

Часть 1:

1. Разработать структуру базы данных для сервиса выдачи автомобилей в аренду.

Таблицы:

1.1. Клиенты (Clients):

- 1) client_id (pk, int, автоинкремент);
- 2) full_name (varchar(60));
- 3) phone_number (varchar(15));
- 4) address (varchar(60));
- 5) email (varchar(60));
- 6) client_status (varchar(15)) (блокирован, не блокирован);

1.2. Автомобили (Cars):

- 1) car_id (pk, int, автоинкремент);
- 2) car_brand (varchar(30));
- 3) car_model (varchar(30));
- 4) car_color (varchar(15));
- 5) car_mileage (int);
- 6) car_price (int);
- 7) car_status (varchar(15)) (свободно, арендовано, ТО);
- 8) category_id (fk references Categories.category_id, int);

1.3. Категории (Categories):

- 1) category_id (pk, int);
- 2) category_status (varchar(15)) (эконом, комфорт, бизнес);

1.4. Заказы (Orders):

- 1) order_id (pk, int, автоинкремент);
- 2) client_id (fk references Clients.client_id, int);
- 3) car_id (fk references Cars.car_id, int);
- 4) order_date_start (TIME);
- 5) order_date_end (TIME);
- 6) order_status (varchar(20)) (в обработке, выполняется, выполнено, отмена);
- 7) order_price (double);

1.5. Сотрудники (Employees):

- 1) employee_id (pk, int, автоинкремент);
- 2) full_name (varchar(60));
- 3) phone_number (varchar(15));
- 4) category_id (fk references Categories.category_id, int);
- 5) order_id (fk references Orders.order_id, int);

1.6. История заказов (Orders_history):

- 1) order_id (fk references Orders.order_id, int);
- 2) client_id (fk references Clients.client_id, int);

- 3) car_id (fk references Cars.car_id, int).
2. Разработать и описать (на русском языке) логику функционирования сервиса.
- 2.1. Клиент заполняет форму, указывая свои данные и предпочитаемые характеристики автомобиля;
 - 2.2. Система проверяет наличие свободных автомобилей и находит подходящий по заданным параметрам;
 - 2.3. Если подходящий автомобиль доступен для аренды, то система создаёт заказ с информацией о клиенте, автомобиле, датах аренды, стоимости заказа;
 - 2.4. Заказ попадает в таблицу заказов (Orders), где его статус меняется на «в обработке»;
 - 2.5. Сотрудник получает заказ в работу и меняет его статус на «выполняется»;
 - 2.6. Когда автомобиль возвращён, сотрудник меняет его статус на «выполнено»;
 - 2.7. Клиент может отслеживать статус своего заказа через личный кабинет или мобильное приложение;
 - 2.8. Когда заказ выполнен, клиент получает уведомление об этом.
3. Описать (на русском языке) логику основных функций и реализовать эту логику в виде SQL-скриптов:
- а) Работа с заказом: создание, закрытие, отмена и т.д.:
Создание: insert into Orders (client_id, car_id, start_date, order_price)
values (client_id_value, car_id_value, start_date_value,
order_price_value);
Закрытие: update Orders set end_date = now(), status = «выполнен»
where client_id = client_id_value;
Отмена: delete from Orders where client_id = client_id_value;
 - б) Работа с клиентом: создание, редактирование, блокировка и т.д.:
Создание: insert into Clients (full_name, phone_number, address, email,
client_status)
select (full_name_value, phone_number_value, address_value,
email_value, «не блокирован»)
from dual
where not exists (select 1 from Clients where client_id = client_id_value
and client_status = «блокирован»);
Редактирование: update Clients set full_name = full_name_value,
phone_number = phone_number_value, address = address_value, email =
email_value where client_id = client_id_value;
Блокировка: update Clients set client_status = «блокирован» where
client_id = client_id_value;

- с) Работа с авто: добавление нового, вывод из эксплуатации, перевод из статуса в статус и т.д.:

Добавление нового авто: insert into Cars (car_brand, car_model, car_color, car_mileage, car_price, car_status, category_id) values (car_brand_value, car_model_value, car_color_value, car_mileage_value, car_price_value, car_status_value, category_id_value);

Вывод авто из эксплуатации: delete from Cars where car_id = car_id_value;

Перевод авто из статуса в статус: update Cars set car_status = car_status_value where car_id = car_id_value;

- d) Сохранение и просмотр истории заказов:

Сохранение в историю заказов: insert into Orders_history (order_id, client_id, car_id) values (order_id_value, client_id_value, car_id_value);

Просмотр истории заказов: select * from Orders_history where client_id = client_id_value;

- e) Финансовый отчет: за период времени сколько и какие автомобили были сданы в аренду, на сколько дней и стоимость аренды, с итоговой стоимостью всех авто.:

```
select sum(c.car_id) as number_of_cars, date_diff(o.order_date_end,
o.order_date_start) as number_of_days, o.order_price, sum(c.car_price)
as total_car_price
from Orders_history as oh
outer join Cars as c on oh.car_id = c.car_id
outer join Orders as o on oh.order_id = o.order_id
where o.order_date_start >= order_date_start_value and
o.order_date_end <= order_date_end_value
group_by c.car_id, c.car_price;
```

- f) Отчет о состоянии автопарка: сколько и какие авто в каком состоянии находятся:

```
select count(car_id), concat(car_color, ' ', car_brand, ' ', car_model) as
car_info, car_status
from Cars
group_by car_status;
```

Часть 2:

Ссылка на видео:

https://drive.google.com/drive/folders/1ZTIdiS_SAoVAPdnMbJ6p4-8oh31TkHuO?usp=drive_link