

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ

Национальный исследовательский ядерный университет «МИФИ»



Институт интеллектуальных кибернетических систем

Кафедра кибернетики (№ 22)

Отчёт о работе по курсу
«Базы данных (теоретические основы баз данных)»
Вариант «Twitch»

Выполнил	Михайлов А.А., Давыдов И.И, Рябов Н.А
Группа	Б22-504/511
Вариант	Twitch
Преподаватель	Петровская А.В
Проверяющий	
Оценка	

Оглавление

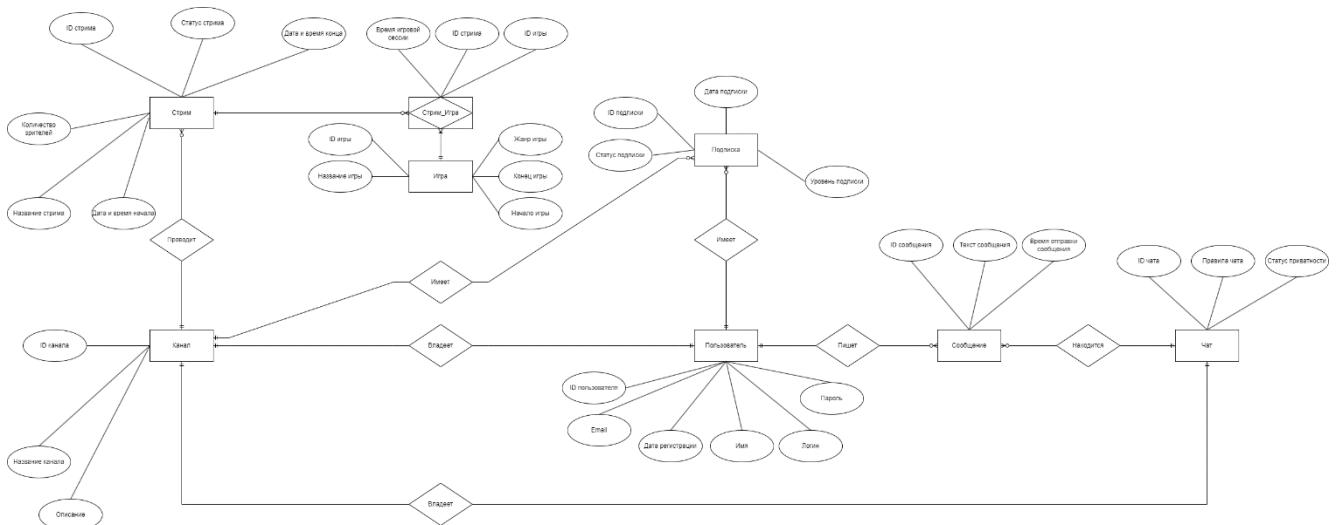
1. Формулировка задания	3
2. Концептуальная модель базы данных	3
2.1. Конкретизация предметной области.....	3
2.2. Описание предметной области.....	4
2.3 Описание атрибутов	4
3. Логическое проектирование.....	6
4. Физическое проектирование	8
4.1 Создание таблиц	8
4.2 Заполнение таблиц.....	10
5. Выполнение запросов	20

1. Формулировка задания

Спроектировать базу данных для стримингового сервиса “Twitch”. База должна содержать информацию о пользователях, подписках, сообщениях, чате, каналах, стримах и игре.

2. Концептуальная модель базы данных

В ходе работы над заданием была проанализирована предметная область, особенности которой отражает построенная концептуальная модель:



2.1. Конкретизация предметной области

Необходимо создать систему, отражающую основные связи стримингового сервиса “Twitch”, смысл которой заключается в обеспечении доступа к видео стримам, находящимся в базе стримов «Twitch», пользователям, которые его используют. Данная система учитывает видеотрансляции, текстовый чат и подписки. Также учитывается еще уровень подписки и статус стрима.

2.2. Описание предметной области

Сервис “Twitch” обеспечивает возможность пользователям смотреть стримы. Пользователи должны иметь возможность просмотра всех возможных стримов, а также возможность просмотра всех стримов пользователей, на которых он подписан. Также у пользователя будет возможность подписаться на какие-то подписки.

Рассмотрим возможности пользователя:

1. Владение каналом, который есть в базе данных “Twitch”.
2. Написание сообщения в общедоступный чат/чат для фолловеров.
3. Подписка на каналы других пользователей.
4. Возможность оформить платную подписку на какого-то стримера.

Исходя из описания возможностей, которые предоставляет стриминговый сервис «Twitch» были выделены следующие сущности:

- Пользователь (User)
- Подписка (Subscription)
- Сообщение(Message)
- Чат (Chat)
- Канал(Channel)
- Стream (Stream)
- Стream-игра (Stream game)
- Игра (Game)

2.3 Описание атрибутов

Для сущности Пользователь (User):

Атрибут	Описание
User_ID	Id пользователя.
Username	Имя пользователя.
Email	Почта пользователя, к которой привязан аккаунт
Registration date	Дата регистрации пользователя
Login	Логин для учетной записи пользователя
Password	Пароль для учетной записи пользователя

Для сущности Подписка (Subscription):

Атрибут	Описание
Id	id подписки.
Subsciptio_status	Статус подписки.
Subsciptio_date	Дата подписки.
Subsciptio_level	Уровень подписки.

Для сущности сообщения (messages):

Атрибут	Описание
Text	Текст сообщения.
Timestamp	Время отправки сообщения.

Для сущности Чат (Chat):

Атрибут	Описание
Id	Id чата.
Rules	Правила чата.
Privacy status	Статус приватности.

Для сущности Канал (Channel):

Атрибут	Описание
Id	Id канала.
Name	Название канала.
Description	Описание канала.

