

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА  
ВЕЛИКОГО»  
ВШ программной инженерии



**ПОЛИТЕХ**

Санкт-Петербургский  
политехнический университет  
Петра Великого

**ОТЧЕТ ЛР1\_2  
ВАРИАНТ 35**

по дисциплине “Системы управления базами данных”

Выполнила  
студентка гр. 3530202/00201

Козлова Е. А.

Руководитель

Прокофьев О. В.

## Table of Contents

<b>Задание 1.1 .....</b>	<b>3</b>
<b>Задание 1.2 .....</b>	<b>4</b>
<b>Задание 1.3 .....</b>	<b>6</b>
<b>Задание 1.4 .....</b>	<b>9</b>
<b>Задание 2.1 .....</b>	<b>12</b>
<b>Задание 2.2 .....</b>	<b>16</b>

## Задание 1.1

### Вариант №35

Предметная область для практических заданий №1.\* №2.\*: **Телевещание**

Пример схемы для задания №1.1,



Рисунок 1. Задание варианта

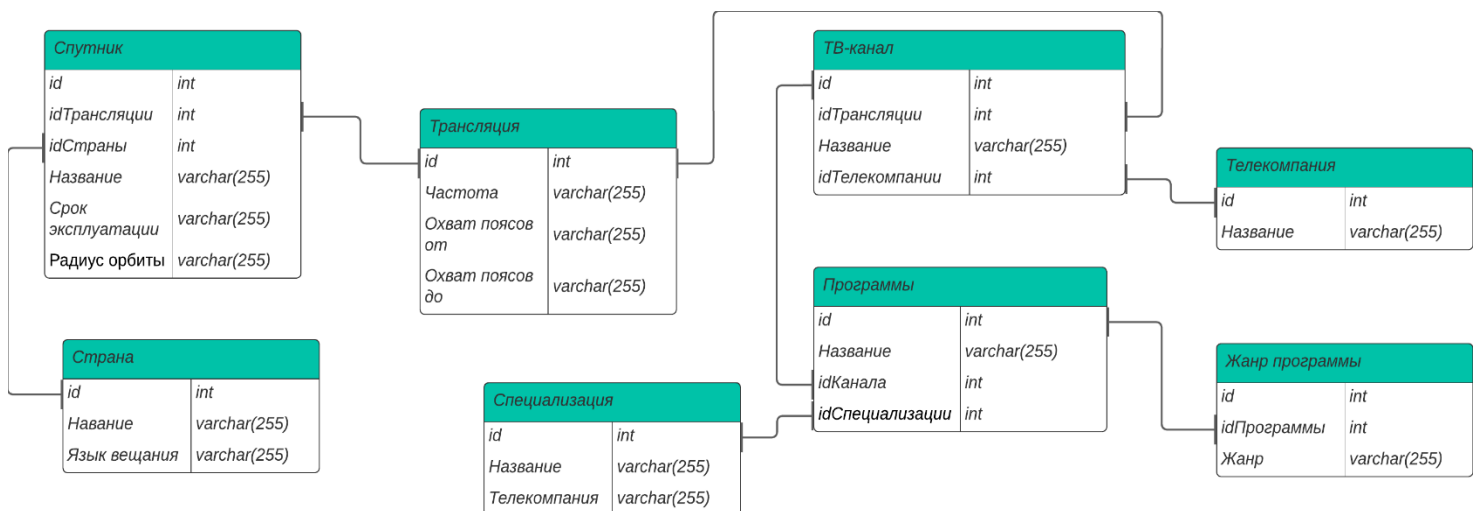


Рисунок 2. Рисунок ER диаграммы

## Задание 1.2

```
CREATE TABLE Satellite (  
    id INT PRIMARY KEY GENERATED ALWAYS AS IDENTITY,  
    idTranslation INT,  
    idCountry INT,  
    Name VARCHAR(255) NOT NULL UNIQUE,  
    Term_operation VARCHAR(255) NOT NULL,  
    Orbit_r VARCHAR(255) NOT NULL  
);  
  
INSERT INTO  
    Satellite (Name, idTranslation, idCountry, Term_operation, Orbit_r)  
VALUES  
    ('Астра 1Н', '2', '4', '15 лет', '36123'),  
    ('Астра 1KR', '3', '4', '15 лет', '36222'),  
    ('Астра 1L', '5', '3', '20 лет', '36789'),  
    ('Астра 2М', '3', '5', '25 лет', '36111'),  
    ('Хот Бёрд 6', '1', '3', '10 лет', '36120'),  
    ('Хот Бёрд 8', '3', '2', '20 лет', '36321'),  
    ('Хот Бёрд 9', '3', '2', '25 лет', '36543'),  
    ('Экспресс AM2', '5', '5', '10 лет', '36712'),  
    ('Экспресс-AM3', '9', '5', '10 лет', '35999'),  
    ('Сириус 4', '6', '4', '10 лет', '36327')  
  
CREATE TABLE Broadcast (  
    id INT PRIMARY KEY GENERATED ALWAYS AS IDENTITY,  
    Frequency VARCHAR(255) NOT NULL,  
    idCountry INT,  
    CofB_fr VARCHAR(255) NOT NULL,  
    CofB_to VARCHAR(255) NOT NULL  
);  
  
INSERT INTO  
    Broadcast (Frequency, idCountry, CofB_fr, CofB_to)  
VALUES  
    ('800 МГц', '1', '3', '6'),  
