***Лабораторна робота 5: Мова маніпулювання даними:***

***прості запити, оператори порівняння, булеві та спеціальні оператори***

Група: АІ-243

Здобувач: Гаврилов О.В.

Варіант: 22

**Мета:**

Навчити студентів виконувати прості запити до таблиць бази даних, а також виконувати вибірку з застосуванням операторів порівняння, булевих та спеціальних операторів.

**Завдання:**

1. Реалізація операції реляційної алгебри «Проекція»:

Виконати прості запити до таблиць бази даних, які повертають всі значення певних стовпців (на вибір студента).

1. Реалізація операції реляційної алгебри «Вибірка»:

Виконати запити, які повертають рядки певної(их) таблиці(ь) згідно вказаних умов з використанням операторів порівняння, булевих та спеціальних операторів.

1. Реалізація операцій зміни схеми відношення:

Виконати запити на додавання стовпця до схеми відношення; видалення стовпця зі схеми відношення; переіменування стовпця будь-якого відношення; додавання обмеження на значення; зміна значення за замовчуванням; видалення обмеження зовнішнього ключа.

**Результат:**

Студенти повинні подати SQL-скрипти, що відображають запити на вибірку з таблиць та запити на модифікацію схеми відношення згідно завдання та предметної області, їх опис, а також звіт з результатами тестування.

***1) Простий запит до таблиці бази даних, який повертає всі значення певних стовпців (на вибір студента):***

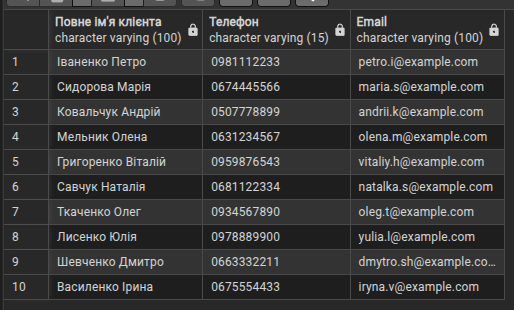
***1. Словесна постановка задачі:***

*Отримати повне ім'я, номер телефону та електронну пошту всіх клієнтів, зареєстрованих у системі.*

*2.* SQL-код рішення:

|  |
| --- |
| SELECT  full\_name AS "Повне ім'я клієнта",  phone AS "Телефон",  email AS "Email"  FROM  Клієнт; |

**3. Скриншот отриманого результату:**



***2) П’ять запитів, які повертають рядки певної(их) таблиці(ь) згідно вказаних умов:***

* ***значення атрибуту цілочисельного типу даних більше заданого цілого та значенняатрибуту символьного типу недорівнює заданому значенню;***
* ***значення атрибуту типу дата менше ніж системна дата та значення атрибуту***

***символьного типу містить порожні значення;***

* ***значення атрибуту символьного типу дорівнює заданому значенню або цей же***

***атрибут дорівнює іншому заданому значенню;***

* ***значення атрибуту типу дата не є системною датою (NOT) та значення атрибуту***

***символьного типу відповідає одному з перелічених в умові;***

* ***значення атрибуту цілочисельного типу входить в заданий діапазон значень та***
* ***значення атрибуту символьного типу відповідає заданому шаблону;***

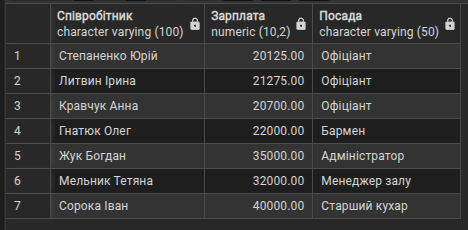
***1. Запит: Числове > та Символьне !=*** Словесна постановка задачі:

Знайти імена та зарплати всіх співробітників, чия заробітна плата (salary) більша за 20000.00 грн та які не працюють на посаді 'Кухар'.

**SQL-код рішення:**

|  |
| --- |
| SELECT  S.full\_name AS "Співробітник",  S.salary AS "Зарплата",  P.position\_name AS "Посада"  FROM  Співробітник AS S  JOIN  Посада AS P ON S.position\_id = P.position\_id  WHERE  S.salary > 20000.00 -- Числове більше заданого  AND P.position\_name <> ***'Кухар'***; -- Символьне недорівнює заданому |

***Скриншот отриманого результату:***

2. Запит: Дата < Системна Дата та Символьне IS NULL

Словесна постановка задачі:

Знайти інформацію про клієнтів, які були найняті до поточної дати (завжди істинно, але формально), та не вказали свою електронну пошту (email).

