## <https://github.com/DariaPanarina/Portfolio-SQL>

## Выбор данных

**1.Выбери столбцы id\_product, name, category, name\_store из таблицы products\_data\_all.**

SELECT

id\_product,

name,

category,

name\_store

FROM

products\_data\_all;

**2. Изучи товары и продажи магазинов в этот день. В таблице products\_data\_all есть информация, как обновлялся каталог товаров каждый день. Напиши запрос к таблице и выбери поля: название продукта (name); цена (price); название магазина (name\_store); дата (date\_upd). Сделай срез по категории (category) и дате (date\_upd). Тебе нужна категория 'молоко и сливки' и Всемирный день молока: '2019-06-01'.**

SELECT

name,

price,

name\_store,

date\_upd

FROM

products\_data\_all

WHERE

category = 'молоко и сливки' AND date\_upd ='2019-06-01';

**3. Выгрузи поля name, price, name\_store, date\_upd категории 'молоко и сливки' за 8, 15, 22 и 29 июня 2019 года. При выборке дат используй конструкцию IN.**

SELECT

name,

price,

name\_store,

date\_upd

FROM

products\_data\_all

WHERE

category ='молоко и сливки' AND date\_upd IN ('2019-06-08','2019-06-15','2019-06-22','2019-06-29');

## Агрегирующие функции

**4. Напиши запрос, который посчитает общее количество строк в таблице products\_data\_all. Назови поле cnt.**

SELECT

COUNT (\*) AS cnt

FROM

products\_data\_all;

**5. Напиши запрос, который посчитает количество строк в таблице products\_data\_all; количество строк в столбце name и количество уникальных продуктов в столбце name. Сохрани результаты в полях cnt, name\_cnt и name\_uniq\_cnt соответственно.**

SELECT

COUNT (\*) AS cnt,

COUNT (name) AS name\_cnt,

COUNT (DISTINCT name) AS name\_uniq\_cnt

FROM

products\_data\_all;

**6.Напиши запрос, который посчитает среднюю цену по всем продуктам из таблицы products\_data\_all. Стоимость указана в столбце price. Результирующее поле назови average.**

SELECT

AVG (price) AS average

FROM

products\_data\_all;

**7. Напиши запрос, который посчитает сумму стоимости всех продуктов в магазине 'Молочные вкусности'. Назови поле summa.**

SELECT

SUM (price) AS summa

FROM

products\_data\_all

WHERE

name\_store='Молочные вкусности';

**8. Напиши запрос, который посчитает разницу между максимальной и минимальной ценой продукта 'Масло топленое Ecotavush 99%, 500 г' в магазине 'ВкусМилк'. Назови поле max\_min\_diff.**

SELECT

MAX(price) - MIN(price) as max\_min\_diff

FROM

products\_data\_all

WHERE

name='Масло топленое Ecotavush 99%, 500 г' AND name\_store='ВкусМилк';

## Изменение типов

**9. Напиши запрос, который посчитает максимальный вес продукта в категории (category) «молоко и сливки». Примени конструкцию CAST AS и назови поле max\_weight.**

SELECT

MAX (CAST(weight AS real)) AS max\_weight

FROM

products\_data\_all

WHERE

category='молоко и сливки';

## Группировка данных

**10. Напиши запрос, который посчитает общее количество уникальных продуктов в каждом магазине (name\_store). Выведи название магазина, общее количество продуктов, количество уникальных продуктов. Назови поля с количеством name\_cnt и name\_uniq\_cnt соответственно.**

SELECT

name\_store,

COUNT(name) AS name\_cnt,

COUNT(DISTINCT name) AS name\_uniq\_cnt

FROM

products\_data\_all

GROUP BY

name\_store;

**11. Напиши запрос, который посчитает разницу максимальной и минимальной цены каждого продукта в категории 'масло сливочное и маргарин' на 10 июня 2019 года. Назови переменную max\_min\_diff. Переведи строковые значения дат в формат даты.**

SELECT

name,

MAX(price)-MIN(price) AS max\_min\_diff

FROM

products\_data\_all

WHERE

category = 'масло сливочное и маргарин' AND date\_upd :: date = '2019-06-10'

GROUP BY

name;

## Сортировка данных

**12. Напиши запрос, который посчитает количество товаров в каждой категории (category) на дату '2019-06-05'. Назови переменную name\_cnt и отсортируй данные по возрастанию количества товаров. Выведи дату, категорию, количество товаров. Назови выбранную дату update\_date.**

Обрати внимание: дату нужно перевести из строкового типа в date.

