ONLY
    DECLARE @id INT
    OPEN curs
    FETCH NEXT FROM curs INTO @id
    UPDATE New_order SET Price = (SELECT Sum(Price*Amount) FROM
Order_row WHERE Order_ID = @id) WHERE ID = @id
    WHILE @@FETCH_STATUS = 0
    BEGIN
        FETCH NEXT FROM curs INTO @id
        UPDATE New_order SET Price = (SELECT Sum(Price*Amount)
FROM Order_row WHERE Order_ID = @id) WHERE ID = @id
    END
    CLOSE curs
    DEALLOCATE curs

```

END

Створення представлень

USE Restaurant;

GO

CREATE VIEW Order_info AS

SELECT New_order.ID, Worker.Surname AS 'Waiter',
Table_number.Number AS 'Table number', New_order.Order_date,
New_order.Order_time, New_order.Price

FROM New_order JOIN Worker ON Waiter_ID = Worker.ID JOIN
Table_number ON Table_ID = Table_number.ID

GO

CREATE VIEW Order_row_info AS

SELECT Order_row.ID, Order_info.ID AS 'Order ID',
Order_info.Waiter,
Order_info.[Table number], Order_info.Order_date,
Order_info.Order_time,
Menu.Food, Menu.Food_type, Order_row.Price, Order_row.Amount,
Worker.Surname AS 'Worker'

FROM Order_row JOIN Order_info ON Order_ID = Order_info.ID
JOIN Menu ON Order_row.Food_ID = Menu.ID LEFT JOIN Worker ON
Order_row.Worker_ID=Worker.ID

GO

DML-запити типу SELECT

USE Restaurant;

--1) вибірка повної інформації про окреме замовлення

```
SELECT New_order.ID, Worker.Surname AS 'Waiter',
Table_number.Number AS 'Table number', New_order.Order_date,
New_order.Order_time, New_order.Price FROM New_order JOIN Worker
ON Waiter_ID = Worker.ID JOIN Table_number ON Table_ID =
Table_number.ID
```

--2) вибірка повної інформації про кожен рядок замовлення

```
SELECT Order_row.ID, Order_info.ID AS 'Order ID',
Order_info.Waiter, Order_info.[Table number],
Order_info.Order_date, Order_info.Order_time, Menu.Food,
Menu.Food_type, Order_row.Price, Order_row.Amount, Worker.Surname
AS 'Worker' FROM Order_row JOIN Order_info ON Order_ID =
Order_info.ID JOIN Menu ON Order_row.Food_ID = Menu.ID JOIN Worker
ON Order_row.Worker_ID=Worker.ID
```

--3) вибірка рецепту до кожної страви

```
SELECT Menu.Food, Ingredient.Ingredient, Recipe.Ingredient_amount,
Ingredient.Unit_of_measurement FROM Ingredient JOIN Recipe ON
Recipe.Ingredient_ID = Ingredient.ID JOIN Menu ON Food_ID =
MENU.ID ORDER BY Recipe.Food_ID
```

--4) вибірка доходу, який припадає на кожного працівника по датам

```
SELECT New_order.Order_date, Worker.Surname, Worker.Name,
SUM(Order_row.Price*Order_row.Amount) AS 'Income', N'Працівник' AS
'Position' FROM Order_row JOIN New_order ON Order_row.Order_ID =
New_order.ID JOIN Worker ON Worker.ID = Order_row.Worker_ID GROUP
BY New_order.Order_date, Worker.Surname, Worker.Name
UNION
```

```
SELECT New_order.Order_date, Worker.Surname, Worker.Name,
SUM(New_order.Price) AS 'Income', N'Офіціант' AS 'Position' FROM
New_order JOIN Worker ON Worker.ID = New_order.Waiter_ID GROUP BY
New_order.Order_date, Worker.Surname, Worker.Name
```

--5) вибірка витрат кожного інгредієнту по датам

