Национальный исследовательский ядерный университет «МИФИ» (Московский Инженерно-Физический Институт)

Кафедра №42 «Криптология и кибербезопасность»

Лабораторная работа №1-2: «Работа с данными. Простые запросы на выборку»

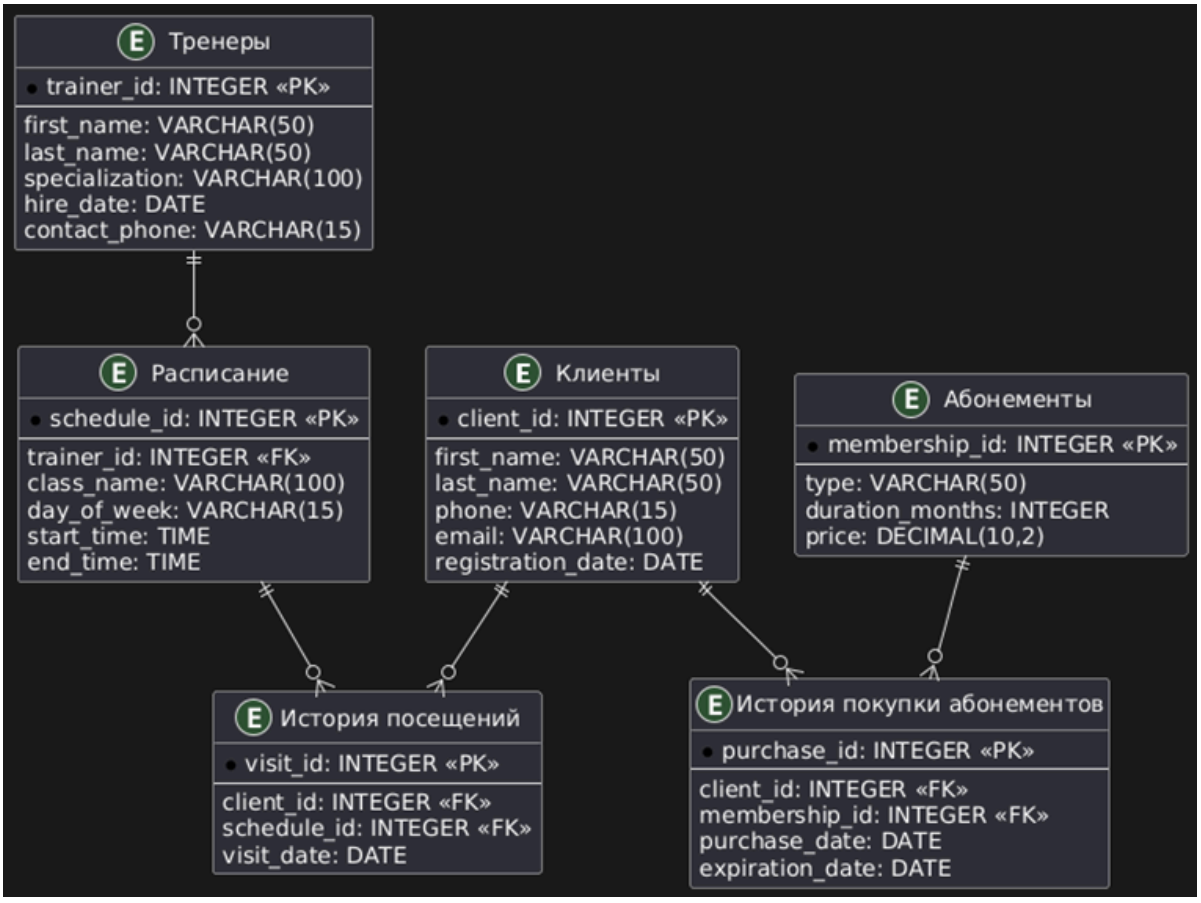
Антон Гатченко Б22-525

2025 г.

Используемая рабочая среда:

- Процессор - AMD Ryzen 5 5600H (laptop), 6c/12t
- Оперативная память – DDR4 16 ГБ
- ОС - Windows 10 Pro 22H2 19045.4780, 64 bit
- SQLite 3.49.1.0

Заполнение таблицы:



Для заполнения таблицы была написана программа на Python (Приложение 1). Для каждой таблицы был создан отдельный класс для соответствия данных разработанной схеме данных (Приложение 2). Для эмуляции реальных данных использовалась библиотека `faker`.

Количество записей в таблице:

Тип	Количество
Клиенты	100
Тренеры	15
Абонементы	8
Занятия	20
Покупки	150
История посещений	200

Запросы к таблице:

Запросы выполнялись посредством кода на Python (Приложение 3). Если результат выдачи довольно большой, то он перенесен в Приложение. Зачастую результатом является только `id`, возможно в паре с каким-то значением. Это было сделано для упрощения вставки результатов запроса в отчет, однако в реальном использовании может быть предпочтительно получать дополнительную информацию, такую как ФИО или даты.

1. ```sql
   SELECT * FROM clients;
   ```

   Возвращает полный список всех клиентов фитнес-центра, включая их идентификаторы, ФИО, контактные данные и дату регистрации. Может быть использован для рассылок о промоакциях или изменениях.

   Результат запроса можно найти в Приложении 4.

2. ```sql
   SELECT * FROM clients WHERE registration_date >= DATE('now', '-30 days');
   ```

   Возвращает список клиентов, зарегистрировавшихся за последние 30 дней. Может быть полезен для анализа рекламных компаний, дополнительной рассылки новым клиентам.

   Результат запроса:

   ```
   [{'client_id': 32},
   {'client_id': 41},
   {'client_id': 45},
   {'client_id': 59},
   {'client_id': 93}]
   ```

3. ```sql
   SELECT * FROM trainers ORDER BY hire_date;
   ```

   Возвращает список тренеров, отсортированный по дате найма (от старых к новым). Может быть полезен для анализа опыта работы тренеров, чтобы составлять отчеты о текучести кадров или эффективности программы найма.

   Результат запроса:

   ```
   [{'trainer_id': 15, 'hire_date': '2020-04-19'},
   {'trainer_id': 10, 'hire_date': '2020-05-12'},
   {'trainer_id': 1, 'hire_date': '2020-06-21'},
   ```
   ```

```
{'trainer_id': 3, 'hire_date': '2020-10-11'},
{'trainer_id': 4, 'hire_date': '2020-10-30'},
{'trainer_id': 6, 'hire_date': '2021-02-04'},
{'trainer_id': 9, 'hire_date': '2021-06-01'},
{'trainer_id': 11, 'hire_date': '2021-08-13'},
{'trainer_id': 2, 'hire_date': '2023-03-30'},
{'trainer_id': 7, 'hire_date': '2023-04-28'},
{'trainer_id': 12, 'hire_date': '2023-10-17'},
{'trainer_id': 13, 'hire_date': '2024-04-03'},
{'trainer_id': 8, 'hire_date': '2024-07-01'},
{'trainer_id': 5, 'hire_date': '2024-08-11'},
{'trainer_id': 14, 'hire_date': '2024-09-05'}]
```

4. ```
   SELECT client_id, COUNT(*) AS visit_count FROM visits GROUP BY client_id;
   ```

   Подсчитывает количество посещений каждого клиента. Может быть использован для дополнительной рассылки неактивным клиентам, для анализа лояльности клиентов.

   Результат запроса можно найти в Приложении 5.

5. ```
   SELECT DISTINCT purchase_date FROM purchases;
   ```

   Возвращает уникальные даты покупок абонементов. Может быть полезен для анализа сезонов продаж и оценки эффективности промоакций.

