Poniżej znajdziesz **podsumowanie** struktury bazy danych w formie opisowej (bez kodu SQL), zawierające:

1. **Wymienienie wszystkich tabel**.
2. **Kolumny wraz z sugerowanymi typami**.
3. **Relacje między tabelami** w postaci 1..N, N..1 czy 1..1.

Możesz traktować to jako finalny opis schematu, który odpowiada Twojemu diagramowi.

## 1. **Users**

**Kolumny (przykładowe typy):**

* id – **UUID** (główny klucz, unikalny identyfikator użytkownika)
* email – **VARCHAR(255)** (unikalny adres e-mail)
* password\_hash – **VARCHAR(255)** (przechowywany hash hasła)
* first\_name – **VARCHAR(50)** (imię)
* last\_name – **VARCHAR(50)** (nazwisko)
* phone\_number – **VARCHAR(20)** (numer telefonu)
* created\_at – **TIMESTAMP** (data utworzenia rekordu)
* updated\_at – **TIMESTAMP** (data ostatniej modyfikacji)
* (jeśli w bazie występuje kolumna *role*) role – **VARCHAR(20)** (np. "user", "admin")

**Relacje:**

* **Users** (1) → (N) **Reservations**  
  (jeden użytkownik może mieć wiele rezerwacji)
* **Users** (1) → (N) **Reviews**  
  (jeden użytkownik może dodać wiele opinii)
* **Users** (1) → (N) **Support\_ticket**  
  (jeden użytkownik może mieć wiele zgłoszeń)
* **Users** (1) → (N) **Admin\_reports**  
  (jeden użytkownik z rolą „admin” może tworzyć wiele raportów)
* (jeśli istnieje osobna tabela *User\_roles*)  
  **Users** (1) → (N) **User\_roles**  
  (jeden użytkownik może mieć wiele wpisów z różnymi rolami)

## 2. **Admin\_reports**

**Kolumny:**

* id – **UUID** (główny klucz)
* admin\_id – **UUID** (klucz obcy wskazujący na **Users**)
* report\_type – **VARCHAR(50)** (typ raportu, np. "inventory", "finance" itp.)
* title – **VARCHAR(100)** (tytuł raportu)
* start\_date – **DATE** (opcjonalna data początkowa zakresu raportu)
* end\_date – **DATE** (opcjonalna data końcowa zakresu raportu)
* content – **TEXT** (zawartość raportu)
* (opcjonalnie) created\_at / updated\_at – **TIMESTAMP**

**Relacje:**

* **Admin\_reports** (N) → (1) **Users**  
  (raport jest tworzony przez jednego użytkownika o roli „admin”)

## 3. **Support\_ticket**

**Kolumny:**

* id – **UUID** (główny klucz)
* user\_id – **UUID** (klucz obcy do **Users**)
* issue\_description – **TEXT** (opis problemu)
* status – **VARCHAR(20)** (np. "open", "in\_progress", "resolved")
* created\_at – **TIMESTAMP**
* updated\_at – **TIMESTAMP**

**Relacje:**

* **Support\_ticket** (N) → (1) **Users**  
  (każde zgłoszenie należy do jednego użytkownika)

## 4. **Reservations**

**Kolumny:**

* id – **UUID** (główny klucz)
* user\_id – **UUID** (klucz obcy do **Users**)
* start\_date – **DATE** (data rozpoczęcia rezerwacji)
* end\_date – **DATE** (data zakończenia rezerwacji)
* status – **VARCHAR(20)** (np. "pending", "confirmed", "canceled", "completed")
* created\_at – **TIMESTAMP**
* updated\_at – **TIMESTAMP**

**Relacje:**

* **Reservations** (N) → (1) **Users**  
  (rezerwacja jest powiązana z jednym użytkownikiem)
* **Reservations** (1) → (N) **Reservation\_items**  
  (jedna rezerwacja może zawierać wiele pozycji sprzętu)
* **Reservations** (1) → (1) **Payments**  
  (jedna rezerwacja może mieć jedną płatność; jeśli chcesz obsługiwać wiele płatności, można zmienić na 1..N)

## 5. **Reservation\_items**

**Kolumny:**

* id – **UUID** (główny klucz)
* reservation\_id – **UUID** (klucz obcy do **Reservations**)
* equipment\_id – **UUID** (klucz obcy do **Equipment**)
* quantity – **INT** (ilość danego sprzętu)
* price\_per\_day – **DECIMAL(10,2)** (cena za sztukę/dzień)
* total\_price – **DECIMAL(10,2)** (łączna cena za tę pozycję)

**Relacje:**

* **Reservation\_items** (N) → (1) **Reservations**  
  (pozycja rezerwacji należy do jednej rezerwacji)
* **Reservation\_items** (N) → (1) **Equipment**  
  (pozycja rezerwacji dotyczy jednego rodzaju sprzętu)