    ('750 МГц', '1', '4', '5'),  
    ('750 МГц', '2', '2', '7'),  
    ('800 МГц', '3', '3', '7'),  
    ('800 МГц', '2', '3', '6'),  
    ('750 МГц', '5', '5', '6'),  
    ('800 МГц', '5', '5', '6'),  
    ('750 МГц', '4', '5', '6'),  
    ('800 МГц', '4', '5', '7'),  
    ('800 МГц', '4', '3', '8')  
  
CREATE TABLE TVchannel (  
    id INT PRIMARY KEY GENERATED ALWAYS AS IDENTITY,  
    idTranslation INT,  
    Name VARCHAR(255) NOT NULL,  
    idTVcompany INT  
);  
  
INSERT INTO  
    TVchannel (idTranslation, Name, idTVcompany)  
VALUES
```

```

('3','CNN','1'),
('2','ESPN','1'),
('5','STAR World','2'),
('1','HBO','2'),
('7','Disney Channel','3'),
('5','Nickelodeon','4'),
('8','FOX','4'),
('9','MTV Music Television','3'),
('5','ESPN','4'),
('3','BBC','5')

```

```

CREATE TABLE Country (
    id INT PRIMARY KEY GENERATED ALWAYS AS IDENTITY,
    Name VARCHAR(255) NOT NULL UNIQUE,
    Language VARCHAR(255) NOT NULL
);

```

```

INSERT INTO
    Country (Name, Language)
VALUES
    ('USA','English'),
    ('Russia','Russian'),
    ('France','French'),
    ('England','English'),
    ('Germany','German')

```

```

CREATE TABLE Specialization (
    id INT PRIMARY KEY GENERATED ALWAYS AS IDENTITY,
    Name VARCHAR(255) NOT NULL UNIQUE,
    TVcompany VARCHAR(255) NOT NULL UNIQUE
);

```

```

INSERT INTO
    Specialization (Name, TVcompany)
VALUES
    ('news','D-Smart'),
    ('sports','Acheloos TV'),
    ('entertainment','2929 Entertainment'),
    ('documentary','Antix Productions')

```

```

CREATE TABLE Programs (
    id INT PRIMARY KEY GENERATED ALWAYS AS IDENTITY,
    Name VARCHAR(255) NOT NULL UNIQUE,
    idChanel INT,
    idSpecialization INT
);

```

```

INSERT INTO
    Programs (Name, idChanel, idSpecialization)
VALUES
    ('News 24','1','1'),
    ('Football','4','2'),
    ('Basketball','4','2'),
    ('Horse Racing','3','2'),
    ('Interns','2','3'),
    ('Truman Show','5','3'),
    ('Animal World','3','4'),
    ('Science from the Inside','1','4'),
    ('The World of Planets','1','4')