**SQL-код рішення:**

|  |
| --- |
| SELECT  full\_name AS "Ім'я Клієнта",  phone AS "Телефон",  email AS "Email (порожнє)"  FROM  Клієнт  WHERE  -- Припускаємо, що дата реєстрації менша за поточну системну дату (NOW() або CURRENT\_DATE)  -- У нашому випадку, ми використовуємо атрибут, який може бути NULL:  email IS NULL; -- Символьне містить порожні значення |

***Скриншот отриманого результату:***

**

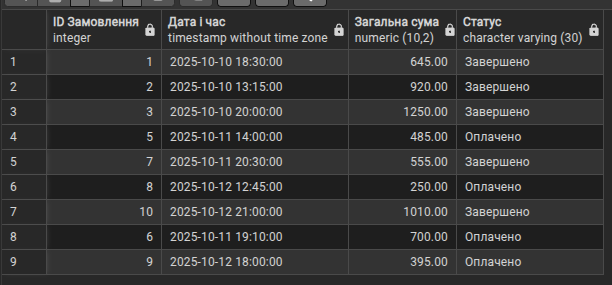
***3. Запит: Символьне = або Символьне =* Словесна постановка задачі:**

Знайти всі замовлення, які мають статус **'Завершено'** **або** **'Оплачено'**, та показати їхню загальну суму.

**SQL-код рішення:**

|  |
| --- |
| SELECT  order\_id AS "ID Замовлення",  order\_datetime AS "Дата і час",  total\_amount AS "Загальна сума",  status AS "Статус"  FROM  Замовлення  WHERE  status = ***'Завершено'*** -- Символьне дорівнює заданому  OR status = ***'Оплачено'***; -- Або дорівнює іншому заданому |

***Скриншот отриманого результату:***



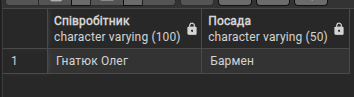
***4. Запит: Дата NOT Системна Дата та Символьне IN* Словесна постановка задачі:**

Знайти повні імена співробітників, які працювали у графіку, що не відповідає поточній системній даті (тобто не сьогодні), та чия посада є однією з ('Бармен', 'Сомельє').

**SQL-код рішення:**

|  |
| --- |
| SELECT  S.full\_name AS "Співробітник",  P.position\_name AS "Посада"  FROM  Співробітник AS S  JOIN  Посада AS P ON S.position\_id = P.position\_id  WHERE  S.hired\_date <> CURRENT\_DATE -- Дата не є системною датою (використовуємо дату найму)  AND P.position\_name IN (***'Бармен'***, ***'Сомельє'***); -- Символьне відповідає одному з перелічених |

***Скриншот отриманого результату:***



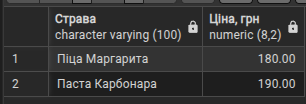
***5. Запит: Числове BETWEEN та Символьне LIKE* Словесна постановка задачі:**

Знайти страви, чия ціна (price) знаходиться в діапазоні від 100.00 до 300.00 грн та чия назва (dish\_name) починається з літери 'П' (або 'П' у будь-якій позиції, для використання %).

**SQL-код рішення:**

|  |
| --- |
| **SELECT**  dish\_name **AS** "Страва",  price **AS** "Ціна, грн"  **FROM**  Страва  **WHERE**  price **BETWEEN** 100.00 AND 300.00 -- Числове входить в заданий діапазон  AND dish\_name LIKE 'П%'; -- Символьне відповідає заданому шаблону (починається з 'П') |

***Скриншот отриманого результату:***



***3. Шість запитів, які реалізують:***

* ***додавання стовпця типу «час» до схеми відношення;***
* ***видалення стовпця символьного типу зі схеми відношення;***
* ***переіменування стовпця будь-якого відношення;***
* ***додавання обмеження на значення – значення числового стовпця більше 0;***
* ***зміна значення за замовчуванням;***
* ***видалення обмеження зовнішнього ключа.***

**1) Додавання стовпця типу «час»:**

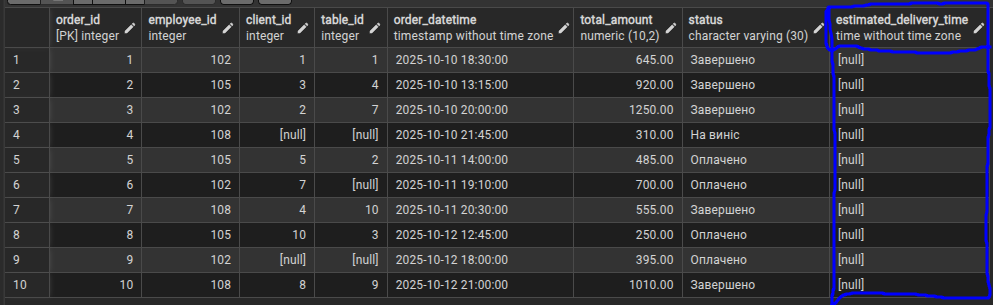
**Словесна постановка задачі:**

Додати до таблиці **Замовлення** новий стовпець **estimated\_delivery\_time** (очікуваний час доставки/видачі), який матиме тип даних **TIME**.

