**Министерство науки и высшего образования Российской Федерации**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**ITMO University**

**Отчет по лабораторной работе № 3**

**По дисциплине** Проектирование баз данных/Базы данных

**Обучающиеся** Наус Михаил Романович, Савинова Алина Константиновна

**Факультет** Факультет технологического менеджмента и инноваций

**Группа** U3275

**Направление подготовки** 27.03.05 Инноватика

**Образовательная программа** Технологии и инновации

**Обучающиеся** Наус М. Р. Савинова А. К.

**Преподаватель** Ромакина О.М.

**Основная часть**

**Цель работы:** овладеть практическими навыками создания запросов для базы данных PostgreSQL 10 (11), добавление новых атрибутов, выборка различных значений.

**Практическое задание:**

Состоит из создания ряда запросов, а именно:

1. Модификация таблицы путем добавления какого-либо атрибута
2. Выборка всех данных в таблице
3. Выборка различных значений какого-либо столбца таблицы
4. Выборка строк таблицы, где значения какого-либо атрибута принадлежат заданному диапазону
5. Выборка строк таблицы, где значения какого-либо атрибута принадлежат заданному набору значений
6. Выборка строк таблицы, где значения какого-либо атрибута соответствуют заданному шаблону
7. Выборка строк таблицы, где значение какого-либо атрибута не пусто
8. Сортировка строк таблицы по двум ключам сортировки
9. Внутреннее (естественное) соединение таблиц
10. Правое соединение таблиц
11. Левое соединение таблиц
12. Полное соединение таблиц
13. Объединение двух таблиц
14. Группировка записей по двум или более полям
15. Вложенный подзапрос
16. Создание модифицируемого представления
17. Создание не модифицируемого представления
18. **Диаграмма IDEF1X.**

Изображение выглядит как текст, снимок экрана, диаграмма, Шрифт

Автоматически созданное описание

(Рисунок 1. Диаграмма IDEF1X)

1. **Модификация таблицы путем добавления какого-либо атрибута.**

Для выполнения этого пункта мы напишем следующий запрос (см. Листинг 1):

-- Модификация таблицы путем добавления какого-либо атрибута

ALTER TABLE Награды ADD COLUMN Дата\_одобрения\_награды DATE;

(Листинг 1)

Далее мы получим следующий вывод (см. Рисунок 2):

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

(Рисунок 2. Вывод таблицы после добавления атрибута)

1. **Выборка всех данных в таблице.**

Для выборки всех данных в таблице (что мы сделали в предыдущем пункте для демонстрации успешного добавления атрибута) используем следующий запрос (см. Листинг 2):

SELECT \* FROM Награды;

(Листинг 2)

После чего получим вывод (см. Рисунок 2), ранее мы уже его видели.

1. **Выборка различных значений какого-либо столбца таблицы.**

Для выполнения этого запроса мы будем использовать выборку различных значений, однако при игнорировании дублирования

(см. Листинг 3):

SELECT DISTINCT Имя\_Фамилия FROM Режиссёр;

(Листинг 3)

После чего получим следующий вывод (см. Рисунок 3):

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

(Рисунок 3)

1. **Выборка строк таблицы, где значения какого-либо атрибута принадлежат заданному диапазону.**

Для выполнения этого запроса мы будем использовать следующий запрос, где будет идти выборка элементов в диапазоне 2000-2019 года получений наград режиссерами (см. Листинг 4):

SELECT \* FROM Награды\_сценариста WHERE Year\_of\_awards BETWEEN 2000 AND 2019;

(Листинг 4)

После чего мы получим следующий вывод в нашем рабочем окружении (см. Рисунок 4):

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

(Рисунок. 4. Вывод данных в ограничениях)

1. **Выборка строк таблицы, где значения какого-либо атрибута принадлежат заданному набору значений.**

Для реализации данного пункта мы будем использовать ключевое слово **IN**, для задания определенного конкретного соответствия с набором данных, составим следующий запрос (см. Листинг 5):

SELECT \* FROM Награды\_актера WHERE Category\_name IN ('personal', 'other');

(Листинг 5)

После чего можем увидеть следующую картину на выходе

(см. Рисунок 5):

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

(Рисунок 5)

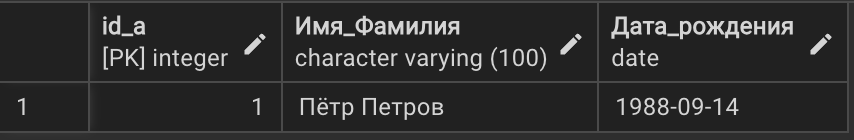
**6.Выборка строк таблицы, где значения какого-либо атрибута соответствуют заданному шаблону.**

