МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»

Факультет Управление и информатика в технологических системах

Кафедра Информационная безопасность

Специальность 10.05.03 «Информационная безопасность автоматизированных систем»

Отчет по практической работе №4

по дисциплине Безопасность Баз Данных

Тема: Применение SQL в приложениях.

Выполнила: студентка 3 курса

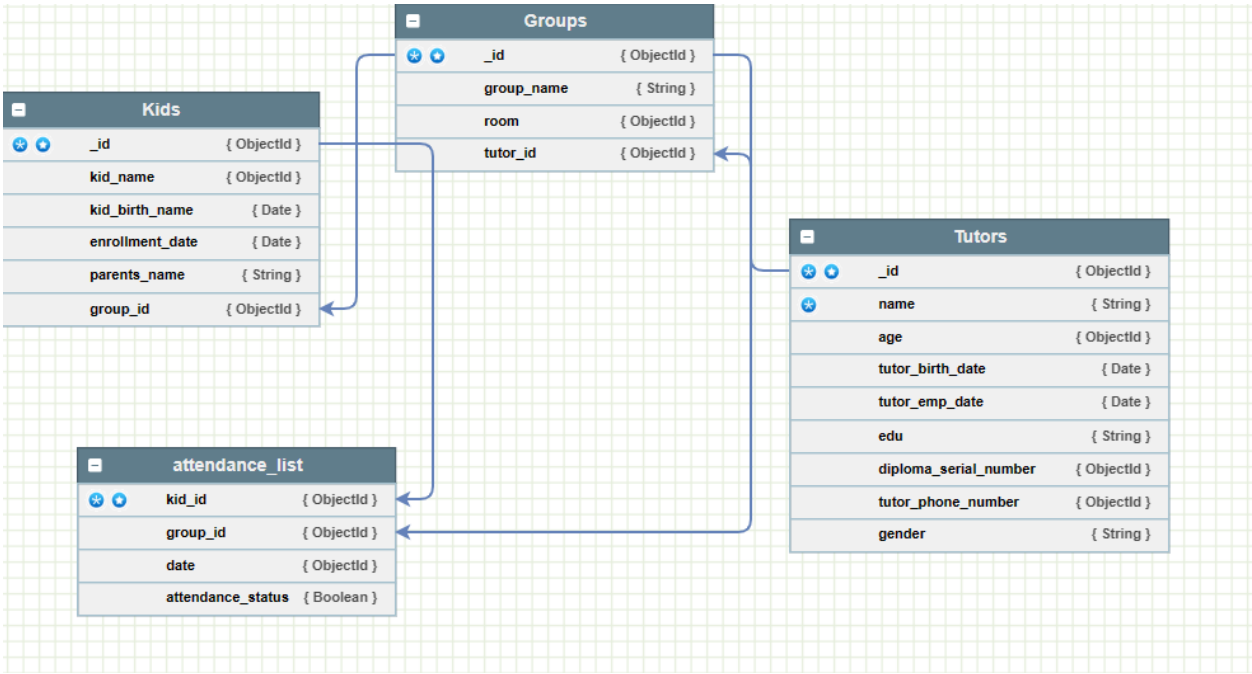
группы УБ-01

Лазарева Маргарита Вячеславовна

Воронеж – 2023

Цель работы: изучить операторы sql для работы с базой данных.

Модель данных для БД «детский сад»:



SQL-представление (SQL view) – это виртуальная таблица, составленная из других таблиц или представлений. Представление не имеет своих собственных данных, а объединяет данные из таблиц или представлений, которые в него входят. Представления создаются с помощью комбинации операторов CREATE VIEW и SELECT. Согласно стандарту SQL-92 представления не могут включать в себя конструкцию ORDER BY, но PostgreSQL такой вариант допускает.

Например, следующий оператор определяет представление под названием GroupNameView, базирующийся на таблице groups:

CREATE VIEW GroupNameView AS SELECT group_name AS GroupName FROM groups ORDER BY group_name;

kindergarten=# select * from GroupNameView;

Для получения отсортированного списка имен художников данное представление можно обработать при помощи следующего запроса:

SELECT * FROM GroupNameView;

Реализация и результат выглядит следующим образом:

```
kindergarten=# CREATE VIEW GroupNameView AS SELECT group_name AS GroupName FROM groups ORDER BY group_name;
CREATE VIEW
kindergarten=# select * from GroupNameView;
               groupname
------------------------------------------
 Fishers
 Flower
 FlowerBloom
 Friends
 Roses
 Sun
 SunFlower
(7 строк)
```

1. Использование представлений для скрытия столбцов и строк

С помощью представлений можно скрыть отдельные столбцы таблиц. Это делается для того, чтобы возвращаемый результат имел более простой вид, а также для предотвращения доступа к конфиденциальным данным. Предположим, что пользователям базы данных нужны только имена и информация об образовании тьюторов, но не их возраст и номера телефонов. Следующий оператор создает представление TutorEducationData, содержащее только эти данные.

