

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего
образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по лабораторной работе «Запросы на выборку и модификацию данных, представления
и индексы в PostgreSQL»
по дисциплине «Проектирование и реализация баз данных»

Автор: Булыга Е.А.
Факультет: ИКТ
Группа: К32421
Преподаватель: Говорова М.М.

itmo

Санкт-Петербург 2023

Содержание

1	Введение	3
2	Модель базы данных	4
3	Выполнение	5
3.1	Запросы	5
3.1.1	Задание 1	5
3.1.2	Задание 2	7
3.1.3	Задание 3	8
3.1.4	Задание 4	9
3.1.5	Задание 5	9
3.1.6	Задание 6	10
3.1.7	Задание 7	11
3.2	Представления	13
3.2.1	Задание 1	13
3.2.2	Задание 2	13
3.3	Модификации данных	16
3.3.1	Задание 1	16
3.3.2	Задание 2	21
3.4	Индексы	23
3.4.1	Простой индекс	23
3.4.2	Составной индекс	24
4	Вывод	25

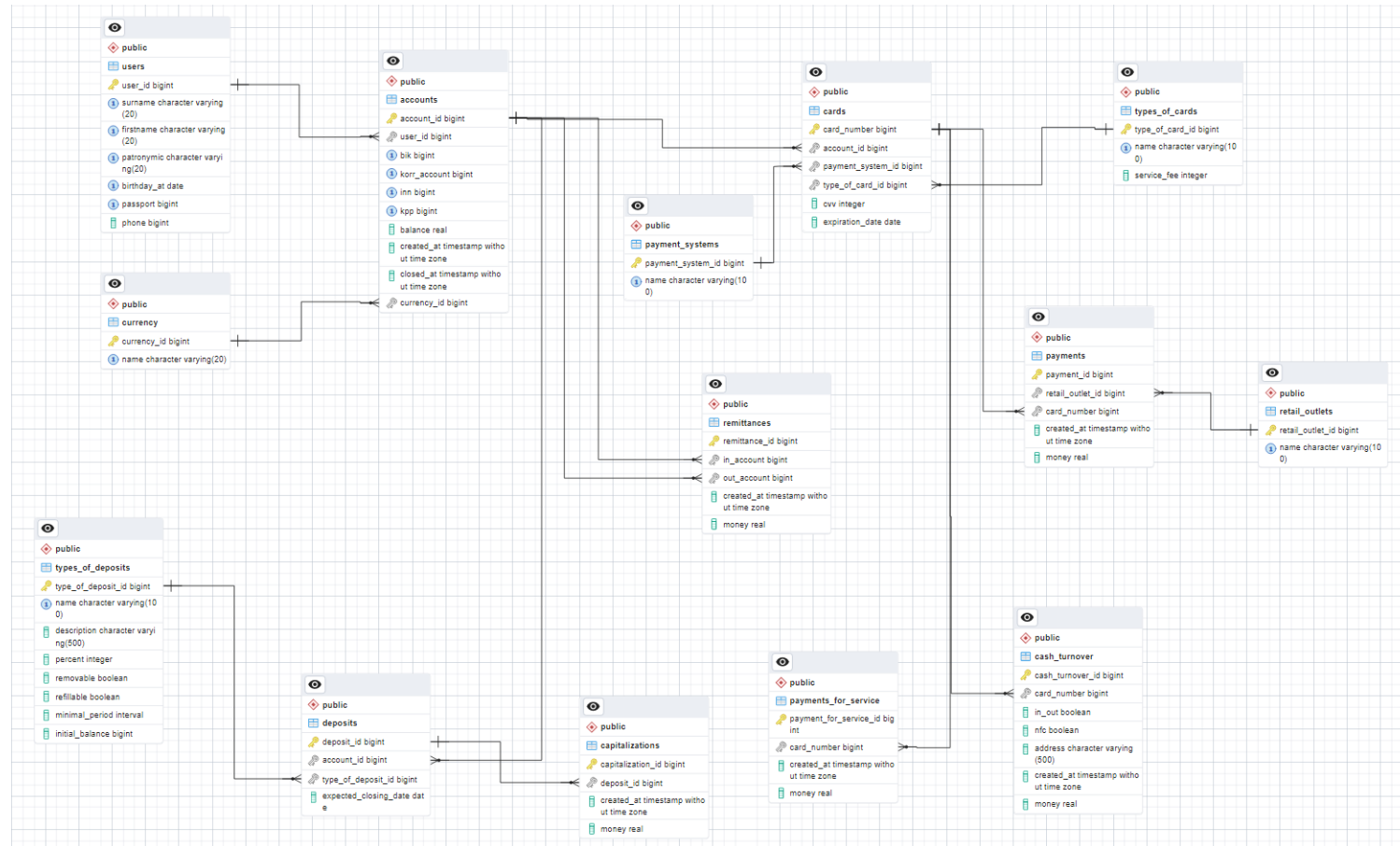
1 Введение

Цель работы: овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.