Для сущности Стим(Stream):

Атрибут	Описание
Id	Id стрима.
Статус	Статус стрима.
Number of viewers	Количество зрителей.
Start date time	Дата и время начала
End date time	Дата и время конца
Title	Название стрима

Для сущности Стима-игра(Stream-game):

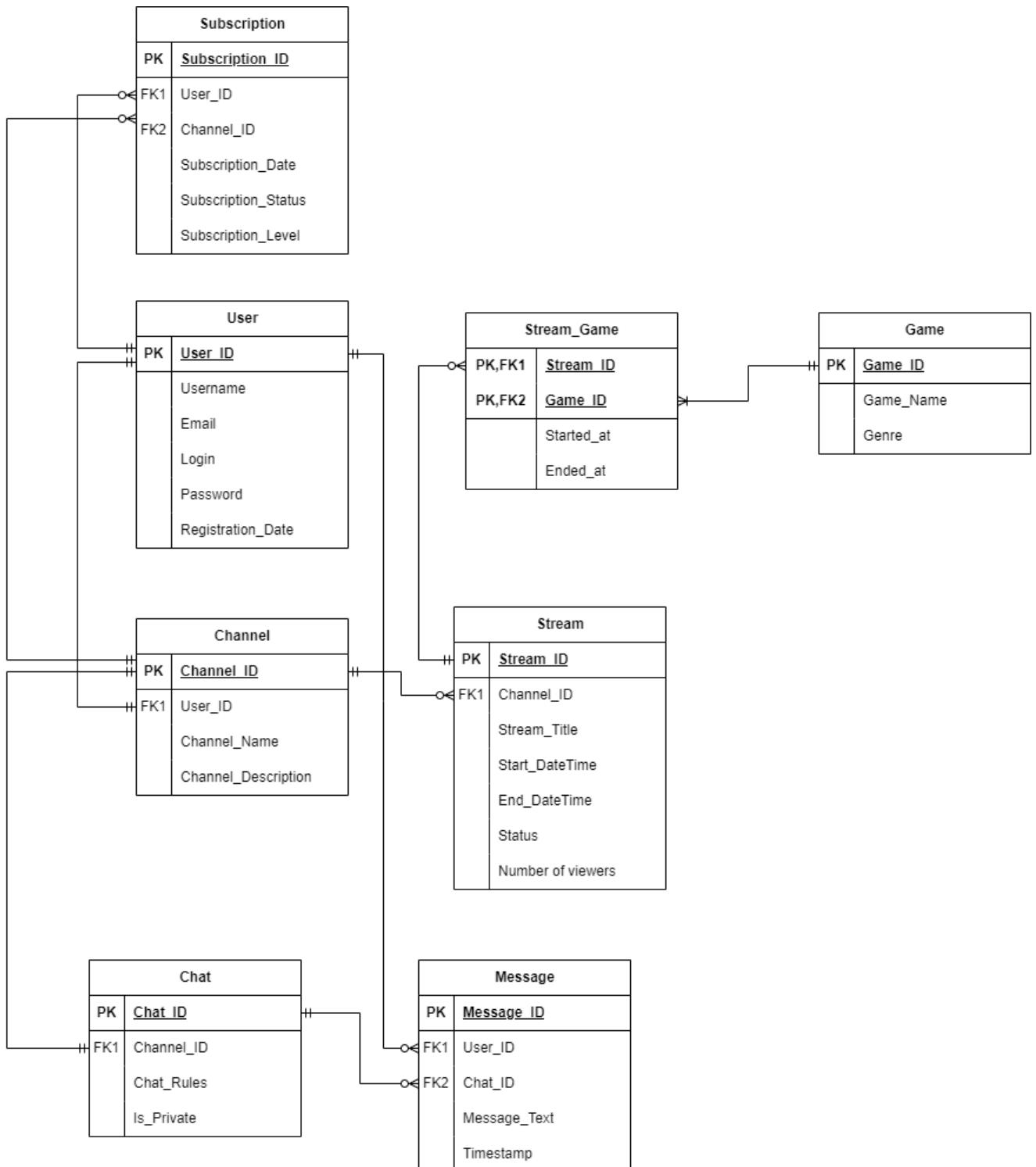
Атрибут	Описание
Id stream	id стрима
Id game	id игры

Для сущности Игра (Game):

Атрибут	Описание
Game name	Название игры
Genre	Жанр игры

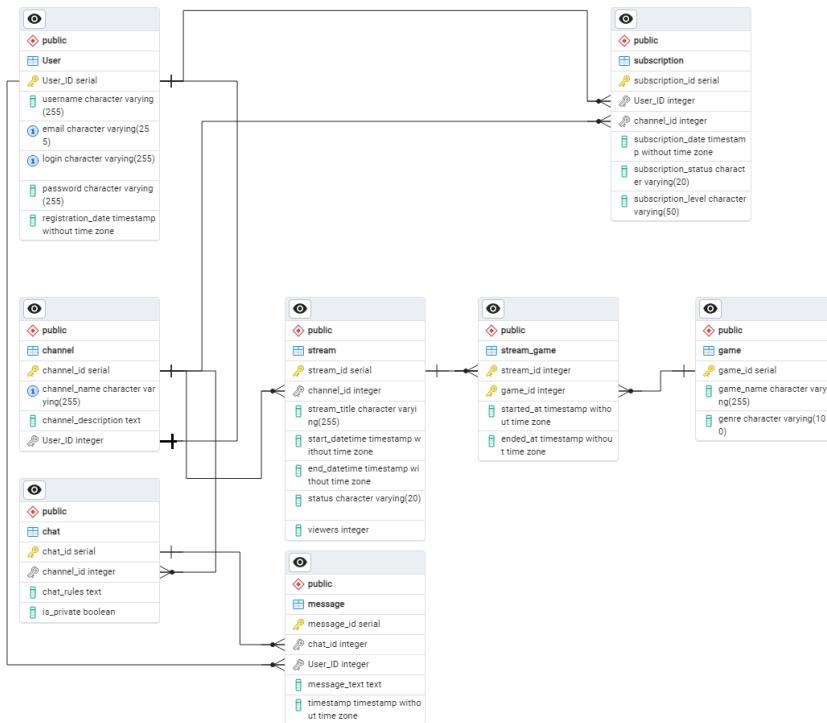
3. Логическое проектирование

Далее, на основе КМПО была разработана логическая модель базы данных, все связи «многие ко многим» были реализованы в виде отдельных таблиц. Данная логическая модель полностью соответствует ЗНФ: каждая таблица содержит неповторяющиеся значения и первичные ключи, при этом каждый неосновной атрибут каждого отношения функционально полно зависит от первичного его ключа.