```

```
CREATE TABLE TVcompany (
    id INT PRIMARY KEY GENERATED ALWAYS AS IDENTITY,
    Name VARCHAR(255) NOT NULL UNIQUE
);
```

```
INSERT INTO
    TVcompany (Name)
VALUES
    ('D-Smart'),
    ('Acheloos TV'),
    ('2929 Entertainment'),
    ('Antix Productions')
```

```
CREATE TABLE Genre (
    id INT PRIMARY KEY GENERATED ALWAYS AS IDENTITY,
    Name VARCHAR(255) NOT NULL UNIQUE,
    idProgram INT UNIQUE
);
```

```
INSERT INTO
    Genre (Name, idProgram)
VALUES
    ('Movie', '5'),
    ('Reportage', '1'),
    ('Live Broadcast', '2'),
    ('TV Series', '6')
```

### Задание 1.3

```
--Вывести все ТВ каналы, которые транслируются на частоте 800 МГц
SELECT
    tvchannel.name AS "channel",
    broadcast.frequency AS "frequency"
FROM
    tvchannel
JOIN broadcast
    ON tvchannel.idtranslation = broadcast.id
WHERE broadcast.frequency = '800 МГц'
```

```
-- Вывести все программы, каналы которых специализируются на спорте
SELECT
    programs.name AS "program",
    tvchannel.name AS "channel",
    specialization.name AS "specialization"
FROM
    programs
JOIN tvchannel
    ON programs.idchannel = tvchannel.id
JOIN specialization
    ON programs.idspecialization = specialization.id
WHERE specialization.name = 'sports';
```

```
--Вывести каналы, которые транслируются на русском языке
SELECT
    tvchannel.name AS "channel",
    country.name AS "country",
    country.language AS "language"
FROM
```

```

        tvchannel
JOIN broadcast
    ON tvchannel.idtranslation = broadcast.id
JOIN satellite
    ON broadcast.id = satellite.idtranslation
JOIN country
    ON satellite.idcountry = country.id
WHERE country.language = 'Russian'

--Вывести каналы, на которых транслируются документальные фильмы на английском
SELECT
tvchannel.name AS "channel",
    country.language AS "language",
    programs.name AS "film"
FROM
    tvchannel
JOIN programs
    ON tvchannel.id = programs.idchannel
JOIN specialization
    ON programs.idspecialization = specialization.id
JOIN broadcast
    ON tvchannel.idtranslation = broadcast.id
JOIN satellite
    ON broadcast.id = satellite.idtranslation
JOIN country
    ON satellite.idcountry = country.id
WHERE country.language = 'German' and specialization.name = 'Documentary'

-- Вывести количество программ, транслирующихся с частотой 750 МГц и специализацией канала новости

WITH ProgSport AS (
    SELECT programs.id
    FROM programs
        JOIN specialization
            ON programs.idspecialization = specialization.id
    WHERE specialization.name = 'news'
    GROUP BY programs.id
), Prog750 AS (
    SELECT programs.id
    FROM programs
        JOIN tvchannel
            ON tvchannel.id = programs.idchannel
        JOIN broadcast
            ON tvchannel.idtranslation = broadcast.id
    WHERE broadcast.frequency = '750 МГц'
    GROUP BY programs.id
)
SELECT count(id) AS ProgSport, count(id) AS Prog750
FROM programs
    WHERE programs.id IN (SELECT programs.id FROM Prog750) OR programs.id IN (SELECT
programs.id FROM ProgSport)
    GROUP BY programs.id

--изменить канал, по которому транслируется программа
UPDATE programs
    SET idchannel = 3
    WHERE name = 'News 24';

UPDATE programs
    SET idchannel = 3

```

```
WHERE name = 'Science from the Inside';
```

```
UPDATE programs  
SET idchannel = 3  
WHERE name = 'Interns';
```

```
--удалить программу Truman Show из таблицы  
DELETE FROM programs WHERE name = 'Truman Show';
```



## Задание 1.4

Уровень изоляции	«Грязное» чтение	Неповторяемое чтение	Фантомное чтение	Аномалия сериализации
Read uncommitted (Чтение незафиксированных данных)	Допускается, но не в PG	Возможно	Возможно	Возможно
Read committed (Чтение зафиксированных данных)	Невозможно	Возможно	Возможно	Возможно
Repeatable read (Повторяемое чтение)	Невозможно	Невозможно	Допускается, но не в PG	Возможно
Serializable (Сериализуемость)	Невозможно	Невозможно	Невозможно	Невозможно