CREATE VIEW TutorEducationData AS SELECT tutor_name, diploma, diploma_serial FROM tutors;

Результаты выполнения оператора SELECT над этим представлением следующие:

```
kindergarten=# CREATE VIEW TutorEducationData AS SELECT tutor_name, diploma, diploma_serial FROM tutors;
CREATE VIEW
kindergarten=# select * from TutorEducationData;
                tutor_name                 |                                 diploma
 | diploma_serial
-------------------------------------------+-------------------------------------------------------------------------+----------------
 Christen Stell                            | Bachelor in International Teacher Education for Primary Schools (ITEPS) |         136789
 Dove Eyre                                 | Bachelor in Intercultural Teacher Education
 |         789061
 Olivia Rodgers                            | Bachelor in International Teacher Education for Primary Schools (ITEPS) |         834567
 Candy Watt                                | Bachelor's Degree in Early Childhood Education Teacher (Lugo Campus)    |         786123
 Lame Starck                               | Bachelor's Degree in Early Childhood Education Teacher (Lugo Campus)    |         458767
 Daniel Michell                            | Bachelor's Degree in Early Childhood Education Teacher (Lugo Campus)    |         786520
 Ima Christen                              | Bachelor of Science in Science Education
 |         675490
(7 строк)      |
```

Можно скрывать от просмотра и строки таблиц. Для этого в определении представления должно присутствовать предложение WHERE. Следующий оператор определяет представление, содержащее имена и информация об образовании тьюторов, устроившихся на работу в детский сад в 2020 году и позже:

CREATE VIEW TutorEducationDataAf2020 AS SELECT tutor_name, diploma, diploma_serial FROM tutors WHERE employment_date > ‘2020-01-01’;

Результирующая таблица будет такая:

```
kindergarten=# CREATE VIEW TutorEducationDataAf2020 AS SELECT tutor_name, diploma, diploma_serial FROM tutors WHERE employment_date > '2020-01-01';
CREATE VIEW
kindergarten=# select * from TutorEducationDataAf2020;
                tutor_name                 |                                 diploma
 | diploma_serial
-------------------------------------------+-------------------------------------------------------------------------+----------------
 Christen Stell                            | Bachelor in International Teacher Education for Primary Schools (ITEPS) |         136789
 Lame Starck                               | Bachelor's Degree in Early Childhood Education Teacher (Lugo Campus)    |         458767
 Ima Christen                              | Bachelor of Science in Science Education
 |         675490
(3 строки)     |
```

Проверим правильность выведенных данных:

```
 tutor_id |                 tutor_name                  | tutor_age | birth_date | employment_date |
             diploma                                    | diploma_serial | tutor_phone_number | gender
----------+---------------------------------------------+-----------+------------+-----------------+
------------------+----------------+--------------------+--------
       21 | Christen Stell                              |        27 | 1995-09-20 | 2020-01-20      |
ry Schools (ITEPS) |         136789 | 89513467890        | woman
       23 | Dove Eyre                                   |        27 | 1995-12-19 | 2018-08-01      |
                  |         789061 | 89004562879        | woman
       24 | Olivia Rodgers                              |        31 | 1991-01-01 | 2019-11-30      |
ry Schools (ITEPS) |         834567 | 89003458976        | woman
       25 | Candy Watt                                  |        30 | 1992-05-24 | 2018-12-02      |
r (Lugo Campus)    |         786123 | 89564873021        | woman
       27 | Lame Starck                                 |        28 | 1994-01-28 | 2022-05-26      |
r (Lugo Campus)    |         458767 | 89503457891        | man
       22 | Daniel Michell                              |        30 | 1992-03-04 | 2019-09-30      |
r (Lugo Campus)    |         786520 | 89203456712        | man
       26 | Ima Christen                                |        33 | 1989-02-12 | 2020-04-30      |
                  |         675490 | 89009325673        | woman
(7 строк)
```

2. Использование представлений для отображения вычисляемых столбцов

Еще одно применение представлений – отображение результатов вычислений, не прибегая к вводу формул пользователем. Например, следующее представление объединяет столбцы diploma и diploma_serial и форматирует результат:

CREATE VIEW TutorEducation AS SELECT tutor_name, (diploma || ':') || diploma_serial AS diploma FROM tutors;

Допустим, пользователь вводит следующий запрос: SELECT * FROM TutorEducation;

Результаты выполнения этого запроса будут следующими:

```
kindergarten=# CREATE VIEW TutorEducation AS SELECT tutor_name, (diploma || ':') || diploma_serial AS diploma FROM tutors;
CREATE VIEW
kindergarten=# SELECT * FROM TutorEducation;
                tutor_name                 |                                  diploma
-------------------------------------------+-------------------------------------------------------------------------------
 Christen Stell                            | Bachelor in International Teacher Education for Primary Schools (ITEPS):136789
 Dove Eyre                                 | Bachelor in Intercultural Teacher Education:789061
 Olivia Rodgers                            | Bachelor in International Teacher Education for Primary Schools (ITEPS):834567
 Candy Watt                                | Bachelor's Degree in Early Childhood Education Teacher (Lugo Campus):786123
 Lame Starck                               | Bachelor's Degree in Early Childhood Education Teacher (Lugo Campus):458767
 Daniel Michell                            | Bachelor's Degree in Early Childhood Education Teacher (Lugo Campus):786520
 Ima Christen                              | Bachelor of Science in Science Education:675490
(7 строк)
```