   Результат запроса можно найти в Приложении 6.

6. ```
   SELECT c.client_id, c.first_name, c.last_name
   FROM clients c
   WHERE NOT EXISTS (
       SELECT 1
       FROM purchases p
       WHERE p.client_id = c.client_id AND p.expiration_date > DATE('now')
   );
   ```

   Находит клиентов, у которых закончился срок действия абонементов. Может быть использован для рассылки выгодных предложений прежним клиентам, а также для анализа лояльности клиентов.

   Результат запроса:

```
[{'client_id': 40}, {'client_id': 43}, {'client_id': 6},
{'client_id': 99}, {'client_id': 8}, {'client_id': 79},
{'client_id': 27}, {'client_id': 23}, {'client_id': 12},
{'client_id': 65}, {'client_id': 34}, {'client_id': 92},
{'client_id': 97}, {'client_id': 19}, {'client_id': 46},
{'client_id': 25}, {'client_id': 82}, {'client_id': 89}]
```
```

7.
```sql
SELECT t.trainer_id, t.first_name, t.last_name, COUNT(v.visit_id) AS
total_visits
FROM trainers t
JOIN schedule s ON t.trainer_id = s.trainer_id
JOIN visits v ON s.schedule_id = v.schedule_id
WHERE v.visit_date >= DATE('now', '-30 days')
GROUP BY t.trainer_id
ORDER BY total_visits DESC;
```

Находит тренеров с наибольшим количеством проведенных занятий за последний месяц. Может быть полезен для оценки эффективности тренеров и распределения бонусов/ сверхурочных, а также планирования расписания.

Результат запроса:

```
[{'trainer_id': 14, 'total_visits': 3},
{'trainer_id': 6, 'total_visits': 3},
{'trainer_id': 2, 'total_visits': 3},
{'trainer_id': 10, 'total_visits': 2},
{'trainer_id': 15, 'total_visits': 1},
{'trainer_id': 11, 'total_visits': 1},
{'trainer_id': 4, 'total_visits': 1}]
```

8.
```sql
SELECT m.type, AVG(m.price) AS average_price
FROM memberships m
JOIN purchases p ON m.membership_id = p.membership_id
GROUP BY m.type;
```

Вычисляет среднюю стоимость абонементов по каждому типу. Может использоваться для планирования ценовой политики, корректировки цен, создания новых абонементов.

Результат запроса:

```
[{'type': 'Премиум', 'average_price': 51691.2},
{'type': 'С бассейном', 'average_price': 19298.0},
{'type': 'С тренером', 'average_price': 21068.131147540982},
{'type': 'Стандарт', 'average_price': 38402.57142857143}]
```

9.
```sql
SELECT c.client_id
FROM clients c
WHERE NOT EXISTS (
    SELECT 1
    FROM visits v
    WHERE v.client_id = c.client_id AND v.visit_date >= DATE('now', '-6
months')
);
```

Находит клиентов, которые не посещали занятия за последние 6 месяцев. Может быть полезен для разработки программ удержания и для акций по мотивации неактивных клиентов.

Результат запроса:

```
[{'client_id': 40}, {'client_id': 67},
{'client_id': 53}, {'client_id': 31},
{'client_id': 2}, {'client_id': 16},
{'client_id': 13}, {'client_id': 21},
{'client_id': 22}, {'client_id': 78},
{'client_id': 37}, {'client_id': 87},
{'client_id': 45}, {'client_id': 88},
{'client_id': 47}, {'client_id': 60},
{'client_id': 7}, {'client_id': 95},
{'client_id': 86}, {'client_id': 73},
{'client_id': 79}, {'client_id': 23},
{'client_id': 77}, {'client_id': 58},
{'client_id': 91}, {'client_id': 32},
{'client_id': 94}, {'client_id': 11},
{'client_id': 34}, {'client_id': 29},
{'client_id': 100}, {'client_id': 26},
{'client_id': 83}, {'client_id': 19},
{'client_id': 48}, {'client_id': 25},
{'client_id': 84}, {'client_id': 82},
{'client_id': 51}]
```

Листинг использованных инструкций SQL

```sql
-- 1. Получение всех клиентов
SELECT * FROM clients;

-- 2. Получение клиентов, зарегистрировавшихся за последние 30 дней
SELECT client_id
FROM clients
WHERE registration_date >= DATE('now', '-30 days');

-- 3. Получение тренеров, отсортированных по дате найма
SELECT trainer_id, hire_date
FROM trainers
ORDER BY hire_date;

-- 4. Подсчет количества посещений для каждого клиента
SELECT client_id, COUNT(*) AS visit_count
FROM visits
GROUP BY client_id;

-- 5. Получение уникальных дат покупок абонементов
SELECT DISTINCT purchase_date
FROM purchases;

-- 6. Получение клиентов с истекшими абонементами
SELECT c.client_id
FROM clients c
WHERE NOT EXISTS (
    SELECT 1
    FROM purchases p
    WHERE p.client_id = c.client_id AND p.expiration_date > DATE('now')
);
```

```sql
-- 7. Получение тренеров с наибольшим количеством проведенных занятий за
последний месяц
SELECT t.trainer_id, COUNT(v.visit_id) AS total_visits
FROM trainers t
JOIN schedule s ON t.trainer_id = s.trainer_id
JOIN visits v ON s.schedule_id = v.schedule_id
WHERE v.visit_date >= DATE('now', '-30 days')
GROUP BY t.trainer_id
ORDER BY total_visits DESC;

-- 8. Получение средней стоимости абонемента по каждому типу
SELECT m.type, AVG(m.price) AS average_price
FROM memberships m
JOIN purchases p ON m.membership_id = p.membership_id
GROUP BY m.type;

-- 9. Получение клиентов, которые не посещали занятия за последние 6 месяцев
SELECT c.client_id
FROM clients c
WHERE NOT EXISTS (
    SELECT 1
    FROM visits v
    WHERE v.client_id = c.client_id AND v.visit_date >= DATE('now', '-6 months')
);
```

Заключение

В ходе выполнения лабораторной работы была продолжена работа над моделью базы данных для фитнес-центра, включающая шесть взаимосвязанных таблиц: клиенты, тренеры, абонементы, расписание, покупки и посещения. Для заполнения таблиц тестовыми данными был написан скрипт на Python. Тестовый набор состоял из записей для 100 клиентов, 15 тренеров, 8 типов абонементов, 20 занятий в расписании, 150 записей о покупках и 200 – о посещениях.

Был реализован набор SQL-запросов для выполнения различных аналитических задач: от получения общего списка клиентов до запросов с агрегацией и фильтрованием. Запросы позволили выявить клиентов с истекшими абонементами, найти неактивных клиентов за последние полгода, определить тренеров с наибольшим количеством проведенных занятий, рассчитать среднюю стоимость абонементов по типам и т.п. Все запросы были успешно протестированы на созданной базе данных.

Реализованные запросы могут быть использованы для решения практических задач администрирования фитнес-центра, таких как анализ активности клиентов, оценка эффективности тренеров и планирование маркетинговых кампаний.

