Костылев МТ-302

11. БД "Гостиница".

Задача – информационная поддержка деятельности гостиницы.

БД должна осуществлять:

* ведение списка постояльцев;
* учёт забронированных мест;
* ведение архива выбывших постояльцев за последний год.

Необходимо предусмотреть:

* Получение списка свободных номеров (по количеству мест и классу);
* Получение списка номеров (мест), освобождающихся сегодня и завтра;
* Выдачу информации по конкретному номеру;
* Автоматизацию выдачи счетов на оплату номера и услуг;
* Получение списка забронированных номеров;
* Проверку наличия брони по имени клиента и/или названию организации.
* Выдачу сотрудника, который работал в нужный день и время.

Сущности и их свойства:

* **Бронь** (id брони, id постояльца, дата въезда, дата выезда, номер забронированного номера, состояние оплаты)
* **Номер** (номер номера, id текущего состояния, кол-во мест в номере, класс номера, стоимость номера)
* **Состояние** (id состояния, код состояние, время смены состояния, дата смены состояния, id работающего сотрудника)
* **Постоялец** (id, фио, телефон, паспортные данные, id последней брони, id организации)
* **Организация** (id, тип организации, название организации)
* **Сотрудник** (id, фио, должность, зарплата, паспортные данные)
* **Сотрудник на смене** (id смены сотрудника, id сотрудника, дата начала смены, время начала смены, время смены)

Отношения:

* **Организация – Постоялец:**

Связь: Один - Много

У одной организации может быть много сотрудников (потенциальных постояльцев), а у постояльца может быть только одна организация.

* **Постоялец – Бронь:**

Связь: Один – Много

У одного постояльца может быть много броней, а за каждой бронью прикреплен только один постоялец.

* **Бронь – Номер:**

Связь: Много – Один

У одного номера может быть много броней, а за каждой бронью прикреплен только один номер.

* **Номер – Состояние:**

Связь: Один – Много

У каждого номера может быть несколько состояний (не противоречащих друг другу), а за каждым состоянием закреплен один номер.

* **Состояние – Сотрудник на смене:**

Связь: Много – Один

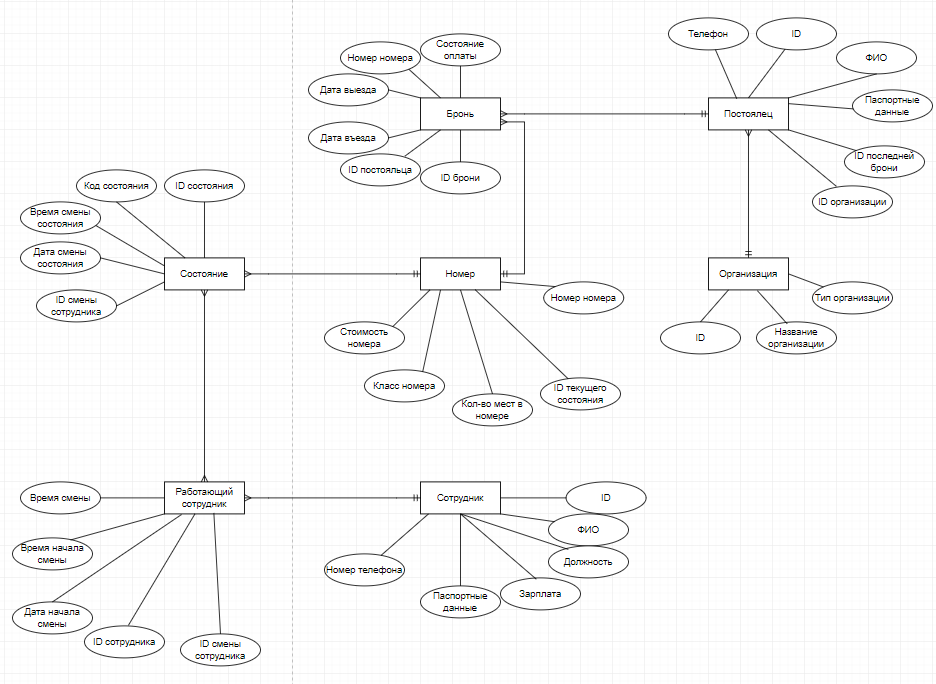
Сотрудник на смене, за смену может сменить несколько состояний, а за каждым состоянием закреплен один сотрудник.

