***Лабораторна робота 3:* Побудова логічної моделі бази даниx**

Група: АІ-243

Здобувач: Гаврилов О.В.

Варіант: 22

**Мета:**

Навчити студентів перетворювати концептуальну модель, створену за допомогою UML, у логічну модель бази даних, представивши її у вигляді ER-діаграми (Entity- Relationship Diagram).

**1) Завдання:**

1. Аналіз концептуальної моделі: переглянути діаграму класів UML, створену в лабораторній роботі 2.
2. Перетворення атрибутів: визначити типи даних атрибутів та їх обмеження.
3. Визначення первинних ключів: визначити первинні ключі для кожної таблиці.
4. Визначення типів зв’язків та зовнішніх ключів: визначити типи зв'язків між сутностями (кардинальність), формалізувати зв’язки за допомогою зовнішніх ключів.
5. Нормалізація схеми даних: привезти схему до нормальної форми Бойса-Кодда.
6. Побудова ER-діаграми: створити ER-діаграму на основі концептуальної моделі, використовуючи відповідну нотацію (Chen, Crow's Foot тощо).
7. Документування переходу: описати процес переходу від UML діаграми до ER- діаграми, включаючи будь-які зміни або спрощення.

2) ***Предметна область:***

*Необхідно розробити базу даних для співробітників та клієнтів ресторану, яка повинна містити наступну інформацію:*

*– персонал ресторану та їх графік роботи;*

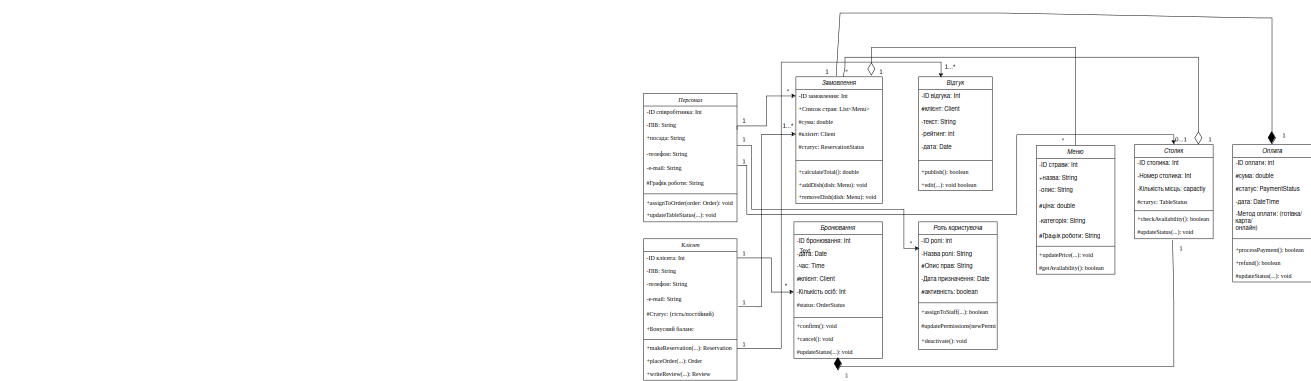
*– меню з зазначенням основних інгредієнтів, часу приготування та вартості;*

*– характеристика залів, столиків та їх закріплення за персоналом;*

*– карти постійних клієнтів, умови їх отримання та переваги.*

*Крім того, структура бази даних повинна надавати можливість зберігати іншу інформацію, що, на думку студента, відноситься до даної предметної області й завдань, розв'язуваним розроблюваною системою.*

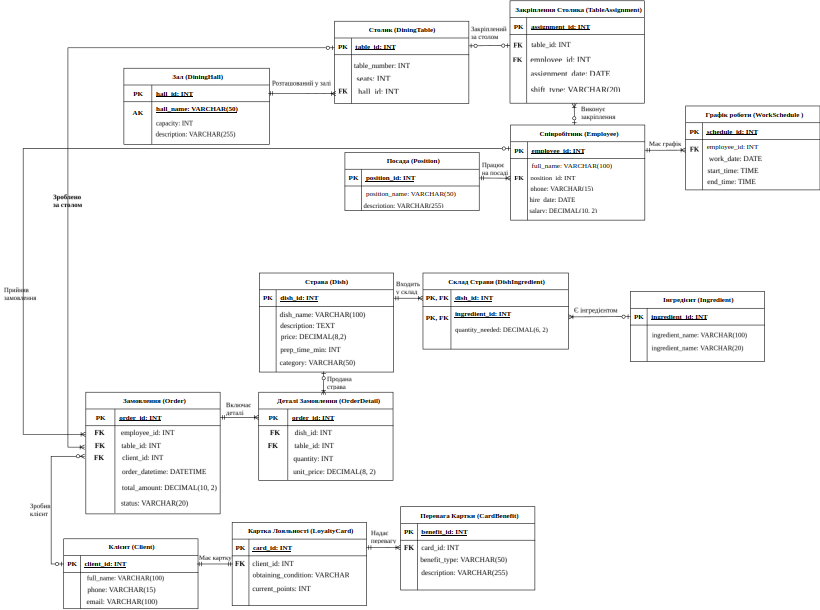
***3) Копія концептуальної моделі, яку створену в лабораторній роботі №2:***