Приложение

1. Исходный код для заполнения таблицы:

```python
import sqlite3
from datetime import datetime, timedelta, date, time
from random import choice, randint
from faker import Faker
from dataclasses import asdict
```

```
from fitness_center import Client, Trainer, Membership, Schedule, Visit, Purchase

class DatabaseFiller:
    def __init__(self, db_path: str):
        self.conn = sqlite3.connect(db_path)
        self.cursor = self.conn.cursor()
        self.fake = Faker('ru_RU')

        self.specializations = ['Фитнес', 'Бассейн', 'Единоборства', 'Йога',
'Пилатес']
        self.membership_types = ['Базовый', 'Стандарт', 'Премиум', 'С бассейном',
'С тренером']
        self.days_of_week = ['Понедельник', 'Вторник', 'Среда', 'Четверг',
'Пятница', 'Суббота', 'Воскресенье']

    def fill_clients(self, num_clients: int):
        clients = [
            Client(
                first_name=self.fake.first_name(),
                last_name=self.fake.last_name(),
                phone=self.fake.unique.phone_number(),
                email=self.fake.unique.email(),
                registration_date=self.fake.date_between(start_date='-2y',
end_date='today')
            )
            for _ in range(num_clients)
        ]
        self._insert_data("clients", clients)

    def fill_trainers(self, num_trainers: int):
        trainers = [
            Trainer(
                first_name=self.fake.unique.first_name(),
                last_name=self.fake.unique.last_name(),
                specialization=choice(self.specializations),
                hire_date=self.fake.date_between(start_date='-5y',
end_date='today'),
                contact_phone=self.fake.unique.phone_number()
            )
            for _ in range(num_trainers)
        ]
        self._insert_data("trainers", trainers)

    def fill_memberships(self, num_memberships: int):
        memberships = [
            Membership(
                type=choice(self.membership_types),
                duration_months=randint(1, 12),
                price=randint(10000, 60000)
            )
            for _ in range(num_memberships)
        ]
        self._insert_data("memberships", memberships)

    def fill_schedule(self, trainer_ids: list[int], num_schedule: int):
        schedule = []
```

```
        for i in range(num_schedule):
            time_ = self.fake.time(pattern='%H:%M')
            schedule.append(
                Schedule(
                    trainer_id=choice(trainer_ids),
                    class_name=choice(self.specializations),
                    day_of_week=choice(self.days_of_week),
                    start_time=datetime.strptime(time_, '%H:%M').time(),
                    end_time=(datetime.strptime(time_, '%H:%M') +
timedelta(hours=1)).time()
                )
            )
        self._insert_data("schedule", schedule)

    def fill_purchases(self, client_ids: list[int], membership_ids: list[int],
num_purchases: int):
        purchases = [
            Purchase(
                client_id=choice(client_ids),
                membership_id=choice(membership_ids),
                purchase_date=self.fake.date_between(start_date='-1y',
end_date='today'),
                expiration_date=self.fake.date_between(start_date='today',
end_date='+1y')
            )
            for _ in range(num_purchases)
        ]
        self._insert_data("purchases", purchases)

    def fill_visits(self, client_ids: list[int], schedule_ids: list[int],
num_visits: int):
        visits = [
            Visit(
                client_id=choice(client_ids),
                schedule_id=choice(schedule_ids),
                visit_date=self.fake.date_between(start_date='-1y',
end_date='today')
            )
            for _ in range(num_visits)
        ]
        self._insert_data("visits", visits)

    @staticmethod
    def convert_value(value):
        if isinstance(value, (datetime, date)):
            return value.strftime('%Y-%m-%d')
        elif isinstance(value, time):
            return value.strftime('%H:%M:%S')
        return value

    def _insert_data(self, table_name: str, data_list: list):
        fields = [field.name for field in
data_list[0].__dataclass_fields__.values() if field.name != "id"]
        placeholders = ", ".join(["?"] * len(fields))
        columns = ", ".join(fields)
```

```
        query = f"INSERT INTO {table_name} ({columns}) VALUES ({placeholders})"

        data_to_insert = [
            tuple(self.convert_value(asdict(item)[field]) for field in fields)
            for item in data_list
        ]

        self.cursor.executemany(query, data_to_insert)
        self.conn.commit()

    def fill_everything(self, num_clients: int, num_trainers: int,
num_memberships: int, num_schedule: int,
                        num_purchases: int, num_visits: int):
        # Заполнение таблиц
        self.fill_clients(num_clients)
        self.fill_trainers(num_trainers)
        self.fill_memberships(num_memberships)

        # Получение ID тренеров для расписания
        self.cursor.execute("SELECT trainer_id FROM trainers")
        trainer_ids = [row[0] for row in self.cursor.fetchall()]
        self.fill_schedule(trainer_ids, num_schedule)

        # Получение ID клиентов и абонементов для покупок
        self.cursor.execute("SELECT client_id FROM clients")
        client_ids = [row[0] for row in self.cursor.fetchall()]
        self.cursor.execute("SELECT membership_id FROM memberships")
        membership_ids = [row[0] for row in self.cursor.fetchall()]
        self.fill_purchases(client_ids, membership_ids, num_purchases)

        # Получение ID расписания для посещений
        self.cursor.execute("SELECT schedule_id FROM schedule")
        schedule_ids = [row[0] for row in self.cursor.fetchall()]
        self.fill_visits(client_ids, schedule_ids, num_visits)

    def close(self):
        self.conn.close()

if __name__ == "__main__":
    NUM_CLIENTS = 100
    NUM_TRAINERS = 15
    NUM_MEMBERSHIPS = 8
    NUM_SCHEDULE = 20
    NUM_PURCHASES = 150
    NUM_VISITS = 200

    filler = DatabaseFiller("sport_club.db")
    filler.fill_everything(NUM_CLIENTS, NUM_TRAINERS, NUM_MEMBERSHIPS,
NUM_SCHEDULE, NUM_PURCHASES, NUM_VISITS)
    filler.close()

    print("База данных успешно заполнена!")
```

2. Исходный код с типами данных для заполнения таблицы:

```python
from dataclasses import dataclass
from datetime import date, time

@dataclass
class Client:
    first_name: str
    last_name: str
    phone: str
    email: str
    registration_date: date
    client_id: int = None  # AUTOINCREMENT, поэтому может быть None при создании

@dataclass
class Trainer:
    first_name: str
    last_name: str
    specialization: str
    hire_date: date
    contact_phone: str
    trainer_id: int = None  # AUTOINCREMENT, поэтому может быть None при создании

@dataclass
class Membership:
    type: str
    duration_months: int
    price: float
    membership_id: int = None  # AUTOINCREMENT, поэтому может быть None при
создании

@dataclass
class Schedule:
    trainer_id: int
    class_name: str
    day_of_week: str
    start_time: time
    end_time: time
    schedule_id: int = None  # AUTOINCREMENT, поэтому может быть None при
создании

@dataclass
class Visit:
    client_id: int
    schedule_id: int
    visit_date: date
    visit_id: int = None  # AUTOINCREMENT, поэтому может быть None при создании

@dataclass
class Purchase:
    client_id: int
    membership_id: int
    purchase_date: date
    expiration_date: date
    purchase_id: int = None  # AUTOINCREMENT, поэтому может быть None при
создании
```