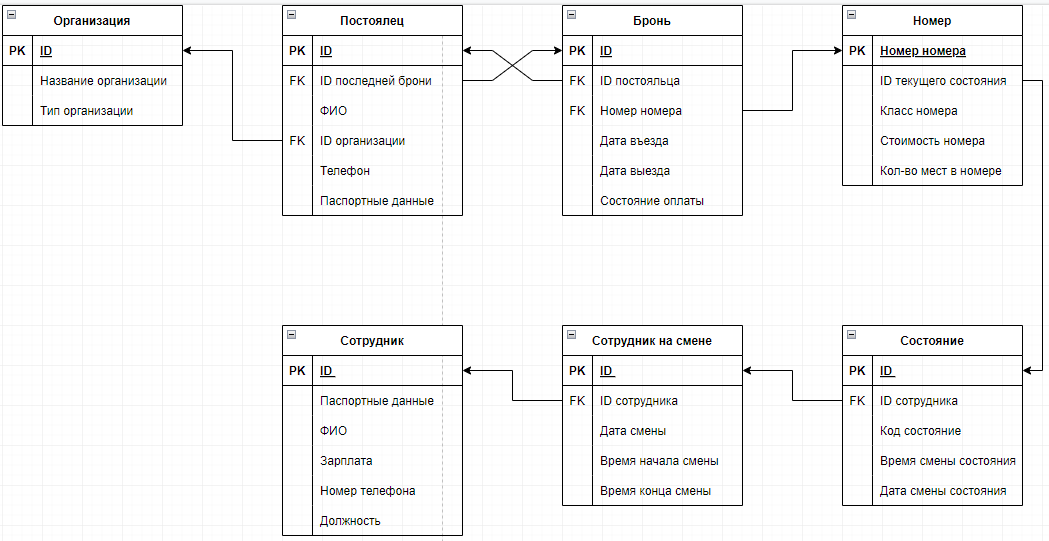
* **Сотрудник на смене – Сотрудник:**

Связь: Много – Один

У сотрудника может быть много смен. Но сотрудник на смене, это только один сотрудник.

РАЗРАБОТКА





|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблица | Свойство | Тип | Ограничения | Описание |
| Бронь | id | INT auto\_inc | Primary key | Ид. |
| id\_resident | INT | Not null | Ид. постояльца |
| check\_out\_date | DATE (TEXT) | Not null | Дата выселения |
| check\_in\_date | DATE (TEXT) | Not null | Дата заселения |
| room | INT | Not null | Номер комнаты |
| prepay | INT | Not null | Сумма предоплаты |
| Номер | number | INT auto\_inc | Primary key | Номер номера |
| class | TEXT | Not null | Класс номера |
| cost | INT | Not null | Стоимость номера |
| number\_of\_bed | TEXT | Not null | Количество спальных мест. Например (2+2+1) |
| Класс номера | class | TEXT | Primary key | Класс номера |
| base\_cost | INT | Not null | Базовая стоимость номера |
| added\_value | INT | Noy null | Добавочная стоимость за место |
| Состояние | status\_code | INT | Not null | Код состояния |
| time\_change | TIME (TEXT) | Not null | Время смены состояния |
| date\_change | DATE (TEXT) | Not null | Дата смены состояния |
| id\_working\_employee | INT | Not null | Ид. Смены работника сменившего состояние. |
| room | INT | Not null | Номер комнаты |
| id | INT auto\_inc | Primary key | Ид. |
| Код состояния | id\_code | INT | Primary key | Ид. кода состояния |
| name\_code | TEXT | Not null | Название состояния |
| Постоялец | name | TEXT | Not null |  |
| name | TEXT | Not null | ФИО |
| id\_organization | INT | Default null | Ид. Организации |
| phone\_number | INT | Default null | Номер телефона |
| passport\_details | INT | Not null | Паспортные данные (серия + номер) |
| id | INT auto\_inc | Primary key | Ид. |
| Организация | name | TEXT | Not null | Название организации |
| type | TEXT | Not null | Тип организации |
| id | INT auto\_inc | Primary key | Ид. |
| Сотрудник | name | TEXT | Not null | ФИО |
| position | TEXT | Not null | Должность |
| salary | INT | Not null | Зарплата |
| phone\_number | INT | Default null | Номер телефона |
| passport\_details | INT | Not null | Паспортные данные (серия + номер) |
| id | INT auto\_inc | Primary key | Ид. |
| Сотрудник на смене | id\_employee | INT | Not null | Ид. сотрудника |
| shift\_start\_date | DATE (TEXT) | Not null | Дата начала смены |
| shift\_start\_time | TIME (TEXT) | Not null | Время начала смены |
| shift\_time | TIME (TEXT) | Default null | Время смены |
| id | INT | Primary key | Ид. сотрудника на смене |
| Должность | position | TEXT | Primary key | Должность |
| wage-rate | INT | Not null | Ставка |