3. Использование представлений для скрытия сложного синтаксиса

Два наиболее распространенных варианта использования представлений в данной ипостаси – это скрытие соединений и скрытие вложенных запросов. Попробуем отобразить, в какую группу ходит каждый из детей. Чтобы отобразить эти сведения, необходимо выполнить одно соединение: соединить таблицы kids и groups. Представление, содержащее это соединение, конструируется с помощью следующего SQL-запроса:

CREATE VIEW KidsGroups AS SELECT k.kid_name AS kid, g.group_name AS group FROM kids K JOIN groups G ON g.group_id = k.group_id;

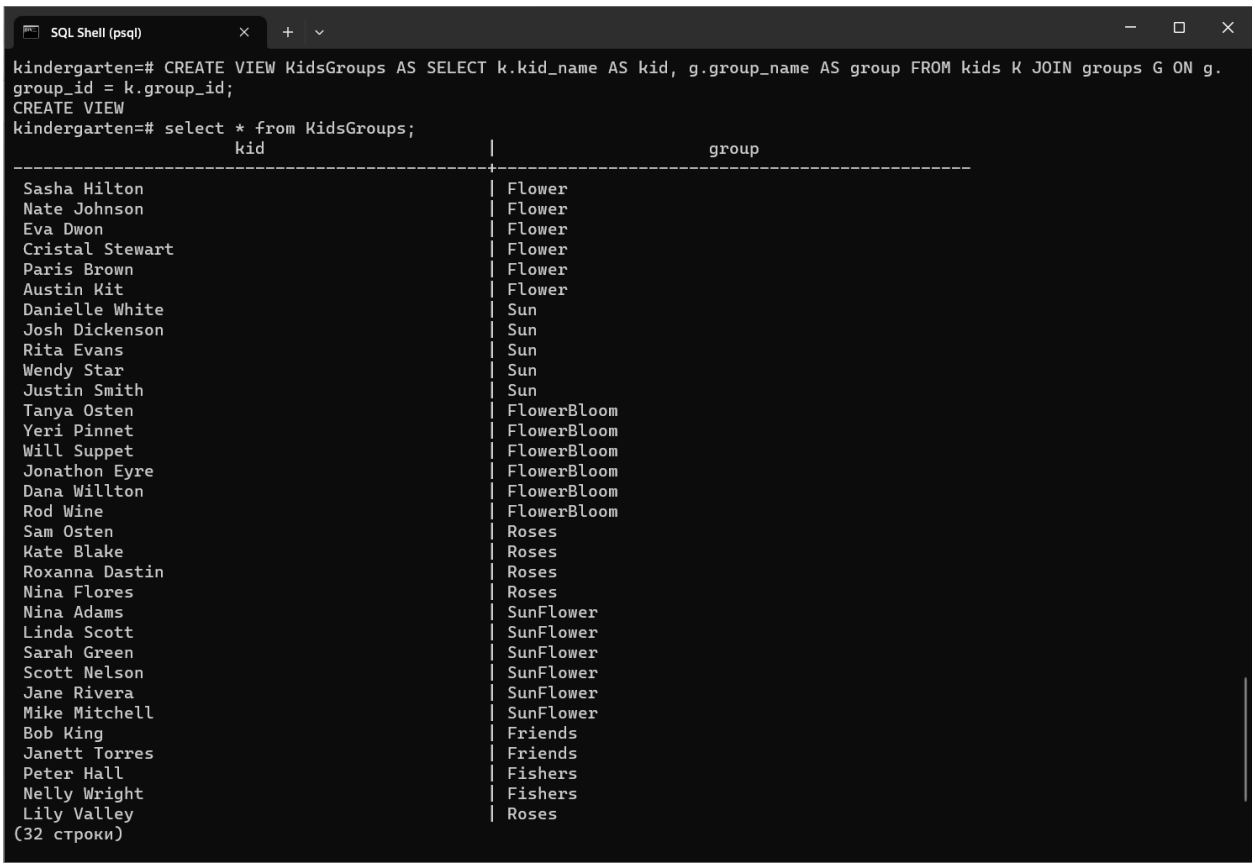
Следующий оператор запрашивает данные из представления KidsGroups:

