

Загрузка данных

The screenshot shows the DBeaver 25.2.4 interface with a PostgreSQL database named 'dz_02'. The left sidebar displays the database structure, including tables like 'customer', 'order_items', 'orders', and 'product'. The main editor shows a SQL script for creating a new database 'dz_02' and tables 'customer' and 'product'. The 'Statistics' tab at the bottom shows the execution details of the 'CREATE TABLE if not exists product' query.

SQL Script:

```

1  --create database if not exists dz_02;
2
3  --create database dz_02;
4
5  drop table if exists customer;
6  drop table if exists product;
7  drop table if exists orders;
8  drop table if exists order_items;
9
10 CREATE TABLE if not exists customer (
11     customer_id int primary key,
12     first_name varchar(64),
13     last_name varchar(64),
14     gender varchar(10),
15     DOB date,
16     job_title varchar(64),
17     job_industry_category varchar(64),
18     wealth_segment varchar(32),
19     deceased_indicator char(1),
20     owns_car varchar(3),
21     address varchar(64),
22     postcode int,
23     state varchar(64),
24     country varchar(64),
25     property_valuation int
26 );
27
28 CREATE TABLE if not exists product (
29     product_id int,
30     brand varchar(64),

```

Statistics:

Name	Value
Updated Rows	0
Execute time	0.0s
Start time	Tue Nov 18 17:38:35 MSK 2025
Finish time	Tue Nov 18 17:38:35 MSK 2025
Query	CREATE TABLE if not exists product (product_id int, brand varchar(64), product_line varchar(64), product_class varchar(64), product_size varchar(64), list_price decimal(10,2), standard_cost decimal(10,2)

DBaver 25.2.4 - customer

Файл Редактирование Навигация Поиск Редактор SQL База данных Окна Справка

SQL Commit Rollback Auto postgres public@dz_02

Базы данных Проекты

Фильтровать соединения по имени

DBaver Sample Database (SQLite)

postgres localhost:5432

Базы данных

dz_01

dz_02

Схемы

public

Таблицы

customer 752K

Колонки

Ограничения

Внешние ключи

Индексы

Зависимости

Ссылки

Секции таблиц

Триггеры

Правила

Policies

order_items 1,2M

orders 1,6M

product 40K

Внешние таблицы

Представления

Мат. представления

Индексы

Функции

Последовательности

Типы данных

Агрегатные функции

Событийные триггеры

Расширения

Хранилище

Системные объекты

Роли

postgres

Администрирование

Системные объекты

Свойства Данные Диаграмма

Просмотр SQL Введите SQL выражение чтобы отфильтровать результаты

	customer_id	first_name	last_name	gender	dob	job_title	job_department
1	1	Laraine	Medendorp	F	1953-10-12	Executive Secretary	Health
2	2	Eli	Bockman	Male	1980-12-16	Administrative Officer	Financ
3	3	Arlin	Dearle	Male	1954-01-20	Recruiting Manager	Proper
4	4	Talbot		Male	1961-10-03		IT
5	5	Sheila-kathryn	Calton	Female	1977-05-13	Senior Editor	n/a
6	6	Curr	Duckhouse	Male	1966-09-16		Retail
7	7	Fina	Merali	Female	1976-02-23		Financ
8	8	Rod	Inder	Male	1962-03-30	Media Manager I	n/a
9	9	Mala	Lind	Female	1973-03-10	Business Systems Development Analyst	Argicu
10	10	Fiorenze	Birdall	Female	1988-10-11	Senior Quality Engineer	Financ
11	11	Uriah	Bisatt	Male	1954-04-30		Proper
12	12	Sawyer	Flattman	Male	1994-07-21	Nuclear Power Engineer	Manuf
13	13	Gabriele	Norcross	Male	1955-02-15	Developer I	Financ
14	14	Rayshell	Kittelman	Female	1983-03-25	Account Executive	Financ
15	15	Erroll	Radage	Male	2000-07-13	Junior Executive	Manuf
16	16	Harlin	Parr	Male	1977-02-27	Media Manager IV	n/a
17	17	Heath	Faraday	Male	1962-03-19	Sales Associate	n/a
18	18	Marjie	Neasham	Female	1967-07-06	Professor	n/a
19	19	Sorcha	Keyson	Female	2001-04-15	Geological Engineer	Manuf
20	20	Basile	Firth	Male	1980-08-13	Project Manager	Manuf
21	21	Mile	Cammocke	Male	1980-09-20	Safety Technician I	Manuf
22	22	Deeanne	Durtnell	Female	1962-12-10		IT
23	23	Olav	Polak	Male	1995-02-10		n/a
24	24	Kim	Skpsey	Female	1977-12-03	Research Assistant I	Argicu
25	25	Geoff	Assaf	Male	1976-12-02	Accounting Assistant III	Financ
26	26	Trixi	Ginnelly	Female	1978-06-10	Editor	Financ
27	27	Garvin	Klees	Male	1978-09-25	Research Nurse	Health
28	28	Fee	Zellmer	Male	1973-09-30	Senior Quality Engineer	Health
29	29	Mona	Sancraft	Female	1968-06-22	Safety Technician III	Manuf
30	30	Darrick	Helleckas	Male	1961-10-18		IT
31	31	Star	Praton	Female	1962-11-24	Staff Accountant III	Teleco
32	32	Marion	Vanichkin	Female	1995-04-20	Legal Assistant	Manuf