```
SELECT New_order.Order_date, Ingredient.Ingredient,
SUM(Order_Row.Amount * Recipe.Ingredient_amount) AS Amount,
Ingredient.Unit_of_measurement FROM New_Order JOIN Order_Row ON
New_Order.ID = Order_Row.Order_ID JOIN Menu ON Menu.ID =
Order_Row.Food_ID JOIN Recipe ON Menu.ID = Recipe.Food_ID JOIN
Ingredient ON Ingredient.ID = Recipe.Ingredient_ID GROUP BY
New_order.Order_date, Ingredient.Ingredient,
Ingredient.Unit_of_measurement
```

--6) вибірка цін з меню і собівартості страв по їхньому складу

```
SELECT Menu.Food, Menu.Price AS 'Menu price', SUM
(Ingredient.PRICE * Recipe.Ingredient_amount) AS 'Recipe price'
FROM Ingredient JOIN Recipe ON Ingredient_ID = Ingredient.ID JOIN
Menu ON Food_ID = Menu.ID GROUP BY Menu.Food, Menu.Price
```

--7) вибірка, скільки яких страв приготував кожен працівник по датам

```
SELECT NEW_ORDER.Order_date, Worker.Surname, Menu.Food,
COUNT(Order_row.Food_ID) AS 'Amount' FROM Worker JOIN Order_row ON
Worker.ID = Worker_ID JOIN Menu ON Menu.ID = Food_ID JOIN
New_order ON Order_row.Order_ID=New_order.ID GROUP BY
NEW_ORDER.Order_date, Worker.Surname, Menu.Food
```

--8) вибірка кількості замовлень на кожен столик

```
SELECT Number AS 'Number of table', COUNT(New_order.ID) AS
'Orders' FROM Table_number JOIN New_order ON Table_ID =
Table_number.ID GROUP BY Number ORDER BY Orders DESC
```

--9) вибірка кількості кожної страви, що було замовлено

```
SELECT Food, SUM(Order_row.Amount) AS 'General amount' FROM Menu
JOIN Order_row ON Menu.ID = Food_ID GROUP BY Food
```

--10) вибірка кількості страв по типам

```
SELECT Food_Type.Food_type, COUNT(Menu.ID) AS 'Amount in menu'
FROM Food_Type JOIN Menu ON Food_Type.Food_type = Menu.Food_type
GROUP BY Food_Type.Food_type
```

--11) вибірка, скільки раз офіціант працював з яким столиком

```
SELECT Worker.Surname, Table_number.Number AS 'Number of table',
COUNT(New_order.Table_ID) AS 'Number of orders' FROM Worker JOIN
New_order ON Worker.ID = Waiter_ID JOIN Table_number ON
Table_number.ID = Table_ID GROUP BY Worker.Surname,
Table_number.Number ORDER BY [Number of orders] DESC
```

--12) вибірка кількості страв по датам замовлень

```
SELECT Menu.Food, New_order.Order_date, SUM(Order_row.Amount) AS
'Amount' FROM Menu JOIN Order_row ON Menu.ID = Food_ID JOIN
New_order ON Order_ID = New_order.ID GROUP BY Menu.Food,
New_order.Order_date
```

--13) вибірка складності рецептів по кількості інгредієнтів для кожної страви

```
SELECT Menu.Food, Menu.Food_type, COUNT(Recipe.ID) AS 'Number of
different ingredients' FROM Menu LEFT JOIN Recipe ON Menu.ID =
Recipe.Food_ID GROUP BY Menu.Food, Menu.Food_type
```

--14) вибірка, скільки раз кожна посада прийняла замовлень по датам