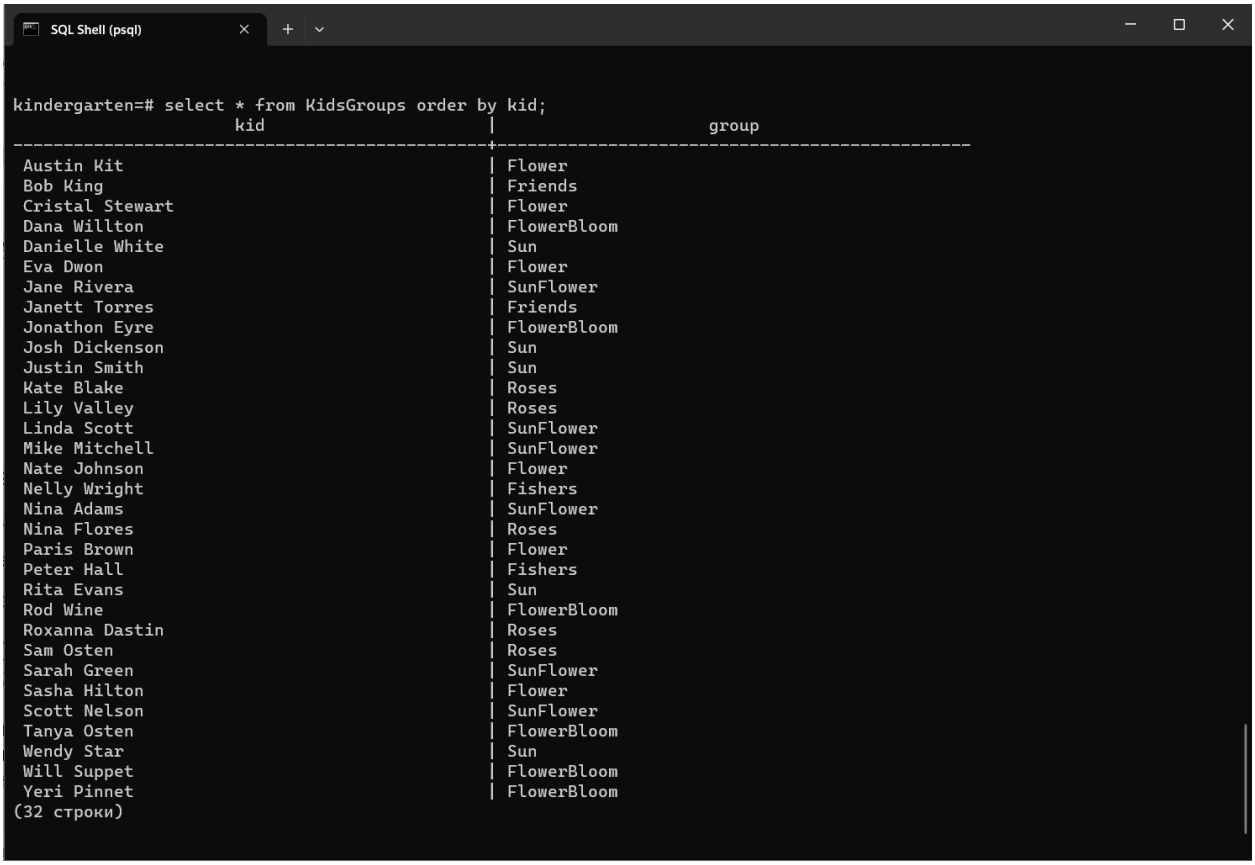
SELECT * FROM KidsGroups;

Следующий оператор запрашивает данные из представления KidsGroups и сортирует результаты по столбцу kid:

SELECT * FROM KidsGroups ORDER BY kid;

Результат получился следующий:





4. Хранимые процедуры

```
kindergarten=# DROP FUNCTION newkid(text);
DROP FUNCTION
kindergarten=# CREATE OR REPLACE FUNCTION NewKid (
kindergarten(# newkidname IN text,
kindergarten(# newgroup OUT text
kindergarten(# )
kindergarten-# AS $NewKid$
kindergarten$# DECLARE new_record RECORD;
kindergarten$# BEGIN
kindergarten$#  FOR new_record IN SELECT kids.kid_name, groups.group_name FROM groups JOIN kids ON groups.group_id=kids.
group_id WHERE kids.kid_name = newkidname
kindergarten$# LOOP
kindergarten$# newgroup := new_record.group_name;
kindergarten$# RAISE NOTICE 'Ребенок % входит в группу %, ', newkidname, newgroup;
kindergarten$# END LOOP;
kindergarten$# END;
kindergarten$# $NewKid$ LANGUAGE plpgsql;
CREATE FUNCTION
kindergarten=# select newkid ('Sasha Hilton');
ЗАМЕЧАНИЕ:  Ребенок Sasha Hilton входит в группу Flower,
 newkid
--------
 Flower
(1 строка)


kindergarten=#
```

