

Poniżej znajduje się opis relacyjnej bazy danych dla systemu zarządzania zadaniami w hotelu, wykonanej w MySQL przy użyciu MariaDB. Opis zawiera tabele, ich pola z typami danych oraz relacje między nimi.

---

## Tabele i pola z typami danych

### *Tabela employee (Pracownicy)*

Przechowuje dane pracowników hotelu.

- **id:** INTEGER (Primary Key, auto-increment) – Unikalny identyfikator pracownika.
- **first\_name:** VARCHAR(50) – Imię pracownika.
- **last\_name:** VARCHAR(50) – Nazwisko pracownika.
- **email:** VARCHAR(100) (unikalny) – Adres e-mail służący jako login.
- **password:** VARCHAR(255) – Zaszyfrowane hasło pracownika.
- **phone\_number:** VARCHAR(15) – Numer telefonu pracownika.
- **role\_id:** INTEGER – Identyfikator roli pracownika (powiązanie z tabelą roles).

### *Tabela roles (Role)*

Zawiera definicje ról pracowników.

- **id:** INTEGER (Primary Key, auto-increment) – Unikalny identyfikator roli.
- **name:** ENUM('Receptionist', 'Manager', 'Housekeeper', 'Maintenance') (unikalny) – Nazwa roli pracownika w hotelu.

### *Tabela rooms (Pokoje)*

Zawiera informacje o pokojach w hotelu.

- **id:** INTEGER (Primary Key, auto-increment) – Unikalny identyfikator pokoju.
- **room\_number:** VARCHAR(10) (unikalny) – Numer pokoju w hotelu.
- **floor:** INTEGER – Numer piętra, na którym znajduje się pokój.
- **bed\_count:** INTEGER – Liczba łóżek w pokoju.
- **price\_per\_night:** DECIMAL(10, 2) – Cena za noc w pokoju.
- **status:** ENUM('available', 'occupied', 'out\_of\_service') – Aktualny status pokoju.

### *Tabela reservations (Rezerwacje)*

Zarządza danymi rezerwacji i informacjami powiązanymi z gośćmi.

- **id:** INTEGER (Primary Key, auto-increment) – Unikalny identyfikator rezerwacji.
- **start\_date:** DATE – Data rozpoczęcia rezerwacji.
- **end\_date:** DATE – Data zakończenia rezerwacji.
- **status:** ENUM('active', 'cancelled', 'completed') – Status rezerwacji.
- **special\_requests:** TEXT – Dodatkowe prośby lub uwagi gościa.
- **modified\_at:** DATETIME – Data ostatniej modyfikacji rezerwacji.

- **catering:** BOOLEAN – Informacja, czy rezerwacja obejmuje catering.
- **guest\_first\_name:** VARCHAR(50) – Imię gościa dokonującego rezerwacji.
- **guest\_last\_name:** VARCHAR(50) – Nazwisko gościa dokonującego rezerwacji.
- **guest\_pesel:** VARCHAR(11) – PESEL gościa.
- **guest\_phone:** VARCHAR(15) – Numer telefonu gościa.
- **invoice\_id:** INTEGER (Foreign Key do invoices(id), nullable) – Identyfikator powiązanej faktury, jeśli istnieje.

### ***Tabela reservation\_rooms (Tabela łącząca rezerwacje i pokoje)***

Umożliwia powiązanie wielu pokoi z jedną rezerwacją.

- **id:** INTEGER (Primary Key, auto-increment) – Unikalny identyfikator powiązania.
- **reservation\_id:** INTEGER (Foreign Key do reservations(id)) – Identyfikator rezerwacji.
- **room\_id:** INTEGER (Foreign Key do rooms(id)) – Identyfikator pokoju.
- **guest\_count:** INTEGER – Liczba gości w pokoju.

### ***Tabela maintenance\_requests (Zgłoszenia serwisowe)***

Przechowuje zgłoszenia serwisowe.

- **id:** INTEGER (Primary Key, auto-increment) – Unikalny identyfikator serwisu.
- **request\_date:** DATETIME – Data zgłoszenia serwisu.
- **description:** TEXT – Opis problemu serwisowego.
- **room\_id:** INTEGER (Foreign Key do rooms(id)) – Identyfikator pokoju, którego dotyczy problem.
- **requester\_id:** INTEGER (Foreign Key do employee(id), nullable) – Identyfikator pracownika zgłaszającego problem.
- **assignee\_id:** INTEGER (Foreign Key do employee(id)) – Identyfikator pracownika wykonującego serwis.
- **status:** ENUM('pending', 'in\_progress', 'completed') – Status serwisu.
- **service\_summary:** TEXT – Podsumowanie wykonanych działań.
- **completion\_date:** DATETIME – Data zakończenia serwisu.