Обновить Save Cancel Экспорт данных ... 200 200+

200 строк получено - 0.0s (0.0s получ.), 2025-11-18 в 17:35:57

postgres dz_02 public customer MSK ru

Temps to drop Thursday

Поиск

17:44 18.11.2025

1. Вывести распределение (количество) клиентов по сферам деятельности, отсортировав результат по убыванию количества.

```

242
243 -- Домашнее задание №3.
244 -- №1.
245 --Вывести распределение (количество) клиентов по сферам деятельности,
246 --отсортировав результат по убыванию количества.
247 select
248     job_industry_category,
249     count(*) as customers_count
250 from
251     customer
252 group by
253     job_industry_category
254 order by
255     customers_count desc;
256

```

customer 1 x

select job_industry_category, cou | Введите SQL выражение чтобы отфильтровать результаты

	AZ job_industry_category	123 customers_count
1	Manufacturing	799
2	Financial Services	774
3	n/a	656
4	Health	602
5	Retail	358
6	Property	267
7	IT	223
8	Entertainment	136
9	Argiculture	113
10	Telecommunications	72

2. Найти общую сумму дохода ($list_price * quantity$) по всем подтвержденным заказам за каждый месяц по сферам деятельности клиентов. Отсортировать результат по году, месяцу и сфере деятельности.

```

262
263 ● select
264     extract (year from o.order_date) as year,
265     extract (month from o.order_date) as month,
266     c.job_industry_category,
267     sum(p_c.list_price*o_i.quantity) as total_revenue
268 from customer c
269 join orders o on c.customer_id = o.customer_id
270 join order_items o_i on o.order_id = o_i.order_id
271 join product_cor p_c on o_i.product_id = p_c.product_id
272 where o.order_status = 'Approved'
273 group by year, month, c.job_industry_category
274 order by year, month, c.job_industry_category;
275
276

```

customer 1 X

select extract (year from o.order_date) as year, extract (month from o.order_date) as month, c.job_industry_category, sum(p_c.list_price*o_i.quantity) as total_revenue from customer c join orders o on c.customer_id = o.customer_id join order_items o_i on o.order_id = o_i.order_id join product_cor p_c on o_i.product_id = p_c.product_id where o.order_status = 'Approved' group by year, month, c.job_industry_category order by year, month, c.job_industry_category;

	123 year	123 month	A-Z job_industry_category	123 total_revenue
1	2 017	1	Argiculture	299 741,43
2	2 017	1	Entertainment	416 965,33
3	2 017	1	Financial Services	2 443 496,22
4	2 017	1	Health	1 921 608,44
5	2 017	1	IT	735 074,61
6	2 017	1	Manufacturing	2 413 462,43
7	2 017	1	n/a	2 278 578
8	2 017	1	Property	575 902,88
9	2 017	1	Retail	1 217 372,32
10	2 017	1	Telecommunications	211 919,63
11	2 017	2	Argiculture	420 197,65
12	2 017	2	Entertainment	399 603,17
13	2 017	2	Financial Services	2 541 873,95
14	2 017	2	Health	1 811 851,88
15	2 017	2	IT	699 719,96
16	2 017	2	Manufacturing	2 807 982,8
17	2 017	2	n/a	1 781 138,61
18	2 017	2	Property	870 581,34
19	2 017	2	Retail	1 098 580,25
20	2 017	2	Telecommunications	237 096,43
21	2 017	3	Argiculture	320 236,94
22	2 017	3	Entertainment	488 610,62
23	2 017	3	Financial Services	2 305 750,75
24	2 017	3	Health	1 909 695,04
25	2 017	3	IT	637 157,44
26	2 017	3	Manufacturing	2 680 021,33
27	2 017	3	n/a	2 004 002,22
28	2 017	3	Property	864 363,39
29	2 017	3	Retail	1 163 784,73
30	2 017	3	Telecommunications	193 538,84
31	2 017	4	Argiculture	334 741,6
32	2 017	4	Entertainment	445 760,41