3. Исходный код для выполнения запросов:

```
import json
import sqlite3
from typing import Any

class DatabaseHelper:
    def __init__(self, db_path: str):
        self.db_path = db_path
        self.conn = sqlite3.connect(self.db_path)
        self.cursor = self.conn.cursor()

    def execute_query(self, query: str, params: tuple = ()) -> list[dict[str,
Any]]:
        self.cursor.execute(query, params)
        columns = [column[0] for column in self.cursor.description]
        return [dict(zip(columns, row)) for row in self.cursor.fetchall()]

    def get_all_clients(self) -> list[dict[str, Any]]:
        query = "SELECT * FROM clients;"
        return self.execute_query(query)

    def get_recent_clients(self) -> list[dict[str, Any]]:
        query = "SELECT client_id FROM clients WHERE registration_date >=
DATE('now', '-30 days');"
        return self.execute_query(query)

    def get_trainers_by_hire_date(self) -> list[dict[str, Any]]:
        query = "SELECT trainer_id, hire_date FROM trainers ORDER BY hire_date;"
        return self.execute_query(query)

    def get_visit_counts_per_client(self) -> list[dict[str, Any]]:
        query = ("SELECT client_id, COUNT(*) AS visit_count "
                 "FROM visits GROUP BY client_id;")
        return self.execute_query(query)

    def get_unique_purchase_dates(self) -> list[dict[str, Any]]:
        query = "SELECT DISTINCT purchase_date FROM purchases;"
        return self.execute_query(query)

    def get_expired_memberships(self) -> list[dict[str, Any]]:
        query = """
        SELECT c.client_id
        FROM clients c
        WHERE NOT EXISTS (
            SELECT 1
            FROM purchases p
            WHERE p.client_id = c.client_id AND p.expiration_date > DATE('now')
        );
        """
        return self.execute_query(query)

    def get_top_trainers_last_month(self) -> list[dict[str, Any]]:
        query = """
        SELECT t.trainer_id, COUNT(v.visit_id) AS total_visits
```

```
        FROM trainers t
        JOIN schedule s ON t.trainer_id = s.trainer_id
        JOIN visits v ON s.schedule_id = v.schedule_id
        WHERE v.visit_date >= DATE('now', '-30 days')
        GROUP BY t.trainer_id
        ORDER BY total_visits DESC;
        """
        return self.execute_query(query)

    def get_average_membership_price_by_type(self) -> list[dict[str, Any]]:
        query = """
        SELECT m.type, AVG(m.price) AS average_price
        FROM memberships m
        JOIN purchases p ON m.membership_id = p.membership_id
        GROUP BY m.type;
        """
        return self.execute_query(query)

    def get_inactive_clients_last_6_month(self) -> list[dict[str, Any]]:
        query = """
        SELECT c.client_id
        FROM clients c
        WHERE NOT EXISTS (
            SELECT 1
            FROM visits v
            WHERE v.client_id = c.client_id AND v.visit_date >= DATE('now', '-6
months')
        );
        """
        return self.execute_query(query)

    def vacuum(self):
        self.conn.execute("VACUUM")
        self.conn.commit()

    def close(self):
        self.conn.close()

def pretty_json(data: dict | list) -> str:
    return "[" + ",\n".join(str(item) for item in data) + "]"

if __name__ == "__main__":
    database_helper = DatabaseHelper("sport_club.db")

    clients = database_helper.get_all_clients()
    print("1. Все клиенты:", json.dumps(clients, ensure_ascii=False, indent=4))

    recent_clients = database_helper.get_recent_clients()
    print("2. Недавно зарегистрированные клиенты:", pretty_json(recent_clients),
sep='\n')

    trainers = database_helper.get_trainers_by_hire_date()
    print("3. Тренеры по дате найма:", pretty_json(trainers), sep='\n')

    visit_counts = database_helper.get_visit_counts_per_client()
```

```
    print("4. Количество посещений для каждого клиента:",
pretty_json(visit_counts), sep='\n')

    unique_purchase_dates = database_helper.get_unique_purchase_dates()
    print("5. Уникальные даты покупок:", pretty_json(unique_purchase_dates),
sep='\n')

    expired_clients = database_helper.get_expired_memberships()
    print("6. Клиенты с истекшими абонементами:", pretty_json(expired_clients),
sep='\n')

    top_trainers = database_helper.get_top_trainers_last_month()
    print("7. Тренеры с наибольшим количеством проведенных занятий за месяц:",
pretty_json(top_trainers), sep='\n')

    avg_prices = database_helper.get_average_membership_price_by_type()
    print("8. Средняя стоимость абонемента по каждому типу:",
pretty_json(avg_prices), sep='\n')

    inactive_clients = database_helper.get_inactive_clients_last_6_month()
    print("9. Клиенты, не посещавшие занятия за последний месяц:",
pretty_json(inactive_clients), sep='\n')

    database_helper.vacuum()
    print("10. База данных оптимизирована.")

    database_helper.close()
```