4. Физическое проектирование

В качестве СУБД для реализации разработанной базы данных был выбран PostgreSQL. Исходя из приведенной логической модели была построена следующая физическая модель базы данных.



4.1 Создание таблиц

1)Создание таблицы Users:

```
CREATE TABLE "User" (
    "User_ID" SERIAL PRIMARY KEY,
    Username VARCHAR(255) NOT NULL,
    Email VARCHAR(255) NOT NULL UNIQUE,
    Login VARCHAR(255) NOT NULL UNIQUE,
    Password VARCHAR(255) NOT NULL,
    Registration_Date TIMESTAMP NOT NULL
);
```

2)Создание таблицы Subscription:

```
CREATE TABLE Subscription (
    Subscription_ID SERIAL PRIMARY KEY,
    "User_ID" INT NOT NULL REFERENCES "User"("User_ID"),
    Channel_ID INT NOT NULL REFERENCES Channel(Channel_ID),
    Subscription_Date TIMESTAMP NOT NULL,
    Subscription_Status VARCHAR(20) CHECK (Subscription_Status IN ('Active', 'Inactive')),
    Subscription_Level VARCHAR(50) CHECK (Subscription_Level IN ('Follower', 'Subscriber'))
);
```

3) Создание таблицы Message:

```
CREATE TABLE Message (
    Message_ID SERIAL PRIMARY KEY,
    Chat_ID INT NOT NULL REFERENCES Chat(Chat_ID),
    "User_ID" INT NOT NULL REFERENCES "User"("User_ID"),
    Message_Text TEXT NOT NULL,
    Timestamp TIMESTAMP NOT NULL
);
```

4) Создание таблицы Chat:

```
CREATE TABLE Chat (
    Chat_ID SERIAL PRIMARY KEY,
    Channel_ID INT NOT NULL REFERENCES Channel(Channel_ID),
    Chat_Rules TEXT,
    Is_Private BOOLEAN DEFAULT FALSE
);
```

5) Создание таблицы Channel:

```
CREATE TABLE Channel (
    Channel_ID SERIAL PRIMARY KEY,
    Channel_Name VARCHAR(255) NOT NULL UNIQUE,
    Channel_Description TEXT,
    "User_ID" INT NOT NULL UNIQUE REFERENCES "User"("User_ID") ON DELETE CASCADE
);
```

6) Создание таблицы Stream:

```
CREATE TABLE Stream (
    Stream_ID SERIAL PRIMARY KEY,
    Channel_ID INT NOT NULL REFERENCES Channel(Channel_ID),
    Stream_Title VARCHAR(255) NOT NULL,
    Start_DateTime TIMESTAMP NOT NULL,
    End_DateTime TIMESTAMP,
    Status VARCHAR(20) CHECK (Status IN ('Active', 'Completed')),
    Viewers INT DEFAULT 0
);
```

7) Создание таблицы Stream-game:

```
CREATE TABLE Stream_Game (
    Stream_ID INT NOT NULL REFERENCES Stream(Stream_ID),
    Game_ID INT NOT NULL REFERENCES Game(Game_ID),
    Started_At TIMESTAMP NOT NULL,
    Ended_At TIMESTAMP,
    PRIMARY KEY (Stream_ID, Game_ID)
);
```

8) Создание таблицы Game:

```
CREATE TABLE Game (
    Game_ID SERIAL PRIMARY KEY,
    Game_Name VARCHAR(255) NOT NULL,
    Genre VARCHAR(100)
);
```

4.2 Заполнение таблиц

Для заполнения базы данных тестовыми данными был написан скрипт на языке программирования Python3 с использованием драйвера взаимодействия с СУБД PostgreSQL psycopg2. Случайные данные для вставки генерировались с помощью библиотеки faker.

Суть работы скрипта заключается в создании строк, содержащих тело запроса, которые далее необходимо применить к базе данных в СУБД. После чего с помощью драйвера psycopg2 осуществляется подключение к базе данных, внутри транзакции исполняется сформированный запрос, после чего подключение закрывается.

Ниже приведены примеры функций заполнения и результаты заполнения:

1) Таблица User:

```
# Функция для вставки данных в таблицу "User"
def insert_users(cur, count=200000):
    # Используем множества для отслеживания уникальных значений
    unique_logins = set()
    unique_emails = set()

    for _ in range(count):
        while True:
            login = fake.user_name()
            email = fake.email()
            if login not in unique_logins and email not in unique_emails:
                unique_logins.add(login)
                unique_emails.add(email)
                break # Уникальные значения найдены, можно выйти из цикла

        username = fake.user_name()
        password = fake.password()
        registration_date = fake.date_time_this_year()

        cur.execute("""
            INSERT INTO "User" (Username, Email, Login, Password, Registration_Date)
            VALUES (%s, %s, %s, %s, %s)
        """, (username, email, login, password, registration_date))
```

	User_ID [PK] integer	username character varying (255)	email character varying (255)	login character varying (255)	password character varying (255)	registration_date timestamp without time zone
1	1	ngonzales	qclarke@example.com	daviddixon	3cB6GH&iC@	2024-09-29 14:02:05
2	2	livers	frances81@example.com	who)Gb7TZxwsB	2024-08-03 11:43:18
3	3	tonywelch	amandaburgess@example.net	xriley	W#1Wisul_#	2024-05-25 16:44:09
4	4	srichardson	ashley44@example.net	dharvey	\$7BeUr8P\$G	2024-10-01 03:30:37
5	5	timothyjohnson	hdavis@example.com	ibooker	ZS(\$A0VhZv	2024-07-14 01:57:48
6	6	laura26	lisa20@example.com	wjohnson	9007!zLqu!	2024-07-20 14:55:31
7	7	brandon70	timothydillon@example.com	xblair	P^24mfYdKj	2024-07-10 07:24:33
8	8	iwilson	jonescathy@example.net	matthew74	Fr(3WiF3oM	2024-11-05 02:46:53
9	9	gthomas	thomas99@example.net	franklinalexis	c0(tC@Os^f	2024-06-24 23:59:13
10	10	rickymitchell	bruce84@example.org	vperry	R6iAoIag*D	2024-05-29 14:36:21
11	11	williamsgreg	lisa49@example.com	robertvasquez	^X2M4Tyal9	2024-04-10 20:36:01
12	12	dustingarrett	zgarcia@example.com	moorewillie	_2EUIkNaa1	2024-08-26 04:46:02
13	13	amandacook	taylorcaroline@example.org	wolson	il!)n0Rg1J	2024-09-23 11:49:34
14	14	russellallison	bsanders@example.org	williamanthony	*T1b\$)Wux*	2024-07-13 09:37:53
15	15	rowedarius	carrie77@example.com	pagedanielle	^^bVLkzW\$6	2024-09-19 18:28:01