Практическое задание:

1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию, часть 2 и 3).
2. Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.
3. Изучить графическое представление запросов и просмотреть историю запросов.
4. Создать простой и составной индексы для двух произвольных запросов и сравнить время выполнения запросов без индексов и с индексами. Для получения плана запроса использовать команду EXPLAIN.

2 Модель базы данных



3 Выполнение

3.1 Запросы

3.1.1 Задание 1

Вывод истории операций по указанному счёту.

```
SELECT
*
FROM
(
    SELECT
        'Капитализация' AS operation,
        money,
        created_at
    FROM
        capitalizations
    WHERE
        deposit_id IN (
            SELECT
                deposit_id
            FROM
                deposits
            WHERE
                account_id = 1
        )
    UNION
    SELECT
        'Снятие' AS operation,
        money,
        created_at
    FROM
        cash_turnover
    WHERE
        card_number IN (
            SELECT
                card_number
            FROM
                cards
            WHERE
                account_id = 1
        )
        AND in_out = 'false'
    UNION
    SELECT
        'Пополнение' AS operation,
        money,
        created_at
    FROM
        cash_turnover
    WHERE
        card_number IN (
            SELECT
                card_number
```

```

        FROM
            cards
        WHERE
            account_id = 1
    )
    AND in_out = 'true'
UNION
SELECT
    'Оплата' AS operation,
    money,
    created_at
FROM
    payments
WHERE
    card_number IN (
        SELECT
            card_number
        FROM
            cards
        WHERE
            account_id = 1
    )
UNION
SELECT
    'Плата за обслуживание' AS operation,
    money,
    created_at
FROM
    payments_for_service
WHERE
    card_number IN (
        SELECT
            card_number
        FROM
            cards
        WHERE
            account_id = 1
    )
UNION
SELECT
    'Перевод со счета' AS operation,
    money,
    created_at
FROM
    remittances
WHERE
    in_account = 1
UNION
SELECT
    'Перевод на счет' AS operation,
    money,
    created_at
FROM
    remittances
WHERE
    out_account = 1
) AS history
ORDER BY
    created_at DESC;

```

	operation text	money real	created_at timestamp without time zone
1	Перевод со счёта	4500	2023-03-21 23:33:48.567741
2	Плата за обслуживание	150	2023-03-21 23:10:18.454695
3	Плата за обслуживание	150	2023-03-21 23:09:49.433517
4	Пополнение	4028	2023-03-21 23:01:16.093885

Рис. 1: История операций указанного счёта

3.1.2 Задание 2

Найти вкладчика, имеющего на текущий день наибольшее количество вкладов.

```

SELECT
  user_id,
  firstname,
  surname
FROM
  (
    SELECT
      users.user_id,
      firstname,
      surname,
      COUNT(*)
    FROM
      (
        SELECT
          user_id
        FROM
          deposits
        LEFT JOIN accounts
          ON deposits.account_id = accounts.account_id
      ) AS acc_deposits
    LEFT JOIN users ON acc_deposits.user_id = users.user_id
    GROUP BY
      users.user_id
  ) AS users_deposits
WHERE
  count =(
    SELECT
      MAX(count)
    FROM
      (
        SELECT
          users.user_id,
          COUNT(*)
        FROM
          (
            SELECT
              user_id
            FROM
              deposits

```

```

        LEFT JOIN accounts
        ON deposits.account_id = accounts.account_id
    ) AS deposits_accounts
    LEFT JOIN users ON deposits_accounts.user_id = users.user_id
GROUP BY
    users.user_id
) AS users_deposits
);

```

	user_id [PK] bigint	firstname character varying (20)	surname character varying (20)
1	1004	Софья	Михайлова