Исходные данные.

id	name	idchannel	idspecification
2	News 24	1	1
3	Football	4	2
4	Basketball	4	2
5	Interns	2	3
6	Animal Planet	3	4
1	Science from the Inside	1	1

(6 ёЁёюь)

1. Read uncommitted
  - 1.1 Грязное чтение

SQL Shell (psql)

Server [localhost]:  
Database [postgres]: teleset  
Port [5432]:  
Username [postgres]:  
Пароль пользователя postgres:  
psql (16.0)  
ПРЕДУПРЕЖДЕНИЕ: Кодовая страница консоли (866) отличается от основной страницы Windows (1251).  
8-битовые (русские) символы могут отображаться некорректно.  
Подробнее об этом смотрите документацию psql, раздел "Notes for Windows users".  
Введите "help", чтобы получить справку.  
  
teleset=# BEGIN transaction isolation level read uncommitted;  
ОШИБКА: ошибка синтаксиса (примерное положение: "uncommitted")  
СТРОКА 1: BEGIN transaction isolation level read uncommitted;  
^  
teleset=# BEGIN transaction isolation level read uncommitted;  
BEGIN  
teleset=# UPDATE programs SET idchannel = idchannel + 1 WHERE idchannel = '4';  
UPDATE 2  
teleset=#

SQL Shell (psql)

Server [localhost]:  
Database [postgres]: teleset  
Port [5432]:  
Username [postgres]:  
Пароль пользователя postgres:  
psql (16.0)  
ПРЕДУПРЕЖДЕНИЕ: Кодовая страница консоли (866) отличается от основной страницы Windows (1251).  
8-битовые (русские) символы могут отображаться некорректно.  
Подробнее об этом смотрите документацию psql, раздел "Notes for Windows users".  
Введите "help", чтобы получить справку.  
  
teleset=# BEGIN transaction isolation level read uncommitted;  
BEGIN  
teleset=# SELECT \* FROM programs;  
id | name | idchannel | idspecification  
---+---+---+---  
2 | News 24 | 1 | 1  
5 | Interns | 2 | 3  
1 | Science from the Inside | 1 | 1  
3 | Football | 4 | 2  
4 | Basketball | 4 | 2  
7 | Animal Planet | 3 | 4  
(6 ёЁёюь)

## 1.2 Неповторяющееся чтение

```
teleset=# BEGIN transaction isolation level read uncommitted;
BEGIN
teleset=# UPDATE programs SET idchannel = idchannel - 1 WHERE idchannel = '3';

UPDATE 4
teleset=# commit;
COMMIT
teleset=#

teleset=# commit;
COMMIT
teleset=# BEGIN transaction isolation level read uncommitted;
BEGIN
teleset=# UPDATE programs SET idchannel = idchannel - 1 WHERE idchannel = '3';

UPDATE 0
teleset=#
```

## 2. Read committed

### 2.1 Грязное чтение

подробнее об этом смотрите документацию psql, раздел "Notes for Windows users".  
Введите "help", чтобы получить справку.

```
teleset=# BEGIN transaction isolation level read committed;
ОШИБКА: ошибка синтаксиса (примерное положение: "transaction")
СТРОКА 1: BEGIN transaction isolation level read committed;
^

teleset=# BEGIN transaction isolation level read committed;
BEGIN
teleset=# UPDATE programs SET idchannel = idchannel + 1 WHERE idchannel = '1';
UPDATE 2
teleset=#
```

подробнее об этом смотрите документацию psql, раздел "Notes for Windows users".  
Введите "help", чтобы получить справку.

```
teleset=# BEGIN transaction isolation level read committed;
BEGIN
teleset=# SELECT * FROM programs;
 id |      name      | idchannel | idspecification
-----+-----+-----+-----
 2 | News 24        |         1 |              1
 5 | Interns        |         2 |              3
 1 | Science from the Inside |         1 |              1
 7 | Animal Planet  |         3 |              4
 3 | Football       |         5 |              2
 4 | Basketball     |         5 |              2
(6 строк)

teleset=#
```