```
SELECT * FROM (
SELECT Position.Position, New_order.Order_date,
COUNT(New_order.ID) AS 'Number of orders' FROM Position JOIN
Worker ON Position.Position = Worker.Position JOIN New_order ON
Worker.ID = New_order.Waiter_ID GROUP BY Position.Position,
New_order.Order_date
UNION
SELECT Position.Position, New_order.Order_date,
COUNT(Order_row.ID) AS 'Number of orders' FROM Position JOIN
Worker ON Position.Position = Worker.Position JOIN Order_row ON
Worker.ID = Order_row.Worker_ID JOIN New_order ON
Order_row.Order_ID = New_order.ID WHERE Position.Position !=
N'Офіціант' GROUP BY Position.Position, New_order.Order_date
) AS table1 ORDER BY Order_date
```

--15) вибірка офіціантів, що працювали між 10 і 30 грудня 2022

```
SELECT DISTINCT Worker.Surname, New_order.Order_date FROM Worker
JOIN New_order ON Waiter_ID = Worker.ID WHERE Order_date BETWEEN
'12-10-2022' AND '12-30-2022'
```

--16) вибірка кількості типів замовлених страв по датам

```
SELECT Food_Type.Food_type, New_order.Order_date,
SUM(Order_row.Amount) AS 'Amount' FROM Food_Type JOIN Menu ON
```

```
Food_Type.Food_Type = Menu.Food_Type JOIN Order_row ON
Order_row.Food_ID = Menu.ID JOIN New_order ON New_order.ID =
Order_row.Order_ID GROUP BY Food_Type.Food_type,
New_order.Order_date
```

--17) вибірка кількості страв, зроблених кожним працівником

```
SELECT Worker.Surname, Menu.Food, COUNT(Order_row.ID) AS 'Count'
FROM Worker JOIN Order_row ON Worker.ID = Worker_ID JOIN Menu ON
Menu.ID = Food_ID GROUP BY Worker.Surname, Menu.Food
```

--18) вибірка кількості замовлень, що прийняв кожен офіціант по датам

```
SELECT Worker.Surname, New_order.Order_date, COUNT(New_order.ID)
AS 'Order count' FROM Worker JOIN New_order ON Worker.ID =
Waiter_ID GROUP BY Worker.Surname, New_order.Order_date
```

--19) вибірка кількості страв на день на кожен столик

```
SELECT Table_number.Number AS 'Number of table',
New_order.Order_date, SUM(Order_row.Amount) 'Number of dishes per
day' FROM Table_number JOIN New_order ON Table_number.ID =
New_order.Table_ID JOIN Order_row ON New_order.ID =
Order_row.Order_ID GROUP BY Table_number.Number,
New_order.Order_date
```

--20) вибірка кількості страв, які приносить офіціант за замовлення