Рис. 2: Вкладчик, имеющий на текущий день наибольшее количество вкладов

3.1.3 Задание 3

Найти вкладчика, имеющего вклады во всех видах валюты на текущий день.

```

SELECT
    users.firstname,
    users.surname
FROM
    (
        SELECT
            user_id,
            COUNT(DISTINCT currency_id)
        FROM
            (
                SELECT
                    user_id,
                    currency_id
                FROM
                    accounts
                WHERE
                    account_id IN (
                        SELECT
                            account_id
                        FROM
                            deposits
                    )
            ) AS acc_deposits
        GROUP BY
            user_id
    ) AS tbl1
LEFT JOIN users ON users.user_id = tbl1.user_id
WHERE
    count =(
        SELECT
            COUNT(*)

```



```

FROM
    currency
)
ORDER BY
    count DESC;

```


	firstname character varying (20) 	surname character varying (20) 
1	Софья	Михайлова

Рис. 3: Вкладчик, имеющий вклады во всех видах валюты

3.1.4 Задание 4

Данные вкладчика, имеющего максимальный вклад в долларах.

```

SELECT
    users.user_id,
    firstname,
    surname
FROM
    accounts
LEFT JOIN users ON accounts.user_id = users.user_id
WHERE
    balance =(
        SELECT
            MAX(balance)
        FROM
            accounts
        WHERE
            currency_id =(
                SELECT
                    currency_id
                FROM
                    currency
                WHERE
                    name = 'USD'
            )
        AND account_id IN (
            SELECT
                account_id
            FROM
                deposits
        )
    );

```

3.1.5 Задание 5

Выведите вклад, являющийся наиболее популярным.

	user_id [PK] bigint	firstname character varying (20)	surname character varying (20)
1	1003	Максим	Морозов

Рис. 4: Вкладчик, имеющий максимальный вклад в долларах

```

SELECT
  name
FROM
  deposits
  LEFT JOIN types_of_deposits
    ON deposits.type_of_deposit_id =
       types_of_deposits.type_of_deposit_id
GROUP BY
  name
HAVING
  COUNT(*) = (
    SELECT
      MAX(count)
    FROM
      (
        SELECT
          COUNT(*)
        FROM
          deposits
          LEFT JOIN types_of_deposits
            ON deposits.type_of_deposit_id =
               types_of_deposits.type_of_deposit_id
        GROUP BY
          name
      ) AS max_counts
  );

```

	name character varying (100)
1	Управляй+

Рис. 5: Наиболее популярный вклад

3.1.6 Задание 6

Вывести список вкладчиков, у которых срок вклада истекает завтра и суммы начислений, которые могут быть ими востребованы.

```

SELECT
  users.user_id,
  firstname,
  surname,
  balance

```

```

FROM
(
    SELECT
        user_id,
        balance
    FROM
        deposits
    LEFT JOIN accounts ON deposits.account_id = accounts.account_id
    WHERE
        expected_closing_date BETWEEN CURRENT_DATE + INTERVAL '1' DAY
        AND CURRENT_DATE + INTERVAL '2' DAY
) AS acc_deposits
LEFT JOIN users ON users.user_id = acc_deposits.user_id;

```

	user_id bigint	firstname character varying (20)	surname character varying (20)	balance real
1	1003	Максим	Морозов	30629.701

Рис. 6: Клиенты, срок вклада которых истекает завтра

3.1.7 Задание 7

Вывести, в каких магазинах чаще всего расплачиваются картами с платежной системой VISA.

```

SELECT
    name
FROM
(
    SELECT
        name,
        COUNT(*)
    FROM
    (
        SELECT
            retail_outlets.name
        FROM
            payments
        LEFT JOIN retail_outlets
            ON payments.retail_outlet_id =
            retail_outlets.retail_outlet_id
        WHERE
            card_number IN (
                SELECT
                    card_number
                FROM
                    cards
                WHERE
                    payment_system_id =(
                        SELECT
                            payment_system_id
                        FROM
                            payment_systems