Здесь мы будем использовать шаблон данных с которым нужно будет сделать соответствие и вывести все удовлетворяющие ему набор данных (см. Листинг 6):

SELECT \* FROM Актёр WHERE Имя\_Фамилия LIKE 'Петр %';

(Листинг 6)

Далее мы получим следующий вывод (см. Рисунок 6):



(Рисунок 6. Вывод данных соответствующих шаблону)

**7. Выборка строк таблицы, где значение какого-либо атрибута не пусто.**

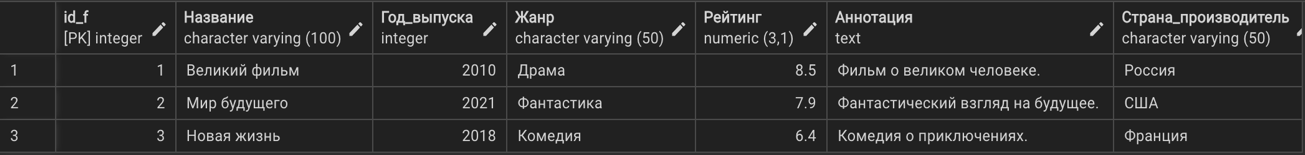
Здесь мы будем использовать уже другую конструкцию из ключевых слов, а именно **IS NOT NULL,** что прямо говорит нам, данные не могут быть пустыми. Давайте составим запрос

(см. Листинг 7):

SELECT \* FROM Фильм WHERE Аннотация IS NOT NULL;

(Листинг 7)

Теперь мы можем увидеть следующий вывод (см. Рисунок 7):



(Рисунок 7. Вывод не пустых данных)

**8. Сортировка строк таблицы по двум ключам сортировки.**

Для выполнения этого задания мы воспользуемся следующим запросом (см. Листинг 8):

SELECT \* FROM Фильмография\_сценариста ORDER BY ID\_F ASC, ID\_S DESC;

(Листинг 8)

После чего получим следующий вывод (см. Рисунок 8):

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

(Рисунок 8)

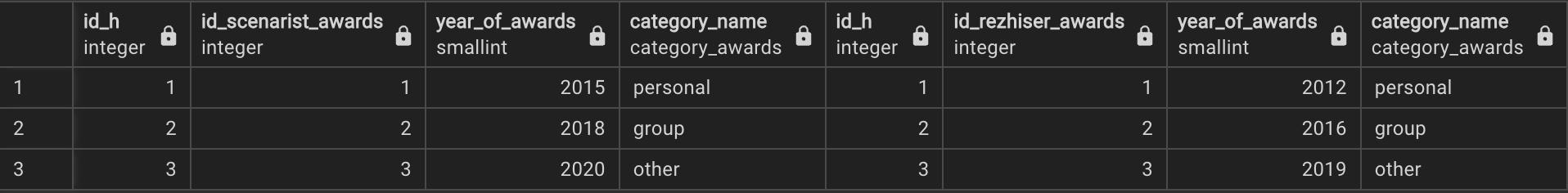
**9. Внутреннее (естественное) соединение таблицы.**

Для выполнения этого запроса мы будем использовать следующий запрос (см. Листинг 9):

SELECT \* FROM Награды\_сценариста INNER JOIN Награды\_режиссера ON Награды\_сценариста.ID\_H = Награды\_режиссера.ID\_H;

(Листинг 9)

После чего получим следующий вывод (см. Рисунок 9):



(Рисунок 9)

**10. Правое соединение таблиц.**

Для выполнения этого задания воспользуемся знаниями о ключевом слове **RIGHT JOIN,** используя его напишем запрос

(см. Листинг 10):

SELECT \* FROM Награды\_сценариста RIGHT JOIN Награды\_фильма ON Награды\_сценариста.ID\_H = Награды\_фильма.ID\_H;

(Листинг 10)

Далее можем лицезреть следующий вывод (см. Рисунок 10):

Изображение выглядит как снимок экрана, черный, текст, линия

Автоматически созданное описание

(Рисунок 10)

**11. Левое соединения таблиц.**

Для выполнения этого пункта, мы будем использовать противоположное ключевое слово – **LEFT JOIN**, благодаря ему напишем следующий запрос (см. Листинг 11):

SELECT \* FROM Награды\_актера LEFT JOIN Награда ON Награды\_актера.ID\_H = Награда.ID\_H;

(Листинг 11)

И получим вот такой вывод (см. Рисунок 11):

Изображение выглядит как текст, снимок экрана, черный, линия

Автоматически созданное описание

(Рисунок 11)