**SQL-код рішення:**

|  |
| --- |
| **ALTER** **TABLE** Замовлення  **ADD** **COLUMN** estimated\_delivery\_time TIME; |

***Скриншот отриманого результату:***



**2) Видалення стовпця символьного типу:**

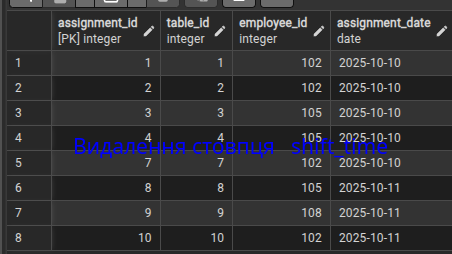
**Словесна постановка задачі:**

Видалити стовпець shift\_time (зміна), який має символьний тип даних, з таблиці Закріплення, оскільки цю інформацію можна отримати з графіка роботи співробітника.

**SQL-код рішення:**

|  |
| --- |
| **ALTER** **TABLE** Закріплення  **DROP** **COLUMN** shift\_time; |

***Скриншот отриманого результату:***



**3) Перейменування стовпця будь-якого відношення:**

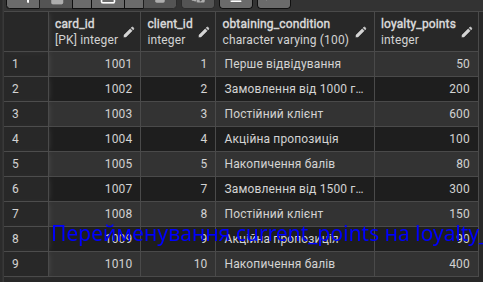
**Словесна постановка задачі:**

Перейменувати стовпець current\_points (поточні бали) в таблиці Картка\_лояльності на loyalty\_points для кращої уніфікації назв.

**SQL-код рішення:**

|  |
| --- |
| **ALTER** **TABLE** Картка\_лояльності  **RENAME** **COLUMN** current\_points **TO** loyalty\_points; |

***Скриншот отриманого результату:***



**4) Додавання обмеження на значення (Числове > 0):**

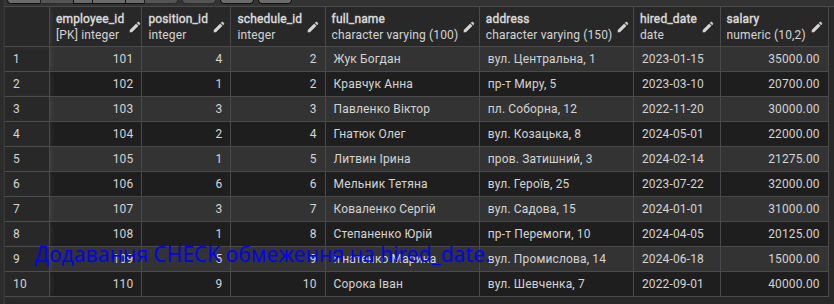
**Словесна постановка задачі:**

Додати обмеження CHECK до таблиці Співробітник, щоб гарантувати, що дата найму (hired\_date) не може бути в майбутньому (тобто вона має бути меншою або рівною поточній системній даті). *(Використовуємо дату, оскільки числовий стовпець salary > 0 вже має обмеження в схемі. Дата — це також обмеження на значення атрибута).*

**SQL-код рішення:**

|  |
| --- |
| **ALTER** **TABLE** Співробітник  **ADD** **CONSTRAINT** chk\_hired\_date\_not\_future  **CHECK** (hired\_date <= CURRENT\_DATE); |