## 6. **Payments**

**Kolumny:**

* id – **UUID** (główny klucz)
* reservation\_id – **UUID** (klucz obcy do **Reservations**)
* amount – **DECIMAL(10,2)** (kwota płatności)
* payment\_method – **VARCHAR(20)** (np. "PayPal", "BLIK", "Visa")
* status – **VARCHAR(20)** (np. "pending", "completed", "failed")
* created\_at – **TIMESTAMP**

**Relacje:**

* **Payments** (1) → (1) **Reservations**  
  (płatność jest przypisana do konkretnej rezerwacji)

## 7. **Reviews**

**Kolumny:**

* id – **UUID** (główny klucz)
* user\_id – **UUID** (klucz obcy do **Users**)
* equipment\_id – **UUID** (klucz obcy do **Equipment**)
* rating – **INT** (wartość oceny, np. 1–5)
* comment – **TEXT** (komentarz użytkownika)
* created\_at – **TIMESTAMP**

**Relacje:**

* **Reviews** (N) → (1) **Users**  
  (opinia jest wystawiana przez jednego użytkownika)
* **Reviews** (N) → (1) **Equipment**  
  (opinia dotyczy jednego rodzaju sprzętu)

## 8. **Equipment**

**Kolumny:**

* id – **UUID** (główny klucz)
* name – **VARCHAR(100)** (nazwa sprzętu)
* category\_id – **UUID** (klucz obcy do **Categories**)
* description – **TEXT** (opis sprzętu)
* price\_per\_day – **DECIMAL(10,2)** (cena za dzień)
* available\_quantity – **INT** (dostępna liczba sztuk)
* created\_at – **TIMESTAMP**
* updated\_at – **TIMESTAMP**

**Relacje:**

* **Equipment** (N) → (1) **Categories**  
  (sprzęt należy do jednej kategorii)
* **Equipment** (1) → (N) **Reservation\_items**  
  (sprzęt może pojawiać się w wielu rezerwacjach)
* **Equipment** (1) → (N) **Reviews**  
  (sprzęt może mieć wiele opinii)

## 9. **Categories**

**Kolumny:**

* id – **UUID** (główny klucz)
* name – **VARCHAR(50)** (nazwa kategorii, np. „Rowery”, „Narty”)
* description – **TEXT** (opis kategorii)

**Relacje:**

* **Categories** (1) → (N) **Equipment**  
  (jedna kategoria może zawierać wiele rodzajów sprzętu)

## 10. (opcjonalnie) **User\_roles**

(Tylko jeśli w Twoim projekcie faktycznie występuje osobna tabela na role. Często wystarczy kolumna *role* w tabeli ***Users***.)

**Kolumny:**

* id – **UUID** (główny klucz)
* user\_id – **UUID** (klucz obcy do **Users**)
* role – **VARCHAR(20)** (np. "user", "admin", "manager")

**Relacje:**

* **User\_roles** (N) → (1) **Users**  
  (jeden rekord w tabeli **User\_roles** jest powiązany z jednym użytkownikiem;  
  jeden użytkownik może mieć wiele ról, jeśli taki jest Twój zamysł)

# Relacje (zestawienie końcowe)

1. **Users** (1) → (N) **Reservations**
2. **Users** (1) → (N) **Support\_ticket**
3. **Users** (1) → (N) **Reviews**
4. **Users** (1) → (N) **Admin\_reports** (tylko dla roli admin)
5. **Reservations** (1) → (N) **Reservation\_items**
6. **Reservation\_items** (N) → (1) **Equipment**
7. **Reservations** (1) → (1) **Payments**
8. **Equipment** (N) → (1) **Categories**
9. **Equipment** (1) → (N) **Reviews**
10. (opcjonalnie) **Users** (1) → (N) **User\_roles**  
    (jeśli role są w osobnej tabeli)

**Uwaga**:

* W niektórych miejscach możesz zmienić relację na 1..N lub 1..1 w zależności od potrzeb (np. płatności).
* Typy danych (VARCHAR, DECIMAL, TIMESTAMP, itp.) można doprecyzować do potrzeb biznesowych.
* Jeśli używasz **ENUM** w PostgreSQL, możesz zdefiniować typy statusów (np. rezerwacji, płatności) jako typy wyliczeniowe.

Dzięki takiej strukturze masz możliwość:

* Wielu sprzętów w jednej rezerwacji (tabela **Reservation\_items**).
* Śledzenia płatności, opinii, zgłoszeń serwisowych, raportów administracyjnych.
* Prostej rozbudowy o dodatkowe pola (np. ubezpieczenia, rabaty) w osobnych tabelach, jeśli zajdzie taka potrzeba.