```

```

        WHERE
            name = 'VISA'
    )
    )
    ) AS retail_outlets_visa
GROUP BY
    name
) AS name_retail_visa
WHERE
count =(
SELECT
    MAX(count)
FROM
    (
        SELECT
            COUNT(*)
        FROM
            (
                SELECT
                    retail_outlets.name
                FROM
                    payments
                LEFT JOIN retail_outlets
                ON payments.retail_outlet_id =
                    retail_outlets.retail_outlet_id
                WHERE
                    card_number IN (
                        SELECT
                            card_number
                        FROM
                            cards
                        WHERE
                            payment_system_id =(
                                SELECT
                                    payment_system_id
                                FROM
                                    payment_systems
                                WHERE
                                    name = 'VISA'
                            )
                    )
            )
        ) AS retail_outlets_visa
    GROUP BY
        name
    ) AS count_retail_visa
);

```


	name
	character varying (100) 
1	Магнит

Рис. 7: Магазины, в которых чаще всего расплачиваются картами VISA

3.2 Представления

3.2.1 Задание 1

Вывести номера карт, с которой и на которую произведен перевод.

```
CREATE VIEW cards_remittances AS
SELECT
    in_card,
    card_number AS out_card,
    created_at,
    money
FROM
(
    SELECT
        card_number AS in_card,
        out_account,
        created_at,
        money
    FROM
        remittances
    LEFT JOIN cards ON remittances.in_account = cards.account_id
) AS card_in_account
LEFT JOIN cards ON card_in_account.out_account = cards.account_id;
```

	in_card bigint	out_card bigint	created_at timestamp without time zone	money real
1	4325492597540086	4788927954158345	2023-03-21 23:33:48.567741	10000
2	4858443003042444	4712227257286898	2023-03-21 23:33:48.567741	4500
3	4325492597540086	4111194292302436	2023-03-21 23:33:48.567741	1500
4	5582651815776708	6011008321403328	2023-03-21 23:33:48.567741	500
5	6011008321403328	4976693980888907	2023-03-21 23:33:48.567741	3000

Рис. 8: Представление переводов между картами

3.2.2 Задание 2

Вывести любимые (т.е. имеющие наибольшее количество оплат) магазины клиентов.

```
CREATE VIEW users_lovely_shops AS
SELECT
    *
FROM
(
    SELECT
        users.user_id,
        firstname,
```

```

        surname,
        name,
        COUNT(*)
FROM
    (
        SELECT
            user_id,
            name
        FROM
            (
                SELECT
                    account_id,
                    name
                FROM
                    (
                        SELECT
                            card_number,
                            name
                        FROM
                            payments
                        LEFT JOIN retail_outlets
                        ON payments.retail_outlet_id =
                            retail_outlets.retail_outlet_id
                    ) AS card_retail_outlet
                LEFT JOIN cards
                ON card_retail_outlet.card_number = cards.card_number
            ) AS accounts_retail_outlets
        LEFT JOIN accounts
        ON accounts_retail_outlets.account_id = accounts.account_id
    ) AS users_retail_outlets
    LEFT JOIN users ON users_retail_outlets.user_id = users.user_id
GROUP BY
    users.user_id,
    firstname,
    surname,
    name
ORDER BY
    user_id,
    count DESC
) AS users_counts
WHERE
    count =(
        SELECT
            MAX(count)
        FROM
            (
                SELECT
                    users.user_id,
                    COUNT(*)
                FROM
                    (
                        SELECT
                            user_id,
                            name
                        FROM
                            (
                                SELECT
                                    account_id,
                                    name
                                FROM

```

```

(
    SELECT
        card_number,
        name
    FROM
        payments
    LEFT JOIN retail_outlets
    ON payments.retail_outlet_id =
        retail_outlets.retail_outlet_id
    ) AS card_retail_outlet
    LEFT JOIN cards
    ON card_retail_outlet.card_number = cards.card_number
    ) AS accounts_retail_outlets
    LEFT JOIN accounts
    ON accounts_retail_outlets.account_id =
        accounts.account_id
    ) AS users_retail_outlets
    LEFT JOIN users ON users_retail_outlets.user_id = users.user_id
GROUP BY
    users.user_id,
    firstname,
    surname,
    name
    ) AS counts
WHERE
    counts.user_id = users_counts.user_id
);

```

	user_id bigint 🔒	firstname character varying (20) 🔒	surname character varying (20) 🔒	name character varying (100) 🔒	count bigint 🔒
1	1001	Егор	Григорьев	Магнит	1
2	1002	София	Николаева	Магнит	1
3	1003	Максим	Морозов	Магнит	1
4	1005	Екатерина	Мельникова	Магнит	2
5	1006	Григорий	Волков	Перекресток	1
6	1007	Мирослав	Лазарев	Перекресток	1
7	1009	Елизавета	Леонтьева	Лента	1

