

Модуль 1

Проектирование и разработка информационной системы.

Дано: описание брифинга и документы заказчика.

Надо: сделать ER-диаграмму для ИС 3-й нормальной формой с ссылочной целостностью.

Результат: pdf файл.

Анализ

Брифинг

1. Приложение для управления гостиницы.
2. Приложение доступно руководителю и сотруднику гостиницы, а также некоторые его функции клиенту и гостям.
3. Приложение исключительно для персонала гостиницы на ПК в офисе, но гостям нужно дать возможность пользоваться функционалом на своих устройствах.
4. Администратор гостиницы имеет полный доступ к информации о бронировании и заселении. Он может управлять номерным фондом и назначать свободные номера, оперативно взаимодействовать с другими отделами для решения вопросов гостей и проведения финансовых операций (выставления счетов).
5. Администратор регистрирует гостей (ищет бронь или создаёт новую), вводит их данные, выбирает нужный тип номера, проверяет доступность и назначает, а после регистрирует оплату и выдаёт карту доступа к номеру.
6. В день выселения в 12:00 карта доступа блокируется и статус номера числится как "грязный".
7. Администратор создаёт график уборки номеров на определённый период (неделя, месяц), распределяя уборщиков по номерам. Этим он устанавливает статус номеров, как "назначен к уборке", а после уборки ставит статус "чистый".
8. Руководитель смотрит статистику загруженности отеля, выручке, наполненности номеров и другие показатели, просматривает за выполнением задач.
9. Загрузка номерного фонда - проданные ночи / общее число номеров.
10. ADR - чистая выручка / проданные ночи.
11. RevPAR - $ADR * \text{загрузка номерного фонда}$

12. Необходима интеграция с PMS.

13. Необходимо обеспечить конфиденциальность и безопасность данных о гостях.

Документы зачкики.

1. Номерной фонд: этаж, номер, категория.

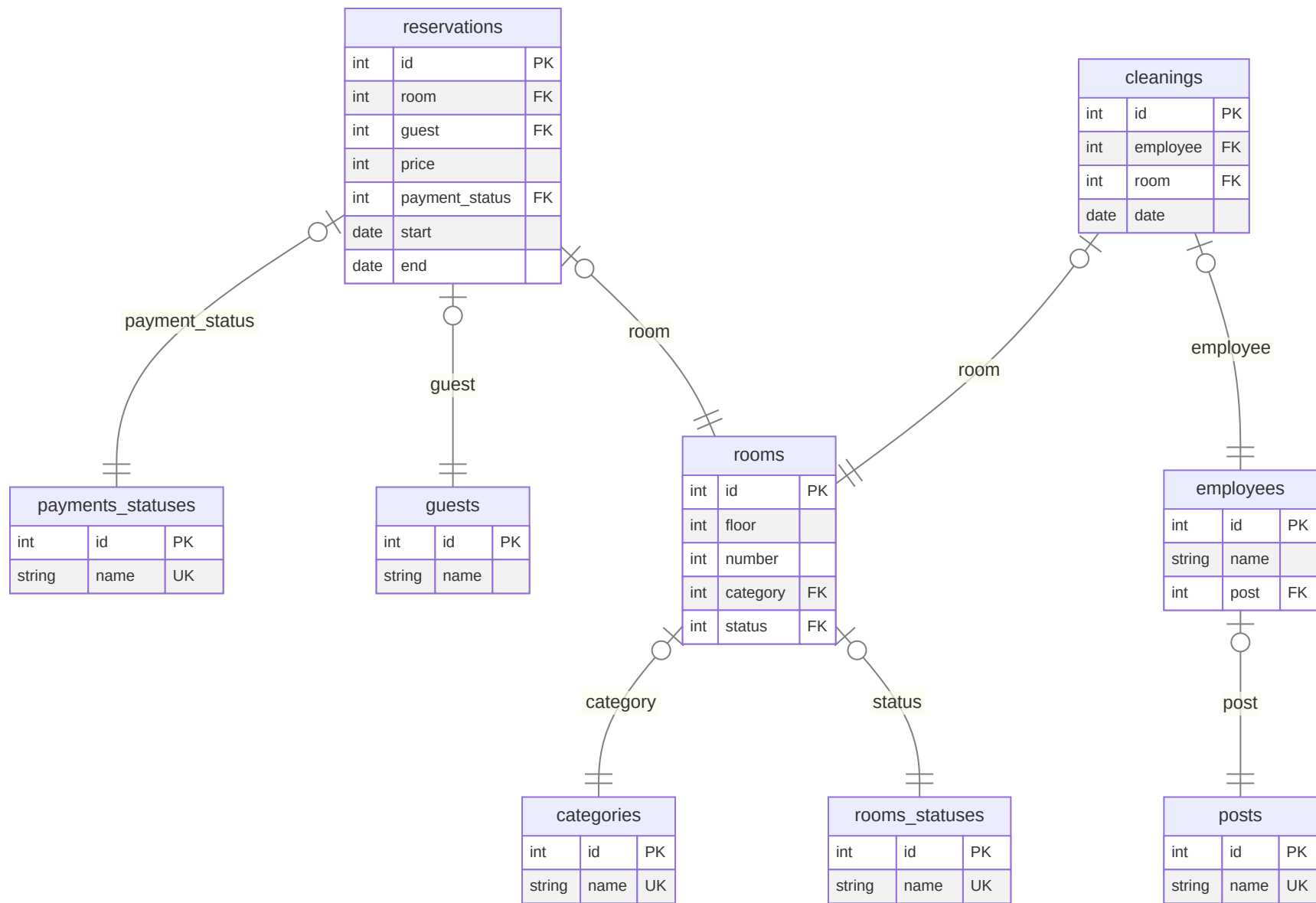
2. Отчёт состояния номерного фонда на дату: этаж, номер, категория, статус, дата выезда.