5. Использование триггеров для проверки допустимости вводимых данных

```
SQL Shell (psql)
kindergarten=# CREATE OR REPLACE FUNCTION xkid() RETURNS trigger AS
kindergarten-# $xkid$
kindergarten$# BEGIN
kindergarten$# IF EXISTS (SELECT * FROM kids WHERE kid_name = NEW.kid_name) THEN
kindergarten$# RAISE EXCEPTION 'Ребенок % уже есть в списках', NEW.kid_name;
kindergarten$# END IF;
kindergarten$# RETURN NEW;
kindergarten$# END;
kindergarten$# $xkid$ LANGUAGE plpgsql;
CREATE FUNCTION
kindergarten=# CREATE TRIGGER xkid
kindergarten-# BEFORE INSERT ON kids
kindergarten-# FOR EACH ROW EXECUTE FUNCTION xkid();
CREATE TRIGGER
kindergarten=# select * from kids;
 kid_id |                kid_name                | group_id | kid_birth_date | enrollment_date |    hometown    | full_age
--------+----------------------------------------+----------+----------------+-----------------+----------------+----------
    101 | Sasha Hilton                           |        1 | 2022-02-23     | 2023-03-25      | Alexander City |        1
    102 | Nate Johnson                           |        1 | 2022-03-07     | 2023-03-12      | Andalusia      |        1
    103 | Eva Dwon                               |        1 | 2022-02-08     | 2023-03-10      | Anniston       |        1
    104 | Cristal Stewart                        |        1 | 2022-01-12     | 2023-02-12      | Alexander City |        1
    105 | Paris Brown                            |        1 | 2022-01-15     | 2023-01-16      | Alexander City |        1
    106 | Austin Kit                             |        1 | 2022-01-04     | 2023-01-16      | Bessemer       |        1
    107 | Danielle White                         |        2 | 2021-03-03     | 2022-02-01      | Auburn         |        2
    108 | Josh Dickenson                         |        2 | 2021-09-09     | 2022-02-02      | Montgomery     |        1
    109 | Rita Evans                             |        2 | 2021-05-21     | 2023-03-05      | Montgomery     |        1
    110 | Wendy Star                             |        2 | 2021-09-30     | 2022-06-04      | Bessemer       |        1
    111 | Justin Smith                           |        2 | 2021-02-28     | 2022-03-01      | Montgomery     |        2
    112 | Tanya Osten                            |        3 | 2020-03-10     | 2021-02-01      | Bessemer       |        3
    113 | Yeri Pinnet                            |        3 | 2020-09-04     | 2021-02-15      | Auburn         |        2
    114 | Will Suppet                            |        3 | 2020-05-13     | 2021-03-06      | Jasper         |        2
    115 | Jonathon Eyre                          |        3 | 2020-09-30     | 2022-08-01      | Jasper         |        2
    116 | Dana Willton                           |        3 | 2020-02-22     | 2022-08-01      | Clanton        |        3
    117 | Rod Wine                               |        3 | 2020-01-30     | 2021-02-02      | Auburn         |        3
    118 | Sam Osten                              |        4 | 2019-03-12     | 2020-04-12      | Montgomery     |        4
    119 | Kate Blake                             |        4 | 2019-01-13     | 2020-02-12      | Clanton        |        4
    120 | Roxanna Dastin                         |        4 | 2019-08-30     | 2020-09-12      | Montgomery     |        3
    122 | Nina Flores                            |        4 | 2019-02-27     | 2020-02-28      | Montgomery     |        4
    123 | Nina Adams                             |        5 | 2018-03-23     | 2022-03-24      | Huntsville     |        5
    124 | Linda Scott                            |        5 | 2018-01-25     | 2019-02-20      | Montgomery     |        5
    125 | Sarah Green                            |        5 | 2018-02-24     | 2019-09-29      | Montgomery     |        5
    126 | Scott Nelson                           |        5 | 2018-01-30     | 2021-08-25      | Huntsville     |        5
    127 | Jane Rivera                            |        5 | 2018-04-26     | 2020-04-05      | Montgomery     |        4
    128 | Mike Mitchell                          |        5 | 2018-03-09     | 2019-04-06      | Montgomery     |        5
    129 | Bob King                               |        6 | 2017-05-14     | 2019-10-15      | Montgomery     |        5
    130 | Janett Torres                          |        6 | 2016-12-16     | 2019-10-16      | Montgomery     |        5
    131 | Peter Hall                             |        7 | 2016-08-18     | 2017-08-31      | Huntsville     |        6
    132 | Nelly Wright                           |        7 | 2016-09-17     | 2018-02-26      | Auburn         |        6
    121 | Lily Valley                            |        4 | 2019-09-20     | 2020-09-21      | Montgomery     |        3
(32 строки)
Сигнал отмены отправлен
kindergarten=# insert into kids (kid_id, kid_name, group_id, kid_birth_date, enrollment_date, hometown) values ('133', 'Hillary John', 1, '2021-06-09', '202
2-09-06', 'Auburn');
INSERT 0 1
kindergarten=# insert into kids (kid_id, kid_name, group_id, kid_birth_date, enrollment_date, hometown) values ('135', 'Lily Valley', 1, '2021-06-09', '2022
-09-06', 'Auburn');
ОШИБКА:  Ребенок Lily Valley                              уже есть в списках
КОНТЕКСТ:  функция PL/pgSQL xkid(), строка 4, оператор RAISE
```

