# ОП 08. Основы проектирования баз данных.

## Общие требования:

**Задание 1.** По указанной предметной области спроектировать ERD. Определите сущности (минимум 4), атрибуты, внешние и первичные ключи. Постройте ERD диаграмму с помощью инструмента <https://app.dbdesigner.net/designer/schema/new>.

Для этого вам необходимо определить основные сущности, представить их атрибуты, свойства и связи. Представленная диаграмма должна быть выполнена в 3 нормальной форме.

Реализовать физическую модель по проектируемой концептуальной модели в PostgreSql. Создайте все необходимые таблицы и связи с помощью кода sql. А также наполните базу данными (достаточно заполнить 1 таблицу). Предоставьте файл-sql с кодом.

***Обязательно*** должны присутствовать связи между таблицами (внешние ключи), а также связь много ко многим.

**Задание 2.** Большинство запросов ориентированы на работу с базой данных по аренде dvd. База данных по аренде DVD представляет собой бизнес-процессы магазина.

* [ERD базы данных](https://www.postgresqltutorial.com/wp-content/uploads/2018/03/dvd-rental-sample-database-diagram.png)
* [Backup-выгрузка базы](https://drive.google.com/file/d/1m7RJA7t8rZEQRq5yfNqiawEx4dgHuJWZ/view?usp=sharing) для работы в pgadmin. Инструкция к восстановлению бекапа доступна [по ссылке](https://drive.google.com/file/d/1HzJr4l8Q2iKuqVo5kw6CK95t86vopl_j/view?usp=sharing).

Для варианта 7:

* [ER-диаграмма](https://drive.google.com/file/d/1F3futo16MWvPTkG65UXpraEAuQl1aGnd/view?usp=sharing)
* [sql-код для создания базы](https://drive.google.com/file/d/1OHY87pjevXsJnWTup0dD89X9IK7up_gc/view).

В базе данных проката DVD 15 таблиц:

* actor - хранит данные актеров, включая имя и фамилию.
* film - хранит данные о фильме, такие как название, год выпуска, продолжительность, рейтинг и т. д.
* film\_actor - хранит отношения между фильмами и актерами.
* category - хранит данные о категориях фильмов.
* film\_category - хранит отношения между фильмами и категориями.
* store - содержит данные о магазине, включая ответственного сотрудника и адрес менеджера.
* inventory - хранит данные инвентаризации.
* rent - хранит данные об аренде.
* payment - хранит платежи покупателя.
* staff - хранит данные о персонале.
* customer - хранит данные о клиентах.
* address - хранит адресные данные для сотрудников и клиентов
* city ​​- хранит названия городов.
* country - хранит названия стран.

# Вариант 1

**1.** Определите сущности, атрибуты, внешние и первичные ключи. Постройте ERD. Создайте все необходимые таблицы и связи с помощью кода sql. Наполните базу данными. Предоставьте файл-sql с кодом.

Предметная область: деятельность Государственной автомобильной инспекции по безопасности дорожного движения города.

**2.** Выполните следующие запросы:

1. Выведите список актеров (только имя и фамилия).

Для этого нужно воспользоваться командой

SELECT DISTINCT <название колонок с именем и фамилией> FROM <название таблицы>

1. Выведите список фильмов с указанием названия и арендной платы, а также языком фильма (словами, например, English). Выполните сортировку по возрастанию арендной платы.

SELECT <название колонок с названием фильма,платой и языком> FROM <название таблицы> ORDER BY <название колонок с арендной платой>;

1. Создайте представление, которое будет определять список клиентов базы, кто еще не делал ни одного заказа в магазине.

CREATE VIEW <название представления>

AS SELECT <поля с клиентами кто не делал заказ>

FROM <название таблицы>

WHERE <поля с клиентами кто не делал заказ>

1. Посчитайте количество заказов, которые сделали клиенты магазина. Информацию выведите в формате id заказчика, email заказчика, количество. Полю количество заказов необходимо указать имя count\_orders. Сортировка по возрастанию для id заказчика.
2. Выведите список фильмов с указанием цены аренды с учетом условия: до 0.99 – «low price», больше 1 до 2.99 – «average», и больше 2.99 – «too expensive».