подробнее об этом смотрите документацию psql, раздел "Notes for Windows users".  
Введите "help", чтобы получить справку.

```
teleset=# BEGIN transaction isolation level read uncommitted;
BEGIN
teleset=# SELECT * FROM programs;
 id |      name      | idchannel | idspecification
-----+-----+-----+-----
 2 | News 24        |         1 |              1
 5 | Interns        |         2 |              3
 1 | Science from the Inside |         1 |              1
 7 | Animal Planet  |         3 |              4
 3 | Football       |         5 |              2
 4 | Basketball     |         5 |              2
(6 строк)

teleset=# SELECT * FROM programs;
 id |      name      | idchannel | idspecification
-----+-----+-----+-----
 5 | Interns        |         2 |              3
 7 | Animal Planet  |         3 |              4
 3 | Football       |         5 |              2
 4 | Basketball     |         5 |              2
 2 | News 24        |         2 |              1
 1 | Science from the Inside |         2 |              1
(6 строк)
```

## 2.2 Неповторяющееся чтение

```
СТРОКА 1: BEGIN transaction isolation level read committed;
^
teleset=# BEGIN transaction isolation level read committed;
BEGIN
teleset=# UPDATE programs SET idchannel = idchannel + 1 WHERE idchannel = '1';
teleset=# UPDATE programs SET idchannel = idchannel + 1 WHERE idchannel = '2';

UPDATE 3
teleset=# COMMIT;
COMMIT
teleset=#

teleset=# BEGIN transaction isolation level read committed;
ПРЕДУПРЕЖДЕНИЕ: транзакция уже выполняется
BEGIN
teleset=# UPDATE programs SET idchannel = idchannel + 1 WHERE idchannel = '2';

UPDATE 0
teleset=# UPDATE programs SET idchannel = idchannel + 1 WHERE idchannel = '2';

UPDATE 0
teleset=#
teleset=#
teleset=# c\
```

### 3. Repeatable read

#### 3.1 Неповторяющееся чтение

```
teleset=# BEGIN transaction isolation level read uncommitted;
BEGIN
teleset=# UPDATE programs SET idchannel = idchannel - 1 WHERE idchannel = '3';

UPDATE 4
teleset=# commit;
COMMIT
teleset=# BEGIN transaction isolation level repeatable read;
BEGIN
teleset=# UPDATE programs SET idchannel = idchannel - 1 WHERE idchannel = '5';

UPDATE 2
teleset=#
```

```
teleset=# SELECT * FROM programs;
id |          name          | idchannel | idspecification
---+-----+-----+-----
3 | Football               | 5         | 2
4 | Basketball             | 5         | 2
7 | Animal Planet          | 2         | 4
5 | Interns                | 2         | 3
2 | News 24                | 2         | 1
1 | Science from the Inside | 2         | 1
(6 æ€Ёюь)

teleset=#
```

#### 3.2 Фантомное чтение

```
teleset=# BEGIN transaction isolation level repeatable read;
BEGIN
teleset=# UPDATE programs SET idchannel = idchannel + 1 WHERE idchannel = '2';
UPDATE 4
teleset=#
```

```
ОШИБКА: ошибка синтаксиса (примерное положение: "UPDATE")
СТРОКА 2: UPDATE programs SET idchannel = idchannel + 1 WHERE idchannel ...
      ^
teleset=# UPDATE programs SET idchannel = idchannel + 1 WHERE idchannel = '
2';
ОШИБКА: текущая транзакция прервана, команды до конца блока транзакции иг
норируются
teleset=#
```

#### 3.3 Аномалия сериализации

```
teleset=# BEGIN transaction isolation level repeatable read;
BEGIN
teleset=# UPDATE programs SET idchannel = (SELECT id FROM programs WHERE id =
5) WHERE idchannel = '4';
UPDATE 6
teleset=# commit;
COMMIT
teleset=#
```

```
ПРЕДУПРЕЖДЕНИЕ: нет незавершённой транзакции
COMMIT
teleset=# BEGIN transaction isolation level repeatable read;
BEGIN
teleset=# UPDATE programs SET idchannel = idchannel + 1 WHERE idchannel =
'4';
ОШИБКА: не удалось сериализовать доступ из-за параллельного изменения
teleset=#
```