3. Постояльцы в гостиной сейчас: этаж, номер, категория, клиент, въезд, выезд.

Вывод

Доступ к БД должен ограничен и нужно сделать API.

Диаграмма



Модуль 2

Сoadминистрирование баз данных и серверов.

Дано: ER-диаграмма из модуля 1, документы заказчика.

Надо: сделать БД на сервере и импортировать данные из файла "Номерной фонд.xlsx", создайте запрос вычисляющий процент загрузки номерного фонда.

Результат: заполненная БД на сервере.

База данных

```
DROP DATABASE IF EXISTS hotel;
CREATE DATABASE hotel DEFAULT CHARACTER SET utf8mb4 DEFAULT COLLATE utf8mb4_unicode_ci;

USE hotel;

CREATE TABLE categories(
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(255) UNIQUE NOT NULL
);

INSERT INTO categories(id, name) VALUES
(1, 'Одноместный стандарт'),
(2, 'Одноместный эконом'),
(3, 'Стандарт двухместный с 2 отдельными кроватями'),
(4, 'Эконом двухместный с 2 отдельными кроватями'),
(5, '3-местный бюджет'),
(6, 'Бизнес с 1 или 2 кроватями'),
(7, 'Двухкомнатный двухместный стандарт с 1 или 2 кроватями'),
(8, 'Студия'),
(9, 'Люкс с 2 спальными кроватями');

CREATE TABLE rooms_statuses(
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(255) UNIQUE NOT NULL
);
```

```
INSERT INTO rooms_statuses(id, name) VALUES
(1, 'Чистый'),
(2, 'Грязный'),
(3, 'Назначен к уборке');
```

```
CREATE TABLE payments_statuses(
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(255) UNIQUE NOT NULL
);
```

```
INSERT INTO payments_statuses(id, name) VALUES
(1, 'Ожидает оплаты'),
(2, 'Оплата отменена'),
(3, 'Оплачено');
```

```
CREATE TABLE posts(
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(255) UNIQUE NOT NULL
);
```

```
INSERT INTO posts(id, name) VALUES
(1, 'Уборщик'),
(2, 'Администратор'),
(3, 'Управляющий');
```

```
CREATE TABLE guests(
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL
);
```

```
CREATE TABLE access_levels(
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(255) UNIQUE NOT NULL
);
```

```
INSERT INTO access_levels(id, name) VALUES
(1, 'Гость'),
(2, 'Уборщик'),
(3, 'Администратор'),
(4, 'Управляющий');
```

```
CREATE TABLE users(
    id INT PRIMARY KEY AUTO_INCREMENT,
    login VARCHAR(255) UNIQUE NOT NULL,
    password VARCHAR(255) NOT NULL,
    access_level INT NOT NULL,
    login_attempts INT NOT NULL DEFAULT 0,
    last_login DATE NULL DEFAULT NULL,
    FOREIGN KEY (access_level) REFERENCES access_levels(id) ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
INSERT INTO users(id, login, password, access_level) VALUES
(1, 'admin', 'super', 3),
(2, 'manager', 'qwerty', 4);
