**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

**ІКНІ**

Кафедра **ПЗ**



**ЗВІТ**

до лабораторної роботи №1

**на тему:** *“Аналіз предметної області та інсталяція СУБД”*

**з дисципліни** *“Бази даних”*

**Лектор:**

аис. кафедри ПЗ

Білоіваненко М. В.

**Виконав:**

ст. гр. ПЗ-33

Юшкевич. А.І.

**Прийняв:**

асис. кафедри ПЗ

Білоіваненко М. В.

«\_\_» \_\_\_\_\_\_\_\_\_ 2024р.

∑=\_\_\_\_\_

Львів – 2024

**Тема.** Аналіз предметної області та інсталяція СУБД.

**Мета.**

1. Ознайомлення з реляційними СУБД промислового рівня. Комерційне ПЗ та системи із відкритим кодом. Історія розвитку систем БД.
2. Поверхневий аналіз предметної області. Опис моделі ПО у встановлених межах. Виявлення важливої для зберігання інформації.
3. Розробка діаграми сутність-зв’язок. Відокремлення об’єктів від атрибутів при моделюванні ПО. Типізація сутностей та зв’язків.

# Теоретичні відомості

### 1. Поняття даних, інформації, знань

* **Дані**: Сировина для інформації; це незакінчені факти або значення, які не мають контексту.
* **Інформація**: Дані, які організовані та мають значення. Інформація є результатом обробки даних і має контекст.
* **Знання**: Інтерпретація та використання інформації на основі досвіду. Знання включає розуміння і застосування інформації для прийняття рішень.

### 2. Типологія моделей даних

* **Модель даних**: Абстрактне представлення об’єктів, подій та зв’язків між ними.
  + **Документна модель**: Описує дані у форматі документів, таких як JSON або XML.
  + **Графова модель**: Визначає дані як графи з вузлами та зв'язками.
  + **Реляційна модель**: Описує дані у вигляді таблиць, які мають реляційні зв'язки.
  + **Об'єктно-орієнтована модель**: Дані представлені як об'єкти з атрибутами та методами.
  + **Семантичні моделі**: Моделі, які акцентують увагу на значенні даних та їх відношеннях.

### 3. Метадані, схема даних

* **Метадані**: Інформація про дані; описують структуру даних, типи, обмеження і зв'язки.
* **Схема даних**: Опис структури даних у базі даних, включаючи типи даних, обмеження і зв'язки між таблицями.

### 4. Цілісність даних

* **Цілісність даних**: Забезпечення правильності та несуперечливості даних. Включає правила для запобігання помилок при введенні або зміні даних.

### 5. Рівні архітектури ПЗ

* **Рівень представлення інформації**: Інтерфейс для взаємодії з користувачем. Забезпечує зручний доступ до інформації.
* **Рівень бізнес-логіки**: Визначає функціональність і обробку даних відповідно до предметної області.
* **Рівень зберігання даних**: Фізичне зберігання даних, управління їх цілісністю та виконання транзакцій.

### 6. Розподіл обов’язків в системах із БД

* **Адміністратор даних**: Управляє даними, планує їх життєвий цикл, розробляє стандарти та бізнес-правила.
* **Адміністратор бази даних**: Фізична реалізація бази даних, забезпечення безпеки, продуктивності та супроводу системи.
* **Розробник логічної бази даних**: Ідентифікація даних, створення сутностей і зв’язків.
* **Розробник фізичної бази даних**: Фізичне втілення бази даних, проектування таблиць, обмежень і методів доступу.
* **Прикладні програмісти**: Розробка програм для взаємодії з базою даних.
* **Користувачі або клієнти**: Кінцеві споживачі інформації з бази даних.

### 7. Функції типової СУБД

* Визначення та ініціалізація структури даних.
* Опис моделі даних та доступ до метаданих.
* Підтримка цілісності даних.
* Авторизація і контроль доступу.
* Паралельне виконання транзакцій.
* Відновлення бази даних.
* Адміністративні утиліти: імпорт/експорт, моніторинг, оптимізація.

### 8. Вимоги до реляційних СУБД

* **Атомарність (Atomicity)**: Транзакція виконується повністю або не виконується зовсім.
* **Узгодженість (Consistency)**: База даних повинна залишатися в узгодженому стані до і після транзакції.
* **Ізольованість (Isolation)**: Зміни в транзакції не видимі іншим користувачам до її завершення.
* **Довговічність (Durability)**: Результати завершених транзакцій зберігаються навіть при збоях системи.