Примечание: Ид. = Идентификатор

**Создание БД на sqlite3 и python**

import sqlite3  
  
connection = sqlite3.connect('hostel.db')  
  
cursor = connection.cursor()  
  
  
# Создание таблиц.  
def create\_tables():  
 # Создаем таблицу комнат.  
 cursor.execute('''CREATE TABLE IF NOT EXISTS Room  
 (number INTEGER NOT NULL PRIMARY KEY,   
 id\_status INTEGER NOT NULL,   
 class TEXT NOT NULL,  
 cost INTEGER NOT NULL,  
 number\_of\_bed TEXT NOT NULL,  
 FOREIGN KEY (id\_status) REFERENCES Status (id)  
 )''')  
  
 # Создаем таблицу постояльцев.  
 cursor.execute('''CREATE TABLE IF NOT EXISTS Resident  
 (id INTEGER NOT NULL PRIMARY KEY,   
 id\_last\_reservation INTEGER NOT NULL,   
 name TEXT NOT NULL,  
 id\_organization INTEGER DEFAULT NULL,  
 phone\_number INTEGER DEFAULT NULL,  
 passport\_details INTEGER NOT NULL,  
 FOREIGN KEY (id\_last\_reservation) REFERENCES Reservation (id),  
 FOREIGN KEY (id\_organization) REFERENCES Organization (id)  
 )''')  
  
 # Создаем таблицу организаций.  
 cursor.execute('''CREATE TABLE IF NOT EXISTS Organization  
 (id INTEGER NOT NULL PRIMARY KEY,   
 name TEXT NOT NULL,   
 type TEXT NOT NULL  
 )''')  
  
 # Создаем таблицу состояний.  
 cursor.execute('''CREATE TABLE IF NOT EXISTS Status  
 (id INTEGER NOT NULL PRIMARY KEY,   
 status\_code INTEGER NOT NULL,   
 time\_change TEXT NOT NULL,  
 date\_change TEXT NOT NULL,  
 id\_working\_employee INTEGER NOT NULL,  
 FOREIGN KEY (id\_working\_employee) REFERENCES Working\_Employee (id)  
 )''')  
  
 # Создаем таблицу броней.  
 cursor.execute('''CREATE TABLE IF NOT EXISTS Reservation  
 (id INTEGER NOT NULL PRIMARY KEY,   
 id\_resident INTEGER NOT NULL,   
 check\_out\_date TEXT NOT NULL,  
 check\_in date TEXT NOT NULL,  
 room\_number INTEGER NOT NULL,  
 prepay INTEGER NOT NULL,  
 FOREIGN KEY (id\_resident) REFERENCES Resident (id),  
 FOREIGN KEY (room\_number) REFERENCES Room (number)  
 )''')  
  
 # Создаем таблицу сотрудников.  
 cursor.execute('''CREATE TABLE IF NOT EXISTS Employee  
 (id INTEGER NOT NULL PRIMARY KEY,   
 name TEXT NOT NULL,   
 position TEXT NOT NULL,  
 salary INTEGER NOT NULL,  
 phone\_number INTEGER DEFAULT NULL,  
 passport\_details TEXT NOT NULL  
 )''')  
  
 # Создаем таблицу работающих сотрудников.  
 cursor.execute('''CREATE TABLE IF NOT EXISTS Working\_Employee  
 (id INTEGER NOT NULL PRIMARY KEY,   
 id\_employee INTEGER NOT NULL,   
 shift\_start\_date TEXT NOT NULL,  
 shift\_start\_time TEXT NOT NULL,  
 shift\_time TEXT DEFAULT NULL,  
 FOREIGN KEY (id\_employee) REFERENCES Employee (id)  
 )''')  
  
  
create\_tables()  
  
connection.commit()  
connection.close()