```

```
CREATE TABLE rooms(
    id INT PRIMARY KEY AUTO_INCREMENT,
    floor INT NOT NULL,
    number INT NOT NULL,
    category INT NOT NULL,
    status INT NOT NULL,
    FOREIGN KEY (category) REFERENCES categories(id) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (status) REFERENCES rooms_statuses(id) ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
INSERT INTO rooms(floor, number, category, status) VALUES
(1, 101, 1, 1),
(1, 102, 1, 1),
(1, 103, 2, 1),
(1, 104, 2, 1),
```

```
(1, 105, 3, 1),
(1, 106, 3, 1),
(1, 107, 4, 1),
(1, 108, 4, 1),
(1, 109, 5, 1),
(2, 201, 6, 1),
(2, 202, 6, 1),
(2, 203, 6, 1),
(2, 204, 7, 1),
(2, 205, 7, 1),
(2, 206, 7, 1),
(2, 207, 1, 1),
(2, 208, 1, 1),
(2, 209, 1, 1),
(3, 301, 8, 1),
(3, 302, 8, 1),
(3, 303, 8, 1),
(3, 304, 9, 1),
(3, 305, 9, 1),
(3, 306, 9, 1);
```

```
CREATE TABLE employees(
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL,
    post INT NOT NULL,
    FOREIGN KEY (post) REFERENCES posts(id) ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
CREATE TABLE cleanings(
    id INT PRIMARY KEY AUTO_INCREMENT,
    room INT NOT NULL,
    employee INT NOT NULL,
    date DATE NOT NULL,
    FOREIGN KEY (room) REFERENCES rooms(id) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (employee) REFERENCES employees(id) ON DELETE CASCADE ON UPDATE CASCADE
```

```
);

CREATE TABLE reservations(
    id INT PRIMARY KEY AUTO_INCREMENT,
    guest INT NOT NULL,
    room INT NOT NULL,
    payment_status INT NOT NULL,
    start DATE NOT NULL,
    end DATE,
    FOREIGN KEY (guest) REFERENCES guests(id) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (room) REFERENCES rooms(id) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (payment_status) REFERENCES payments_statuses(id) ON DELETE CASCADE ON UPDATE CASCADE
);
```

Модуль 3

Проектирование и разработка информационных систем.

Дано: Требования к разработке.pdf.

Надо: разработать формы авторизации (логин и пароль) и изменения пароля (при первом входе), сообщения пользователю об успехе и неуспехе, реализовать блокировку (если неверных попыток входа 3 или последняя активность была месяц назад).

Результат: приложение с двумя формами и логикой, подключённое к БД.

Модуль 4

Проектирование и разработка информационных систем.

Дано: api_info.pdf, TransferSimulator.exe, ТестКейс.docx, Требования к разработке.pdf, Настройка ПК для эмулятора.pdf.

Надо: разработать форму и логику для валидации ФИО гостя получаемого от Simulator.

Результат: приложение формой для валидации и логикой, подключённое к симулятору.