```
SELECT New_order.ID, Worker.Surname, New_order.Order_date,
New_order.Order_time, SUM(Order_row.Amount) AS 'Number of dishes'
FROM Worker JOIN New_order ON Worker.ID = New_order.Waiter_ID JOIN
```

Order_row ON Order_row.Order_ID = New_order.ID GROUP BY
 New_order.ID, Worker.Surname, New_order.Order_date,
 New_order.Order_time

Додаток Б. Заповнення таблиць реляційної бази даних

Food_Type

Food_type
Закуска
Салат
Суп
М'ясо та птиця
Десерт
Безалкогольний напій

Menu

Food	Food_type	Price
Тартар з яловичини	Закуска	200
Асорті сала	Закуска	300
Кобб з куркою	Салат	100
Мільфей з лососем	Салат	200
Грибна юшка	Суп	100
Рибний суп	Суп	200
Борщ з яловичиною	Суп	200
Томлена яловичина	М'ясо та птиця	200
Січений стейк	М'ясо та птиця	200
Куряче філе в карі соусі	М'ясо та птиця	300
Морозиво	Десерт	100
Мармуровий чізкейк	Десерт	200
Апельсиновий фреш	Безалкогольний напій	100
Чай смородиновий	Безалкогольний напій	200

Американо	Безалкогольний напій	200
-----------	----------------------	-----

Unit_of_measurement

Unit_of_measurement
кг
л
шт.

Ingredient

Ingredient	Unit_of_measurement	PRICE
Сало	кг	200
Яловичина	кг	1999
Картопля	кг	10
Філе курки	кг	150
Лосось	кг	2000
Біле морозиво	кг	80
Апельсин	кг	90
Яйце	шт.	10
Гриби	кг	100

Recipe

Food_ID	Ingredient_ID	Ingredient_amount
2	1	0,2
1	2	0,35
1	8	1
3	4	0,3

3	8	1
5	3	0,1
6	3	0,1
7	3	0,1

Position

Position
Офіціант
Кухар
Шеф-кухар
Бармен

Worker

Name	Surname	Position
Іван	Шевченко	Офіціант
Петро	Нестайко	Офіціант
Богдан	Іванов	Шеф-кухар
Степан	Бандера	Кухар
Всеволод	Терентій	Кухар
Злотодан	Омельяненко	Бармен

Table_number

Number
2
3
5
6

7
8
12
13
15
27

New_order

Waiter_ID	Table_ID	Order_date	Order_time
1	1	18.12.2022	19:00
1	5	18.12.2022	19:10
1	3	18.12.2022	19:20
2	7	18.12.2022	19:30
2	5	18.12.2022	19:40
2	3	18.12.2022	19:50
1	2	18.12.2022	20:00
1	7	18.12.2022	20:10
2	9	18.12.2022	20:20
1	10	18.12.2022	20:30
2	3	18.12.2022	20:40
2	5	18.12.2022	20:50
2	3	19.12.2022	19:00
1	2	19.12.2022	19:10
1	6	19.12.2022	19:20
2	7	19.12.2022	19:30
1	2	19.12.2022	19:40
2	3	19.12.2022	19:50
2	5	19.12.2022	20:00

2	1	19.12.2022	20:10
1	5	19.12.2022	20:20
1	3	19.12.2022	20:30
1	7	19.12.2022	20:40
1	5	19.12.2022	20:50
2	3	20.12.2022	19:00
2	2	20.12.2022	19:10
2	7	20.12.2022	19:20
1	9	20.12.2022	19:30
1	6	20.12.2022	19:40
2	7	20.12.2022	19:50
1	2	20.12.2022	20:00
2	3	20.12.2022	20:10
2	5	20.12.2022	20:20

Order_row

Order_ID	Food_ID	Amount	Worker_ID
1	1	1	3
1	3	2	4
1	2	3	5
1	5	1	3
2	6	3	3
2	4	2	3
2	8	1	3
3	7	2	4
3	9	1	4
4	15	2	4
5	13	1	5

5	14	1	4
5	12	1	5
6	10	1	5
6	11	1	5
7	4	4	5
7	8	3	5
7	7	2	5
8	9	1	4
9	15	3	3
9	14	2	4
9	12	1	5
10	10	2	4
10	11	1	3
11	4	2	5
12	8	1	3
12	1	2	4
12	3	3	5
13	2	2	5
14	5	1	5
14	6	2	3
14	4	3	4
15	11	2	4
16	4	1	4
16	8	1	3
17	7	2	4
18	9	1	5
18	15	2	3
18	14	1	3
18	12	2	3

19	2	3	3
19	5	2	4
19	6	2	4
20	4	3	4
20	11	2	5
20	4	1	4
20	8	2	5
20	7	3	5
21	11	2	5
21	4	1	5
21	8	1	5
22	7	2	4
23	9	1	3
23	15	3	4
23	14	2	5
23	12	1	4
23	2	2	3
23	5	1	4
24	6	2	3
24	4	1	4
24	12	1	5
24	2	1	3
24	5	1	3
25	6	1	3
26	4	1	3
27	11	1	4
27	4	3	5
27	8	2	5
28	7	1	5

28	11	1	5
29	4	2	4
30	8	1	3
30	7	3	4
31	9	2	5
32	15	1	4
32	14	2	3
32	12	1	4
33	10	2	3
33	11	1	4
33	4	1	5