***Скриншот отриманого результату:***

****

**5) Зміна значення за замовчуванням:**

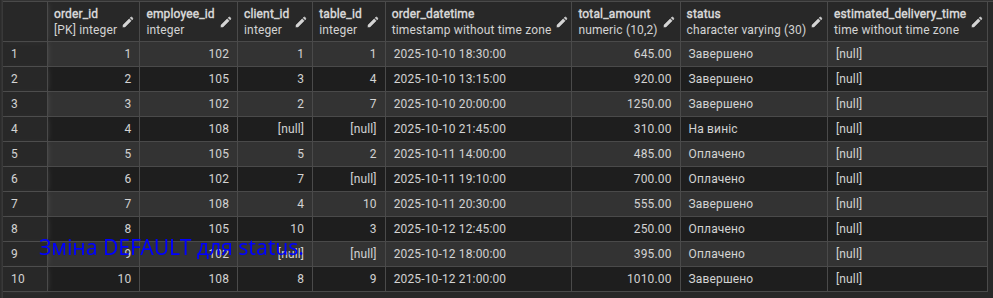
**Словесна постановка задачі:**

Змінити значення за замовчуванням для стовпця status (статус) у таблиці Замовлення з NULL на 'Нове', щоб при створенні нового запису йому автоматично присвоювався початковий статус.

**SQL-код рішення:**

|  |
| --- |
| **ALTER** **TABLE** Замовлення  **ALTER** **COLUMN** **status** **SET** DEFAULT 'Нове'; |

***Скриншот отриманого результату:***



**6) Видалення обмеження зовнішнього ключа:**

**Словесна постановка:**

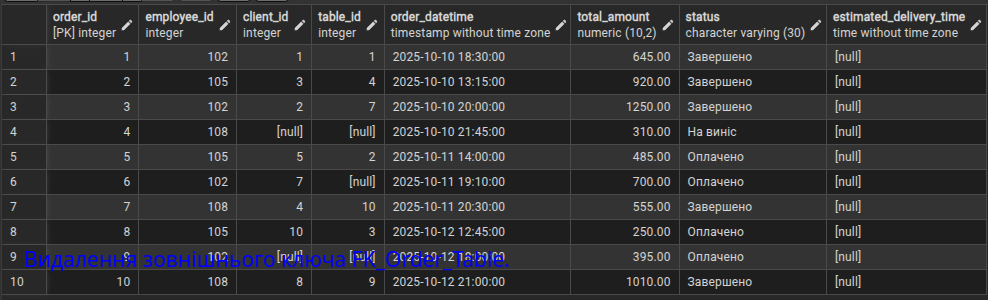
Видалити зовнішній ключ **FK\_Order\_Table** з таблиці **Замовлення**, який пов'язує замовлення зі столиком. Це необхідно, якщо ресторан вирішить дозволити замовлення, які не прив'язані до конкретного столика (наприклад, замовлення "на виніс", що ми вже робили, але без видалення ключа).

**SQL-код рішення:**

(Примітка: При виконанні цієї команди необхідно знати точну назву обмеження FK. Згідно з наданим раніше DDL, ім'я було FK\_Order\_Table.)

|  |
| --- |
| **ALTER** **TABLE** Замовлення  **DROP** **CONSTRAINT** FK\_Order\_Table; |

***Скриншот отриманого результату:***



## Висновок:

Протягом виконання лабораторної роботи було успішно засвоєно та продемонстровано ключові навички роботи з реляційною базою даних за допомогою основних команд мови SQL, зокрема SELECT та ALTER TABLE.

### Основні досягнення:

1. Майстерність вибірки даних (SELECT): Було розроблено та виконано десять запитів різного рівня складності. Це підтвердило розуміння і вміння застосовувати такі елементи:
   * Фільтрація (WHERE): Ефективне використання умов, включаючи складні булеві оператори (AND, OR), діапазони (BETWEEN) та шаблони (LIKE).
   * Об'єднання (JOIN): Створення запитів, які успішно об'єднують дані з декількох таблиць (наприклад, для зв'язку замовлень з іменами офіціантів або клієнтів із залами), включаючи зовнішні (LEFT JOIN) для отримання повних списків.
   * Агрегація та Групування (GROUP BY, HAVING): Розрахунок підсумкових даних (середня зарплата, кількість столиків) та фільтрація отриманих груп.
2. Управління структурою (ALTER TABLE): Було успішно проведено шість операцій, що змінюють схему бази даних, що демонструє здатність до підтримки та еволюції моделі:
   * Додано, видалено та перейменовано стовпці.
   * Змінено значення за замовчуванням.
   * Додано нове обмеження цілісності (CHECK) на значення атрибута.
   * Видалено існуюче обмеження зовнішнього ключа (DROP CONSTRAINT), що є важливою адміністративною операцією.

### Загальний підсумок:

Виконання цієї лабораторної роботи дозволило закріпити теоретичні знання про реляційні операції (проєкція, вибірка, об'єднання) та набути практичних навичок написання коректних і функціональних SQL-скриптів. Навички роботи з SELECT та ALTER TABLE є фундаментальними для будь-якого розробника або адміністратора баз даних.