6. Модуль TABLEFUNC. Сводная таблица дат поступления детей в детский сад по годам:

```
kindergarten=# SELECT DATE_TRUNC('year', enrollment_date) AS YEAR,
kindergarten-# SUM(CASE WHEN EXTRACT(MONTH FROM enrollment_date) = 1 THEN 1 ELSE 0 END) AS JAN,
kindergarten-# SUM(CASE WHEN EXTRACT(MONTH FROM enrollment_date) = 2 THEN 1 ELSE 0 END) AS FEB,
kindergarten-# SUM(CASE WHEN EXTRACT(MONTH FROM enrollment_date) = 3 THEN 1 ELSE 0 END) AS MAR,
kindergarten-# SUM(CASE WHEN EXTRACT(MONTH FROM enrollment_date) = 4 THEN 1 ELSE 0 END) AS APR,
kindergarten-# SUM(CASE WHEN EXTRACT(MONTH FROM enrollment_date) = 5 THEN 1 ELSE 0 END) AS MAY,
kindergarten-# SUM(CASE WHEN EXTRACT(MONTH FROM enrollment_date) = 6 THEN 1 ELSE 0 END) AS JUN,
kindergarten-# SUM(CASE WHEN EXTRACT(MONTH FROM enrollment_date) = 7 THEN 1 ELSE 0 END) AS JUL,
kindergarten-# SUM(CASE WHEN EXTRACT(MONTH FROM enrollment_date) = 8 THEN 1 ELSE 0 END) AS AUG,
kindergarten-# SUM(CASE WHEN EXTRACT(MONTH FROM enrollment_date) = 9 THEN 1 ELSE 0 END) AS SEP,
kindergarten-# SUM(CASE WHEN EXTRACT(MONTH FROM enrollment_date) = 10 THEN 1 ELSE 0 END) AS OCT,
kindergarten-# SUM(CASE WHEN EXTRACT(MONTH FROM enrollment_date) = 11 THEN 1 ELSE 0 END) AS NOV,
kindergarten-# SUM(CASE WHEN EXTRACT(MONTH FROM enrollment_date) = 12 THEN 1 ELSE 0 END) AS DEC
kindergarten-# FROM kids
kindergarten-# GROUP BY 1
kindergarten-# ORDER BY 1;
          year          | jan | feb | mar | apr | may | jun | jul | aug | sep | oct | nov | dec
------------------------+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
 2017-01-01 00:00:00+03 |   0 |   0 |   0 |   0 |   0 |   0 |   0 |   1 |   0 |   0 |   0 |   0
 2018-01-01 00:00:00+03 |   0 |   1 |   0 |   0 |   0 |   0 |   0 |   0 |   0 |   0 |   0 |   0
 2019-01-01 00:00:00+03 |   0 |   1 |   0 |   1 |   0 |   0 |   0 |   0 |   1 |   2 |   0 |   0
 2020-01-01 00:00:00+03 |   0 |   2 |   0 |   2 |   0 |   0 |   0 |   0 |   2 |   0 |   0 |   0
 2021-01-01 00:00:00+03 |   0 |   3 |   1 |   0 |   0 |   0 |   0 |   1 |   0 |   0 |   0 |   0
 2022-01-01 00:00:00+03 |   0 |   2 |   2 |   0 |   0 |   1 |   0 |   2 |   1 |   0 |   0 |   0
 2023-01-01 00:00:00+03 |   2 |   1 |   4 |   0 |   0 |   0 |   0 |   0 |   0 |   0 |   0 |   0
(7 строк)


kindergarten=#
```

