**Лабораторная работа №2**

**Мащёнский Дмитрий**

1. *Вывести количество товаров для каждой категории.*  
   SELECT COUNT(\*), category\_id FROM products GROUP BY category\_id;
2. *Вывести список пользователей, которые не оплатили хотя бы 1 заказ полностью.*  
     
   SELECT users.login FROM users  
   JOIN orders ON users.id=orders.users\_id AND orders.status=’C’
3. *Удалить товары, которые ни разу не были куплены (отсутствуют информация в таблице basket).*

DELETE FROM products

WHERE products.id NOT IN (

SELECT basket.product\_id

FROM basket

)

1. *Вывести заказы, которые оплачены только частично*.  
     
   SELECT orders.id, SUM(payments\_orders.value) FROM orders   
   JOIN payments\_orders ON orders.status!= 'D'

AND payments\_orders.value>'0.00'

AND payments\_orders.order\_id=orders.id

1. *Вывести для каждого покупателя количество его заказов по статусам.*

SELECT u.id,

COUNT(\*) filter (WHERE orders.status = 'N' ) ,

COUNT(\*) filter (WHERE orders.status = 'D' ) ,

COUNT(\*) filter (WHERE orders.status = 'P' ) ,

COUNT(\*) filter (WHERE orders.status = 'F' ) ,

COUNT(\*) filter (WHERE orders.status = 'C'),

FROM users u

RIGHT JOIN orders ON orders.user\_id = u.id

GROUP BY u.id;

1. *Вывести средний чек для выполненных заказов (status=*'*F*'*).*

SELECT AVG(price::numeric::float) FROM orders WHERE status='F '

1. *Вывести топ-10 самых продаваемых товаров по суммарной прибыли (заказы для этих товаров должны быть полностью оплачены).*

SELECT name,

SUM(basket.price) AS basket\_value

FROM products

RIGHT JOIN basket ON basket.product\_id = products.id,

(

SELECT id, price

SUM(payment\_orders.value) AS paid\_value

FROM

orders

RIGHT JOIN payment\_orders ON payment\_orders.order\_id = orders.id

GROUP BY

id, payment\_orders.value

) AS paid\_orders

JOIN basket ON basket.order\_id = paid\_orders.id

WHERE paid\_orders.price <= paid\_orders.paid\_value

GROUP BY

id

ORDER BY

basket\_value DESC

LIMIT 10;

1. *Вывести список товаров, которые лежат пока только в корзине и не привязаны к заказу, и их количества не хватает на складе для продажи (считается, что товар списывается со склада только тогда, когда заказ будет доставлен).*

SELECT \* FROM products

RIGHT JOIN basket ON basket.product\_id = products.id and basket.order\_id IS NULL

GROUP BY id

HAVING SUM(basket.quantity)<SUM(products.quantity)

1. *Вывести список пользователей, которые "бросили" свои корзины (не оформили заказ) за последние 30 дней.*SELECT \* FROM users

JOIN orders ON orders.user\_id=users.id AND orders.status=’N’

JOIN payments\_orders ON orders.id=payments\_orders.order\_id

AND payments\_orders.create\_at>=(SELECT NOW()-INTERVAL '30' DAY)

ORDER BY users.id

1. *Добавить скидку 10% на все товары, которые покупались (статус заказов "P" или "F") не более 10 раз.*

UPDATE products

SET discount=discount\* 1.1

WHERE products.id IN

(SELECT basket.product\_id

FROM basket

RIGHT JOIN orders ON basket.order\_id=orders.id

WHERE orders.status SIMILAR TO 'P|F'

GROUP BY basket.product\_id

HAVING COUNT(\*) <= 10)

1. *Вывести количество заказов оплаченных полностью с внутреннего счета.*

SELECT COUNT(\*) FROM orders

JOIN payments\_orders

ON orders.id=payments\_orders.order\_id

WHERE

orders.status=’P’ AND

payments\_orders.payment\_id IS NULL AND

payments\_orders.from\_account IS TRUE

GROUP BY orders.id

1. *Сделать скидку 50% (без доставки, только на товары) на новые заказы (статус "N") для VIP-пользователей.*

*UPDATE orders*

*SET discount = discount\*1.5*

*WHERE orders.user\_id IN*

*(SELECT user.id FROM users*

*RIGHT JOIN accounts ON accounts.user\_id = users.id*

*WHERE*

*accounts.is\_vip = true*

*GROUP BY users.id*

*) AND orders.status = 'N';*

1. *Вывести самую популярную доставку и самый популярный способ оплаты (результат из 2 записей)*

SELECT top1.\*, top2.\*

FROM

(

SELECT delivery.\*,

COUNT(delivery\_orders.\*) AS orders\_count

FROM delivery

LEFT JOIN delivery\_orders ON delivery\_orders.delivery\_id = delivery.id

ORDER BY

orders\_count DESC

GROUP BY delivery.id

LIMIT 1

) AS top1,

(

SELECT payments.\*,

COUNT(payments\_orders.\*) AS orders\_count

FROM payments

LEFT JOIN payments\_orders ON payments\_orders.payment\_id = payments.id

ORDER BY

orders\_count DESC

GROUP BY payments.id

LIMIT 1

) AS top2;

1. *Удалить пустые категории.*

DELETE FROM categories WHERE id NOT IN (SELECT products.category\_id FROM products)

1. *Вывести список пользователей, которые были оплачены полностью не более чем через час, с момента добавления первого товара в корзину*

*.* SELECT orders.\*,

MIN(b.created\_at) ,

SUM(payments\_orders.value),

MAX(payments\_orders.created\_at),

FROM orders

LEFT JOIN payments\_orders ON payments\_orders.order\_id = orders.id

LEFT JOIN basket b ON b.order\_id = orders.id

WHERE

orders.price = (SELECT SUM(payments\_orders.value) FROM payments\_otders) AND

(SELECT date\_part( '4', (SELECT MAX(payments\_orders.create\_at)-MIN(basket.create\_at) FROM payments\_orders, basket)) ) <= 1

GROUP BY orders.id;

1. *Вернуть все деньги пользователей на внутренний счет для заказов, которые были оплачены (как внутренним счетом, так и некоторым способом оплаты) и были отменены (status = "С").*

*WITH acc AS (*

*SELECT*

*SUM(orders.price) AS price,*

*SUM(payments\_orders.value) AS paid\_value*

*FROM orders*

*LEFT JOIN payments\_orders ON payments\_orders.order\_id = orders.id*

*WHERE*

*orders.status = 'C'*

*GROUP BY orders.id*

*HAVING SUM(price) <= SUM(payments\_orders.value)*

*) UPDATE accounts SET value = value + price FROM acc ;*