Рис. 9: Представление любимых магазинов клиентов

3.3 Модификации данных

3.3.1 Задание 1

Выполнить капитализацию вклада. В зависимости от внесения данных в таблицу о капитализации изменить баланс на счёте.

```
DO $$ DECLARE i RECORD;
BEGIN FOR i IN (
    SELECT
        deposit_id
    FROM
        deposits
) LOOP IF (
    (
        SELECT
            accounts.balance
        FROM
            (
                SELECT
                    deposits.account_id,
                    types_of_deposits.percent
                FROM
                    deposits
                LEFT JOIN types_of_deposits
                ON deposits.type_of_deposit_id =
                    types_of_deposits.type_of_deposit_id
                WHERE
                    deposits.deposit_id = i.deposit_id
            ) AS tbl1
        LEFT JOIN accounts ON tbl1.account_id = accounts.account_id
    ) > 0
) THEN INSERT INTO capitalizations (deposit_id, money)
VALUES
    (
        i.deposit_id,
        (
            SELECT
                (
                    accounts.balance * tbl1.percent / 100 / 12
                )
            FROM
                (
                    SELECT
                        deposits.account_id,
                        types_of_deposits.percent
                    FROM
                        deposits
                    LEFT JOIN types_of_deposits
                    ON deposits.type_of_deposit_id =
                        types_of_deposits.type_of_deposit_id
                    WHERE
                        deposits.deposit_id = i.deposit_id
                    ) AS tbl1
                LEFT JOIN accounts ON tbl1.account_id = accounts.account_id
            )
        )
    );
```



```

ELSE INSERT INTO capitalizations (deposit_id, money)
VALUES
(
    i.deposit_id,
    (
        SELECT
            0.01
        FROM
            (
                SELECT
                    deposits.account_id,
                    types_of_deposits.percent
                FROM
                    deposits
                LEFT JOIN types_of_deposits
                ON deposits.type_of_deposit_id =
                    types_of_deposits.type_of_deposit_id
                WHERE
                    deposits.deposit_id = i.deposit_id
            ) AS tbl1
        LEFT JOIN accounts ON tbl1.account_id = accounts.account_id
    )
);
END IF;
END LOOP;
FOR i IN (
    SELECT
        account_id
    FROM
        deposits
) LOOP
UPDATE
    accounts
SET
    balance = balance + (
        SELECT
            money
        FROM
            (
                SELECT
                    capitalizations.created_at,
                    money
                FROM
                    capitalizations
                LEFT JOIN deposits
                ON capitalizations.deposit_id = deposits.deposit_id
                WHERE
                    account_id = i.account_id
            ) AS tbl1
        ORDER BY
            tbl1.created_at DESC
        LIMIT
            1
    )
WHERE
    accounts.account_id = i.account_id;
END LOOP;
END;
$$;

```

	capitalization_id [PK] bigint	deposit_id bigint	created_at timestamp without time zone	money real
2	2	2	2023-03-21 21:03:48.170912	25
3	3	3	2023-03-21 21:03:48.170912	125
4	4	4	2023-03-21 21:03:48.170912	62.5
5	5	5	2023-03-21 21:03:48.170912	62.5
6	6	6	2023-03-21 21:03:48.170912	29.166666
7	7	1	2023-03-21 21:03:52.823346	0.00005833333
8	8	2	2023-03-21 21:03:52.823346	25.0625
9	9	3	2023-03-21 21:03:52.823346	125.520836
10	10	4	2023-03-21 21:03:52.823346	62.65625
11	11	5	2023-03-21 21:03:52.823346	62.760418
12	12	6	2023-03-21 21:03:52.823346	29.336805
13	13	1	2023-03-21 21:45:36.563468	0.00005867361
14	14	2	2023-03-21 21:45:36.563468	25.125156
15	15	3	2023-03-21 21:45:36.563468	126.04384
16	16	4	2023-03-21 21:45:36.563468	62.81289
17	17	5	2023-03-21 21:45:36.563468	63.02192
18	18	6	2023-03-21 21:45:36.563468	29.507936
19	25	2	2023-04-10 02:19:33.360139	25.250782
20	26	4	2023-04-10 02:19:33.360139	63.126953
21	27	5	2023-04-10 02:19:33.360139	63.5471
22	28	6	2023-04-10 02:19:33.360139	29.852194
23	29	1	2023-04-10 02:19:33.360139	0.000042398668
24	30	3	2023-04-10 02:19:33.360139	127.0942

