**Цель работы**: Изучить возможности операторов SELECT и VIEW SQL для создания запросов и представлений.

**Задание**

1. Используя оператор SELECT языка SQL, создать для базы данных, которая была создана в лабораторной работе №3, следующие однотабличные и многотабличные запросы (МySQL или PostgreSQL):

* запрос, выводящий все строки таблицы;
* запрос с порядком столбцов таблицы, отличным от исходного;
* запрос, использующий WHERE с составным условием;
* запросы с использованием IN,  BETWEEN,  LIKE в условии;
* запрос с использованием какой-либо из агрегатных функций COUNT, SUM, AVG, MAX, MIN;
* Запросы, демонстрирующие использование команд ORDER BY, GROUP BY и HAVING;
* запросы, соединяющие таблицы с помощью JOIN и без него;
* запросы с подзапросами (с использованием агрегатных функций в подзапросе; с использованием подзапросов, возвращающие единственное и множественные значения; с использованием подзапросов, демонстрирующие использование операторов ALL и ANY).

1. Создать для базы данных, которая была создана в лабораторной работе №3, обновляемое и не обновляемое представления. Просмотреть хранимые данные, используя представления. Используя обновляемое представление внести изменение в таблицу. Просмотреть полученный результат (MySQL или PostgreSQL).
2. Создать запросы, сформулированные в описании предметной области курсовой работы (MySQL или PostgreSQL).

**Задание 1 (PostgreSQL) :**

- запрос, выводящий все строки таблицы

select\* from "Musikalnaya\_gruppa";

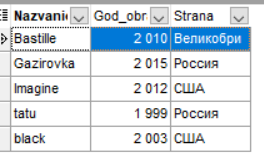


Рисунок 1 - запрос, выводящий все строки таблицы «Музыкальная\_группа»

- запрос с порядком столбцов таблицы, отличным от исходного;

select "Nazvanie\_gruppi", "Nazvanie\_pesni", "Avtor\_teksta", "Data\_sozdania", "Yazik\_pesni" from "Pesnya";

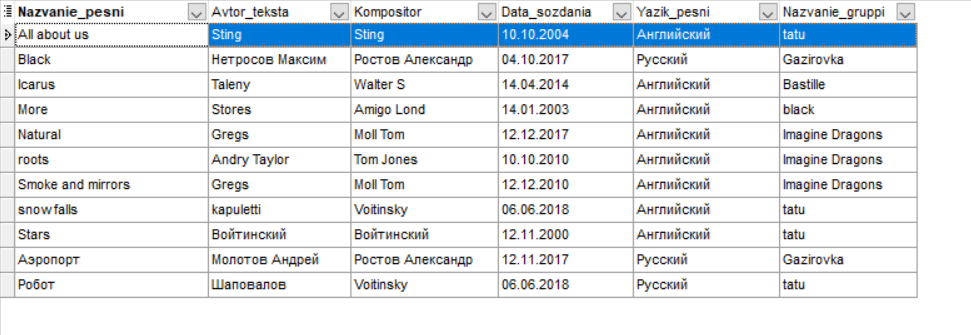
Рисунок 2 — стандартный порядок столбцов таблицы «Песня»



Рисунок 3 - запрос на отличный от исходного порядок столбцов таблицы «Песня»

- запрос, использующий WHERE с составным условием (вывести исполнителей, чей возраст не превышает 20);

select "Psevdonim\_Familia", "Vozrast" from "Ispolnitel" where "Vozrast"<20;

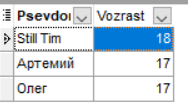


Рисунок 4 — запрос, выводящий всех исполнителей не старше 20 лет

- запросы с использованием IN,  BETWEEN,  LIKE в условии;

select "Nazvanie\_gruppi","Nazvanie\_pesni" from "Pesnya" where "Nazvanie\_gruppi" In ('tatu','black');