### ***Tabela housekeeping\_tasks (Zlecenia sprzątania)***

Zarządza zadaniami sprzątania.

- **id:** INTEGER (Primary Key, auto-increment) – Unikalny identyfikator zlecenia.
- **employee\_id:** INTEGER (Foreign Key do employee(id)) – Identyfikator przypisanej pokojówki.
- **room\_id:** INTEGER (Foreign Key do rooms(id)) – Identyfikator pokoju do posprzątania.
- **request\_date:** DATETIME – Data dodania zlecenia.
- **completion\_date:** DATETIME – Data zakończenia zlecenia.
- **status:** ENUM('pending', 'in\_progress', 'completed', 'declined') – Status zlecenia.

- **description:** TEXT – Opis zlecenia sprzątania.

### ***Tabela invoices (Faktury)***

Przechowuje informacje o fakturach.

- **id:** INTEGER (Primary Key, auto-increment) – Unikalny identyfikator faktury.
- **issue\_date:** DATE – Data wystawienia faktury.
- **pdf\_file:** VARCHAR(255) – Ścieżka lub URL do pliku PDF z fakturą.
- **company\_nip:** VARCHAR(20) (nullable) – Numer NIP dla faktury.
- **company\_name:** VARCHAR(100) (nullable) – Nazwa firmy dla faktury.
- **company\_address:** VARCHAR(255) (nullable) – Adres rozliczeniowy dla faktury.

### ***Tabela reports (Raporty)***

Przechowuje raporty generowane przez system.

- **id:** INTEGER (Primary Key, auto-increment) – Unikalny identyfikator raportu.
- **report\_file:** VARCHAR(255) – Ścieżka lub URL do pliku raportu.
- **created\_at:** DATETIME – Data utworzenia raportu.
- **report\_type:** ENUM('employee\_statistics', 'general\_report') – Typ raportu.
- **created\_by:** INTEGER (Foreign Key do employee(id)) – Identyfikator pracownika, który utworzył raport.

---

## **Relacje między tabelami**

### **1. employee ↔ roles:**

- Jeden pracownik ma jedną rolę, a jedna rola może być przypisana do wielu pracowników.
- Klucz obcy: role\_id w employee odwołuje się do roles(id) (relacja wiele-do-jednego).

### **2. reservation\_rooms ↔ reservations:**

- Jedna rezerwacja może obejmować wiele pokoi poprzez tabelę łączącą.
- Klucz obcy: reservation\_id w reservation\_rooms odwołuje się do reservations(id) (relacja wiele-do-jednego).

### **3. reservation\_rooms ↔ rooms:**

- Jeden pokój może być powiązany z wieloma rezerwacjami poprzez tabelę łączącą.
- Klucz obcy: room\_id w reservation\_rooms odwołuje się do rooms(id) (relacja wiele-do-jednego).

### **4. maintenance\_requests ↔ employee:**

- Jeden pracownik może zgłosić wiele serwisów (requester\_id) lub być przypisany do wielu serwisów (assignee\_id).
- Klucze obce: requester\_id i assignee\_id w maintenance\_requests odwołują się do employee(id) (relacje wiele-do-jednego).

### **5. maintenance\_requests ↔ rooms:**

- Jeden pokój może mieć wiele zgłoszeń serwisowych.
- Klucz obcy: room\_id w maintenance\_requests odwołuje się do rooms(id) (relacja wiele-do-jednego).

**6. housekeeping\_tasks ↔ employee:**

- Jeden pracownik (pokojówka) może być przypisany do wielu zleceń sprzątnia.
- Klucz obcy: employee\_id w housekeeping\_tasks odwołuje się do employee(id) (relacja wiele-do-jednego).

**7. housekeeping\_tasks ↔ rooms:**

- Jeden pokój może mieć wiele zleceń sprzątnia.
- Klucz obcy: room\_id w housekeeping\_tasks odwołuje się do rooms(id) (relacja wiele-do-jednego).

**8. reports ↔ employee:**

- Jeden pracownik może utworzyć wiele raportów.
- Klucz obcy: created\_by w reports odwołuje się do employee(id) (relacja wiele-do-jednego).

**9. reservations ↔ invoices:**

- Jedna rezerwacja może mieć jedną fakturę.
- Klucz obcy: invoice\_id w reservations odwołuje się do invoices(id) (relacja jeden-do-jednego).