Рис. 10: Таблица информации о капитализации ДО изменений

	capitalization_id [PK] bigint	deposit_id bigint	created_at timestamp without time zone	money real
8	8	2	2023-03-21 21:03:52.823346	25.0625
9	9	3	2023-03-21 21:03:52.823346	125.520836
10	10	4	2023-03-21 21:03:52.823346	62.65625
11	11	5	2023-03-21 21:03:52.823346	62.760418
12	12	6	2023-03-21 21:03:52.823346	29.336805
13	13	1	2023-03-21 21:45:36.563468	0.00005867361
14	14	2	2023-03-21 21:45:36.563468	25.125156
15	15	3	2023-03-21 21:45:36.563468	126.04384
16	16	4	2023-03-21 21:45:36.563468	62.81289
17	17	5	2023-03-21 21:45:36.563468	63.02192
18	18	6	2023-03-21 21:45:36.563468	29.507936
19	25	2	2023-04-10 02:19:33.360139	25.250782
20	26	4	2023-04-10 02:19:33.360139	63.126953
21	27	5	2023-04-10 02:19:33.360139	63.5471
22	28	6	2023-04-10 02:19:33.360139	29.852194
23	29	1	2023-04-10 02:19:33.360139	0.000042398668
24	30	3	2023-04-10 02:19:33.360139	127.0942
25	31	2	2023-04-10 04:50:28.298747	25.31391
26	32	4	2023-04-10 04:50:28.298747	63.28477
27	33	5	2023-04-10 04:50:28.298747	63.81188
28	34	6	2023-04-10 04:50:28.298747	30.02633
29	35	1	2023-04-10 04:50:28.298747	0.000042575328
30	36	3	2023-04-10 04:50:28.298747	127.62376

Рис. 11: Таблица информации о капитализации ПОСЛЕ изменений

	account_id [PK] bigint	user_id bigint	bik bigint	korr_account bigint	inn bigint	kpp bigint	balance real	created_at timestamp without time zone	closed_at timestamp without time zone	currency_id bigint
1	18	1004	894654555	534367697934728200	6167889821	784008799	15314.851	2023-03-20 20:59:45.921653	[null]	2
2	19	1004	844284389	758139909173454800	7958829463	985457131	5147.371	2023-03-20 20:59:45.921653	[null]	1
3	14	1002	246421530	524734098810322940	7750790048	500106076	0.010218078	2023-03-20 20:59:45.921653	[null]	3
4	16	1003	996388490	145996352135976960	5751759279	595750776	30629.701	2023-01-11 00:00:00	[null]	3
5	8	1008	499891421	767708826257377300	8400231192	712134374	0	2023-03-20 20:59:45.921653	[null]	1
6	11	1002	292601058	481120053472307200	9411145356	932626750	0	2023-03-20 20:59:45.921653	[null]	1
7	5	1005	715100225	184958250330902530	1264561546	604749868	8178	2023-03-20 20:59:45.921653	[null]	1
8	7	1007	929291628	776627963145224200	7059468784	118882167	1362	2023-03-20 20:59:45.921653	[null]	1
9	12	1005	915962263	276832715666973700	1555084199	180277873	10	2023-03-20 20:59:45.921653	[null]	1
10	6	1006	105239847	413554577544376300	9748776026	283810052	2122	2023-03-20 20:59:45.921653	[null]	1
11	10	1001	184743352	776627963145924200	9920565600	579837625	9898	2023-03-20 20:59:45.921653	[null]	1
12	9	1009	564491538	224519537363562500	4163270577	961499602	4383	2023-03-20 20:59:45.921653	[null]	1
13	1	1001	591731039	346706729259861000	1404612055	260765666	8228	2023-03-20 20:59:45.921653	[null]	2
14	4	1004	196242436	227536073608902660	5104831302	571539492	7948	2023-03-20 20:59:45.921653	[null]	2
15	3	1003	824449147	352748740687090700	1258368255	682661582	2882	2023-03-20 20:59:45.921653	[null]	2
16	2	1002	567592410	776627963145124200	7693145261	614293120	1140	2023-03-20 20:59:45.921653	[null]	2
17	13	1005	182023823	715698208383594500	8033117404	962073087	2429	2023-03-20 20:59:45.921653	[null]	3
18	15	1003	985510713	548534998839915500	7681555595	924725342	10125.563	2023-03-20 20:59:45.921653	[null]	3
19	17	1004	125499342	774970428740290600	4425790645	339362742	25313.908	2023-03-20 20:59:45.921653	[null]	3