### 4. Serializable

#### 4.1 Аномалия сериализации

```
teleset=# BEGIN transaction isolation level serializable;
BEGIN
teleset=# UPDATE programs SET idchannel = (SELECT id FROM programs WHERE id =
'4') WHERE idchannel = '5';
UPDATE 6
teleset=# COMMIT;
COMMIT
teleset=#
```

```
teleset=# commit;
ROLLBACK
teleset=# BEGIN transaction isolation level serializable;
BEGIN
teleset=# UPDATE programs SET idchannel = idchannel + 1 WHERE idchannel = '5
';
ОШИБКА: не удалось сериализовать доступ из-за параллельного изменения
teleset=#
```

## 5. Триггер

```
1 CREATE OR REPLACE FUNCTION idCH_chek() RETURNS trigger AS $$
2 BEGIN
3     IF NEW.idchanel < 0 THEN
4         RAISE EXCEPTION 'id chanel must be > 0';
5     END IF;
6     IF NEW.idspecialization < 0 THEN
7         RAISE EXCEPTION 'id specialization must be > 0';
8     END IF;
9     RETURN NEW;
10 END;
11 $$ LANGUAGE plpgsql;
12 CREATE OR REPLACE TRIGGER idCH_chek BEFORE INSERT OR UPDATE ON programs
13 FOR EACH ROW EXECUTE FUNCTION idCH_chek();
14 UPDATE programs
15     SET idchanel = -3, idspecialization = 0
16     WHERE id = 6;
17 SELECT * FROM programs
```

Data Output   Messages   Notifications

UPDATE 0

Query returned successfully in 179 msec.

## Задание 2.1

### 1. Создание пользователя и представления

```
1 CREATE USER test WITH
2 NOSUPERUSER
3 NOCREATEDB
4 INHERIT
5 LOGIN
6 NOREPLICATION
7 CONNECTION LIMIT -1
8 PASSWORD '1234';
9 GRANT SELECT ON TABLE tvchanel TO test;
10 GRANT SELECT, UPDATE(name) ON TABLE tvcompany TO test;
11 GRANT SELECT, UPDATE, INSERT ON TABLE programs TO test;
12 GRANT SELECT ON teleprog TO test;
13
14 CREATE VIEW teleprog AS
15 SELECT tvchanel.name AS tvchanel, programs.name AS programs
16 FROM programs
17 JOIN tvchanel
18     ON programs.idchanel = tvchanel.id
19
20 CREATE VIEW sputnik AS
21 SELECT satellite.id AS idSputnik, satellite.name AS Sputnik, broadcast.id AS idBroadcast , country.name AS Country
22 FROM satellite
23 JOIN broadcast
24     ON satellite.idtranslation = broadcast.id
25 JOIN country
26     ON broadcast.idcountry = country.id
27
```

tvchannel character varying (255)	programs character varying (255)	idsputnik integer	sputnik character varying (255)	idbroadcast integer	country character varying (255)
ESPN	Interns	1	Астра 1H	2	USA
CNN	Science from the Inside	2	Астра 1KR	3	Russia
CNN	The World of Planets	3	Астра 1L	5	Russia
HBO	Football	4	Астра 2M	3	Russia
HBO	Basketball	5	Хот Бёрд 6	1	USA
HBO	Horse Racing	6	Хот Бёрд 8	3	Russia
HBO	Animal World	7	Хот Бёрд 9	3	Russia
HBO	News 24	8	Экспресс AM2	5	Russia
		9	Экспресс-AM3	9	England
		10	Сириус 4	6	Germany

2. Проверка подлинности доступа к таблицам
  - 2.2. GRANT SELECT ON TABLE tvchannel TO test;

На просмотр доступ есть:

```
Telesett=> SELECT * FROM tvchannel;
 id | idtranslation |      name      | idtvcompany
-----+-----+-----+-----
  1 |          3 | CNN            |          1
  2 |          2 | ESPN           |          1
  3 |          5 | STAR World     |          2
  4 |          1 | HBO            |          2
  5 |          7 | Disney Channel |          3
  6 |          5 | Nickelodeon    |          4
  7 |          8 | FOX            |          4
  8 |          9 | MTV Music Television |          3
  9 |          5 | ESPN           |          4
 10 |          3 | BBC            |          5
(10 строк)
```