Рисунок 5 — запрос, выводящий название песен двух групп (tatu и black)

select "Nazvanie\_gruppi", "Nazvanie", "Mesto\_vistupleniya", "Data\_vistuplenia"

from "Vistuplenie"

where "Data\_vistuplenia" between '06.06.2018' and '11.11.2018';



Рисунок 6 — все выступления

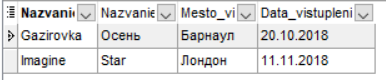


Рисунок 7 — запрос, выводящий все выступления в диапазоне 06.06.2018-11.11.2018

select "Psevdonim\_Familia", "Amplua", "Identifikazionnij\_nomer\_ispolnitelya" from "Ispolnitel"

where "Psevdonim\_Familia" like '%а';

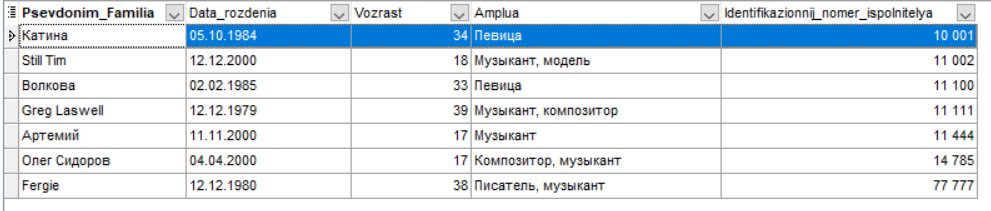


Рисунок 8 — все исполнители

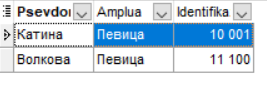


Рисунок 9 — запрос, выводящий всех исполнителей, чей псевдоним заканчивается на символ «а»

- запрос с использованием какой-либо из агрегатных функций COUNT, SUM, AVG, MAX, MIN;

select COUNT("Nazvanie\_pesni")

from "Pesnya"

where "Nazvanie\_gruppi" = 'tatu';

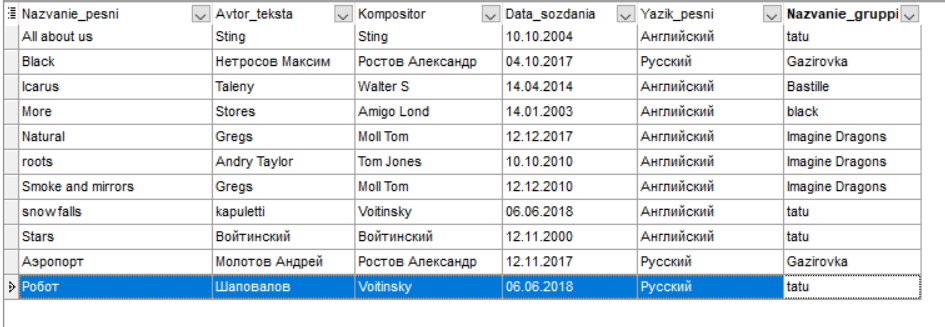


Рисунок 10 — таблица «Песня»



Рисунок 10 — запрос, считающий количество песен у группы tatu

select SUM("Vozrast")

from "Ispolnitel";



Рисунок 11 — запрос, выводящий общую сумму возраста всех исполнителей

select AVG("Vozrast")

from "Ispolnitel";



Рисунок 12 — запрос, выводящий средний возраст исполнителей

select MAX("Vozrast")

from "Ispolnitel";



Рисунок 13 — запрос, выводящий максимальный возраст среди исполнителей

select MIN("Vozrast")

from "Ispolnitel";



Рисунок 14 — запрос, выводящий минимальный возраст среди исполнителей

- Запросы, демонстрирующие использование команд ORDER BY, GROUP BY и HAVING;

select "Psevdonim\_Familia", "Vozrast"

from "Ispolnitel"

order by "Vozrast" ASC;