7. Словарь метаданных

Получим список ограничений:

```
SQL Shell (psql)
kindergarten=# select * from information_schema.table_constraints;
 constraint_catalog | constraint_schema |              constraint_name               | table_catalog | table_schema |        table_name        | constra
int_type | is_deferrable | initially_deferred | enforced | nulls_distinct
-------------------+-------------------+--------------------------------------------+---------------+--------------+--------------------------+--------
---------+---------------+--------------------+----------+----------------
 kindergarten       | pg_catalog        | pg_proc_oid_index                          | kindergarten  | pg_catalog   | pg_proc                  | PRIMARY
 KEY      | NO            | NO                 | YES      |
 kindergarten       | pg_catalog        | pg_proc_proname_args_nsp_index             | kindergarten  | pg_catalog   | pg_proc                  | UNIQUE
          | NO            | NO                 | YES      | YES
 kindergarten       | pg_catalog        | pg_type_oid_index                          | kindergarten  | pg_catalog   | pg_type                  | PRIMARY
 KEY      | NO            | NO                 | YES      |
 kindergarten       | pg_catalog        | pg_type_typname_nsp_index                  | kindergarten  | pg_catalog   | pg_type                  | UNIQUE
          | NO            | NO                 | YES      | YES
 kindergarten       | pg_catalog        | pg_attribute_relid_attnam_index            | kindergarten  | pg_catalog   | pg_attribute             | UNIQUE
          | NO            | NO                 | YES      | YES
 kindergarten       | pg_catalog        | pg_attribute_relid_attnum_index            | kindergarten  | pg_catalog   | pg_attribute             | PRIMARY
 KEY      | NO            | NO                 | YES      |
 kindergarten       | pg_catalog        | pg_class_oid_index                         | kindergarten  | pg_catalog   | pg_class                 | PRIMARY
 KEY      | NO            | NO                 | YES      |
 kindergarten       | pg_catalog        | pg_class_relname_nsp_index                 | kindergarten  | pg_catalog   | pg_class                 | UNIQUE
          | NO            | NO                 | YES      | YES
 kindergarten       | pg_catalog        | pg_attrdef_adrelid_adnum_index             | kindergarten  | pg_catalog   | pg_attrdef               | UNIQUE
          | NO            | NO                 | YES      | YES
 kindergarten       | pg_catalog        | pg_attrdef_oid_index                       | kindergarten  | pg_catalog   | pg_attrdef               | PRIMARY
 KEY      | NO            | NO                 | YES      |
 kindergarten       | pg_catalog        | pg_constraint_conrelid_contypid_conname_index | kindergarten  | pg_catalog   | pg_constraint            | UNIQUE
          | NO            | NO                 | YES      | YES
 kindergarten       | pg_catalog        | pg_constraint_oid_index                    | kindergarten  | pg_catalog   | pg_constraint            | PRIMARY
 KEY      | NO            | NO                 | YES      |
 kindergarten       | pg_catalog        | pg_inherits_relid_seqno_index              | kindergarten  | pg_catalog   | pg_inherits              | PRIMARY
 KEY      | NO            | NO                 | YES      |
 kindergarten       | pg_catalog        | pg_index_indexrelid_index                  | kindergarten  | pg_catalog   | pg_index                 | PRIMARY
 KEY      | NO            | NO                 | YES      |
 kindergarten       | pg_catalog        | pg_operator_oid_index                      | kindergarten  | pg_catalog   | pg_operator              | PRIMARY
 KEY      | NO            | NO                 | YES      |
 kindergarten       | pg_catalog        | pg_operator_oprname_l_r_n_index            | kindergarten  | pg_catalog   | pg_operator              | UNIQUE
          | NO            | NO                 | YES      | YES
 kindergarten       | pg_catalog        | pg_opfamily_am_name_nsp_index              | kindergarten  | pg_catalog   | pg_opfamily              | UNIQUE
          | NO            | NO                 | YES      | YES
 kindergarten       | pg_catalog        | pg_opfamily_oid_index                      | kindergarten  | pg_catalog   | pg_opfamily              | PRIMARY
```

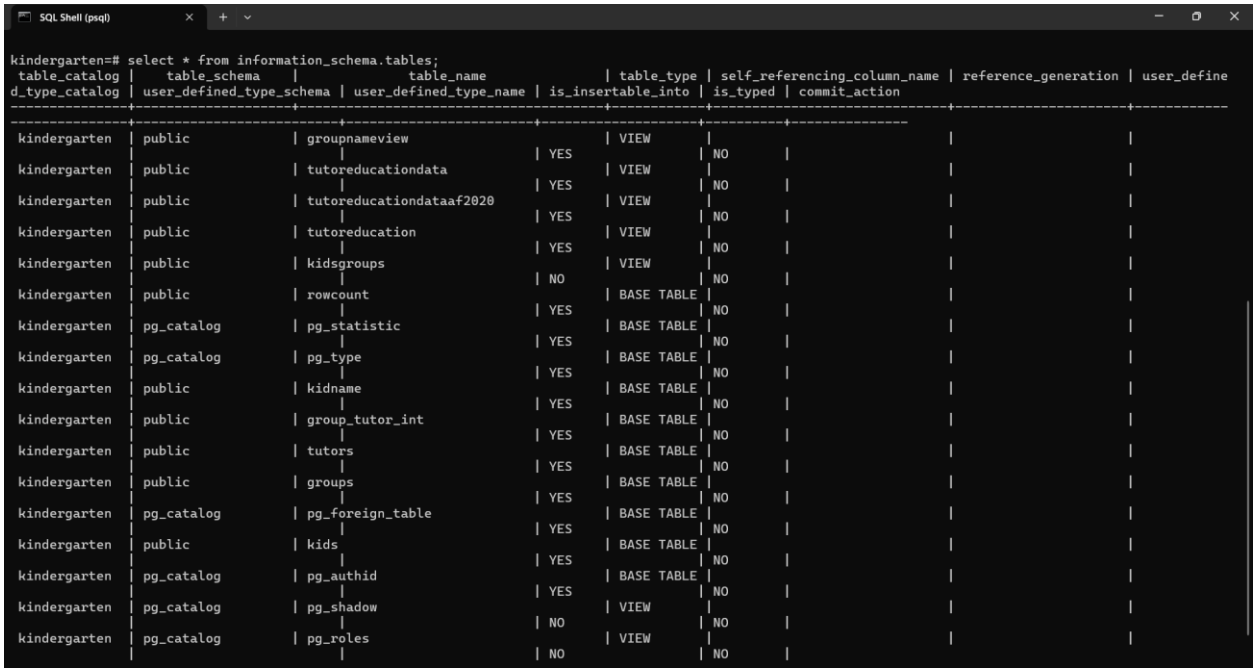
Получим список внешних ключей:

```
kindergarten=# select * from information_schema.referential_constraints;
 constraint_catalog | constraint_schema |     constraint_name     | unique_constraint_catalog | unique_constraint_schema | unique_constraint_name | match_op
tion | update_rule | delete_rule
--------------------+-------------------+-------------------------+---------------------------+--------------------------+------------------------+---------
-----+-------------+-------------
 kindergarten       | public            | group_tutor_int_tutorfk | kindergarten              | public                   | tutorpk                | NONE
     | NO ACTION   | CASCADE
 kindergarten       | public            | group_tutor_fk          | kindergarten              | public                   | tutorpk                | NONE
     | NO ACTION   | NO ACTION
 kindergarten       | public            | tutor_group_int_groupfk | kindergarten              | public                   | grouppk                | NONE
     | NO ACTION   | CASCADE
 kindergarten       | public            | kid_group_fk            | kindergarten              | public                   | grouppk                | NONE
     | NO ACTION   | NO ACTION
 kindergarten       | public            | kid_group_fk            | kindergarten              | public                   | grouppk                | NONE
     | NO ACTION   | NO ACTION
 kindergarten       | public            | kid_fk                  | kindergarten              | public                   | kidpk                  | NONE
     | NO ACTION   | NO ACTION
(6 строк)
```