На какие-либо изменения нет:

```
Telesett=> UPDATE tvchannel SET name = 'TNT' WHERE idtvcompany = 4;
ОШИБКА: нет доступа к таблице tvchannel
```

### 2.3. GRANT SELECT, UPDATE(name) ON TABLE tvcompany TO test;

На просмотр и изменения таблицы доступ есть:

```
Telesett=> SELECT * FROM tvcompany;
id |          name
----+-----
 1 | D-Smart
 2 | Acheloos TV
 3 | 2929 Entertainment
 4 | Antix Productions
(4 ÷)

Telesett=> UPDATE tvcompany SET name = 'ROSTELECOM' WHERE id = 3;
UPDATE 1
Telesett=> SELECT * FROM tvcompany;
id |          name
----+-----
 1 | D-Smart
 2 | Acheloos TV
 4 | Antix Productions
 3 | ROSTELECOM
(4 ÷)
```

На удаление и добавление новых строчек нет доступа:

```
Telesett=> DELETE FROM tvcompany WHERE id = 1;
ОШИБКА:  нет доступа к таблице tvcompany
```

### 2.4. GRANT SELECT, UPDATE, INSERT ON TABLE programs TO test;

На просмотр , изменения и добавление строчек таблицы доступ есть:

```
Telesett=> SELECT * FROM programs;
id |          name          | idchanel | idspecialization
----+-----+-----+-----
 5 | Interns                 |         2 |              3
 8 | Science from the Inside |         1 |              4
 9 | The World of Planets    |         1 |              4
 2 | Football                |         4 |              2
 3 | Basketball              |         4 |              2
 4 | Horse Racing            |         4 |              2
 7 | Animal World            |         4 |              4
 1 | News 24                 |         4 |              1
```

```

Telesett=> INSERT INTO programs(name, idchannel, idspecialization) VALUES ('
Comedy', 3, 3);
INSERT 0 1
Telesett=> SELECT * FROM programs;

```

id	name	idchannel	idspecialization
5	Interns	2	3
8	Science from the Inside	1	4
9	The World of Planets	1	4
2	Football	4	2
3	Basketball	4	2
4	Horse Racing	4	2
7	Animal World	4	4
1	News 24	4	1
11	Comedy	3	3

На удаление доступа нет:

```

Telesett=> DELETE FROM programs WHERE id = 9;
ОШИБКА: нет доступа к таблице programs

```

### 3. Выборки:

```

Telesett=> SELECT programs.id, programs.name FROM programs JOIN tvchannel ON
programs.idchannel = tvchannel.id JOIN tvcompany ON tvchannel.idtvcompany = tvco
mpany.id WHERE tvcompany.name = 'D-Smart';

```

id	name
5	Interns
8	Science from the Inside
9	The World of Planets

(3 ёёёюш)

### 4. Вывод макрокоманды psql /dp (флаги привилегий пользователей)

```

Telesett-> \dp

```

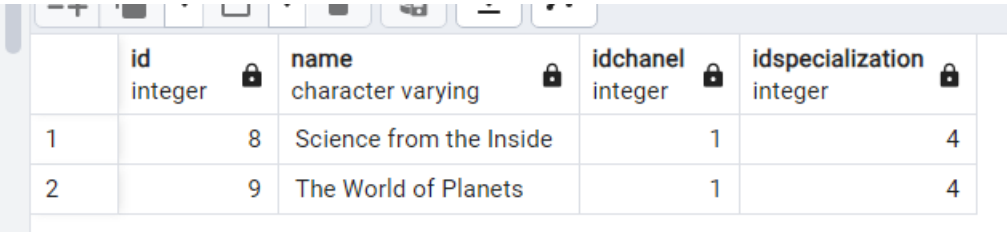
Схема	Имя	Тип	Права доступа	Права для столбцов	Политики
public	broadcast	таблица			
public	broadcast_id_seq	последовательность			
public	country	таблица			
public	country_id_seq	последовательность			
public	genre	таблица			
public	genre_id_seq	последовательность			
public	programs	таблица	postgres=arwdDxt/postgres+ test=arw/postgres		
public	programs_id_seq	последовательность			
public	satellite	таблица			
public	satellite_id_seq	последовательность			
public	specialization	таблица			
public	specialization_id_seq	последовательность			
public	sputnik	представление			
public	teleprog	представление	postgres=arwdDxt/postgres+ test=r/postgres		
public	tvchannel	таблица	postgres=arwdDxt/postgres+ test=r/postgres		
public	tvchannel_id_seq	последовательность			
public	tvcompany	таблица	postgres=arwdDxt/postgres+ test=r/postgres	name: test=w/postgres	+
public	tvcompany_id_seq	последовательность			

