Витрина RFM

1. Требования к витрине.
2. Создать витрину данных в схеме analysis
3. Имя таблицы – dm\_rfm\_segments
4. Структура витрины следующая:

* user\_id
* recency (число от 1 от 5)
* frequency (число от 1 до 5)
* monetary\_value (число от 1 до 5)

1. Данные нужны, начиная с начала 2022 года
2. Обновление данных не требуется
3. Данные нужны только с успешных заказов (статус заказа Closed)
4. Структура исходных данных.

Для RFM-анализа по выполненным заказам, необходимы таблицы orders (необходимые столбцы: order\_id, order\_ts, payment, user\_id, status), orderstatuses (необходимые столбцы: id, key), users(id).

По столбцам orders(order\_id) и orderstatuses(id) происходит объединение таблиц.

Recency считается по order\_ts.

Frequency считается по user\_id.

Monetary\_value считается по payment.

1. Качество данных.

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица | Объект | Анализ | Вывод |
| production.  orders | order\_id | PRIMARY KEY | Повторений данных нет |
| order\_ts | NOT NULL | Отсутствие пустых записей |
| MIN - 2022-02-12 02:41:28.000  MAX - 2022-03-14 02:38:26.000 | Все данные лежит в нужном временном диапазоне |
| payment | NOT NULL | Отсутствие пустых записей |
| MIN – 60  MAX - 6360 | Отсутствуют нулевые данные |
| user\_id | NOT NULL | Отсутствие пустых записей |
| status | NOT NULL | Отсутствие пустых записей |
| MIN – 4  MAX - 5 | Отсутствие нулевых записей |
| production.  orderstatuses | id | PRIMARY KEY | Повторений данных нет |
| key | NOT NULL | Отсутствие пустых записей |

1. Скрипты создания
   1. Создание изображений в analysis (views.sql):

**CREATE** **OR** **REPLACE** **VIEW** analysis.orderitems **AS**

**SELECT** \*

**FROM** production.orderitems;

**CREATE** **OR** **REPLACE** **VIEW** analysis.orders **AS**

**SELECT** \*

**FROM** production.orders;

**CREATE** **OR** **REPLACE** **VIEW** analysis.orderstatuses **AS**

**SELECT** \*

**FROM** production.orderstatuses;

**CREATE** **OR** **REPLACE** **VIEW** analysis.products **AS**

**SELECT** \*

**FROM** production.products;

**CREATE** **OR** **REPLACE** **VIEW** analysis.users **AS**

**SELECT** \*

**FROM** production.users;

* 1. Создание витрины (datamart\_ddl.sql)

**CREATE** **TABLE** analysis.dm\_rfm\_segments(

user\_id **int4** **NOT** **NULL**,

recency **smallint** **NOT** **NULL**,

frequency **smallint** **NOT** **NULL**,

monetary\_value **smallint** **NOT** **NULL**,

**CONSTRAINT** dm\_rfm\_segments\_recency\_check **CHECK** (recency >= 1 **AND** recency <= 5),

**CONSTRAINT** dm\_rfm\_segments\_frequency\_check **CHECK** (frequency >= 1 **AND** frequency <= 5),

**CONSTRAINT** dm\_rfm\_segments\_monetary\_value\_check **CHECK** (monetary\_value >= 1 **AND** monetary\_value <= 5)

)

* 1. Создание отдельных таблиц показателей

**CREATE** **TABLE** analysis.tmp\_rfm\_recency (

user\_id **INT** **NOT** **NULL** **PRIMARY** **KEY**,

recency **INT** **NOT** **NULL** **CHECK**(recency >= 1 **AND** recency <= 5)

);

**CREATE** **TABLE** analysis.tmp\_rfm\_frequency (

user\_id **INT** **NOT** **NULL** **PRIMARY** **KEY**,

frequency **INT** **NOT** **NULL** **CHECK**(frequency >= 1 **AND** frequency <= 5)

);

**CREATE** **TABLE** analysis.tmp\_rfm\_monetary\_value (

user\_id **INT** **NOT** **NULL** **PRIMARY** **KEY**,

monetary\_value **INT** **NOT** **NULL** **CHECK**(monetary\_value >= 1 **AND** monetary\_value <= 5)

);

* 1. Заполнение таблицы с показателем recency (tmp\_rfm\_recency.sql)

**INSERT** **INTO** analysis.tmp\_rfm\_recency

**WITH** orders\_with\_status **AS**(

**SELECT** user\_id, order\_ts, **key**,

**ROW\_NUMBER**() **OVER**(**PARTITION** **BY** user\_id **ORDER** **BY** order\_ts **DESC**) user\_number

**FROM** analysis.orders o

**JOIN** analysis.orderstatuses os **ON** o.status = os.id

**WHERE** **key** = **'Closed'** **AND** **date\_part**(**'year'**, order\_ts) >= 2022

), last\_order **AS** (

**SELECT** \*,

**ROW\_NUMBER**() **OVER**(**ORDER** **BY** order\_ts) **as** order\_ts\_number

**FROM** orders\_with\_status

**WHERE** user\_number = 1

)

**SELECT** user\_id, (

**CASE**

**WHEN** order\_ts\_number <= **CEIL**((**SELECT** **MAX**(order\_ts\_number) **FROM** last\_order) / 5) **THEN** 1

**WHEN** order\_ts\_number <= **CEIL**((**SELECT** **MAX**(order\_ts\_number) **FROM** last\_order) \* 2 / 5) **THEN** 2

**WHEN** order\_ts\_number <= **CEIL**((**SELECT** **MAX**(order\_ts\_number) **FROM** last\_order) \* 3 / 5) **THEN** 3

**WHEN** order\_ts\_number <= **CEIL**((**SELECT** **MAX**(order\_ts\_number) **FROM** last\_order) \* 4 / 5) **THEN** 4

**ELSE** 5

**END**) **as** recency

**FROM** last\_order

* 1. Заполнение таблицы с показателем frequency (tmp\_rfm\_frequency.sql)