2) Таблица Subscription:

```
# Функция для вставки данных в таблицу Subscription
def insert_subscriptions(cur, count=500000, min_followers=50, max_followers=10000):
    # Получаем списки пользователей и каналов
    cur.execute('SELECT "User_ID" FROM "User"')
    user_ids = [row[0] for row in cur.fetchall()]
    print(f"Fetched {len(user_ids)} users")

    cur.execute('SELECT Channel_ID FROM Channel')
    channel_ids = [row[0] for row in cur.fetchall()]
    print(f"Fetched {len(channel_ids)} channels")

    existing_subscriptions = set()
    subscriptions = []
    remaining_subscriptions = count

    # Шаг 1: добавляем случайное количество подписчиков для каждого канала (в диапазоне от min_followers до max_followers)
    for i, channel_id in enumerate(channel_ids, 1):
        # Определяем случайное количество подписчиков для текущего канала
        num_followers = random.randint(min_followers, min(max_followers, len(user_ids)))
        chosen_users = random.sample(user_ids, num_followers)

        for user_id in chosen_users:
            if (user_id, channel_id) not in existing_subscriptions:
                subscription_date = fake.date_time_this_year()
                subscription_status = random.choice(['Active', 'Inactive'])
                subscription_level = random.choice(['Follower', 'Subscriber'])

                subscriptions.append(
                    (user_id, channel_id, subscription_date, subscription_status, subscription_level)
                )
                existing_subscriptions.add((user_id, channel_id))
                remaining_subscriptions -= 1

        # Вставляем подписки пакетами по 1000 записей для ускорения
        if len(subscriptions) >= 1000:
            execute_values(cur, """
                INSERT INTO Subscription ("User_ID", Channel_ID, Subscription_Date, Subscription_Status, Subscription_Level)
                VALUES %s
            """, subscriptions)
            subscriptions.clear()

    # Прерываем цикл, если достигли нужного количества подписок
    if remaining_subscriptions <= 0:
        print("Reached target subscription count.")
        break

    # Шаг 2: распределяем оставшиеся подписки случайным образом по всем каналам
    while remaining_subscriptions > 0:
        user_id = random.choice(user_ids)
        channel_id = random.choice(channel_ids)

        if (user_id, channel_id) not in existing_subscriptions:
            existing_subscriptions.add((user_id, channel_id))
            subscription_date = fake.date_time_this_year()
            subscription_status = random.choice(['Active', 'Inactive'])
            subscription_level = random.choice(['Follower', 'Subscriber'])
```

```

        subscriptions.append(
            (user_id, channel_id, subscription_date, subscription_status, subscription_level)
        )
        remaining_subscriptions -= 1

        # Вставка остатка подписок пакетами по 1000
        if len(subscriptions) >= 1000:
            execute_values(cur, """
                INSERT INTO Subscription ("User_ID", Channel_ID, Subscription_Date, Subscription_Status, Subscription_Level)
                VALUES %s
            """, subscriptions)
            subscriptions.clear()

        # Вставка оставшихся подписок, если их меньше 1000
        if subscriptions:
            execute_values(cur, """
                INSERT INTO Subscription ("User_ID", Channel_ID, Subscription_Date, Subscription_Status, Subscription_Level)
                VALUES %s
            """, subscriptions)

    print("All subscriptions inserted successfully!")

```

	subscription_id [PK] integer	User_ID integer	channel_id integer	subscription_date timestamp without time zone	subscription_status character varying (20)	subscription_level character varying (50)
1	1	18365	1	2024-02-19 15:44:37	Inactive	Follower
2	2	157994	1	2024-07-25 18:00:14	Inactive	Follower
3	3	178540	1	2024-06-11 00:37:05	Inactive	Follower
4	4	47938	1	2024-01-02 14:06:49	Active	Follower
5	5	142773	1	2024-06-27 20:19:27	Inactive	Subscriber
6	6	91932	1	2024-11-10 18:54:27	Inactive	Subscriber
7	7	12544	1	2024-10-21 06:04:00	Active	Subscriber
8	8	163905	1	2024-03-10 19:57:07	Active	Subscriber
9	9	2292	1	2024-02-02 10:41:16	Inactive	Follower
10	10	161060	1	2024-03-06 23:20:59	Active	Follower
11	11	72292	1	2024-08-31 17:01:36	Active	Follower
12	12	20966	1	2024-05-05 17:13:56	Active	Follower
13	13	90791	1	2024-01-10 11:02:22	Inactive	Follower
14	14	136416	1	2024-05-29 13:13:14	Active	Follower
15	15	115382	1	2024-09-22 21:32:29	Inactive	Follower