(18 строк)

## Задание 2.2

-- Вывод всех программы, которые транслируются по специализациям с четным id

```
CREATE OR REPLACE FUNCTION del_prog()
  RETURNS TABLE (id int, name varchar(255), idchannel int, idspecialization int) AS
$function$
DECLARE
  prog RECORD;
BEGIN
  CREATE TABLE tmp(
    id int,
    name varchar(255),
    idchannel int,
    idspecialization int
  );
  FOR prog IN SELECT * FROM programs
  LOOP
    IF prog.idspecialization % 2 = 0 THEN
      INSERT INTO tmp (id, name, idchannel, idspecialization)
      VALUES
        (prog.id, prog.name, prog.idchannel, prog.idspecialization);
    END IF;
  END LOOP;
  RETURN QUERY SELECT * FROM tmp;
  DROP TABLE tmp;
END;
$function$
LANGUAGE plpgsql;
SELECT * FROM del_prog();
```



	id integer	name character varying	idchannel integer	idspecialization integer
1	8	Science from the Inside	1	4
2	9	The World of Planets	1	4

-- Удаление программы, котрые показываются после 3 канала

```
CREATE OR REPLACE FUNCTION del_prog1()
  RETURNS void AS
$function$
DECLARE
  CURRENT_ROW RECORD;
  prog_cursor CURSOR FOR SELECT * FROM programs WHERE (idchannel > 3);
BEGIN
  OPEN prog_cursor;
  LOOP
    FETCH prog_cursor INTO CURRENT_ROW;
```



```

EXIT WHEN NOT FOUND;
DELETE FROM programs WHERE id = CURRENT_ROW.id;
END LOOP;
CLOSE prog_cursor;
END;
$function$
LANGUAGE plpgsql;
SELECT * FROM del_prog1()
SELECT * FROM programs

```

до

	id [PK] integer	name character varying (255)	idchanel integer	idspecialization integer
1	5	Interns	2	3
2	8	Science from the Inside	1	4
3	9	The World of Planets	1	4
4	2	Football	4	2
5	3	Basketball	4	2
6	4	Horse Racing	4	2
7	7	Animal World	4	4
8	1	News 24	4	1
9	11	Comedy	3	3

после

	id [PK] integer	name character varying (255)	idchanel integer	idspecialization integer
1	5	Interns	2	3
2	8	Science from the Inside	1	4
3	9	The World of Planets	1	4
4	11	Comedy	3	3

```

-- Изменение программы, ее канала и специализации
CREATE OR REPLACE FUNCTION add_prog(idf int, namef varchar(255), idchanelf int,
idspecializationf int)
RETURNS void AS
$function$
BEGIN
    IF namef IS null THEN
        RAISE 'name must be filled';
    END IF;
    IF idchanelf < 1 THEN
        RAISE 'the channel ID must be greater than 0';
    END IF;
    IF idspecializationf < 1 THEN

```

```

        RAISE 'the specialization ID must be greater than 0';
    END IF;
    UPDATE programs SET name = namef, idchannel = idchanelf, idspecialization =
idspecializationf
    WHERE ID = idf;
END;
$function$
LANGUAGE plpgsql;
SELECT * FROM add_prog(11,'Sled serial',2,3);
SELECT * FROM programs;

```

	id [PK] integer	name character varying (255)	idchannel integer	idspecialization integer
1	5	Interns	2	3
2	8	Science from the Inside	1	4
3	9	The World of Planets	1	4
4	11	Sled serial	2	3