### 9. Властивості транзакції

* Транзакція є неподільним набором операцій, які або завершуються успішно (зміна даних у базі), або відхиляються повністю (жодна зміна не застосовується).

### 10. Реляційна модель даних

* Дані представляються у вигляді таблиць (реляцій), які мають ряди і стовпці. Таблиці можуть бути пов'язані між собою за допомогою ключів.

### 11. Множинність та модальність зв’язків

* **Множинність**: Кількість записів однієї таблиці, що можуть бути пов'язані з записами іншої таблиці (наприклад, один до багатьох).
* **Модальність**: Визначає обов'язковість або необов'язковість участі у зв'язку (наприклад, обов'язково чи необов'язково).

### 12. Мета нормалізації схеми БД

* **Нормалізація**: Процес організації даних у базі даних для зменшення надлишковості і забезпечення цілісності даних. Це включає розбиття таблиць на менші, уникання аномалій і створення зв'язків між таблицями.

# Завдання

1. Вибір сервера БД з виявленням його 2 переваг та 2 недоліків. Інсталяція ПЗ, виконання демонстраційного запиту.
2. Вибір індивідуальної предметної області для виконання робіт. Виявлення 5 зв’язаних сутностей ПО та їхніх важливих властивостей. Встановлення типів відношень.
3. Графічне створення концептуальної ER-діаграми ПО.
4. Текстовий опис 3 правил бізнес-логіки для коректного керування даними в обраній ПО.
5. Текстовий опис 3 форм табличної вибірки даних відповідно до розглянутого бізнес-процесу.