3) Таблица Message:

```
# Функция для вставки данных в таблицу Message
def insert_messages(cur, count=200000):
    cur.execute('SELECT Chat_ID FROM Chat')
    chat_ids = [row[0] for row in cur.fetchall()]

    cur.execute('SELECT "User_ID" FROM "User"')
    user_ids = [row[0] for row in cur.fetchall()]

    for _ in range(count):
        chat_id = random.choice(chat_ids)
        user_id = random.choice(user_ids)
        message_text = fake.sentence()
        timestamp = fake.date_time_this_year()
        cur.execute("""
            INSERT INTO Message (Chat_ID, "User_ID", Message_Text, Timestamp)
            VALUES (%s, %s, %s, %s)
        """, (chat_id, user_id, message_text, timestamp))
```

	message_id [PK] integer	chat_id integer	User_ID integer	message_text text	timestamp timestamp without time zone
1	1	73265	68198	Shake concern may.	2024-11-11 00:14:37
2	2	43059	185754	Pay large institution until social fish.	2024-03-08 23:51:39
3	3	55550	91567	People certainly safe traditional.	2024-09-29 00:24:16
4	4	184387	61610	Character factor deal attorney wait.	2024-10-24 20:21:55
5	5	130742	25943	Student difference score space available loss.	2024-05-18 12:03:59
6	6	44533	113900	Raise medical expert.	2024-09-09 11:50:00
7	7	75292	64647	Decade close unit rich pull.	2024-11-03 15:10:12
8	8	53515	114871	Phone daughter opportunity.	2024-08-07 11:14:35
9	9	129308	115305	She my back loss yet than.	2024-07-18 08:00:24
10	10	49697	35723	Behavior choose model moment beautiful girl show.	2024-07-31 04:11:47
11	11	72627	32609	Early chair year hard write gas it assume.	2024-01-29 18:31:13
12	12	138437	114815	Relate respond act move wish treatment.	2024-08-25 09:35:33
13	13	50806	115656	Write store president.	2024-03-28 23:22:28
14	14	151819	4993	Fill throw lot anything.	2024-08-22 08:00:40
15	15	24463	85255	Air modern once them develop arm involve.	2024-03-04 05:49:58

4) Таблица Chat:

```

# Функция для вставки данных в таблицу Chat
def insert_chats(cur):
    cur.execute('SELECT Channel_ID FROM Channel')
    channel_ids = [row[0] for row in cur.fetchall()]

    for channel_id in channel_ids:
        chat_rules = f"Rules for channel {channel_id}"
        is_private = random.choice([True, False])
        cur.execute("""
            INSERT INTO Chat (Channel_ID, Chat_Rules, Is_Private)
            VALUES (%s, %s, %s)
        """, (channel_id, chat_rules, is_private))

```

	chat_id [PK] integer	channel_id integer	chat_rules text	is_private boolean
1	1	1	Rules for channel 1	false
2	2	2	Rules for channel 2	false
3	3	3	Rules for channel 3	true
4	4	4	Rules for channel 4	false
5	5	5	Rules for channel 5	false
6	6	6	Rules for channel 6	false
7	7	7	Rules for channel 7	true
8	8	8	Rules for channel 8	true
9	9	9	Rules for channel 9	true
10	10	10	Rules for channel 10	true
11	11	11	Rules for channel 11	true
12	12	12	Rules for channel 12	false
13	13	13	Rules for channel 13	true
14	14	14	Rules for channel 14	true
15	15	15	Rules for channel 15	true

5) Таблица Channel:

```
# Функция для вставки данных в таблицу Channel
def insert_channels(cur):
    cur.execute('SELECT "User_ID" FROM "User"')
    user_ids = [row[0] for row in cur.fetchall()]
    random.shuffle(user_ids)

    # Извлечение всех существующих имен каналов
    cur.execute('SELECT Channel_Name FROM Channel')
    existing_channel_names = set(row[0] for row in cur.fetchall())

    for user_id in user_ids:
        while True:
            channel_name = fake.company()
            if channel_name not in existing_channel_names:
                existing_channel_names.add(channel_name)
                break # Найдено уникальное имя канала

        channel_description = fake.text()

        cur.execute("""
            INSERT INTO Channel (Channel_Name, Channel_Description, "User_ID")
            VALUES (%s, %s, %s)
        """, (channel_name, channel_description, user_id))
```

	channel_id [PK] integer	channel_name character varying (255)	channel_description text	User_ID integer
1	1	Powers, Hernandez and Raymond	Dream around hand information.	140204
2	2	Nelson Ltd	Clear bit garden move public. Myself range research above attorney most stop. Receive plant bed suggest return happen.	71750
3	3	Sims Inc	Campaign kind heart door resource. Impact choose likely ground. Air specific step war inside.	131286
4	4	Johnson-Larsen	Since appear thank population anyone common form. Daughter clear future. Personal chance public stop class speak place.	49821
5	5	Thomas Inc	Husband federal process their born argue. Nothing almost process hand onto.	97693
6	6	Guerra PLC	Recent beat turn several degree structure. Admit our last. Event school capital rule college.	22896
7	7	Gibbs, Johnson and Martin	Interesting return provide view water. Page interesting different first partner establish.	192229
8	8	Matthews and Sons	Here we real entire north. Food movie talk accept available. Physical our determine finally.	47979
9	9	Shields PLC	Their remain imagine green. Already voice claim which industry relate never.	15357
10	10	Coleman LLC	When sell land challenge. Executive describe lawyer accept sport side class.	184223
11	11	Pena-Carter	Cultural wait Republican garden.	78427
12	12	Ochoa LLC	Attention may half poor brother. Most make war officer. Specific serve development.	9369
13	13	Jones-Bernard	Nature a your adult although wall. Travel executive enjoy.	66110
14	14	Nicholson-Reed	Month affect worker simple career customer. Necessary street watch increase arrive even should. Research someone decide difference pretty.	122340
15	15	Ward and Sons	Speech course page these war choice it cost. Miss various fine.	14778

6) Таблица Stream:

```
# ФУНКЦИЯ ДЛЯ ВСТАВКИ ДАННЫХ В ТАБЛИЦУ Stream
def insert_streams(cur, count=100000):
    cur.execute('SELECT Channel_ID FROM Channel')
    channel_ids = [row[0] for row in cur.fetchall()]

    for _ in range(count):
        channel_id = random.choice(channel_ids)
        stream_title = fake.sentence(nb_words=4)
        start_time = fake.date_time_this_year()
        end_time = start_time + timedelta(hours=random.randint(a: 1, b: 8))
        status = random.choice(['Active', 'Completed'])
        viewers = random.randint(a: 0, b: 10000)
        cur.execute("""
            INSERT INTO Stream (Channel_ID, Stream_Title, Start_DateTime, End_DateTime, Status, Viewers)
            VALUES (%s, %s, %s, %s, %s, %s)
        """, (channel_id, stream_title, start_time, end_time, status, viewers))
```