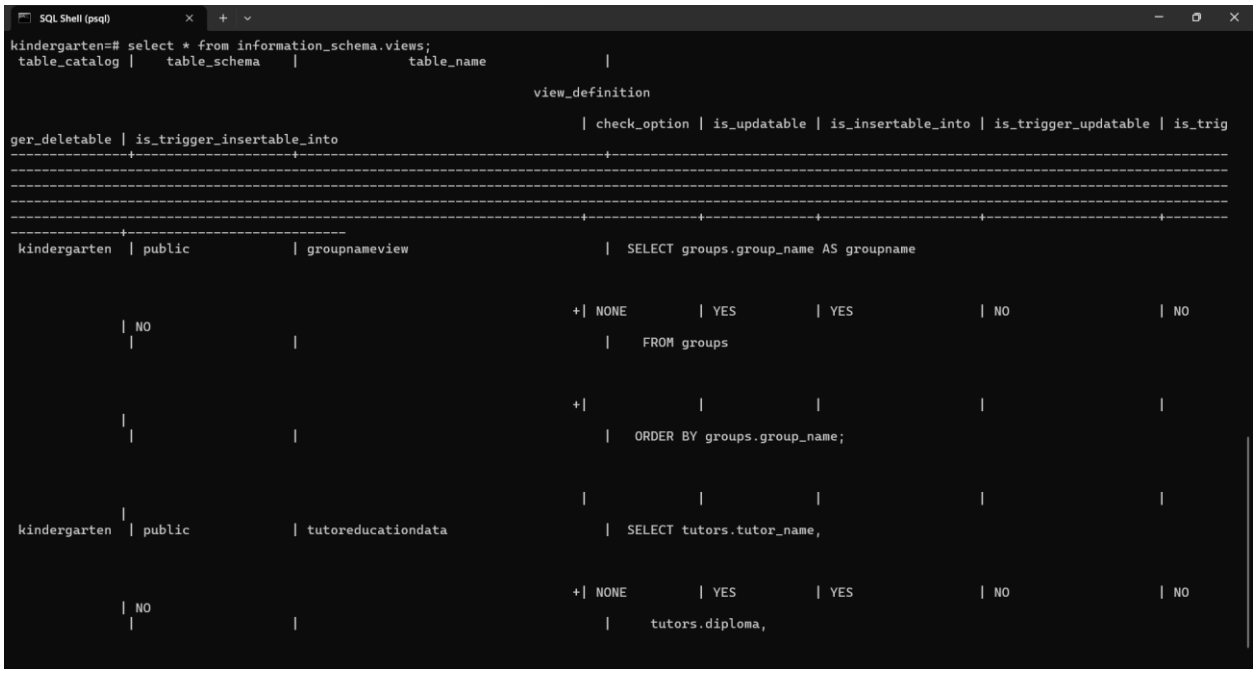
Получим список последовательностей:

```
kindergarten=# select * from information_schema.sequences;
 sequence_catalog | sequence_schema | sequence_name | data_type | numeric_precision | numeric_precision_radix | numeric_scale | start_value | minimum_value
|    maximum_value    | increment | cycle_option
------------------+-----------------+---------------+-----------+-------------------+-------------------------+---------------+-------------+---------------
+---------------------+-----------+--------------
 kindergarten     | public          | seq_group     | bigint    |                64 |                       2 |             0 | 1           | 0
| 9223372036854775807 | 1         | NO
 kindergarten     | public          | seq_tutor     | bigint    |                64 |                       2 |             0 | 21          | 20
| 9223372036854775807 | 1         | NO
 kindergarten     | public          | seq_kid       | bigint    |                64 |                       2 |             0 | 101         | 101
| 9223372036854775807 | 1         | NO
(3 строки)
```

Получим список таблиц:



Получим список представлений:



Получим список хранимых процедур:

Получим список триггеров:

Вывод: в данной практической работе мы работали с представлениями, хранимыми процедурами и триггерами.