Обновить Save Cancel Экспорт данных ... 200

120 ... 120 строк получено - 0.0s, 2025-11-30 в 15:16:10

3. Вывести количество уникальных онлайн-заказов для всех брендов в рамках подтвержденных заказов клиентов из сферы IT. Включить бренды, у которых нет онлайн-заказов от IT-клиентов, — для них должно быть указано количество 0.

```

276
277 -- №3.
278 --Вывести количество уникальных онлайн-заказов для всех брендов
279 --в рамках подтвержденных заказов клиентов из сферы IT.
280 --Включить бренды, у которых нет онлайн-заказов от IT-клиентов,
281 --- для них должно быть указано количество 0.
282
283 select
284     p_c.brand,
285     count(distinct o.order_id) as online_order_brand
286 from product_cor p_c
287 left join order_items o_i on p_c.product_id = o_i.product_id
288 left join orders o on o_i.order_id = o.order_id
289 left join customer c
290     on o.customer_id = c.customer_id
291     and c.job_industry_category = 'IT'
292     and o.order_status = 'Approved'
293     and o.online_order = 'True'
294 group by p_c.brand
295 order by online_order_brand desc, p_c.brand;
296
297
298
299

```

product_cor 1 ×

select p_c.brand, count(distinct o.order_id) as online_order_brand from product_cor p_c left join order_items o_i on p_c.product_id = o_i.product_id left join orders o on o_i.order_id = o.order_id left join customer c on o.customer_id = c.customer_id and c.job_industry_category = 'IT' and o.order_status = 'Approved' and o.online_order = 'True' group by p_c.brand order by online_order_brand desc, p_c.brand;

AZ brand	123 online_order_brand
OHM Cycles	3 899
Solex	3 885
Giant Bicycles	3 755
WeareA2B	3 685
Trek Bicycles	2 568
Norco Bicycles	2 208

4. Найти по всем клиентам: сумму всех заказов (общего дохода), максимум, минимум и количество заказов, а также среднюю сумму заказа по каждому клиенту. Отсортировать результат по убыванию суммы всех заказов и количества заказов. Выполнить двумя способами: используя только GROUP BY и используя только оконные функции. Сравнить результат.

```

select
    c.first_name,
    c.last_name,
    sum(o_i.item_list_price_at_sale) as total_revenue,
    max(o_i.item_list_price_at_sale) as max_order_amount,
    min(o_i.item_list_price_at_sale) as min_order_amount,
    count(distinct o.order_id) as orders_count,
    avg(o_i.item_list_price_at_sale) as avg_order_amount
from
    customer c
    join orders o on c.customer_id = o.customer_id
    join order_items o_i on o.order_id = o_i.order_id
group by
    c.customer_id,
    c.first_name,
    c.last_name
order by
    total_revenue desc,
    orders_count desc;

```

customer 1

select c.first_name, c.last_name, sum(o_i.item_list_price_at_sale) as total_revenue, max(o_i.item_list_price_at_sale) as max_order_amount, min(o_i.item_list_price_at_sale) as min_order_amount, count(distinct o.order_id) as orders_count, avg(o_i.item_list_price_at_sale) as avg_order_amount