	stream_id [PK] integer	channel_id integer	stream_title character varying (255)	start_datetime timestamp without time zone	end_datetime timestamp without time zone	status character varying (20)	viewers integer
1	1	3456	Positive day.	2024-01-04 22:05:19	2024-01-05 06:05:19	Active	7311
2	2	159684	Specific what another item hear.	2024-03-08 08:12:04	2024-03-08 13:12:04	Completed	327
3	3	103232	Month he might.	2024-01-28 14:49:58	2024-01-28 16:49:58	Active	8166
4	4	185875	Figure after message free.	2024-05-05 17:53:20	2024-05-06 01:53:20	Active	4847
5	5	11468	Everybody discover direction father.	2024-05-03 14:45:31	2024-05-03 19:45:31	Completed	9989
6	6	161379	Realize owner true international bit.	2024-02-01 15:01:50	2024-02-01 19:01:50	Active	9928
7	7	3664	Hair enter enter century.	2024-10-22 01:36:36	2024-10-22 04:36:36	Completed	686
8	8	191134	Art education carry.	2024-07-01 13:17:55	2024-07-01 19:17:55	Active	2700
9	9	159860	While authority.	2024-07-05 06:51:47	2024-07-05 09:51:47	Active	8981
10	10	82721	Paper wonder especially last.	2024-03-27 08:56:46	2024-03-27 15:56:46	Active	7121
11	11	187532	Even choice occur know.	2024-06-04 20:55:16	2024-06-04 21:55:16	Active	6267
12	12	88950	Particular human.	2024-07-19 11:34:48	2024-07-19 18:34:48	Active	3622
13	13	54693	After forget music.	2024-06-21 12:19:05	2024-06-21 18:19:05	Active	9365
14	14	7380	Mission commercial tough store.	2024-09-04 01:11:13	2024-09-04 05:11:13	Completed	1904
15	15	1920	Camera management.	2024-07-12 03:28:49	2024-07-12 09:28:49	Active	6514

7) Таблица Game:

```
# Функция для вставки данных в таблицу Game
def insert_games(cur, count=100000):
    genres = ['Action', 'Adventure', 'Puzzle', 'RPG', 'Strategy']

    for _ in range(count):
        game_name = fake.word().capitalize() + ' ' + fake.word().capitalize()
        genre = random.choice(genres)
        cur.execute("""
            INSERT INTO Game (Game_Name, Genre)
            VALUES (%s, %s)
        """, (game_name, genre))
```

	game_id [PK] integer	game_name character varying (255)	genre character varying (100)
1	1	Second Big	RPG
2	2	Require Difficult	Strategy
3	3	Life Along	Strategy
4	4	Respond Necessary	Adventure
5	5	Fight Machine	Puzzle
6	6	After Get	Puzzle
7	7	Less Staff	Puzzle
8	8	Voice Tonight	RPG
9	9	Election This	Strategy
10	10	Change Adult	Strategy
11	11	Bring Ok	Adventure
12	12	Simple Whom	Action
13	13	Sort Would	Adventure
14	14	Power Man	Puzzle
15	15	Rule Wall	Puzzle

8) Таблица Stream_Game:

```
# Функция для вставки данных в таблицу Stream_Game
def insert_stream_games(cur, count=100000):
    # Получаем все возможные Stream_ID и Game_ID
    cur.execute('SELECT Stream_ID FROM Stream')
    stream_ids = [row[0] for row in cur.fetchall()]

    cur.execute('SELECT Game_ID FROM Game')
    game_ids = [row[0] for row in cur.fetchall()]

    # Множество для отслеживания уникальных комбинаций (Stream_ID, Game_ID)
    stream_games = set()

    for _ in range(count):
        while True:
            stream_id = random.choice(stream_ids)
            game_id = random.choice(game_ids)

            # Проверка на уникальность пары (stream_id, game_id)
            if (stream_id, game_id) not in stream_games:
                stream_games.add((stream_id, game_id))
                break # Уникальная комбинация найдена, выходим из цикла

# Генерация временных меток
started_at = fake.date_time_this_year()
ended_at = started_at + timedelta(hours=random.randint(1, 8))

# Вставка данных
cur.execute("""
    INSERT INTO Stream_Game (Stream_ID, Game_ID, Started_At, Ended_At)
    VALUES (%s, %s, %s, %s)
""", (stream_id, game_id, started_at, ended_at))
```

	stream_id [PK] integer	game_id [PK] integer	started_at timestamp without time zone	ended_at timestamp without time zone
1	25760	21776	2024-02-18 18:45:59	2024-02-18 21:45:59
2	84792	30527	2024-07-28 23:15:46	2024-07-29 06:15:46
3	2156	99945	2024-07-23 19:25:22	2024-07-24 00:25:22
4	54583	41786	2024-10-13 03:25:19	2024-10-13 05:25:19
5	6295	75412	2024-04-14 18:57:09	2024-04-14 23:57:09
6	96216	53230	2024-03-03 14:03:43	2024-03-03 15:03:43
7	49644	71172	2024-10-10 21:21:41	2024-10-11 00:21:41
8	43590	51063	2024-05-22 13:32:41	2024-05-22 16:32:41
9	74266	75321	2024-04-02 18:34:44	2024-04-03 02:34:44
10	3032	21202	2024-09-10 09:44:17	2024-09-10 15:44:17
11	66103	74438	2024-06-23 04:13:53	2024-06-23 10:13:53
12	6650	19768	2024-07-14 12:25:09	2024-07-14 14:25:09
13	23666	37344	2024-01-03 09:18:28	2024-01-03 14:18:28
14	76756	41171	2024-07-14 19:14:05	2024-07-15 03:14:05
15	887	60902	2024-08-28 11:51:12	2024-08-28 18:51:12

5. Выполнение запросов

В этом разделе приведены запросы, которые были написаны к разработанной базе данных в рамках курса.