# Хід роботи

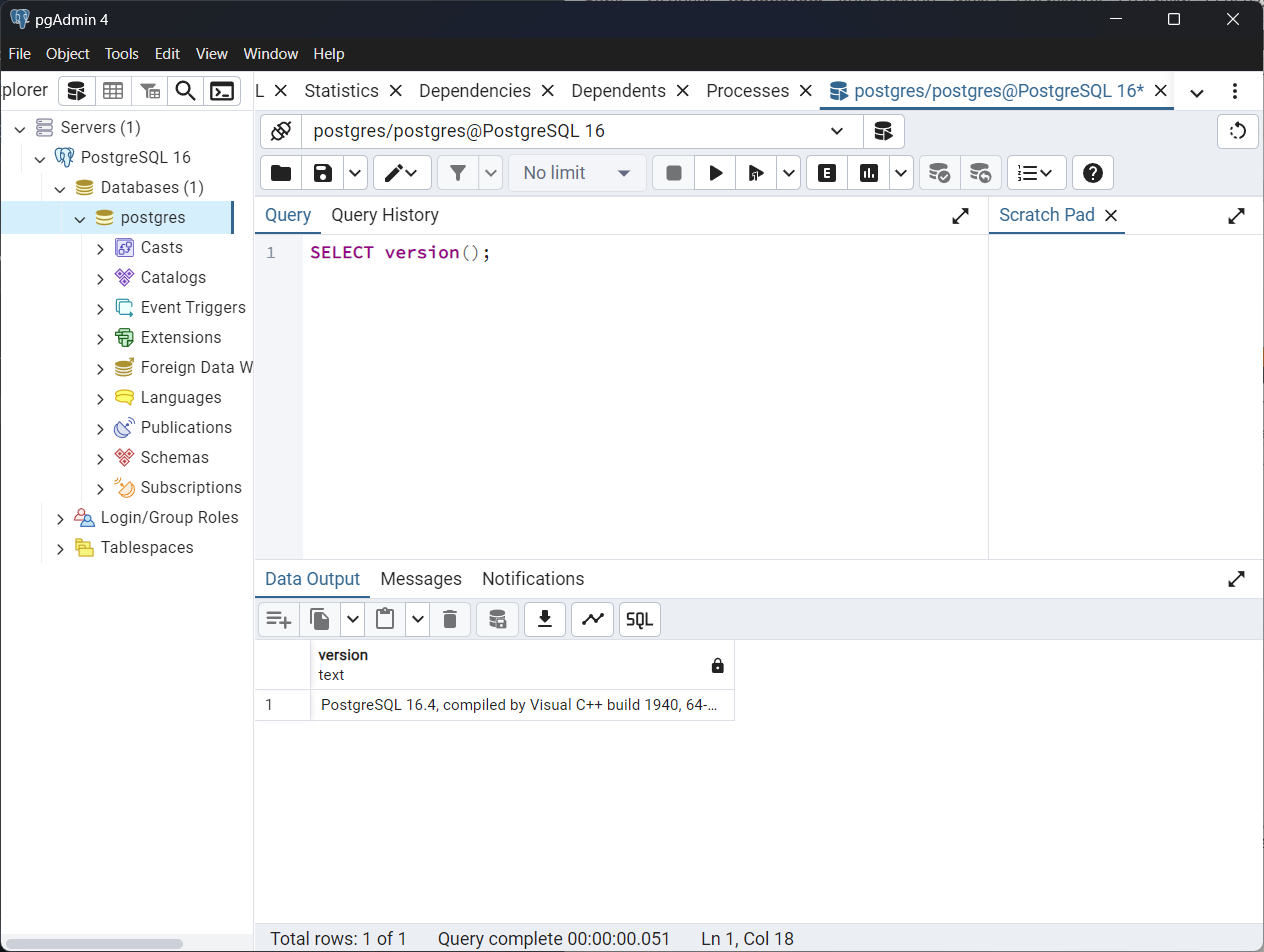
Обрана БД: **PostgreSQL**

**Переваги:**

1. ***Підтримка NoSQL-функціональності***: крім реляційних таблиць, PostgreSQL підтримує роботу з документами у форматі JSON, ключ-значення (hstore), а також графові бази даних через індекси GiST та GIN.
2. ***Масштабованість та паралельні запити****:*PostgreSQL добре масштабується як по вертикалі (на одному сервері), так і по горизонталі (на кількох серверах), що робить його придатним для великих баз даних.

**Недоліки:**

1. *Високе споживання ресурсів:* PostgreSQL вимагає більше ресурсів для роботи в порівнянні з іншими СУБД, як-от MySQL. Це може бути проблемою для малопотужних серверів або віртуальних машин.
2. *Складність налаштування:* Налаштування та оптимізація PostgreSQL можуть бути складними для новачків у порівнянні з простішими СУБД.



*Рис. 1. Знімок екрану графічного інтерфейсу ПЗ із активним*

*підключенням до створеної БД та тестовий запит*

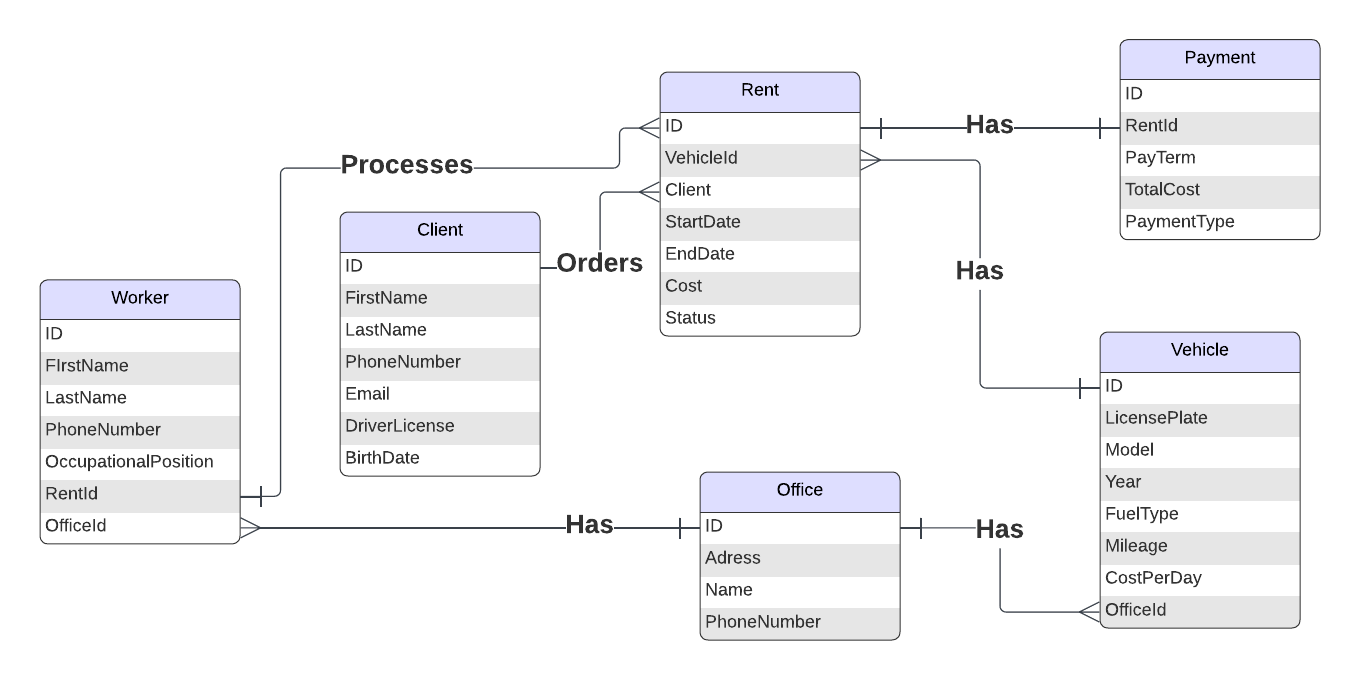
**Вибрана предметна область:** База даних для фірми з прокату автомобілів.