**

***4) Перелік атрибутів з зазначенням типів даних та обмежень. Представити у вигляді таблиці:***

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Сутність | | | Властивість | | | Тип даних | | | | Ключ | | | Обмеження на  значення |
| Персонал | | | staff\_id | | | INT | | | | PK | | | PRIMARY KEY, AUTO\_INCREMENT, NOT NULL |
| full\_name | | | VARCHAR(100) | | | |  | | | NOT NULL |
| position | | | VARCHAR(50) | | | |  | | | NOT NULL, CHECK (position IN ('офіціант','кухар','адміністратор')) |
| phone | | | VARCHAR(15) | | | |  | | | UNIQUE, NOT NULL |
| hire\_date | | | DATE | | | |  | | | NOT NULL |
| salary | | | DECIMAL(10,2) | | | |  | | | NOT NULL, CHECK (salary > 0) |
| Графік роботи | | | schedule\_id | | | INT | | | | PK | | | AUTO\_INCREMENT, NOT NULL |
| staff\_id | | | INT | | | | FK | | | Staff(staff\_id), NOT NULL |
| work\_date | | | DATE | | | |  | | | NOT NULL |
| shift\_start | | | TIME | | | |  | | | NOT NULL |
| shift\_end | | | TIME | | | |  | | | NOT NULL |
| Меню | | | dish\_id | | | INT | | | | PK | | | AUTO\_INCREMENT, NOT NULL |
|  | | | dish\_name | | | VARCHAR(100) | | | |  | | | UNIQUE, NOT NULL |
|  | | | cook\_time | | | INT | | | |  | | | CHECK (cook\_time > 0) |
|  | | price | | | DECIMAL(8,2) | | | |  | | | CHECK (price > 0) | |
| Інгредієнти | | ingredient\_id | | | INT | | | | PK | | | AUTO\_INCREMENT, NOT NULL | |
|  | | ingredient\_name | | | VARCHAR(100) | |  | | | | | UNIQUE, NOT NULL | |
| Склад страви | | dish\_id | | | INT | | | | FK | | | Dish(dish\_id), NOT NULL | |
|  | | ingredient\_id | | | INT | | | | FK | | | Ingredient(ingredient\_id), NOT NULL | |
|  | | quantity | | | VARCHAR(50) | | | |  | | | NOT NULL | |
| Зали | | hall\_id | | | INT | | | | PK | | | AUTO\_INCREMENT, NOT NULL | |
|  | | hall\_name | | | VARCHAR(50) | | | |  | | | NOT NULL, UNIQUE | |
|  | | capacity | | | INT | | | |  | | | NOT NULL, CHECK (capacity > 0) | |
| Столики | | table\_id | | | INT | | | | PK | | | AUTO\_INCREMENT, NOT NULL | |
|  | | hall\_id | | | INT | | | | FK | | | Hall(hall\_id), NOT NULL | |
|  | | seats | | | INT | | | |  | | | NOT NULL, CHECK (seats > 0) | |
|  | | staff\_id | | | INT | | | | FK | | | Staff(staff\_id), NULL (стіл може бути не закріплений) | |
| Клієнти | | client\_id | | | INT | | | | PK | | | AUTO\_INCREMENT, NOT NULL | |
|  | | full\_name | | | VARCHAR(100) | | | |  | | | NOT NULL | |
|  | | phone | | | VARCHAR(15) | | | |  | | | UNIQUE | |
|  | | email | | | VARCHAR(100) | | | |  | | | UNIQUE | |
| Карти клієнтів | | card\_id | | | INT | | | | PK | | | AUTO\_INCREMENT, NOT NULL | |
|  | client\_id | | | INT | | | | FK | | | Клієнти(client\_id), NOT NULL | | |
|  | | issue\_date | | | DATE | | | |  | | | NOT NULL | |
|  | | conditions | | | TEXT | | | |  | | | NOT NULL | |
|  | | benefits | | | TEXT | | | |  | | | NOT NULL | |

***5) Перелік атрибутів з зазначенням типів даних та обмежень. Представити у вигляді таблиці:***



***6) Опис перевірки схеми на відповідність нормальній формі Бойса-Кодда.***

****Персонал (Staff)****

Знаходиться в 1НФ: усі атрибути атомарні.