1)Получение списка всех пользователей с датой регистрации

```
SELECT Username, Registration_Date FROM "User";
```

	username character varying (255) 	registration_date timestamp without time zone 
1	ngonzales	2024-09-29 14:02:05
2	livers	2024-08-03 11:43:18
3	tonywelch	2024-05-25 16:44:09
4	srichardson	2024-10-01 03:30:37
5	timothyjohnson	2024-07-14 01:57:48
6	laura26	2024-07-20 14:55:31
7	brandon70	2024-07-10 07:24:33
8	iwilson	2024-11-05 02:46:53
9	gthomas	2024-06-24 23:59:13
10	rickymitchell	2024-05-29 14:36:21

2)Получение всех каналов и их владельцев.

```
SELECT Channel_Name, Username AS Owner
FROM Channel
JOIN "User" ON Channel."User_ID" = "User"."User_ID";
```

	channel_name character varying (255) 	owner character varying (255) 
1	Powers, Hernandez and Raymond	travisjohnson
2	Nelson Ltd	millerbrian
3	Sims Inc	msmith
4	Thomas Inc	kelleyedward
5	Matthews and Sons	ogaines
6	Coleman LLC	santosjoshua
7	Pena-Carter	rhodesdaniel
8	Ochoa LLC	allenbaird
9	Jones-Bernard	imaldonado
10	Nicholson-Reed	jocelynclark
11	Ward and Sons	smata
12	Harris-Williams	dguerrero
13	Sanchez, Webb and Roy	wross
14	Burns, Chapman and Franklin	carrie79
15	Davis, Cohen and Young	bgood

3)Получение всех активных стримов с их количеством зрителей.

```
SELECT Stream_Title, Viewers
FROM Stream
WHERE Status = 'Active';
```

	stream_title character varying (255)	viewers integer	
1	Positive day.	7311	
2	Month he might.	8166	
3	Figure after message free.	4847	
4	Realize owner true international bit.	9928	
5	Art education carry.	2700	
6	While authority.	8981	
7	Paper wonder especially last.	7121	
8	Even choice occur know.	6267	
9	Particular human.	3622	
10	After forget music.	9365	
11	Camera management.	6514	
12	Industry both.	1440	
13	Available off exactly improve cause.	6891	
14	Want where.	5017	
15	Practice home past.	9956	

4)Получение всех подписок пользователя по имени.

```
SELECT Username, Channel_Name, Subscription_Date
FROM Subscription
JOIN "User" ON Subscription."User_ID" = "User"."User_ID"
JOIN Channel ON Subscription.Channel_ID = Channel.Channel_ID
WHERE Username = 'michael58';
```

	username character varying (255)	channel_name character varying (255)	subscription_date timestamp without time zone	
1	michael58	Johnson-Larsen	2024-01-05 00:42:32	
2	michael58	Pena-Carter	2024-10-02 11:33:03	
3	michael58	Ochoa LLC	2024-11-11 05:29:28	
4	michael58	King, Duke and Lopez	2024-07-12 01:03:18	
5	michael58	Cook and Sons	2024-04-02 10:36:54	
6	michael58	Roy Inc	2024-08-08 18:44:46	
7	michael58	Huynh and Sons	2024-07-31 12:22:23	
8	michael58	Mcguire PLC	2024-09-13 01:30:25	
9	michael58	Weaver Ltd	2024-09-07 18:30:13	
10	michael58	Brown, Tate and Trevino	2024-02-10 16:49:06	

5)Получение всех сообщений из конкретного чата, отсортированных по времени.

```
SELECT Message_Text, Timestamp, Username
FROM Message
JOIN "User" ON Message."User_ID" = "User"."User_ID"
WHERE Chat_ID = 4
ORDER BY Timestamp;
```

	message_text text	timestamp timestamp without time zone	username character varying (255)
1	Professor will everybody.	2024-05-19 20:04:57	patrick49
2	Eye suggest agreement ok should.	2024-10-04 19:12:27	graykimberly

6)Количество подписчиков у каждого канала

```
SELECT Channel_Name, COUNT(Subscription_ID) AS Subscriber_Count
FROM Subscription
JOIN Channel ON Subscription.Channel_ID = Channel.Channel_ID
GROUP BY Channel_Name;
```

	channel_name character varying (255)	subscriber_count bigint
1	Burns PLC	2967
2	Brooks PLC	4057
3	Snow, Potter and Harris	4390
4	Larsen, Rose and Gray	3897
5	Romero-Landry	8714
6	Brooks-Stone	1563
7	Gonzalez-Johnson	7564
8	Chang Inc	2949
9	Evans-Martin	3510
10	Scott Inc	2638

7)Самые популярные игры по количеству стримов.

```
SELECT Game_Name, COUNT(Stream_Game.Stream_ID) AS Stream_Count
FROM Game
JOIN Stream_Game ON Game.Game_ID = Stream_Game.Game_ID
GROUP BY Game_Name
ORDER BY Stream_Count DESC;
```

	game_name character varying (255) 	stream_count bigint 
1	Compare Person	9
2	Enter Particular	8
3	Brother Shoulder	8
4	Reality Rich	8
5	Heart Movie	8
6	Charge Real	8
7	Customer Song	8
8	Southern Series	7
9	Laugh Own	7
10	Talk Type	7

8)Получение суммарного количества зрителей со всех стримов конкретного канала.

```
SELECT Channel_Name, SUM(Viewers) AS Total_Viewers
FROM Stream
JOIN Channel ON Stream.Channel_ID = Channel.Channel_ID
WHERE Channel_Name = 'Anderson, OConnor and Lara'
GROUP BY Channel_Name;
```

	channel_name character varying (255) 	total_viewers bigint 
1	Anderson, OConnor and Lara	561

9)Количество сообщений в каждом чате.

```
SELECT Channel_Name, COUNT(Message_ID) AS Message_Count
FROM Message
JOIN Chat ON Message.Chat_ID = Chat.Chat_ID
JOIN Channel ON Chat.Channel_ID = Channel.Channel_ID
GROUP BY Channel_Name;
```

	channel_name character varying (255)	message_count bigint
1	Abbott-Alexander	2
2	Abbott-Bradshaw	2
3	Abbott-Bryant	1
4	Abbott-Carr	1
5	Abbott-Cunningham	2
6	Abbott-Dean	1
7	Abbott-Evans	1
8	Abbott-Finley	2
9	Abbott-Frye	1
10	Abbott-Garcia	1