**INSERT** **INTO** analysis.tmp\_rfm\_frequency

**WITH** orders\_with\_amount **AS**(

**SELECT** user\_id,

**COUNT**(user\_id) **as** amount

**FROM** analysis.orders o

**JOIN** analysis.orderstatuses os **ON** o.status = os.id

**WHERE** **key** = **'Closed'** **AND** **date\_part**(**'year'**, order\_ts) >= 2022

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

), orders\_with\_amount\_and\_numbers **AS** (

**SELECT** \*,

**ROW\_NUMBER**() **OVER**(**ORDER** **BY** amount) **as** amount\_number

**FROM** orders\_with\_amount

)

**SELECT** user\_id, (

**CASE**

**WHEN** amount\_number <= **CEIL**((**SELECT** **MAX**(amount\_number) **FROM** orders\_with\_amount\_and\_numbers) / 5) **THEN** 1

**WHEN** amount\_number <= **CEIL**((**SELECT** **MAX**(amount\_number) **FROM** orders\_with\_amount\_and\_numbers) \* 2 / 5) **THEN** 2

**WHEN** amount\_number <= **CEIL**((**SELECT** **MAX**(amount\_number) **FROM** orders\_with\_amount\_and\_numbers) \* 3 / 5) **THEN** 3

**WHEN** amount\_number <= **CEIL**((**SELECT** **MAX**(amount\_number) **FROM** orders\_with\_amount\_and\_numbers) \* 4 / 5) **THEN** 4

**ELSE** 5

**END**) **as** frequency

**FROM** orders\_with\_amount\_and\_numbers

* 1. Заполнение таблицы с показателем monetary\_value (tmp\_rfm\_monetary\_value.sql)

**INSERT** **INTO** analysis.tmp\_rfm\_monetary\_value

**WITH** orders\_with\_sum **AS**(

**SELECT** user\_id,

**SUM**(payment) **as** payment

**FROM** analysis.orders o

**JOIN** analysis.orderstatuses os **ON** o.status = os.id

**WHERE** **key** = **'Closed'** **AND** **date\_part**(**'year'**, order\_ts) >= 2022

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

), orders\_with\_sum\_and\_numbers **AS** (

**SELECT** \*,

**ROW\_NUMBER**() **OVER**(**ORDER** **BY** payment) **as** amount\_payment

**FROM** orders\_with\_sum

)

**SELECT** user\_id, (

**CASE**

**WHEN** amount\_payment <= **CEIL**((**SELECT** **MAX**(amount\_payment) **FROM** orders\_with\_sum\_and\_numbers) / 5) **THEN** 1

**WHEN** amount\_payment <= **CEIL**((**SELECT** **MAX**(amount\_payment) **FROM** orders\_with\_sum\_and\_numbers) \* 2 / 5) **THEN** 2

**WHEN** amount\_payment <= **CEIL**((**SELECT** **MAX**(amount\_payment) **FROM** orders\_with\_sum\_and\_numbers) \* 3 / 5) **THEN** 3

**WHEN** amount\_payment <= **CEIL**((**SELECT** **MAX**(amount\_payment) **FROM** orders\_with\_sum\_and\_numbers) \* 4 / 5) **THEN** 4

**ELSE** 5

**END**) **as** monetary\_value

**FROM** orders\_with\_sum\_and\_numbers

* 1. Заполнение витрины данных (datamart\_query.sql)

**INSERT** **INTO** analysis.dm\_rfm\_segments

**SELECT** \*

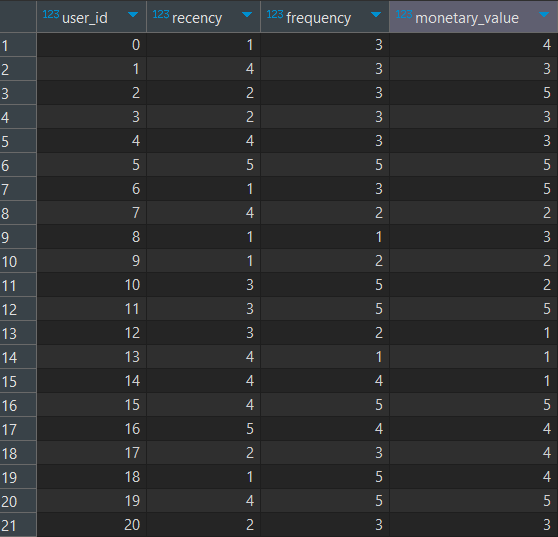
**FROM** analysis.tmp\_rfm\_recency

**JOIN** analysis.tmp\_rfm\_frequency **USING**(user\_id)

**JOIN** analysis.tmp\_rfm\_monetary\_value **USING**(user\_id)

**ORDER** **BY** user\_id;

Вид итоговой витрины:



1. Изменение представления orders, в связи с введением новой таблицы production.OrderStatusLog (orders\_view.sql)

**CREATE** **OR** **REPLACE** **VIEW** analysis.orders **AS**

**SELECT** order\_id, order\_ts, user\_id, bonus\_payment, payment, **cost**, bonus\_grant, status\_id **as** status

**FROM** production.orders

**JOIN** production.orderstatuslog **USING**(order\_id)