

# Звіт до лабораторної роботи №1

**Тема:** Розробка концептуальної моделі бази даних для системи каршерингу.

## 1. Опис предметної області та вимоги

Система розробляється для автоматизації сервісу оренди автомобілів (Car Sharing). Головна мета — забезпечити взаємодію між користувачами та парком автомобілів, фіксувати поїздки та фінансові операції.

### Функціональні вимоги:

- **Управління користувачами:** Реєстрація клієнтів, зберігання контактних даних та інформації про водійське посвідчення.
- **Управління автопарком:** Облік автомобілів з чіткою прив'язкою до їх поточного місцезнаходження (локації) та технічного стану (статусу).
- **Оренда:** Фіксація факту бронювання та використання авто з прив'язкою до конкретного користувача та машини.
- **Фінанси:** Облік вартості поїздок та історії оплат.
- **Зворотний зв'язок:** Можливість залишати відгуки, які прив'язані до конкретного авто та користувача.

## 2. Сутності та атрибути

На основі аналізу вимог виділено наступні сутності. Для кожної сутності визначено первинні ключі (PK) та зовнішні ключі (FK), необхідні для зв'язків.

1. **USER** — інформація про клієнта.
  - a. `user_id` (PK): Унікальний ідентифікатор клієнта.
  - b. `name`: ПІБ користувача.
  - c. `email, phone_number`: Контактні дані.
  - d. `driver_license_number`: Номер посвідчення (для валідації права керування).
2. **CAR** — транспортний засіб.
  - a. `car_id` (PK): Унікальний ідентифікатор авто.
  - b. `license_plate`: Державний номерний знак.
  - c. `brand, model, year`: Технічні характеристики.
  - d. `price_per_hour`: Тарифна ставка.
  - e. `location_id` (FK): Посилання на поточну локацію.

- f. `status_id` (FK): Посилання на поточний статус (вільна, в ремонті тощо).
- 3. **RENT** — факт використання авто.
  - a. `rent_id` (PK): Унікальний номер поїздки.
  - b. `start_time`, `end_time`: Час початку та завершення.
  - c. `total_cost`: Розрахована вартість.
  - d. `status`: Стан поїздки (активна, завершена, скасована).
  - e. `user_id` (FK): Хто орендував.
  - f. `car_id` (FK): Яку машину орендували.
- 4. **PAYMENT** — фінансові транзакції.
  - a. `payment_id` (PK): Унікальний номер транзакції.
  - b. `amount`: Сума оплати.
  - c. `payment_method`: Спосіб оплати.
  - d. `payment_date`: Дата та час транзакції.
  - e. `rent_id` (FK): До якої поїздки відноситься оплата.
- 5. **REVIEW** — оцінка сервісу.
  - a. `review_id` (PK): Ідентифікатор відгуку.
  - b. `rating`: Числова оцінка (наприклад, 1-5).
  - c. `comment`: Текстовий коментар.
  - d. `user_id` (FK): Автор відгуку.
  - e. `car_id` (FK): Оцінюване авто.
- 6. **CAR\_LOCATION** — довідник паркувальних зон.
  - a. `location_id` (PK): Ідентифікатор локації.
  - b. `city`: Місто.
  - c. `address`: Адреса або координати зони.
- 7. **CAR\_STATUS** — довідник можливих станів.
  - a. `status_id` (PK): Ідентифікатор статусу.
  - b. `status_name`: Назва статусу (Available, Rented, Maintenance).

### 3. Опис зв'язків та бізнес-правила

- **CAR\_LOCATION — CAR (1:M)**: Одна локація може містити багато автомобілів, але кожен конкретний автомобіль в один момент часу знаходиться лише в одній локації.
- **CAR\_STATUS — CAR (1:M)**: Один статус (наприклад, "В ремонті") може бути присвоєний багатьом машинам, але машина завжди має лише один поточний статус.
- **USER — RENT (1:M)**: Один користувач може здійснити безліч поїздок, але кожна поїздка прив'язана до одного облікового запису користувача.

- **CAR — RENT (1:M):** Автомобіль бере участь у багатьох поїздках протягом свого життєвого циклу, але конкретний запис про оренду стосується однієї машини.
- **RENT — PAYMENT (1:M):** Одна поїздка може мати кілька спроб оплати або транзакцій, але кожна транзакція прив'язана до конкретної поїздки.
- **USER/CAR — REVIEW (1:M):** Користувачі залишають відгуки про конкретні автомобілі. Один відгук чітко посилається на одного користувача та одне авто.

#### 4. ER діаграма