Рисунок 15 — сортировка исполнителей по возрастанию возраста

select "Psevdonim\_Familia", "Vozrast"

from "Ispolnitel"

order by "Psevdonim\_Familia" DESC;

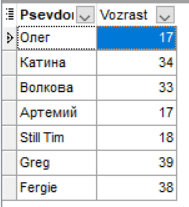


Рисунок 16 — сортировка по убыванию (поле — Псевдоним\_Фамилия)

select "Nazvanie\_gruppi", count(\*)

from "Pesnya"

group by "Nazvanie\_gruppi";

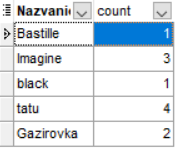


Рисунок 17 — запрос, выводящий количество песен каждой группы

select "Nazvanie\_gruppi", count(\*)

from "Pesnya"

where "Yazik\_pesni" = 'Русский'

group by "Nazvanie\_gruppi";

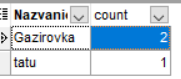


Рисунок 18 - запрос, выводящий количество русскоязычных песен у групп

select "Nazvanie\_gruppi", count(\*)

from "Pesnya"

where "Data\_sozdania" between '10.01.2000' and '20.12.2018'

group by "Nazvanie\_gruppi"

having count(\*) = 4;



Рисунок 19 — запрос, который определяет группу, у которой за период 10.01.2000 — 20.12.2018 было выпущено 4 песни и выводит название группы и количество этих песен

- запросы, соединяющие таблицы с помощью JOIN и без него;

С помощью условия в разделе WHERE :

select "Nazvanie\_pesni", "Strana", "Avtor\_teksta", "Yazik\_pesni"

from "Musikalnaya\_gruppa", "Pesnya"

where "Musikalnaya\_gruppa"."Nazvanie\_gruppi" = "Pesnya"."Nazvanie\_gruppi";



Рисунок 20 — запрос, выводящий название песни, автора текста и язык песни (из таблицы «Песня»), а также страну (таблица «Музыкальная группа»)

- С помощью оператора JOIN :

select "Nazvanie\_pesni", "Strana", "Avtor\_teksta", "Yazik\_pesni"

from "Musikalnaya\_gruppa" join "Pesnya"

on "Musikalnaya\_gruppa"."Nazvanie\_gruppi" = "Pesnya"."Nazvanie\_gruppi";

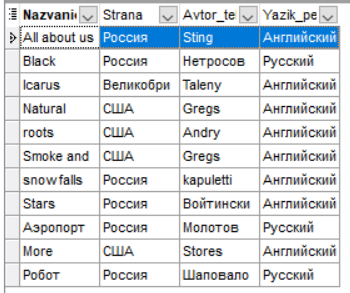


Рисунок 21 - запрос, выводящий название песни, автора текста и язык песни (из таблицы «Песня»), а также страну (таблица «Музыкальная группа»)

- Соединение трех таблиц (Музыкальная\_группа, Песня, Место\_в\_последнем\_хит\_параде) :

select "Nazvanie\_pesni", "Strana", "Avtor\_teksta", "Yazik\_pesni", "Nomer\_v\_hit\_parade"

from ("Musikalnaya\_gruppa" join "Pesnya" on "Musikalnaya\_gruppa"."Nazvanie\_gruppi" = "Pesnya"."Nazvanie\_gruppi")

join "Mesto\_v\_poslednem\_hit\_parade" on "Mesto\_v\_poslednem\_hit\_parade"."Nazvanie\_gruppi" = "Pesnya"."Nazvanie\_gruppi";



Рисунок 22 - запрос, выводящий название песни, автора текста и язык песни (из таблицы «Песня»), страну (таблица «Музыкальная группа»), номер в последнем хит-параде (таблица «Место в последнем хит-параде»)

- Запросы с подзапросами (с использованием агрегатных функций в подзапросе; с использованием подзапросов, возвращающие единственное и множественные значения; с использованием подзапросов, демонстрирующие использование операторов ALL и ANY).