4. Выдача всех клиентов со всеми данными


```
[
    {
        "client_id": 1,
        "first_name": "Лариса",
        "last_name": "Куликов",
        "phone": "+7 (596) 607-88-85",
        "email": "kabanovazhanna@example.org",
        "registration_date": "2024-12-23"
    },
    {
        "client_id": 2,
        "first_name": "Константин",
        "last_name": "Галкина",
        "phone": "8 345 717 7937",
        "email": "miroslav_2020@example.com",
        "registration_date": "2024-03-29"
    },
    {
        "client_id": 3,
        "first_name": "Илья",
        "last_name": "Красильникова",
        "phone": "+7 (444) 681-4123",
        "email": "lihachevprokofi@example.net",
        "registration_date": "2024-10-11"
    },
```

```
    {
        "client_id": 4,
        "first_name": "Евстафий",
        "last_name": "Кудрявцева",
        "phone": "8 (710) 328-16-20",
        "email": "solovevegor@example.net",
        "registration_date": "2023-11-29"
    },
    {
        "client_id": 5,
        "first_name": "Кира",
        "last_name": "Суворова",
        "phone": "+7 400 675 45 01",
        "email": "zahar69@example.org",
        "registration_date": "2024-10-08"
    },
    {
        "client_id": 6,
        "first_name": "Егор",
        "last_name": "Журавлева",
        "phone": "+7 496 561 0791",
        "email": "darja91@example.com",
        "registration_date": "2024-11-06"
    },
    {
        "client_id": 7,
        "first_name": "Доброслав",
        "last_name": "Куликов",
        "phone": "8 756 101 54 19",
        "email": "prokofi_1995@example.com",
        "registration_date": "2024-03-28"
    },
    {
        "client_id": 8,
        "first_name": "Герасим",
        "last_name": "Копылова",
        "phone": "+74202547801",
        "email": "tverdislav66@example.net",
        "registration_date": "2024-08-28"
    },
    {
        "client_id": 9,
        "first_name": "Нестор",
        "last_name": "Никифоров",
        "phone": "85413483434",
        "email": "alekse67@example.com",
        "registration_date": "2023-09-20"
    },
    {
        "client_id": 10,
        "first_name": "Прохор",
        "last_name": "Зайцев",
        "phone": "+7 610 276 3769",
        "email": "ippolit2005@example.net",
        "registration_date": "2023-05-04"
    },
```

```
    {
        "client_id": 11,
        "first_name": "Евгения",
        "last_name": "Панфилова",
        "phone": "+7 (201) 737-3808",
        "email": "viktorija86@example.com",
        "registration_date": "2024-06-27"
    },
    {
        "client_id": 12,
        "first_name": "Гурий",
        "last_name": "Носов",
        "phone": "+7 553 615 51 75",
        "email": "dobromisl_04@example.org",
        "registration_date": "2024-12-09"
    },
    {
        "client_id": 13,
        "first_name": "Валентин",
        "last_name": "Гришина",
        "phone": "+7 (490) 973-8873",
        "email": "irakli_1988@example.org",
        "registration_date": "2024-03-31"
    },
    {
        "client_id": 14,
        "first_name": "Аполлинарий",
        "last_name": "Лихачева",
        "phone": "+79365388337",
        "email": "evgraf2008@example.net",
        "registration_date": "2023-07-13"
    },
    {
        "client_id": 15,
        "first_name": "Валентина",
        "last_name": "Иванова",
        "phone": "+7 (818) 190-6302",
        "email": "venediktjudin@example.net",
        "registration_date": "2024-10-22"
    },
    {
        "client_id": 16,
        "first_name": "Панкрат",
        "last_name": "Герасимов",
        "phone": "+7 (254) 130-06-40",
        "email": "jaropolk19@example.com",
        "registration_date": "2024-12-05"
    },
    {
        "client_id": 17,
        "first_name": "Лора",
        "last_name": "Гуляева",
        "phone": "8 068 406 77 73",
        "email": "dorofe05@example.net",
        "registration_date": "2023-05-01"
    },
```

```
    {
        "client_id": 18,
        "first_name": "Анжелика",
        "last_name": "Осипова",
        "phone": "+7 275 329 8732",
        "email": "stanimir39@example.org",
        "registration_date": "2023-05-04"
    },
    {
        "client_id": 19,
        "first_name": "Станимир",
        "last_name": "Туров",
        "phone": "+7 (050) 401-17-49",
        "email": "uljan_00@example.com",
        "registration_date": "2024-09-14"
    },
    {
        "client_id": 20,
        "first_name": "Филарет",
        "last_name": "Максимова",
        "phone": "+7 (330) 411-88-13",
        "email": "naum81@example.com",
        "registration_date": "2023-09-18"
    },
    {
        "client_id": 21,
        "first_name": "Мокей",
        "last_name": "Гурьева",
        "phone": "8 302 911 63 59",
        "email": "dshestakova@example.net",
        "registration_date": "2024-02-02"
    },
    {
        "client_id": 22,
        "first_name": "Август",
        "last_name": "Данилова",
        "phone": "+78084684804",
        "email": "odintsovantip@example.net",
        "registration_date": "2025-02-01"
    },
    {
        "client_id": 23,
        "first_name": "Адам",
        "last_name": "Назаров",
        "phone": "8 031 554 5392",
        "email": "ignati21@example.org",
        "registration_date": "2023-10-11"
    },
    {
        "client_id": 24,
        "first_name": "Софон",
        "last_name": "Егоров",
        "phone": "8 969 902 8659",
        "email": "prokl1999@example.org",
        "registration_date": "2024-08-25"
    },
```

```
    {
        "client_id": 25,
        "first_name": "Тимур",
        "last_name": "Шилов",
        "phone": "+76964036444",
        "email": "kostinmilovan@example.org",
        "registration_date": "2023-10-16"
    },
    {
        "client_id": 26,
        "first_name": "Никанор",
        "last_name": "Суханова",
        "phone": "88272751286",
        "email": "lnikonov@example.net",
        "registration_date": "2024-04-02"
    },
    {
        "client_id": 27,
        "first_name": "Виссарион",
        "last_name": "Моисеева",
        "phone": "8 113 723 49 02",
        "email": "miheevaverjan@example.org",
        "registration_date": "2023-04-08"
    },
    {
        "client_id": 28,
        "first_name": "Пимен",
        "last_name": "Носова",
        "phone": "+7 714 678 6334",
        "email": "vera_58@example.com",
        "registration_date": "2024-01-03"
    },
    {
        "client_id": 29,
        "first_name": "Андроник",
        "last_name": "Родионова",
        "phone": "+7 (864) 388-17-70",
        "email": "orehovaanzhela@example.net",
        "registration_date": "2024-12-23"
    },
    {
        "client_id": 30,
        "first_name": "София",
        "last_name": "Гордеева",
        "phone": "84155541745",
        "email": "anzhelika_1979@example.com",
        "registration_date": "2023-12-28"
    },
    {
        "client_id": 31,
        "first_name": "Ким",
        "last_name": "Воробьев",
        "phone": "87561846053",
        "email": "stanimir_44@example.org",
        "registration_date": "2023-09-26"
    },
```

```
    {
        "client_id": 32,
        "first_name": "Викентий",
        "last_name": "Панов",
        "phone": "81279184054",
        "email": "artemevakim@example.net",
        "registration_date": "2025-03-27"
    },
    {
        "client_id": 33,
        "first_name": "Сигизмунд",
        "last_name": "Сидорова",
        "phone": "+7 324 360 54 86",
        "email": "dementevavarvara@example.com",
        "registration_date": "2024-07-10"
    },
    {
        "client_id": 34,
        "first_name": "Никифор",
        "last_name": "Пономарева",
        "phone": "+7 (304) 290-7514",
        "email": "luchezar_04@example.com",
        "registration_date": "2023-10-31"
    },
    {
        "client_id": 35,
        "first_name": "Дарья",
        "last_name": "Назарова",
        "phone": "+7 (884) 888-2362",
        "email": "ekaterina_1984@example.com",
        "registration_date": "2025-01-18"
    },
    {
        "client_id": 36,
        "first_name": "Маргарита",
        "last_name": "Дорофеева",
        "phone": "+7 759 597 3049",
        "email": "mefodi85@example.com",
        "registration_date": "2024-11-04"
    },
    {
        "client_id": 37,
        "first_name": "Станислав",
        "last_name": "Захаров",
        "phone": "82593984173",
        "email": "svjatoslav_89@example.net",
        "registration_date": "2024-05-01"
    },
    {
        "client_id": 38,
        "first_name": "Ульяна",
        "last_name": "Носков",
        "phone": "81220278593",
        "email": "zpetuhova@example.com",
        "registration_date": "2024-01-23"
    },
```

```
    {
        "client_id": 39,
        "first_name": "Глафира",
        "last_name": "Никитин",
        "phone": "+77385260631",
        "email": "margarita2005@example.net",
        "registration_date": "2025-02-06"
    },
    {
        "client_id": 40,
        "first_name": "Парамон",
        "last_name": "Авдеев",
        "phone": "8 (250) 875-0047",
        "email": "vadim44@example.org",
        "registration_date": "2023-11-22"
    },
    {
        "client_id": 41,
        "first_name": "Нестор",
        "last_name": "Ершов",
        "phone": "8 600 007 2523",
        "email": "hariton_1970@example.com",
        "registration_date": "2025-03-18"
    },
    {
        "client_id": 42,
        "first_name": "Андрей",
        "last_name": "Воронов",
        "phone": "+7 227 726 2798",
        "email": "nkovaleva@example.net",
        "registration_date": "2024-06-17"
    },
    {
        "client_id": 43,
        "first_name": "Влас",
        "last_name": "Белов",
        "phone": "8 (582) 466-7008",
        "email": "serafim2003@example.org",
        "registration_date": "2024-04-21"
    },
    {
        "client_id": 44,
        "first_name": "Федот",
        "last_name": "Якушев",
        "phone": "+7 (848) 454-31-61",
        "email": "kopilovosip@example.com",
        "registration_date": "2023-06-06"
    },
    {
        "client_id": 45,
        "first_name": "Михаил",
        "last_name": "Игнатова",
        "phone": "8 (501) 580-0348",
        "email": "galina_94@example.com",
        "registration_date": "2025-03-16"
    },
```

```
    {
        "client_id": 46,
        "first_name": "Людмила",
        "last_name": "Чернов",
        "phone": "87304259804",
        "email": "vorontsovagalina@example.com",
        "registration_date": "2023-11-12"
    },
    {
        "client_id": 47,
        "first_name": "Соломон",
        "last_name": "Комиссаров",
        "phone": "8 (477) 270-5967",
        "email": "mstislav_2014@example.org",
        "registration_date": "2024-04-05"
    },
    {
        "client_id": 48,
        "first_name": "Платон",
        "last_name": "Фомин",
        "phone": "+76080339535",
        "email": "dobromislrjabov@example.net",
        "registration_date": "2024-02-05"
    },
    {
        "client_id": 49,
        "first_name": "Парфен",
        "last_name": "Ковалева",
        "phone": "8 611 157 0470",
        "email": "evlampi_17@example.com",
        "registration_date": "2024-11-27"
    },
    {
        "client_id": 50,
        "first_name": "Савватий",
        "last_name": "Галкина",
        "phone": "8 (333) 333-8095",
        "email": "kulikovanatoli@example.com",
        "registration_date": "2024-07-27"
    },
    {