	AZ first_name	AZ last_name	total_revenue	max_order_amount	min_order_amount	orders_count	avg_order_amount
1	Jillie	Fyndon	19 071,32	2 005,66	230,91	14	1 362,23714285
2	Hercule		18 349,27	1 992,93	290,62	13	1 411,48230769
3	Jeffry	Slowly	18 052,68	2 091,47	360,4	12	1 504,
4	Tye	Doohan	17 898,46	2 091,47	1 057,51	10	1 789,8
5	Melantha	Pickburn	17 258,94	2 083,94	183,86	11	1 568,99454545
6	Raffaello	Godleman	17 160,24	2 005,66	183,86	12	1 430,
7	Kynthia	Purcer	17 133,93	2 091,47	688,63	11	1 557,
8	Ericka	Eggers	17 035,83	1 977,36	71,16	13	1 310,44846153
9	Deana	Rathbourne	16 199,24	2 083,94	183,86	13	1 246,09538461
10	Herc	McIlhone	16 122,34	2 091,47	290,62	12	1 343,52833333
11	Ammamaria	Standbridge	15 836	2 091,47	586,45	10	1 583,

```

327 with order_level as (
328     select
329         c.customer_id,
330         c.first_name,
331         c.last_name,
332         o.order_id,
333         sum(o_i.item_list_price_at_sale) as order_amount
334     from customer c
335     join orders o on c.customer_id = o.customer_id
336     join order_items o_i on o.order_id = o_i.order_id
337     group by
338         c.customer_id, c.first_name, c.last_name, o.order_id
339 ),
340 customer_agr as (
341     select
342         o_l.customer_id,
343         o_l.first_name,
344         o_l.last_name,
345         o_l.order_id,
346         o_l.order_amount,
347         sum(o_l.order_amount) over (partition by o_l.customer_id) as total_revenue,
348         max(o_l.order_amount) over (partition by o_l.customer_id) as max_order_amount,
349         min(o_l.order_amount) over (partition by o_l.customer_id) as min_order_amount,
350         count(o_l.order_amount) over (partition by o_l.customer_id) as orders_count,
351         avg(o_l.order_amount) over (partition by o_l.customer_id) as avg_order_amount
352     from order_level o_l
353 )
354 select distinct
355     c_a.customer_id,
356     c_a.first_name,
357     c_a.last_name,
358     c_a.total_revenue,
359     c_a.max_order_amount,
360     c_a.min_order_amount,
361     c_a.orders_count,
362     c_a.avg_order_amount
363 from customer_agr c_a
364 order by
365     total_revenue desc,
366     orders_count desc;

```

	AZ first_name	AZ last_name	123 total_revenue	123 max_order_amount	123 min_order_amount	123 orders_count	123 avg_order_amount
1	Jillie	Fyndon	19 071,32	2 005,66	230,91	14	1 362,23714285
2	Hercule		18 349,27	1 992,93	290,62	13	1 411,48230769
3	Jeffry	Slowly	18 052,68	2 091,47	360,4	12	1 504,
4	Tye	Doohan	17 898,46	2 091,47	1 057,51	10	1 789,8
5	Melantha	Pickburn	17 258,94	2 083,94	183,86	11	1 568,99454545
6	Raffaello	Godleman	17 160,24	2 005,66	183,86	12	1 430,
7	Kynthia	Purcer	17 133,93	2 091,47	688,63	11	1 557,
8	Ericka	Eggers	17 035,83	1 977,36	71,16	13	1 310,44846153
9	Deana	Rathbourne	16 199,24	2 083,94	183,86	13	1 246,09538461
10	Herc	McIlhone	16 122,34	2 091,47	290,62	12	1 343,52833333
11	Ammamaria	Standbridge	15 826	2 091,47	586,45	10	1 58,
12	Glynnis	Sailor	15 447,92	1 977,36	230,91	12	1 287,32666666
13	Barrett	Lindley	15 370,81	2 091,47	569,56	9	1 707,86777777
14	Delores	Ashcroft	15 091,91	2 005,66	544,05	11	1 371,99181818
15	Rozamond	Franceschino	15 071,26	1 997,68	183,86	10	1 507,1
16	Keeley	Kruger	14 949,91	1 810	441,49	12	1 245,82583333