10) Средняя длина сообщений в чате.

```
SELECT Chat_ID, AVG(LENGTH(Message_Text)) AS Avg_Message_Length
FROM Message
GROUP BY Chat_ID;
```

	chat_id integer	avg_message_length numeric
1	68453	26.000000000000000000
2	156494	29.500000000000000000
3	83264	60.000000000000000000
4	35599	50.000000000000000000
5	151812	56.000000000000000000
6	103729	37.500000000000000000
7	69906	38.000000000000000000
8	174416	43.000000000000000000
9	44127	34.000000000000000000
10	96868	58.000000000000000000

Более сложные запросы с подзапросами и фильтрацией:

11) Получение пользователей, подписанных на конкретный канал.

```
SELECT Username
FROM "User"
WHERE "User_ID" IN (
    SELECT "User_ID"
    FROM Subscription
    WHERE Channel_ID = (SELECT Channel_ID FROM Channel WHERE Channel_Name = 'Paul and Sons')
);
```

	username character varying (255) 
1	timothy65
2	gwilliams
3	alecsantos
4	patricia00
5	shannon93
6	xyoung
7	nolantracy
8	michael73
9	aprilmiller
10	allisonpratt

12)Получение всех каналов с количеством подписчиков и количеством стримов.

```
SELECT Channel_Name,
       COUNT(DISTINCT Subscription.Subscription_ID) AS Subscriber_Count,
       COUNT(DISTINCT Stream.Stream_ID) AS Stream_Count
FROM Channel
LEFT JOIN Subscription ON Channel.Channel_ID = Subscription.channel_ID
LEFT JOIN Stream ON Channel.Channel_ID = Stream.Channel_ID
GROUP BY Channel_Name
ORDER BY
    Subscriber_Count DESC;
```

	channel_name character varying (255) 	subscriber_count bigint 	stream_count bigint 
1	Villa, Henderson and Davis	9958	0
2	Stewart, Kim and Townsend	9876	0
3	Brooks-Lewis	9786	0
4	Warren-Jackson	9466	0
5	Ibarra, Rodriguez and Rodgers	9422	0
6	Nelson LLC	9127	0
7	Silva-Barr	9126	1
8	Paul and Sons	8833	1
9	Taylor-Wilkinson	8823	1
10	Wheeler, Brown and Mills	8787	0

13)Получение подписчиков, которые подписаны на более чем один канал.

```
SELECT "User"."User_ID", Username, COUNT(Subscription_ID) AS Subscription_Count
FROM "User"
JOIN Subscription ON "User"."User_ID" = Subscription."User_ID"
GROUP BY "User"."User_ID", Username
HAVING COUNT(Subscription_ID) > 1;
```

	User_ID [PK] integer	username character varying (255)	subscription_count bigint
1	151812	bowenjames	6
2	109863	lsheppard	4
3	79030	tyrone04	4
4	180028	englishjustin	3
5	174416	lisa64	3
6	86580	aparks	3
7	162195	michael04	2
8	185153	iwalsh	6
9	196633	pbryan	3
10	157514	jonathan16	3

14)Проверка подписок для обновления статуса на "Inactive".

```
SELECT *
FROM Subscription
WHERE Subscription_Date < (CURRENT_DATE - INTERVAL '6 months');
```

	subscription_id [PK] integer	User_ID integer	channel_id integer	subscription_date timestamp without time zone	subscription_status character varying (20)	subscription_level character varying (50)
1	4	54757	1	2024-05-02 21:15:38	Inactive	Subscriber
2	5	142161	1	2024-03-23 01:51:01	Inactive	Follower
3	7	110825	1	2024-05-12 09:36:48	Inactive	Subscriber
4	8	119365	1	2024-03-22 06:15:24	Inactive	Subscriber
5	12	43172	1	2024-03-28 23:37:37	Active	Follower
6	15	67320	1	2024-03-07 09:27:36	Inactive	Follower
7	23	66557	1	2024-03-22 21:19:26	Inactive	Follower
8	24	41481	1	2024-02-26 22:13:27	Active	Subscriber
9	27	170381	1	2024-01-01 11:52:41	Inactive	Follower
10	30	125901	1	2024-02-26 08:11:02	Inactive	Subscriber

15)Просмотр сообщений, которые старше полгода.

```
SELECT *
FROM Message
WHERE Timestamp < (CURRENT_DATE - INTERVAL '6 months');
```

	message_id [PK] integer	chat_id integer	User_ID integer	message_text text	timestamp timestamp without time zone
1	3	138815	50696	Expect church entire answer value interesting.	2024-01-01 19:54:53
2	5	29965	181893	Its bad institution different rise never office.	2024-01-26 21:05:28
3	6	104613	170674	Mission maintain maintain not indeed.	2024-04-29 02:34:00
4	9	181544	55657	Record hard economy security staff bill cut energy.	2024-05-03 23:33:58
5	10	55690	6978	Pass could miss despite.	2024-01-13 16:30:56
6	23	71193	95269	Head into if team.	2024-03-08 07:34:50
7	24	27425	12506	Real beat everything such career institution recent get.	2024-05-02 17:26:31
8	25	67611	84386	Wrong as common nation note surface power.	2024-03-05 23:00:50
9	26	70011	183738	Professional hand boy recent early position.	2024-01-05 00:11:12
10	32	81583	7294	Response medical worker visit wide so.	2024-02-04 04:12:38

16)Топ-10 каналов по количеству подписчиков.

```
SELECT
    Channel.Channel_Name,
    COUNT(Subscription.Subscription_ID) AS Subscriber_Count
FROM
    Channel
LEFT JOIN
    Subscription ON Channel.Channel_ID = Subscription.Channel_ID
GROUP BY
    Channel.Channel_ID, Channel.Channel_Name
ORDER BY
    Subscriber_Count DESC
LIMIT 10;
```

	channel_name character varying (255)	subscriber_count bigint
1	Villa, Henderson and Davis	9958
2	Stewart, Kim and Townsend	9876
3	Brooks-Lewis	9786
4	Warren-Jackson	9466
5	Ibarra, Rodriguez and Rodgers	9422
6	Nelson LLC	9127
7	Silva-Barr	9126
8	Paul and Sons	8833
9	Taylor-Wilkinson	8823
10	Wheeler, Brown and Mills	8787