- Запрос с использованием агрегатных функций в подзапросе; с использованием подзапросов, возвращающие единственное значение.

Репертуар наиболее популярной группы :

select "Pesnya"."Nazvanie\_pesni", "Musikalnaya\_gruppa"."Nazvanie\_gruppi"

from ("Musikalnaya\_gruppa" join "Pesnya" on "Musikalnaya\_gruppa"."Nazvanie\_gruppi" = "Pesnya"."Nazvanie\_gruppi")

join "Mesto\_v\_poslednem\_hit\_parade" on "Mesto\_v\_poslednem\_hit\_parade"."Nazvanie\_gruppi" = "Pesnya"."Nazvanie\_gruppi"

where "Nomer\_v\_hit\_parade" =

( select min("Nomer\_v\_hit\_parade")

from "Mesto\_v\_poslednem\_hit\_parade"

);

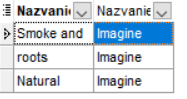


Рисунок 23 — запрос с подзапросом, возвращающим одно значение

- Запрос с подзапросом, возвращающим множественные значения.

Получить информацию об исполнителях, возраст которых не превышает 20.

SELECT\*

from "Ispolnitel"

where "Psevdonim\_Familia" in

( select "Psevdonim\_Familia"

from "Ispolnitel"

where "Vozrast" <= 20);

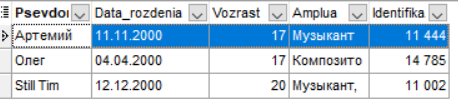


Рисунок 24 — запрос с подзапросом, возвращающим несколько значений

- Запрос с использованием подзапроса, демонстрирующего использование операторов ALL и ANY.

Вывести информацию об исполнителях, возраст которых равен 20 :

SELECT\*

from "Ispolnitel"

where "Psevdonim\_Familia" = ANY

( select "Psevdonim\_Familia"

from "Ispolnitel"

where "Vozrast" = 20);

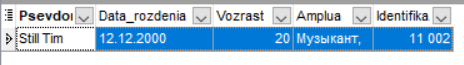


Рисунок 25 - Запрос с использованием подзапроса, демонстрирующего использование оператора ANY

Вывести группы, в состав которых входят исполнители со средним возрастом, не превышающим 20. Вывести псевдонимы исполнителей :

select "Musikalnaya\_gruppa"."Nazvanie\_gruppi", "Ispolnitel"."Psevdonim\_Familia"

from ("Ispolnitel" join "Musikalnaya\_gruppa\_ispolnitel" on "Musikalnaya\_gruppa\_ispolnitel" ."Identifikazionnij\_nomer\_ispolnitelya" = "Ispolnitel"."Identifikazionnij\_nomer\_ispolnitelya") join "Musikalnaya\_gruppa" on "Musikalnaya\_gruppa\_ispolnitel"."Nazvanie\_gruppi" = "Musikalnaya\_gruppa"."Nazvanie\_gruppi"

where "Vozrast" <= ALL

(select AVG("Vozrast")

from "Ispolnitel"

group by "Psevdonim\_Familia");

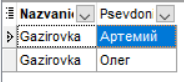


Рисунок 26 - Запрос с использованием подзапроса, демонстрирующего использование оператора ALL

**Задание 2 (PostgreSQL) :**

Создать для базы данных, которая была создана в лабораторной работе №3, обновляемое и не обновляемое представления. Просмотреть хранимые данные, используя представления. Используя обновляемое представление внести изменение в таблицу. Просмотреть полученный результат.

В EMS SQL Manager for MySQL для работы с представлениями существует удобный интерфейс, который доступен из подменю «Представления» выбранной базы данных. Для сохранения представления используется команда «Компилировать». Так же можно использовать для создания представления инструмент SQL-скрипт.