«low price»

SELECT <название колонок с названием фильма,платой>

FROM <название таблицы>

WHERE <название колонок с платой> < 0.99

«average»

SELECT <название колонок с названием фильма,платой>

FROM <название таблицы>

WHERE <название колонок с платой> > 1 AND <название колонок с платой> < 2.99

«too expensive»

SELECT <название колонок с названием фильма,платой>

FROM <название таблицы>

WHERE <название колонок с платой> > 2.99

1. Создайте функцию, которая будет возвращать максимальную цену аренды фильма.

SELECT MAX(<название колонок с платой>) FROM <название таблицы>

# Вариант 2

**1.** Определите сущности, атрибуты, внешние и первичные ключи. Постройте ERD. Создайте все необходимые таблицы и связи с помощью кода sql. Наполните базу данными. Предоставьте файл-sql с кодом.

Предметная область: приемная комиссия университета.

**2.** Выполните следующие запросы:

* 1. Выведите список сотрудников (необходимые поля - имя и фамилия, email).

SELECT DISTINCT <название колонок с именем,фамилией и email> FROM <название таблицы>

* 1. Выведите список фильмов, ограничьте выборку десятью записями. Отсортируйте по названию (по алфавиту). Обратите внимание, что выполнить сортировку необходимо именно внутри изначально ограниченной выборки.

SELECT \* FROM <название таблицы> LIMIT 10 ORDER BY <название столбика с именами фильмов>

* 1. Выведите список сотрудников и указание города сотрудника (Указание города словами. Например, у сотрудника Hillyer - город Lethbridge).

SELECT <имена сотрудников и город сотрудника> FROM <название таблицы>

* 1. Выведите сотрудников и сумму, на которую они обработали платежи клиентов. Выполните сортировку по возрастанию.

SELECT <имена сотрудников и сумма> FROM <название таблицы> ORDER BY <название столбика с суммой>

* 1. Посчитайте сумму платежей клиентов, данному полю укажите название sum\_rental. Выведите список id клиентов только тех, кто сделал платежей на сумму более 100.

SELECT COUNT(<столбик с платежами>),id FROM <название таблицы>

WHERE <поле с суммой заказа> > 100

* 1. Выполните запрос для нахождения списка фильмов, рейтинг которых G. Выведите название фильма и вторая колонка, которая будет содержать строку *'Фильм демонстрируется без ограничений'*.

SELECT <название фильма> + *Фильм демонстрируется без ограничений'*. FROM <название таблицы>

WHERE <рейтинг> = «G»

# Вариант 3

**1.** Определите сущности, атрибуты, внешние и первичные ключи. Постройте ERD. Создайте все необходимые таблицы и связи с помощью кода sql. Наполните базу данными. Предоставьте файл-sql с кодом.

Предметная область: агентство недвижимости.

**2.** Выполните следующие запросы:

1. Выведите список городов, с указанием только id и названия города.

SELECT <название столбиков с id и названием городов> FROM <название таблицы>

1. Выполните выборку по нахождению городов, которые находятся в Brazil. Ограничьте выборку в 10 записей.

SELECT <название столбиков с названием городов> FROM <название таблицы> LIMIT 10

WHERE <название столбиков с названием стран> = «Brazil»

1. Создайте представление, которое будет содержать информацию по активным сотрудникам. Представление должно быть защищено от вставки записей, в которых active = 0.

CREATE VIEW <название представления>

AS SELECT <поля с клиентами кто не делал заказ>

FROM <название таблицы>

WHERE <поля активности клиентов>

1. Выведите список названий категорий и количество фильмов, которые находятся в базе магазина.

SELECT DISTINCT <название>

1. Выведите топ-10 стран, в которых наибольшее количество клиентов магазина. Необходимо вывести название страны и количество.