Рис. 12: Таблица информации о счетах ДО изменений

	account_id [PK] bigint	user_id bigint	bik bigint	korr_account bigint	inn bigint	kpp bigint	balance real	created_at timestamp without time zone	closed_at timestamp without time zone	currency_id bigint
1	15	1003	985510713	548534998839915500	7681555595	924725342	10150.877	2023-03-20 20:59:45.921653	[null]	3
2	17	1004	125499342	774970428740290600	4425790645	339362742	25377.193	2023-03-20 20:59:45.921653	[null]	3
3	18	1004	894654555	534367697934728200	6167889821	784008799	15378.662	2023-03-20 20:59:45.921653	[null]	2
4	19	1004	844284389	758139909173454800	7958829463	985457131	5177.3975	2023-03-20 20:59:45.921653	[null]	1
5	14	1002	246421530	524734098810322940	7750790048	500106076	0.010260654	2023-03-20 20:59:45.921653	[null]	3
6	16	1003	996388490	145996352135976960	5751759279	595750776	30757.324	2023-01-11 00:00:00	[null]	3
7	8	1008	499891421	767708826257377300	8400231192	712134374	0	2023-03-20 20:59:45.921653	[null]	1
8	11	1002	292601058	481120053472307200	9411145356	932626750	0	2023-03-20 20:59:45.921653	[null]	1
9	5	1005	715100225	184958250330902530	1264561546	604749868	8178	2023-03-20 20:59:45.921653	[null]	1
10	7	1007	929291628	776627963145224200	7059468784	118882167	1362	2023-03-20 20:59:45.921653	[null]	1
11	12	1005	915962263	276832715666973700	1555084199	180277873	10	2023-03-20 20:59:45.921653	[null]	1
12	6	1006	105239847	413554577544376300	9748776026	283810052	2122	2023-03-20 20:59:45.921653	[null]	1
13	10	1001	184743352	776627963145924200	9920565600	579837625	9898	2023-03-20 20:59:45.921653	[null]	1
14	9	1009	564491538	224519537363562500	4163270577	961499602	4383	2023-03-20 20:59:45.921653	[null]	1
15	1	1001	591731039	346706729259861000	1404612055	260765666	8228	2023-03-20 20:59:45.921653	[null]	2
16	4	1004	196242436	227536073608902660	5104831302	571539492	7948	2023-03-20 20:59:45.921653	[null]	2
17	3	1003	824449147	352748740687090700	1258368255	682661582	2882	2023-03-20 20:59:45.921653	[null]	2
18	2	1002	567592410	776627963145124200	7693145261	614293120	1140	2023-03-20 20:59:45.921653	[null]	2
19	13	1005	182023823	715698208383594500	8033117404	962073087	2429	2023-03-20 20:59:45.921653	[null]	3

Рис. 13: Таблица информации о счетах ПОСЛЕ изменений

3.3.2 Задание 2

Удалить запись о последней оплате по конкретной карте, если на балансе недостаточно средств.

Добавим строчку об оплате картой, на которой баланс равен нулю.

```
INSERT INTO payments (  
    retail_outlet_id, card_number, money  
)  
VALUES  
    (1, 5577386779412118, 1000);
```

	payment_id [PK] bigint	retail_outlet_id bigint	card_number bigint	created_at timestamp without time zone	money real
1	1	4	5222830188437785	2023-03-21 23:55:46.311901	1263
2	2	3	4712227257286898	2023-03-21 23:56:35.386301	1707
3	3	3	4308268238089647	2023-03-21 23:57:08.35444	3830
4	4	4	6011008321403328	2023-03-21 23:58:08.844465	1868
5	5	3	4325492597540086	2023-03-21 23:58:49.997165	1602
6	6	3	5481196642315710	2023-03-21 23:59:59.378677	3882
7	7	2	4976693980888907	2023-03-22 00:00:46.190846	1240
8	8	3	4788927954158345	2023-03-22 00:01:30.711293	4533
9	9	2	5481196642315710	2023-04-10 01:55:27.55253	1205
10	10	1	5577386779412118	2023-04-10 06:36:41.144919	1000