        "client_id": 51,
        "first_name": "Егор",
        "last_name": "Юдина",
        "phone": "8 (341) 532-9729",
        "email": "vladimirovisidor@example.com",
        "registration_date": "2023-06-07"
    },
    {
        "client_id": 52,
        "first_name": "Назар",
        "last_name": "Панова",
        "phone": "8 (413) 887-9398",
        "email": "sinklitikija_95@example.org",
        "registration_date": "2023-05-24"
    },
```

```
    {
        "client_id": 53,
        "first_name": "Мстислав",
        "last_name": "Блохина",
        "phone": "+7 (251) 503-38-51",
        "email": "drozdovapraskovja@example.com",
        "registration_date": "2023-08-11"
    },
    {
        "client_id": 54,
        "first_name": "Ювеналий",
        "last_name": "Силина",
        "phone": "8 (532) 303-6786",
        "email": "djachkovaviktorija@example.net",
        "registration_date": "2023-08-02"
    },
    {
        "client_id": 55,
        "first_name": "Кира",
        "last_name": "Калашникова",
        "phone": "83368608976",
        "email": "ilja_2012@example.org",
        "registration_date": "2023-05-01"
    },
    {
        "client_id": 56,
        "first_name": "Павел",
        "last_name": "Гуляева",
        "phone": "8 (204) 932-71-57",
        "email": "tihon2012@example.org",
        "registration_date": "2025-02-09"
    },
    {
        "client_id": 57,
        "first_name": "Раиса",
        "last_name": "Силин",
        "phone": "8 (806) 328-99-94",
        "email": "anisimovuljan@example.com",
        "registration_date": "2025-01-12"
    },
    {
        "client_id": 58,
        "first_name": "Радим",
        "last_name": "Некрасов",
        "phone": "8 (085) 341-2801",
        "email": "sorokinstepan@example.com",
        "registration_date": "2023-12-11"
    },
    {
        "client_id": 59,
        "first_name": "Чеслав",
        "last_name": "Никонова",
        "phone": "+7 164 967 4852",
        "email": "mironmaksimov@example.net",
        "registration_date": "2025-03-25"
    },
```

```
    {
        "client_id": 60,
        "first_name": "Ладимир",
        "last_name": "Кудряшов",
        "phone": "+7 306 453 0068",
        "email": "wnikiforov@example.org",
        "registration_date": "2023-07-22"
    },
    {
        "client_id": 61,
        "first_name": "Герман",
        "last_name": "Панфилова",
        "phone": "8 (178) 270-9788",
        "email": "boleslavtsvetkov@example.com",
        "registration_date": "2024-03-16"
    },
    {
        "client_id": 62,
        "first_name": "Савва",
        "last_name": "Сергеев",
        "phone": "+7 (050) 852-0620",
        "email": "hantonov@example.net",
        "registration_date": "2023-07-31"
    },
    {
        "client_id": 63,
        "first_name": "Любосмысл",
        "last_name": "Громова",
        "phone": "8 865 285 67 03",
        "email": "aterentev@example.net",
        "registration_date": "2023-10-22"
    },
    {
        "client_id": 64,
        "first_name": "Андрон",
        "last_name": "Селезнева",
        "phone": "8 486 506 2830",
        "email": "krasilnikovauljana@example.com",
        "registration_date": "2025-02-05"
    },
    {
        "client_id": 65,
        "first_name": "Аким",
        "last_name": "Орлов",
        "phone": "+7 (620) 343-65-64",
        "email": "juri1970@example.org",
        "registration_date": "2024-10-06"
    },
    {
        "client_id": 66,
        "first_name": "Василиса",
        "last_name": "Лихачева",
        "phone": "+76780897432",
        "email": "german2015@example.com",
        "registration_date": "2025-01-08"
    },
```

```
    {
        "client_id": 67,
        "first_name": "Никифор",
        "last_name": "Александров",
        "phone": "+75492316102",
        "email": "feoktist_52@example.org",
        "registration_date": "2024-06-01"
    },
    {
        "client_id": 68,
        "first_name": "Ия",
        "last_name": "Евсеева",
        "phone": "8 309 698 60 22",
        "email": "fade_2005@example.com",
        "registration_date": "2024-11-09"
    },
    {
        "client_id": 69,
        "first_name": "Твердислав",
        "last_name": "Филатов",
        "phone": "+74609765184",
        "email": "filimon_34@example.com",
        "registration_date": "2025-01-06"
    },
    {
        "client_id": 70,
        "first_name": "Самсон",
        "last_name": "Прохоров",
        "phone": "+7 698 110 5441",
        "email": "gusevkliment@example.com",
        "registration_date": "2024-07-09"
    },
    {
        "client_id": 71,
        "first_name": "Парамон",
        "last_name": "Виноградов",
        "phone": "8 (340) 890-6831",
        "email": "rogovsevastjan@example.org",
        "registration_date": "2024-08-24"
    },
    {
        "client_id": 72,
        "first_name": "Агафон",
        "last_name": "Давыдов",
        "phone": "+7 (994) 743-10-54",
        "email": "irina1973@example.com",
        "registration_date": "2024-12-11"
    },
    {
        "client_id": 73,
        "first_name": "Демид",
        "last_name": "Мамонтова",
        "phone": "+70582234058",
        "email": "anzhela16@example.net",
        "registration_date": "2023-08-26"
    },
```

```
    {
        "client_id": 74,
        "first_name": "Владимир",
        "last_name": "Сысоева",
        "phone": "+7 (440) 820-26-57",
        "email": "ovchinnikovgrigori@example.net",
        "registration_date": "2024-09-19"
    },
    {
        "client_id": 75,
        "first_name": "Устин",
        "last_name": "Шестаков",
        "phone": "8 (672) 515-96-59",
        "email": "nikita_2007@example.com",
        "registration_date": "2023-04-08"
    },
    {
        "client_id": 76,
        "first_name": "Маргарита",
        "last_name": "Шаров",
        "phone": "+7 605 988 4606",
        "email": "glafira_2021@example.org",
        "registration_date": "2023-05-09"
    },
    {
        "client_id": 77,
        "first_name": "Мокей",
        "last_name": "Назарова",
        "phone": "+7 (444) 880-32-66",
        "email": "nikita_45@example.net",
        "registration_date": "2023-08-20"
    },
    {
        "client_id": 78,
        "first_name": "Ярополк",
        "last_name": "Данилова",
        "phone": "8 360 236 34 27",
        "email": "zhuravlevmilovan@example.org",
        "registration_date": "2025-02-02"
    },
    {
        "client_id": 79,
        "first_name": "Прохор",
        "last_name": "Миронов",
        "phone": "+7 410 895 64 83",
        "email": "alekse_32@example.net",
        "registration_date": "2024-07-15"
    },
    {
        "client_id": 80,
        "first_name": "Октябрина",
        "last_name": "Мамонтов",
        "phone": "+7 459 663 61 96",
        "email": "arseni91@example.net",
        "registration_date": "2024-11-08"
    },
```

```
    {
        "client_id": 81,
        "first_name": "Леон",
        "last_name": "Третьяков",
        "phone": "+7 (119) 587-75-47",
        "email": "cheslav_1975@example.net",
        "registration_date": "2024-06-21"
    },
    {
        "client_id": 82,
        "first_name": "Дементий",
        "last_name": "Юдин",
        "phone": "+7 (619) 905-04-31",
        "email": "fokinermil@example.net",
        "registration_date": "2024-01-13"
    },
    {
        "client_id": 83,
        "first_name": "Спартак",
        "last_name": "Туров",
        "phone": "+7 (696) 645-1222",
        "email": "platon_2022@example.net",
        "registration_date": "2023-05-21"
    },
    {
        "client_id": 84,
        "first_name": "Константин",
        "last_name": "Ширяев",
        "phone": "+7 (137) 400-71-33",
        "email": "rodionovanatalja@example.org",
        "registration_date": "2023-08-29"
    },
    {
        "client_id": 85,
        "first_name": "Ратмир",
        "last_name": "Стрелков",
        "phone": "83617071691",
        "email": "naina_1993@example.net",
        "registration_date": "2023-06-23"
    },
    {
        "client_id": 86,
        "first_name": "Ким",
        "last_name": "Лапина",
        "phone": "+7 (879) 332-53-60",
        "email": "sofron62@example.net",
        "registration_date": "2024-06-09"
    },
    {
        "client_id": 87,
        "first_name": "Аполлон",
        "last_name": "Зиновьева",
        "phone": "+7 627 209 26 39",
        "email": "ladislav56@example.net",
        "registration_date": "2024-07-24"
    },
```

```
    {
        "client_id": 88,
        "first_name": "Евлампий",
        "last_name": "Казакова",
        "phone": "+7 401 638 20 75",
        "email": "angelina60@example.com",
        "registration_date": "2024-01-16"
    },
    {
        "client_id": 89,
        "first_name": "Иннокентий",
        "last_name": "Юдина",
        "phone": "8 536 295 2533",
        "email": "ygalkin@example.com",
        "registration_date": "2024-01-25"
    },
    {
        "client_id": 90,
        "first_name": "Андрон",
        "last_name": "Агафонов",
        "phone": "+7 (625) 528-04-20",
        "email": "vkapustina@example.net",
        "registration_date": "2024-09-14"
    },
    {
        "client_id": 91,
        "first_name": "Аркадий",
        "last_name": "Орехова",
        "phone": "+74573194418",
        "email": "cbobilev@example.org",
        "registration_date": "2023-07-12"
    },
    {
        "client_id": 92,
        "first_name": "Милен",
        "last_name": "Ситников",
        "phone": "8 241 860 6814",
        "email": "hristofor_18@example.com",
        "registration_date": "2024-03-29"
    },
    {
        "client_id": 93,
        "first_name": "Елисей",
        "last_name": "Пахомов",
        "phone": "8 159 585 6572",
        "email": "sergesharapov@example.com",
        "registration_date": "2025-03-05"
    },
    {
        "client_id": 94,
        "first_name": "Клавдия",
        "last_name": "Панова",
        "phone": "+7 (864) 574-2691",
        "email": "ruben_1985@example.com",
        "registration_date": "2025-01-08"
    },
```

```
    {
        "client_id": 95,
        "first_name": "Ирина",
        "last_name": "Лазарев",
        "phone": "+7 (942) 603-66-20",
        "email": "semendementev@example.com",
        "registration_date": "2023-06-30"
    },
    {
        "client_id": 96,
        "first_name": "Дементий",
        "last_name": "Гордеев",
        "phone": "+7 628 144 22 72",
        "email": "alevtina_1997@example.com",
        "registration_date": "2025-01-10"
    },
    {
        "client_id": 97,
        "first_name": "Корнил",
        "last_name": "Субботина",
        "phone": "8 758 957 92 01",
        "email": "xagafonova@example.net",
        "registration_date": "2025-01-25"
    },
    {
        "client_id": 98,
        "first_name": "Зинаида",
        "last_name": "Туров",
        "phone": "+7 795 261 30 23",
        "email": "gorbunovaglafira@example.org",
        "registration_date": "2024-12-30"
    },
    {
        "client_id": 99,
        "first_name": "Фадей",
        "last_name": "Зуева",
        "phone": "87151141231",
        "email": "androniksimonov@example.com",
        "registration_date": "2025-01-17"
    },
    {
        "client_id": 100,
        "first_name": "Вениамин",
        "last_name": "Соколов",
        "phone": "+7 515 546 6939",
        "email": "prov32@example.com",
        "registration_date": "2024-05-25"
    }
]
```