SELECT <название стран> FROM <название таблицы> ORDER BY <кол-во клиенов> LIMIT 10

1. Модифицируйте запрос задания 5. Добавьте отображение новой колонки status, в зависимости от количества выводит строку: >=36 - 'many', меньше - 'few'.

SELECT <название стран и колонка status>

FROM <название таблицы>

ORDER BY <кол-во клиенов> LIMIT 10

WHERE <колонки status> >=36;

# Вариант 4

**1.** Определите сущности, атрибуты, внешние и первичные ключи. Постройте ERD. Создайте все необходимые таблицы и связи с помощью кода sql. Наполните базу данными. Предоставьте файл-sql с кодом.

Предметная область: авиакомпания.

**2.** Выполните следующие запросы:

1. Выведите список стран, которые начинаются на букву “S”.

SELECT <НАЗВАНИЯ СТРАН>

FROM<СТОЛБЕЦ СО СТРАНАМИ>

WHERE <СТРОЧКА СО СТРАНАМИ> LIKE ‘S%’

1. Выведите имя, фамилию и название адреса сотрудников компании.

SELECT <ИМЯ,ФАМИЛИЯ,АДРЕС>

FROM<СТОЛБЕЦ СО СТРАНАМИ>

1. Создайте представление, в котором реализуйте запрос и получение выборки всех фильмов (доступные колонки - название фильма, год релиза и имя языка фильма).

CREATE VIEW <название представления>

AS SELECT <пназвание фильма, год релиза и имя языка фильма >

FROM <название таблицы>

1. Выведите фамилию сотрудника, второе поле будет значением MD5-хешем фамилии в шестнадцатеричном виде.
2. Посчитайте сумму платежей клиентов, данному полю укажите название sum\_rental. Выведите список id клиентов только тех, кто сделал платежей на сумму более 200. Округлите значение суммы до ближайшего целого. Отсортируйте выборку по убыванию поля sum\_rental.
3. Модифицируйте запрос задания b. Соедините имя и фамилию в одно единое поле, разделив данные пробелом. И установите название колонки - fio.

# Вариант 5

**1.** Определите сущности, атрибуты, внешние и первичные ключи. Постройте ERD. Создайте все необходимые таблицы и связи с помощью кода sql. Наполните базу данными. Предоставьте файл-sql с кодом.

Предметная область: колледж.

**2.** Выполните следующие запросы:

1. Выведите только названия городов, ограничьте выборку городов, только теми, что начинаются на “D”.

SELECT <НАЗВАНИЯ ГОРОДОВ>

FROM<СТОЛБЕЦ С ГОРОДАМИ>

WHERE <СТРОЧКА С ГОРОДАМИ> LIKE ‘D%’

1. Выведите информацию о магазинах. В выборке должны быть колонки - id магазина, имя и фамилия менеджера, адрес магазина.

SELECT ID SHOP, NAME AND FIRST NAME MANAGER, ADRESS

FROM SHOP

1. Создайте представление, которое будет содержать информацию по активным клиентам. Представление должно быть защищено от вставки записей, в которых active = 0.

CREATE VIEW ACTIV CLIENT

SELECT ACTIV CLIENT

FROM CLIENT

WHERE < discontinued = 1>

1. Выведите информацию о платежах покупателей - id и время года, когда был сделан заказ (весна, осень, ...). Колонка должна иметь название season.

SELECT ID, SEASON

FROM PAYMENT

1. Модифицируйте запрос задания d, чтобы было подсчитано количество платежей в каждом сезоне.

SELECT PAYMENT COUNT(order\_t.SalespersonID) AS TotalOrderQuantity

FROM salesperson\_t join order\_t on (ordert\_t.SalespersonID = salesperson\_t.SalespersonID )

group by order\_t.SalespersonID

1. Вычислите значения MD5-хеша следующей единой строки - id клиента, имени и номера платежа. Выполните выборку для всех платежей клиентов.



# Вариант 6

**1.** Определите сущности, атрибуты, внешние и первичные ключи. Постройте ERD. Создайте все необходимые таблицы и связи с помощью кода sql. Наполните базу данными. Предоставьте файл-sql с кодом.