Результирующие таблицы идентичны

5. Найти имена и фамилии клиентов с топ-3 минимальной и топ-3 максимальной суммой транзакций за весь период (учесть клиентов, у которых нет заказов, приняв их сумму транзакций за 0).

```
-- №5.
--Найти имена и фамилии клиентов с топ-3 минимальной
--и топ-3 максимальной суммой транзакций за весь период
--(учесть клиентов, у которых нет заказов, приняв их сумму транзакций за 0).

with customer_sum as (
    select
        c.customer_id,
        c.first_name,
        c.last_name,
        sum(oi.item_list_price_at_sale) as total_revenue
    from customer c
    left join orders o      on c.customer_id = o.customer_id
    left join order_items oi on o.order_id   = oi.order_id
    group by
        c.customer_id, c.first_name, c.last_name
),
ranked as (
    select
        customer_id,
        first_name,
        last_name,
        total_revenue,
        rank() over (order by total_revenue asc nulls last) as r_min,
        rank() over (order by total_revenue desc nulls last) as r_max
    from customer_sum
)
select
    customer_id,
    first_name,
    last_name,
    total_revenue
from ranked
where r_min <= 3
     or r_max <= 3
order by total_revenue, customer_id;
```

customer 1 X

WITH customer_sum AS (SELECT c.customer_id, ... Введите SQL выражение чтобы отфильтровать результаты

	123 customer_id	A-Z first_name	A-Z last_name	123 total_revenue
1	3 292	Hamlen	Slograve	60,34
2	2 532	Milli	Hubbert	71,49
3	2 274	Nada	Reinert	142,98
4	1 597	Jeffry	Slowly	18 052,68
5	1 129	Hercule		18 349,27
6	2 183	Jillie	Fyndon	19 071,32

6. Вывести только вторые транзакции клиентов (если они есть) с помощью оконных функций. Если у клиента меньше двух транзакций, он не должен попасть в результат.

```
-- #6.
--Вывести только вторые транзакции клиентов (если они есть) с помощью оконных функций.
--Если у клиента меньше двух транзакций, он не должен попасть в результат.
with ordered_orders as (
    select
        c.customer_id,
        c.first_name,
        c.last_name,
        o.order_id,
        o.order_date,
        row_number() over (
            partition by c.customer_id
            order by o.order_date
        ) as row_num
    from customer c
    join orders o on c.customer_id = o.customer_id
)
select
    customer_id,
    first_name,
    last_name,
    order_id,
    order_date
from ordered_orders
where row_num = 2;
```

	123 customer_id	AZ first_name	AZ last_name	123 order_id	order_date
1	1	Laraine	Medendorp	13 424	2017-02-21
2	2	Eli	Bockman	6 743	2017-06-11
3	3	Arlin	Dearle	15 188	2017-03-24
4	4	Talbot		14 648	2017-06-18
5	5	Sheila-kathryn	Calton	19 993	2017-04-28
6	6	Curr	Duckhouse	8 204	2017-02-06
7	7	Fina	Merali	18 549	2017-02-24
8	8	Rod	Inder	19 844	2017-01-28
9	9	Mala	Lind	2 979	2017-03-06
10	10	Fiorenze	Birdall	10 250	2017-07-13
11	11	Uriah	Bisatt	16 846	2017-06-02
12	12	Sawyer	Flattman	12 242	2017-07-23
13	13	Gabriele	Norcross	8 905	2017-02-16
14	14	Rayshell	Kittelman	8 486	2017-08-16
15	15	Erroll	Radage	434	2017-03-10
16	16	Harlin	Parr	5 083	2017-05-10
17	17	Heath	Faraday	10 775	2017-05-01
18	18	Marjie	Neasham	3 777	2017-05-11
19	19	Sorcha	Keyson	14 850	2017-03-25

7. Вывести имена, фамилии и профессии клиентов, а также длительность максимального интервала (в днях) между двумя последовательными заказами. Исключить клиентов, у которых только один или меньше заказов.

```

432 -- №7.
433 -- Вывести имена, фамилии и профессии клиентов,
434 -- а также длительность максимального интервала (в днях) между двумя последовательными заказами
435 -- Исключить клиентов, у которых только один или меньше заказов.
436 with ordered_orders as (
437     select
438         c.customer_id,
439         c.first_name,
440         c.last_name,
441         c.job_title,
442         o.order_date,
443         lag(o.order_date) over (
444             partition by c.customer_id
445             order by o.order_date
446         ) as prev_order_date
447     from customer c
448     join orders o on c.customer_id = o.customer_id
449 ),
450 intervals as (
451     select
452         customer_id,
453         first_name,
454         last_name,
455         job_title,
456         (order_date - prev_order_date) as diff_days
457     from ordered_orders
458     where prev_order_date is not null
459 )
460 select
461     customer_id,
462     first_name,
463     last_name,
464     job_title,
465     max(diff_days) as max_interval_days
466 from intervals
467 group by
468     customer_id,
469     first_name,
470     last_name,
471     job_title
472 order by
473     max_interval_days desc,
474     customer_id;
475