1. **Шість пов'язаних сутностей та їх атрибути:**

* *Автомобіль:* ID, Номерний знак, Модель, Рік випуску, Тип палива, Пробіг, Вартість прокату за добу, ID Офісу.
* *Клієнт:* ID, Ім'я, Прізвище, Номер телефону, Email, Водійське посвідчення, Дата народження.
* *Оренда:* ID, Автомобіль, Клієнт, Дата початку, Дата завершення, Вартість, Статус оренди.
* *Співробітник:* ID, Ім'я, Прізвище, Номер телефону, Посада, ID Офісу.
* *Офіс:* ID, Адреса, Назва офісу, Номер телефону.
* *Платіж:* ID, Оренда, Дата платежу, Сума, Спосіб оплати.

1. **Типи відносин:**

* *Оренда - Автомобіль:* Один до Багатьох (один автомобіль може мати багато оренд за різні періоди).
* *Оренда - Клієнт:* Один до Багатьох (один клієнт може брати в оренду кілька автомобілів).
* *Оренда - Співробітник:* Один до Багатьох (один співробітник може оформляти кілька оренд).
* *Офіс - Співробітник:* Один до Багатьох (один офіс може мати кілька співробітників).
* *Платіж - Оренда:* Один до Одного (одна оренда може мати один платіж).
* *Автомобіль - Офіс:* Один до Багатьох (один офіс може мати кілька автомобілів).



*Рис. 2. ER-діаграма*

1. **Опис бізнес-логіки:**
2. **Правило перевірки наявності автомобіля:**
   * Автомобіль можна здати в оренду лише, якщо він доступний на вибрані дати. Якщо автомобіль уже заброньований на ці дати, оренду оформити не можна.
   * **Реалізація:** При створенні нової оренди система перевіряє наявність вільних автомобілів у базі на вибраний період.
3. **Правило розрахунку вартості оренди:**
   * Загальна вартість оренди залежить від кількості днів оренди та вартості автомобіля за добу.
   * **Реалізація:** При оформленні оренди система автоматично розраховує загальну вартість на основі кількості днів та ставки за день.
4. **Правило перевірки водійського стажу:**
   * Оренду автомобіля можуть оформити лише клієнти з водійським посвідченням, отриманим щонайменше за рік до моменту оренди.
   * **Реалізація:** При оформленні замовлення система перевіряє дату видачі водійського посвідчення клієнта.

**Опис вибірок:**

1. **Вибірка доступних автомобілів на певну дату:**
   * Показує автомобілі, які доступні для оренди на вибрану дату. Виводяться: ID автомобіля, модель, номерний знак, рік випуску, тип палива, вартість прокату за добу.
2. **Історія оренд клієнта:**
   * Показує всі оренди, які оформляв конкретний клієнт. Виводяться: ID оренди, автомобіль, дати початку та завершення, загальна вартість оренди.
3. **Завантаженість автомобіля за певний період:**
   * Показує всі оренди конкретного автомобіля за вибраний період. Виводяться: ID оренди, дати початку та завершення, клієнт, сума, статус оренди.

**Висновки:** під час виконання цієї лабораторної роботи я ознайомився з реляційними СУБД промислового рівня, комерційним ПЗ та системами із відкритим кодом. Дізнався про історію розвитку систем БД. Поверхнево проаналізував предметну область транспортної компанії по перевезенню вантажів. Описав модель ПО у встановлених межах. Виявив важливу для зберігання інформацію. Розробив діаграму сутність-зв’язок. Вибрав систему управління базами даний, а саме PostgreSQL, та встановив на ПК.