Предметная область: зоопарк.

**2.** Выполните следующие запросы:

1. Подсчитайте сколько городов, название которых начинаются на букву “V”.

SELECT <НАЗВАНИЯ ГОРОДОВ>

FROM<СТОЛБЕЦ С ГОРОДАМИ>

WHERE <СТРОЧКА С ГОРОДАМИ> LIKE ‘V%’

1. Выведите список фильмов, которые относятся к категории “Travel”.

SELECT <СПИСОК КАТЕГОРИЙ ФИЛЬМОВ>

FROM<ФИЛЬМЫ>

WHERE CATEGORY !=’TRAVEL’

1. Создайте представление top5\_category\_film, которое будет выводить список пяти самых популярных категорий фильмов в базе компании.

CREATE VIEW CATEGORY FILM

SELECT CATEGORY FILM

FROM FILM

WHERE TOP<6;

1. Выведите имена первых 10 клиентов и сумму, на которую они выполнили платежи. Выполните сортировку по возрастанию, колонка с суммой должна иметь название sum\_orders.

SELECT <ИМНА КЛИЕНТЫ, СУММА ЗАКУПОК>

FROM <КЛИЕНТЫ>

WHERE <КОЛ-ВО КЛИЕНТОВ>= ‘10’

ORDER BY SUM ORDERS

1. Найдите названия городов, в которых клиентов будет больше чем 1.

SELECT cities.name,

COUNT(users.cities\_id) FROM cities

LEFT JOIN users

ON users.cities\_id = cities.id

GROUP BY cities.name

HAVING COUNT(users.cities\_id) > 1

1. Выведите список стран в верхнем регистре.

SELECT \*, UPPER(COUNTRY) AS UPPER\_COUNTRY

FROM COUNTRY

# Вариант 7

**1.** Определите сущности, атрибуты, внешние и первичные ключи. Постройте ERD. Создайте все необходимые таблицы и связи с помощью кода sql. Наполните базу данными. Предоставьте файл-sql с кодом.

Предметная область: выбирается самостоятельно.

**2.** Выполните следующие запросы:

Запросы выполняются к базе [Northind](https://drive.google.com/file/d/1OHY87pjevXsJnWTup0dD89X9IK7up_gc/view).

* Выберите все записи из таблицы customers, но только колонки имя контакта и город.

SELECT <ИМЯ КОНТАКТА И ГОРОД>

FROM customers

* Найти активные продукты из категории Beverages и Seafood, которых в продаже менее 20 единиц. Вывести наименование продуктов, кол-во единиц в продаже, имя контакта поставщика и его телефонный номер.

SELECT < НАМЕНОВАНИЕ, КОЛ-ВО, ИМЯ, НОМЕР ТЕЛЕФОНА >

FROM Beverages, Seafood

LIMIT 20

* Создать представление продуктов, которые сняты с производства (discontinued = 1) содержащее все колонки. Представление должно быть защищено от вставки записей, в которых discontinued = 0.

CREATE VIEW <название таблицы>

AS SELECT <поля с товарами снятых с производства>

FROM <название таблицы>

WHERE < discontinued = 1>

* Необходимо посчитать количество единиц товара по категориям (категория отображается названием).

SELECT category.name as category, count(\*) AS cnt

FROM product

INNER JOIN category on product.category\_id = category.category\_id

GROUP BY category.category\_id

* Выведите имя и фамилию сотрудника, у которого самое большое количество символов в поле ФИО.

**SELECT** \*

**FROM** customers

**WHERE** **LENGTH** (contact\_name) =

(**SELECT** **MAX** (**LENGTH** (contact\_name)) **FROM** customers);

* Выведите список сотрудников - имя, фамилия, и регион. Последнее поле в случае, когда равно Null, должно быть установлено значением “Неизвестно”,

**SELECT** first\_name, last\_name, **COALESCE**(region, 'unknown') **AS** region

**FROM** employees