```

	customer_id	first_name	last_name	job_title	max_interval_days
1	1 584	Susanetta		Legal Assistant	357
2	1 810	Royall	Terris	Geological Engineer	330
3	2 128	Gregorius	Cockram	Data Coordinator	330
4	3 316	Stoddard	Giacomoni	Structural Analysis Engineer	330
5	3 156	Bearnard	Letixier		329
6	3 222	Caralie	Sellors	Senior Editor	321
7	335	Debee	Martynov	Senior Editor	320
8	316	Genni	Larway	Environmental Specialist	314
9	2 085	Carolynn	Samsin	Pharmacist	310
10	2 146	Timmie	Lenden		310
11	3 024	Franz	Craddy		310
12	92	Jodee	Judkins	Recruiting Manager	306
13	1 633	Ashia	Muzzi	Mechanical Systems Engineer	306
14	2 586	Heywood	Sollett	Tax Accountant	305
15	2 541	Cleveland	Islep	Software Engineer II	299
16	510	Sheilah	Blackmore		297
17	520	Jazmin	Neumann	VP Quality Control	297
18	2 108	Michel	O'Halligan	Software Test Engineer IV	297
19	3 159	Jesus	MacShirie		296
20	2 172	Emmery	Angrock	Information Systems Manager	292
21	2 329	Donny	Stiven	Computer Systems Analyst II	292
22	2 122	Osborne	Nawton	Registered Nurse	291
23	2 500	Micky	Living	Senior Developer	291
24	2 106	Antons	Ley	Project Manager	290
25	2 218	Madelena	Blincoe	Senior Sales Associate	286

8. Найти топ-5 клиентов (по общему доходу) в каждом сегменте благосостояния (wealth_segment). Вывести имя, фамилию, сегмент и общий доход. Если в сегменте менее 5 клиентов, вывести всех.

```
-- №8.
-- Найти топ-5 клиентов (по общему доходу) в каждом сегменте благосостояния (wealth_segment).
-- Вывести имя, фамилию, сегмент и общий доход. Если в сегменте менее 5 клиентов, вывести всех.
with customer_revenue as (
    select
        c.customer_id,
        c.first_name,
        c.last_name,
        c.wealth_segment,
        sum(o_i.item_list_price_at_sale) as total_revenue
    from customer c
    join orders o on c.customer_id = o.customer_id
    join order_items o_i on o.order_id = o_i.order_id
    group by
        c.customer_id,
        c.first_name,
        c.last_name,
        c.wealth_segment
),
ranked as (
    select
        customer_id,
        first_name,
        last_name,
        wealth_segment,
        total_revenue,
        rank() over (
            partition by wealth_segment
            order by total_revenue desc
        ) as rnk
    from customer_revenue
)
select
    customer_id,
    first_name,
    last_name,
    wealth_segment,
    total_revenue
from ranked
where rnk <= 5
order by
    wealth_segment,
    rnk,
    customer_id;
```

	customer_id	first_name	last_name	wealth_segment	total_revenue
1	1 597	Jeffrey	Slowly	Affluent Customer	18 052,68
2	941	Tye	Doohan	Affluent Customer	17 898,46
3	2 788	Melantha	Pickburn	Affluent Customer	17 258,94
4	1 887	Kynthia	Purcer	Affluent Customer	17 133,93
5	1 302	Ericka	Eggers	Affluent Customer	17 035,83
6	936	Raffaello	Godleman	High Net Worth	17 160,24
7	1 103	Glynnis	Sailor	High Net Worth	15 447,92
8	999	Dido	Leyburn	High Net Worth	14 662,14
9	1 460	Morley	Shutt	High Net Worth	14 633,24
10	2 476	Hal	Braddon	High Net Worth	14 578,69
11	2 183	Jillie	Fyndon	Mass Customer	19 071,32
12	1 129	Hercule		Mass Customer	18 349,27
13	1 140	Deana	Rathbourne	Mass Customer	16 199,24
14	1 317	Barrett	Lindley	Mass Customer	15 370,81
15	2 762	Rozamond	Franceschino	Mass Customer	15 071,26