5. Выдача количества посещений для каждого клиента


```
[{'client_id': 1, 'visit_count': 1},
{'client_id': 3, 'visit_count': 3},
{'client_id': 4, 'visit_count': 5},
```

```
{'client_id': 5, 'visit_count': 2},
{'client_id': 6, 'visit_count': 1},
{'client_id': 7, 'visit_count': 1},
{'client_id': 8, 'visit_count': 5},
{'client_id': 9, 'visit_count': 5},
{'client_id': 10, 'visit_count': 3},
{'client_id': 12, 'visit_count': 2},
{'client_id': 13, 'visit_count': 1},
{'client_id': 14, 'visit_count': 2},
{'client_id': 15, 'visit_count': 2},
{'client_id': 16, 'visit_count': 1},
{'client_id': 17, 'visit_count': 1},
{'client_id': 18, 'visit_count': 3},
{'client_id': 19, 'visit_count': 2},
{'client_id': 20, 'visit_count': 1},
{'client_id': 21, 'visit_count': 2},
{'client_id': 22, 'visit_count': 1},
{'client_id': 24, 'visit_count': 4},
{'client_id': 25, 'visit_count': 2},
{'client_id': 26, 'visit_count': 1},
{'client_id': 27, 'visit_count': 2},
{'client_id': 28, 'visit_count': 4},
{'client_id': 29, 'visit_count': 2},
{'client_id': 30, 'visit_count': 2},
{'client_id': 32, 'visit_count': 1},
{'client_id': 33, 'visit_count': 4},
{'client_id': 35, 'visit_count': 3},
{'client_id': 36, 'visit_count': 1},
{'client_id': 37, 'visit_count': 2},
{'client_id': 38, 'visit_count': 2},
{'client_id': 39, 'visit_count': 4},
{'client_id': 40, 'visit_count': 2},
{'client_id': 41, 'visit_count': 2},
{'client_id': 42, 'visit_count': 5},
{'client_id': 43, 'visit_count': 3},
{'client_id': 44, 'visit_count': 7},
{'client_id': 45, 'visit_count': 1},
{'client_id': 46, 'visit_count': 2},
{'client_id': 48, 'visit_count': 1},
{'client_id': 49, 'visit_count': 3},
{'client_id': 50, 'visit_count': 1},
{'client_id': 51, 'visit_count': 1},
{'client_id': 52, 'visit_count': 5},
{'client_id': 54, 'visit_count': 3},
{'client_id': 55, 'visit_count': 2},
{'client_id': 56, 'visit_count': 2},
{'client_id': 57, 'visit_count': 4},
{'client_id': 58, 'visit_count': 1},
{'client_id': 59, 'visit_count': 3},
{'client_id': 60, 'visit_count': 2},
{'client_id': 61, 'visit_count': 2},
{'client_id': 62, 'visit_count': 2},
{'client_id': 63, 'visit_count': 3},
{'client_id': 64, 'visit_count': 6},
{'client_id': 65, 'visit_count': 3},
{'client_id': 66, 'visit_count': 1},
```

```
{'client_id': 68, 'visit_count': 1},
{'client_id': 69, 'visit_count': 1},
{'client_id': 70, 'visit_count': 2},
{'client_id': 71, 'visit_count': 4},
{'client_id': 72, 'visit_count': 2},
{'client_id': 73, 'visit_count': 2},
{'client_id': 74, 'visit_count': 2},
{'client_id': 75, 'visit_count': 1},
{'client_id': 76, 'visit_count': 2},
{'client_id': 77, 'visit_count': 1},
{'client_id': 78, 'visit_count': 1},
{'client_id': 79, 'visit_count': 1},
{'client_id': 80, 'visit_count': 4},
{'client_id': 81, 'visit_count': 1},
{'client_id': 82, 'visit_count': 1},
{'client_id': 83, 'visit_count': 1},
{'client_id': 84, 'visit_count': 1},
{'client_id': 85, 'visit_count': 2},
{'client_id': 86, 'visit_count': 1},
{'client_id': 89, 'visit_count': 2},
{'client_id': 90, 'visit_count': 3},
{'client_id': 92, 'visit_count': 3},
{'client_id': 93, 'visit_count': 2},
{'client_id': 94, 'visit_count': 1},
{'client_id': 95, 'visit_count': 2},
{'client_id': 96, 'visit_count': 2},
{'client_id': 97, 'visit_count': 4},
{'client_id': 98, 'visit_count': 2},
{'client_id': 99, 'visit_count': 3}]
```


