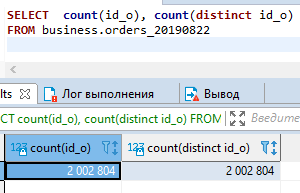
**1. Залить в свою БД данные по продажам (часть таблицы Orders в csv, исходник здесь**[**https://drive.google.com/drive/folders/1C3HqIJcABblKM2tz8vPGiXTFT7MisrML?usp=sharing**](https://drive.google.com/drive/folders/1C3HqIJcABblKM2tz8vPGiXTFT7MisrML?usp=sharing)**, интересует orders\_20190822 )**

*# проверка уникальности поля id\_o (все ли строки импортированы)*

**SELECT** **count**(id\_o), **count**(**distinct** id\_o)

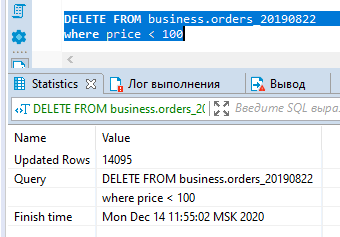
**FROM** business.orders\_20190822



*# удаление заказов с подозрительно низкой суммой. Большие суммы не трогаю – все-таки оптовики и юр. лица под вопросом*

**DELETE** **FROM** business.orders\_20190822

**where** price < 100



**2. Проанализировать, какой период данных выгружен**

*# я не смогла победить dbeaver, чтоб даты импортировались нормально. Пришлось импортировать в формате VARCHAR и преобразовывать в дату с помощью STR\_TO\_DATE*

**ALTER** **TABLE** business.orders\_20190822 **ADD** **COLUMN** new\_o\_date **DATE**

**UPDATE** business.orders\_20190822 **SET** new\_o\_date = STR\_TO\_DATE(o\_date, '%d.%m.%Y') **WHERE** new\_o\_date **IS** **NULL**

**ALTER** **TABLE** business.orders\_20190822 **DROP** o\_date

*# начальная дата и конечная дата*

**SELECT** **MIN**(new\_o\_date), **MAX**(new\_o\_date)

**FROM** business.orders\_20190822



*# все ли дни попали (731 день, т.к. 2016 год – високосный)*

**SELECT** **count**(**distinct** new\_o\_date)

**FROM** business.orders\_20190822



**3. Посчитать кол-во строк, кол-во заказов и кол-во уникальных пользователей, кот совершали заказы.**

**SELECT** **COUNT**(\*) **AS** total\_rows, **COUNT**(id\_o) **AS** total\_orders, **count**(**distinct** user\_id) **AS** uniq\_user

**FROM** business.orders\_20190822

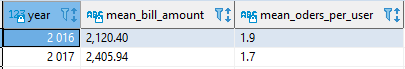


**4. По годам посчитать средний чек, среднее кол-во заказов на пользователя, сделать вывод , как изменялись это показатели Год от года.**

**SELECT** **YEAR**(new\_o\_date) **AS** year, **FORMAT**(**sum**(price)/**COUNT**(id\_o), 2) **AS** mean\_bill\_amount, **FORMAT**(**COUNT**(id\_o)/**COUNT**(**DISTINCT** user\_id), 1) **AS** mean\_oders\_per\_user

**FROM** business.orders\_20190822

**GROUP** **BY** year



**5. Найти кол-во пользователей, кот покупали в одном году и перестали покупать в следующем.**

**SELECT** **COUNT**(o1.user\_id) **AS** users\_2016\_only

**FROM**

(**SELECT** **DISTINCT** user\_id **FROM** business.orders\_20190822 **WHERE** **YEAR**(new\_o\_date) = 2016) **AS** o1

**LEFT** **JOIN**

(**SELECT** **DISTINCT** user\_id **FROM** business.orders\_20190822 **WHERE** **YEAR**(new\_o\_date) = 2017) **AS** o2

**ON** o1.user\_id = o2.user\_id

**WHERE** o2.user\_id **IS** **NULL**



**6. Найти ID самого активного по кол-ву покупок пользователя.**

**SELECT** user\_id, **COUNT**(id\_o) **AS** total\_orders

**FROM** business.orders\_20190822

**GROUP** **BY** user\_id

**ORDER** **BY** total\_orders **DESC** **LIMIT** 1



**7. Найти коэффициенты сезонности по месяцам.**

*# для своего удобства я сделала сначала представление с необходимыми данными:*

**CREATE** **VIEW** per\_month\_amount **AS**

**SELECT**

**MONTH**(o\_2016.new\_o\_date) **AS** month\_num,

o\_2016.total\_2016,

o\_2017.total\_2017

**FROM**

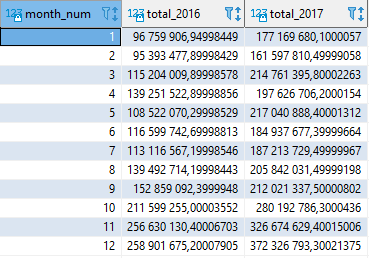
(**SELECT** new\_o\_date, **sum**(price) **AS** total\_2016 **FROM** business.orders\_20190822 **WHERE** **YEAR**(new\_o\_date) = 2016 **GROUP** **BY** **MONTH**(new\_o\_date)) **AS** o\_2016

**JOIN**

(**SELECT** new\_o\_date, **sum**(price) **AS** total\_2017 **FROM** business.orders\_20190822 **WHERE** **YEAR**(new\_o\_date) = 2017 **GROUP** **BY** **MONTH**(new\_o\_date)) **AS** o\_2017

**ON** **MONTH**(o\_2016.new\_o\_date) = **MONTH**(o\_2017.new\_o\_date)

**SELECT** \* **FROM** per\_month\_amount



*# потом использовала это представление для войны за коэффициент. Мне кажется я очень все усложнила, и долго не могла победить это помесячное деление на среднее – выводился только первый месяц. Теперь эта пятиэтажная конструкция хотя бы работает!*

**SELECT**

month\_num,

total\_2016,

total\_2016/(**SELECT** **AVG**(total\_2016) **FROM** per\_month\_amount) index\_2016,

total\_2017,

total\_2017/(**SELECT** **AVG**(total\_2017) **FROM** per\_month\_amount) index\_2017

**FROM** per\_month\_amount