Воспользуемся SQL-скриптом для создания представления для базы данных «Музыкальные группы». Представление выводит информацию о исполнителях, возраст которых более 20.

create view Vozrast\_more\_20 as

select "Psevdonim\_Familia", "Vozrast"

from "Ispolnitel"

where "Vozrast" > 20

with check option;



Рисунок 27 - данные, которые отображает представление Vozrast\_more\_20

Следующий оператор UPDATE выполняет обновление представления Vozrast\_more\_20. Рисунок 28 демонстрирует данные, которые отображает представление после обновления.

Update "vozrast\_more\_20" set "Vozrast" = 40

where "Vozrast" = 39;



Рисунок 28 - данные, которые отображает представление Vozrast\_more\_20 после обновления

Следующее представление не является обновляемым. Представление выводит псевдоним и возраст исполнителей, возраст которых больше среднего :

create view "vozrast\_more\_25" as

select "Psevdonim\_Familia", "Vozrast"

from "Ispolnitel"

where "Vozrast" > 25

update vozrast\_more\_25 set "Vozrast" = '15'

where "Vozrast" = '34'

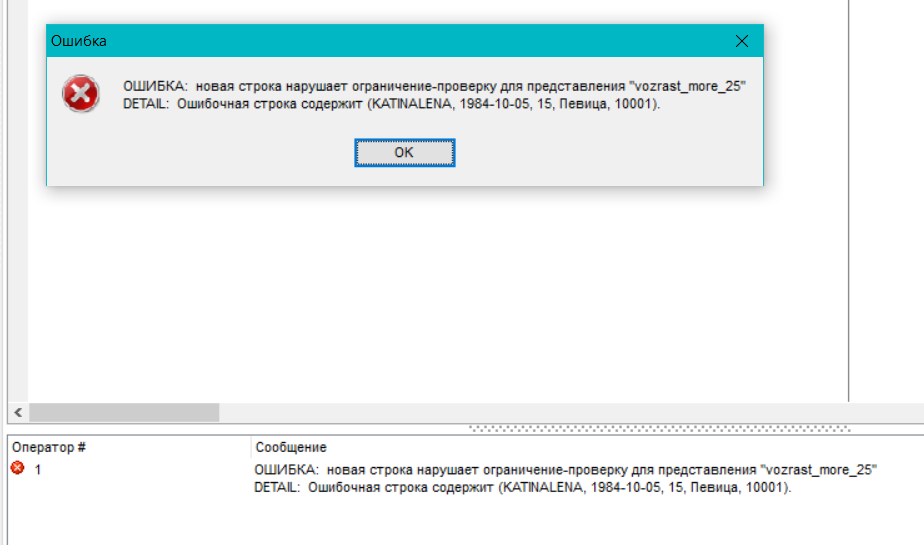


Рисунок 29 — сообщение об ошибке обновления

**Задание 3 (PostgreSQL) :**

Создать запросы, сформулированные в описании предметной области курсовой работы (MySQL или PostgreSQL).

Автор текста, композитор и дата создания песни с данным названием? В репертуар какой группы она входит?

select "Nazvanie\_pesni","Avtor\_teksta", "Kompositor", "Data\_sozdania", "Nazvanie\_gruppi"

from "Pesnya"

where "Nazvanie\_pesni" = 'All about us';



Рисунок 30 — запрос, выводящий автора текста, композитора, дату создания песни с данным названием, а также название группы, в репертуар которой данная песня входит

Репертуар наиболее популярной группы?

select "Pesnya".\*

from ("Pesnya" join "Musikalnaya\_gruppa" on "Pesnya"."Nazvanie\_gruppi" = "Musikalnaya\_gruppa"."Nazvanie\_gruppi") join "Mesto\_v\_poslednem\_hit\_parade" on "Mesto\_v\_poslednem\_hit\_parade"."Nazvanie\_gruppi" = "Musikalnaya\_gruppa"."Nazvanie\_gruppi"

where "Nomer\_v\_hit\_parade" =

(select min("Nomer\_v\_hit\_parade")

from "Mesto\_v\_poslednem\_hit\_parade");