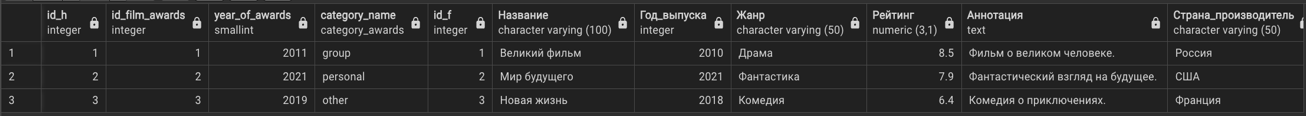
**12. Полное соединение таблиц.**

Для выполнения этого пункта нам потребуются знания, как работает **FULL OUTER JOIN,** после чего мы сможем составить следующий запрос (см. Листинг 12):

SELECT \* FROM Награды\_фильма FULL OUTER JOIN Фильм ON Награды\_фильма.ID\_Film\_Awards = Фильм.ID\_F;

(Листинг 12)

После чего получим следующий вывод (см. Рисунок 12):



(Рисунок 12)

**13. Объединение двух таблиц.**

Для выполнения этого пункта мы составим запрос с помощью ключевого слова **UNION** (см. Листинг 13):

SELECT \* FROM Актёр UNION SELECT \* FROM Роль\_в\_кино;

(Листинг 13)

Ну и можем ознакомиться с полученным выводом (см. Рисунок 13):

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

(Рисунок 13)

**14. Группировка записей по двум или более полям.**

Для выполнения этого запроса мы воспользуемся ключевым словом – **GROUP BY,** составим же запрос (см. Листинг 14):

SELECT Название, Рейтинг, COUNT(\*) FROM Фильм GROUP BY Название, Рейтинг;

(Листинг 14)

После можем ознакомиться с выводом (см. Рисунок 14):

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

(Рисунок 14)

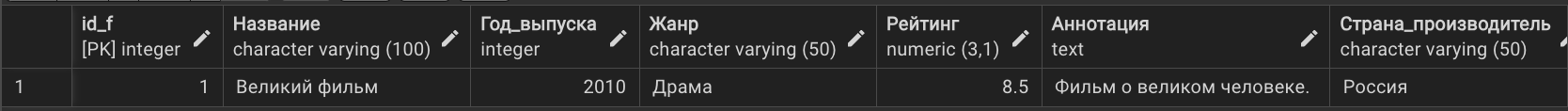
**15. Вложенный подзапрос.**

Для выполнения этого пункта мы воспользуемся еще одним запросом для упрощения нашей жизни, давайте посмотрим на следующий запрос (см. Листинг 15):

SELECT \* FROM Фильм WHERE Рейтинг = (SELECT MAX(Рейтинг) FROM Фильм);

(Листинг 15)

Получим такой вот вывод (см. Рисунок 15):



(Рисунок 15)

**16. Создание модифицируемого представления.**

Чтобы выполнить этот пункт нам потребуется знание ключевого слова **VIEW,** мы будем использовать его в следующем запросе

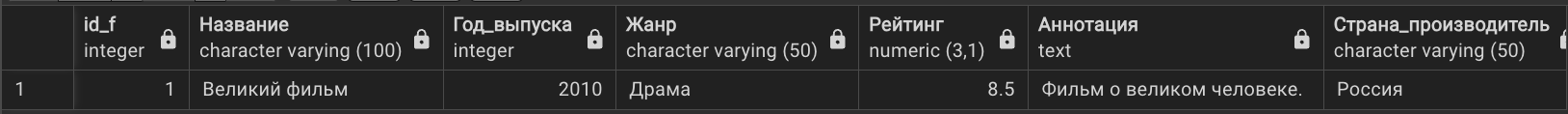
(см. Листинг 16):

CREATE VIEW View\_Фильмы\_Россия AS SELECT \* FROM Фильм WHERE Страна\_производитель = 'Россия'

WITH CHECK OPTION;

(Листинг 16)

Чтобы получить вывод, воспользуемся аналогичной командой, как из пункта 2 (см. Рисунок 16):



(Рисунок 16)

**17. Создание не модифицируемого представления.**

Тут все проще, потому что представления по умолчанию не модифицируемые (см. Листинг 17):

CREATE VIEW View\_Награды\_актера AS SELECT ID\_H, ID\_Actor\_Awards, Year\_of\_awards, Category

FROM Награды\_актера;

(Листинг 17)

Ознакомимся с данными (см. Рисунок 17):

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

(Рисунок 17)

**Вывод**

В ходе работы мы смогли научиться строит работающие запросы для баз данных и получать нужные данные различными способами. Также мы ознакомились с довольно широким спектром команд, которые позволяют манипулировать с выборкой данных так, как нам нужно. С полным кодом этой лабораторной работы можно ознакомиться на гитхабе у одного из участников команды: <https://github.com/fallayn/Database_Laboratory_No_4.git>