Знаходиться в 2НФ: немає складеного ключа → автоматично виконується.

Знаходиться в 3НФ: немає транзитивних залежностей між неключовими атрибутами.

Знаходиться в НФБК: ключ staff\_id є простим, тому умова виконується автоматично.

****Графік роботи (Schedule)****

1НФ: атомарність дотримана.

2НФ: неключові атрибути (work\_date, shift\_start, shift\_end) залежать від складеного ключа (schedule\_id, staff\_id) повністю.

3НФ: немає транзитивних залежностей.

НФБК: умова виконується, оскільки жоден неключовий атрибут не визначає частину ключа.

****Зали (Hall)****

1НФ: атомарність є.

2НФ: ключ простий → автоматично.

3НФ: залежність hall\_name → capacity не порушує 3НФ, оскільки hall\_name є ключем.

НФБК: виконується.

****Столики (Table)****

1НФ: дотримана.

2НФ: неключові атрибути залежать від усього ключа table\_id.

3НФ: транзитивних залежностей немає.

НФБК: виконується.

****Меню (Dish)****

1НФ–НФБК: простий ключ dish\_id, атрибути залежать від нього повністю.

****Клієнти (Client)****

1НФ–НФБК: простий ключ client\_id, аномалій немає.

****Карти клієнтів (LoyaltyCard)****

Увага: помилка в атрибутах — client\_id повторюється. Потрібно виправити на:

card\_id: INT (PK)

client\_id: INT (FK)

Після виправлення: умови НФБК виконуються.

****Склад страви (DishIngredient)****

1НФ: атомарність є.

2НФ: неключові атрибути залежать від повного складеного ключа (dish\_id, ingredient\_id).

3НФ: транзитивних залежностей немає.

НФБК: виконується.

****Висновок:**** Усі відношення (після виправлення LoyaltyCard) відповідають нормальній формі Бойса-Кодда.

***7)Опис процесу переходу від UML-діаграми до ER-діаграми, включаючи будь-які зміни або спрощення.***

#### ****Основні перетворення для ресторану:****

1. ****Класи → Таблиці:****
   * Client → Клієнти
   * Staff → Персонал
   * Dish → Страви
   * Hall → Зали
   * Order → Замовлення
2. ****Спрощення успадкування:****
   * Концептуальна модель мала клас User з підкласами Client та Staff
   * У логічній моделі створено окремі таблиці Клієнти та Персонал без спільного батьківського класу
   * Це спрощення зроблено для практичності реалізації
3. ****Трансформація зв'язків:****

****«1 до N»:****

* Зал (1) — Столик (N) → стовпець hall\_id у таблиці Столики
* Клієнт (1) — Замовлення (N) → стовпець client\_id у таблиці Замовлення

****«N до M» через проміжні таблиці:****

* Замовлення (N) — Страва (M) → створено таблицю Склад\_замовлення
* Страва (N) — Інгредієнт (M) → створено таблицю Склад\_страви

#### ****Спрощення для ресторанної сфери:****

1. ****Об'єднання пов'язаних понять:****
   * Графік роботи персоналу об'єднано в єдину таблицю Графік\_роботи
   * Програму лояльності спрощено до таблиці Карти\_клієнтів
2. ****Видалення надмірної деталізації:****
   * Складні ієрархії меню (напої, основні страви, десерти) об'єднано в єдину таблицю Страви
   * Типи столів (віп, звичайні) не виокремлено окремими сутностями
3. ****Оптимізація для бізнес-процесів:****
   * Таблиця Замовлення містить лише необхідні для обслуговування поля
   * У таблиці Страви додано поле cook\_time для планування роботи кухні

#### ****Результат для ресторану:****

Отримана логічна модель оптимально підходить для автоматизації роботи ресторану:

* Відстеження замовлень клієнтів
* Управління робочим графіком персоналу
* Контроль складу та інгредієнтів
* Керування схемою залу та столиків

Модель готова для фізичної реалізації в СУБД та розробки програмного забезпечення для ресторану.

***Висновок:***

*У ході виконання роботи було проаналізовано предметну область, побудовано ER-діаграму сутностей та зв’язків, визначено атрибути з типами даних і ключами. Схему перевірено на відповідність нормальним формам аж до форми Бойса–Кодда (BCNF). У результаті проєктування отримано логічну структуру бази даних, яка забезпечує відсутність надлишковості, аномалій оновлення та підтримує цілісність і узгодженість даних.*