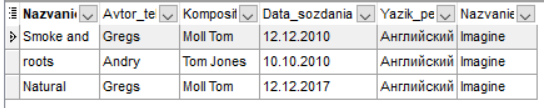


Рисунок 31 — запрос, выводящий репертуар самой популярной группы

Цена билета на последний концерт указанной группы?

select "Musikalnaya\_gruppa"."Nazvanie\_gruppi", "Vistuplenie"."Data\_vistuplenia", "Vistuplenie"."Mesto\_vistupleniya", "Zena"."Zena\_bileta"

from ("Vistuplenie" join "Zena" on "Vistuplenie"."Mesto\_vistupleniya" = "Zena"."Mesto\_vistupleniya") join "Musikalnaya\_gruppa" on "Vistuplenie"."Nazvanie\_gruppi" = "Musikalnaya\_gruppa"."Nazvanie\_gruppi"

where "Data\_vistuplenia" =

(select MAX("Data\_vistuplenia")

from "Vistuplenie"

where "Vistuplenie"."Nazvanie\_gruppi" = 'tatu'

);

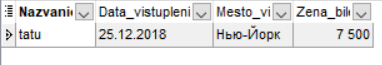


Рисунок 32 - запрос, выводящий цену на последний концерт указанной группы

Состав исполнителей группы с заданным названием, их возраст и амплуа?

select "Musikalnaya\_gruppa"."Nazvanie\_gruppi", "Ispolnitel"."Psevdonim\_Familia", "Ispolnitel"."Vozrast", "Ispolnitel"."Amplua"

from ("Ispolnitel" join "Musikalnaya\_gruppa\_ispolnitel" on "Musikalnaya\_gruppa\_ispolnitel" ."Identifikazionnij\_nomer\_ispolnitelya" = "Ispolnitel"."Identifikazionnij\_nomer\_ispolnitelya") join "Musikalnaya\_gruppa" on "Musikalnaya\_gruppa\_ispolnitel"."Nazvanie\_gruppi" = "Musikalnaya\_gruppa"."Nazvanie\_gruppi"

where "Musikalnaya\_gruppa"."Nazvanie\_gruppi" = 'tatu';

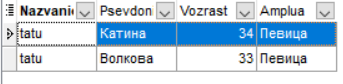


Рисунок 33 — запрос, выводящий по заданному названию группы ее исполнителей, их возраст и амплуа

Место и продолжительность гастролей группы с заданным названием?

select "Mesto","Data\_nachala","Data\_okonchaniya"

from "Gastroli"

where "Nazvanie\_gruppi" = 'tatu';



Рисунок 34 — запрос, выводящий место и продолжительность гастролей с заданным названием группы

В каких группах средний возраст исполнителей не превышает 20 лет?

select "Musikalnaya\_gruppa"."Nazvanie\_gruppi", "Ispolnitel"."Psevdonim\_Familia", "Ispolnitel"."Vozrast"

from ("Ispolnitel" join "Musikalnaya\_gruppa\_ispolnitel" on "Musikalnaya\_gruppa\_ispolnitel" ."Identifikazionnij\_nomer\_ispolnitelya" = "Ispolnitel"."Identifikazionnij\_nomer\_ispolnitelya") join "Musikalnaya\_gruppa" on "Musikalnaya\_gruppa\_ispolnitel"."Nazvanie\_gruppi" = "Musikalnaya\_gruppa"."Nazvanie\_gruppi"

where "Vozrast" <= ALL

(select AVG("Vozrast")

from "Ispolnitel"

group by "Psevdonim\_Familia");

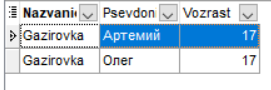


Рисунок 35 — запрос, выводящий группы, в состав которых входят исполнители, средний возраст которых не превышает 20 лет