6. Выдача уникальных дат покупок


```
[{'purchase_date': '2024-08-13'},
{'purchase_date': '2024-08-05'},
{'purchase_date': '2024-06-10'},
{'purchase_date': '2024-10-26'},
{'purchase_date': '2024-12-21'},
{'purchase_date': '2024-10-15'},
{'purchase_date': '2024-09-10'},
{'purchase_date': '2024-07-27'},
{'purchase_date': '2024-11-01'},
{'purchase_date': '2024-10-14'},
{'purchase_date': '2024-10-02'},
{'purchase_date': '2025-03-01'},
{'purchase_date': '2025-01-17'},
{'purchase_date': '2024-09-24'},
{'purchase_date': '2025-01-21'},
{'purchase_date': '2024-08-14'},
{'purchase_date': '2024-10-07'},
{'purchase_date': '2024-07-16'},
{'purchase_date': '2024-08-24'},
{'purchase_date': '2024-10-17'},
{'purchase_date': '2024-12-15'},
{'purchase_date': '2025-02-20'},
{'purchase_date': '2024-11-07'},
```

```
{'purchase_date': '2024-06-24'},
{'purchase_date': '2024-07-14'},
{'purchase_date': '2024-09-18'},
{'purchase_date': '2025-01-23'},
{'purchase_date': '2024-07-09'},
{'purchase_date': '2024-04-22'},
{'purchase_date': '2024-06-08'},
{'purchase_date': '2024-06-06'},
{'purchase_date': '2024-11-22'},
{'purchase_date': '2024-08-04'},
{'purchase_date': '2025-03-19'},
{'purchase_date': '2024-07-25'},
{'purchase_date': '2024-12-18'},
{'purchase_date': '2024-12-09'},
{'purchase_date': '2025-03-02'},
{'purchase_date': '2025-03-18'},
{'purchase_date': '2024-05-16'},
{'purchase_date': '2024-09-27'},
{'purchase_date': '2024-11-03'},
{'purchase_date': '2024-11-02'},
{'purchase_date': '2025-02-16'},
{'purchase_date': '2025-02-23'},
{'purchase_date': '2024-04-14'},
{'purchase_date': '2024-04-13'},
{'purchase_date': '2024-10-19'},
{'purchase_date': '2024-12-26'},
{'purchase_date': '2024-10-27'},
{'purchase_date': '2025-02-14'},
{'purchase_date': '2024-07-11'},
{'purchase_date': '2024-05-09'},
{'purchase_date': '2024-06-28'},
{'purchase_date': '2025-03-03'},
{'purchase_date': '2024-03-30'},
{'purchase_date': '2024-08-10'},
{'purchase_date': '2025-01-04'},
{'purchase_date': '2024-10-20'},
{'purchase_date': '2024-08-22'},
{'purchase_date': '2024-10-21'},
{'purchase_date': '2025-01-31'},
{'purchase_date': '2024-05-07'},
{'purchase_date': '2024-12-14'},
{'purchase_date': '2025-03-10'},
{'purchase_date': '2025-01-05'},
{'purchase_date': '2024-09-04'},
{'purchase_date': '2024-04-24'},
{'purchase_date': '2024-07-24'},
{'purchase_date': '2024-05-30'},
{'purchase_date': '2025-01-14'},
{'purchase_date': '2024-04-07'},
{'purchase_date': '2024-07-03'},
{'purchase_date': '2024-04-19'},
{'purchase_date': '2024-08-20'},
{'purchase_date': '2024-09-01'},
{'purchase_date': '2025-01-13'},
{'purchase_date': '2025-01-18'},
{'purchase_date': '2024-09-25'},
```

```
{'purchase_date': '2024-05-29'},
{'purchase_date': '2024-04-23'},
{'purchase_date': '2024-08-18'},
{'purchase_date': '2024-04-03'},
{'purchase_date': '2024-07-01'},
{'purchase_date': '2024-10-05'},
{'purchase_date': '2024-08-07'},
{'purchase_date': '2025-02-02'},
{'purchase_date': '2024-09-15'},
{'purchase_date': '2024-10-01'},
{'purchase_date': '2024-07-13'},
{'purchase_date': '2025-02-08'},
{'purchase_date': '2024-08-21'},
{'purchase_date': '2024-08-17'},
{'purchase_date': '2024-09-08'},
{'purchase_date': '2024-04-04'},
{'purchase_date': '2024-07-28'},
{'purchase_date': '2025-03-22'},
{'purchase_date': '2024-09-16'},
{'purchase_date': '2024-11-11'},
{'purchase_date': '2025-02-18'},
{'purchase_date': '2024-09-09'},
{'purchase_date': '2024-08-08'},
{'purchase_date': '2024-04-18'},
{'purchase_date': '2024-07-29'},
{'purchase_date': '2024-06-22'},
{'purchase_date': '2024-07-30'},
{'purchase_date': '2025-03-20'},
{'purchase_date': '2025-03-14'},
{'purchase_date': '2024-04-28'},
{'purchase_date': '2025-03-16'},
{'purchase_date': '2025-01-02'},
{'purchase_date': '2024-06-19'},
{'purchase_date': '2024-10-24'},
{'purchase_date': '2024-07-08'},
{'purchase_date': '2024-05-11'},
{'purchase_date': '2025-01-15'},
{'purchase_date': '2024-08-30'},
{'purchase_date': '2024-10-29'},
{'purchase_date': '2024-04-16'},
{'purchase_date': '2024-07-02'},
{'purchase_date': '2024-05-28'},
{'purchase_date': '2024-09-05'},
{'purchase_date': '2024-10-12'},
{'purchase_date': '2025-03-27'},
{'purchase_date': '2024-07-04'},
{'purchase_date': '2024-12-13'}]
```