Рис. 14: Таблица платежей в торговых точках ДО изменений

Удалим запись, если она последняя по данной карте и баланс меньше стоимости покупки.

```
DELETE FROM  
    payments  
WHERE  
    payment_id = (  
        SELECT  
            MAX(payment_id)  
        FROM  
            payments  
        WHERE  
            card_number = 5577386779412118  
    )  
AND money > (  
    SELECT  
        balance  
    FROM  
        accounts  
    RIGHT JOIN cards ON accounts.account_id = cards.account_id  
    WHERE  
        card_number = 5577386779412118  
);
```

	payment_id [PK] bigint	retail_outlet_id bigint	card_number bigint	created_at timestamp without time zone	money real
1	1	4	5222830188437785	2023-03-21 23:55:46.311901	1263
2	2	3	4712227257286898	2023-03-21 23:56:35.386301	1707
3	3	3	4308268238089647	2023-03-21 23:57:08.35444	3830
4	4	4	6011008321403328	2023-03-21 23:58:08.844465	1868
5	5	3	4325492597540086	2023-03-21 23:58:49.997165	1602
6	6	3	5481196642315710	2023-03-21 23:59:59.378677	3882
7	7	2	4976693980888907	2023-03-22 00:00:46.190846	1240
8	8	3	4788927954158345	2023-03-22 00:01:30.711293	4533
9	9	2	5481196642315710	2023-04-10 01:55:27.55253	1205

Рис. 15: Таблица платежей в торговых точках ПОСЛЕ изменений

3.4 Индексы

3.4.1 Простой индекс

Выведем на экран данные о счетах пользователя с идентификатором 1004.

17.04.2023 00:08:15	4	118 msec
Дата	Rows affected	Продолжительность

CopyCopy to Query Editor

```
SELECT * FROM accounts WHERE user_id=1004;
```

Сообщения

Запрос выполнен успешно. Общее время выполнения: 118 msec.
обработано строк: 4.

Рис. 16: Информация о выполнении запроса вывода на экран таблицы счетов

Добавим индекс по столбцу: *user_id*. Снова выведем данные о счетах того же пользователя.

17.04.2023 00:11:02	4	187 msec
Дата	Rows affected	Продолжительность

CopyCopy to Query Editor

```
SELECT * FROM accounts WHERE user_id=1004;
```

Сообщения

Запрос выполнен успешно. Общее время выполнения: 187 msec.
обработано строк: 4.

Рис. 17: Информация о выполнении запроса вывода на экран таблицы счетов

Добавление индекса привело к заметному увеличению времени выполнения запроса, что связано с малым количеством строк в таблице.

3.4.2 Составной индекс

Выведем на экран данные о вкладах, у которых номер счета больше 10, а название вклада - "Вклад Подари жизнь".

15.05.2023 17:39:16
Дата

2
Rows affected

126 msec
Продолжительность

Copy

Copy to Query Editor

```
osit_id=deposits.type_of_deposit_id WHERE account_id>10 AND name='Вклад Подари жиз
```

Сообщения

Запрос выполнен успешно. Общее время выполнения: 126 msec. обработано строк: 2.

Рис. 18: Информация о выполнении запроса вывода на экран таблицы счетов

Добавим индекс по столбцу *account_id* в таблице *deposits* и индекс по столбцу *name* в таблице *types_of_deposits*. Снова выведем данные.

15.05.2023 17:39:00
Дата

2
Rows affected

314 msec
Продолжительность

Copy

Copy to Query Editor

```
osit_id=deposits.type_of_deposit_id WHERE account_id>10 AND name='Вклад Подари жиз
```

Сообщения

Запрос выполнен успешно. Общее время выполнения: 314 msec. обработано строк: 2.

Рис. 19: Информация о выполнении запроса вывода на экран таблицы счетов

Добавление индекса привело также к увеличению времени выполнения запроса, что снова связано с малым количеством строк в таблицах.

4 Вывод

В процессе выполнения лабораторной работы были освоены практические навыки создания представлений, написания запросов на выборку и модификацию данных и индексов.