SELECT

category,

'2019-06-05'AS update\_date,

COUNT(name) AS name\_cnt

FROM

products\_data\_all

WHERE

date\_upd :: date = '2019-06-05'

GROUP BY

category,

update\_date

ORDER BY

name\_cnt ASC;

**13. Напиши запрос, который выведет топ-5 самых дорогих продуктов по убыванию. Выведи название продукта и стоимость. Дай переменной имя max\_price.**

SELECT

name,

MAX(price) AS max\_price

FROM

products\_data\_all

GROUP BY

name

ORDER BY

max\_price DESC

LIMIT

5;

## Отношения между таблицами

**14. Напиши запрос, который выведет:**

**номер транзакции — id\_transaction из таблицы transactions;**

**название категории — category из таблицы products;**

**название продукта — name из таблицы products.**

**Условие присоединения: значения в полях products.id\_product и transactions.id\_product равны. Названия полей результирующей таблицы: id\_transaction, category, name.**

**Выведи 10 строк. Отсортируй данные по возрастанию номера транзакции.**

SELECT

transactions.id\_transaction AS id\_transaction,

products.category AS category,

products.name AS name

FROM

transactions

INNER JOIN products ON products.id\_product = transactions.id\_product

ORDER BY

id\_transaction ASC

LIMIT

10;

**15. Выбери транзакции (покупки), при которых приобретались продукты категории 'масло сливочное и маргарин', причём только 20 июня 2019 года.**

**Из таблиц transactions и products получи:**

**день и время — date из таблицы transactions;**

**номер транзакции — id\_transaction из таблицы transactions;**

**название категории — category из таблицы products;**

**название продукта — name из таблицы products.**

**Присоедини к transactions данные таблицы products методом INNER JOIN по полю id\_product.**

**В результирующей таблице выведи четыре поля: date, id\_transaction, category, name.**

SELECT

transactions.date AS date,

transactions.id\_transaction AS id\_transaction,

products.category AS category,

products.name AS name

FROM

transactions

INNER JOIN products ON products.id\_product = transactions.id\_product

WHERE

products.category='масло сливочное и маргарин' AND CAST(transactions.date as date) = '2019-06-20';

**16. Авторы есть, а книг нет — это тревожит. А все ли продукты из products продаются в продуктовых магазинах products\_stores?**

**Напиши запрос, который выберет:**

**id\_product из таблицы products;**

**name из таблицы products;**

**id\_store из таблицы products\_stores.**

**Присоедини таблицу products\_stores к таблице products методом LEFT JOIN по полю id\_product.**

**Полям результирующей таблицы нужно дать имена: id\_product, name, id\_store.**

**SELECT**

products.id\_product AS id\_product,

products.name AS name,

products\_stores.id\_store AS id\_store

FROM

products

LEFT JOIN products\_stores ON products\_stores.id\_product=products.id\_product;

**17. Методом RIGHT JOIN выведи даты, на которые нет данных о транзакциях, но есть информация о погоде:**

**1) Получи дату (date) из таблицы weather. Обрати внимание: дату нужно привести к правильному типу конструкцией CAST.**

**2) Присоедини таблицу weather к таблице transactions методом RIGHT JOIN по полю date.**

**3) Составь срез данных в блоке WHERE: выбери только пустые даты из таблицы transactions конструкцией IS NULL.**

**4) Выведи поле date результирующей таблицы.**

SELECT

CAST(weather.date AS date) AS date

FROM

transactions

RIGHT JOIN weather ON CAST(weather.date AS date) = CAST(transactions.date AS date)

WHERE transactions.date IS NULL ;

**18. Найди информацию о погоде и названиях купленных товаров за все дни (включая те, когда покупок не было). Отсортируй данные по убыванию даты и выведи первые 30 строк.**

**Выведи дату (date), температуру (temp), наличие дождя (rain) и название товара (name).**

**Получи:**

**date из таблицы weather;**

**temp из weather;**

**rain из weather;**

**name из products.**

**Способом LEFT JOIN присоедини к weather таблицу transactions по полю date. Затем присоедини к transactions и таблицу products по полю id\_product**.

SELECT

CAST(weather.date AS date),

weather.temp,

weather.rain,

products.name

FROM

weather

LEFT JOIN transactions ON CAST(weather.date AS date) = CAST(transactions.date AS date)

LEFT JOIN products ON products.id\_product = transactions.id\_product

ORDER BY

weather.date DESC

LIMIT 30;

**19. За 5 июня 2019 года выведи номер транзакции; название магазина, где она произошла; категорию и название приобретённого продукта.**

**Выбери:**

**id\_transaction из таблицы transactions;**

**name\_store из таблицы stores;**

**category из таблицы products;**

**name из таблицы products.**

**Методом JOIN соедини таблицу transactions с таблицей products по полю id\_product; затем соедини transactions со stores по полю id\_store.**

**В блоке WHERE сделай срез по дате '2019-06-05'.**

**Выведи поля: id\_transaction, name\_store, category, name.**

SELECT

transactions.id\_transaction AS id\_transaction,

stores.name\_store AS name\_store,

products.category AS category,

products.name AS name

FROM

transactions

INNER JOIN products ON products.id\_product=transactions.id\_product

INNER JOIN stores ON stores.id\_store=transactions.id\_store

WHERE

CAST(transactions.date AS date) = '2019-06-05'