МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Институт цифровых технологий, электроники и физики

Кафедра вычислительной техники и электроники (ВТиЭ)

Лабораторная работа № 5

**Однотабличные запросы в SQL. Агрегатные функции.**

**Вариант 12.**

Выполнил студент 595 гр.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Лаптев

Проверил:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Я.С. Сергеева

Лабораторная работа защищена

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Барнаул 2021

1. **Создание БД.**

Создание БД производилось в программе MySQL Workbench 8.0. На вкладке File был выбран пункт New Model, после чего открылось окно модели.

Изображение выглядит как текст

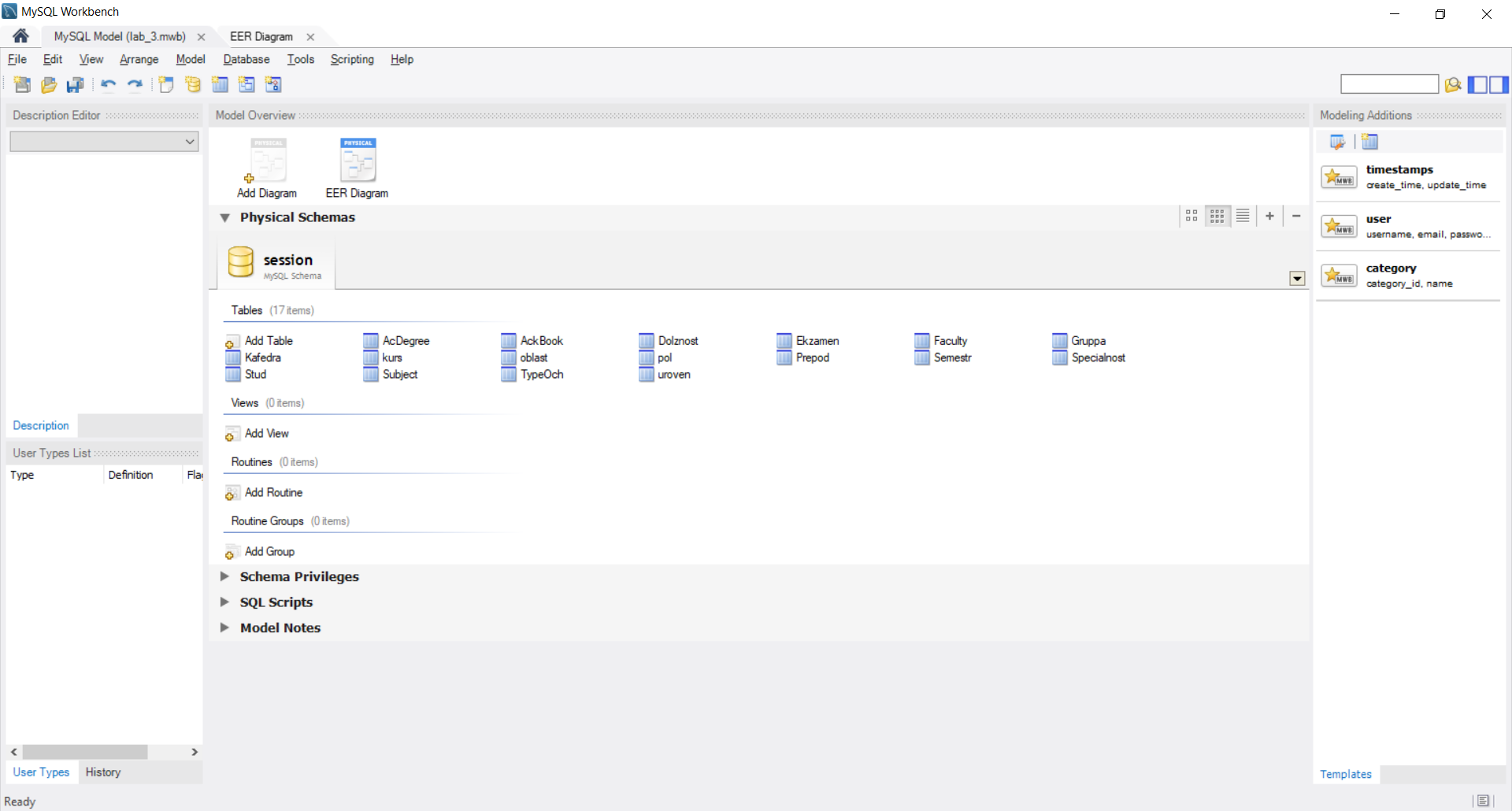
Автоматически созданное описание

*Рис. 1. Окно модели в MySQL Workbench.*

Модель была переименована в «session».

1. **Создание таблиц в БД.**

Далее в окне модели было выбрано поле Add Table на вкладке Table. После чего было создано некоторое количество таблиц с названиями, соответствующими названиям таблиц в концептуальной модели. В каждой таблице добавлены столбцы с нужными именами.

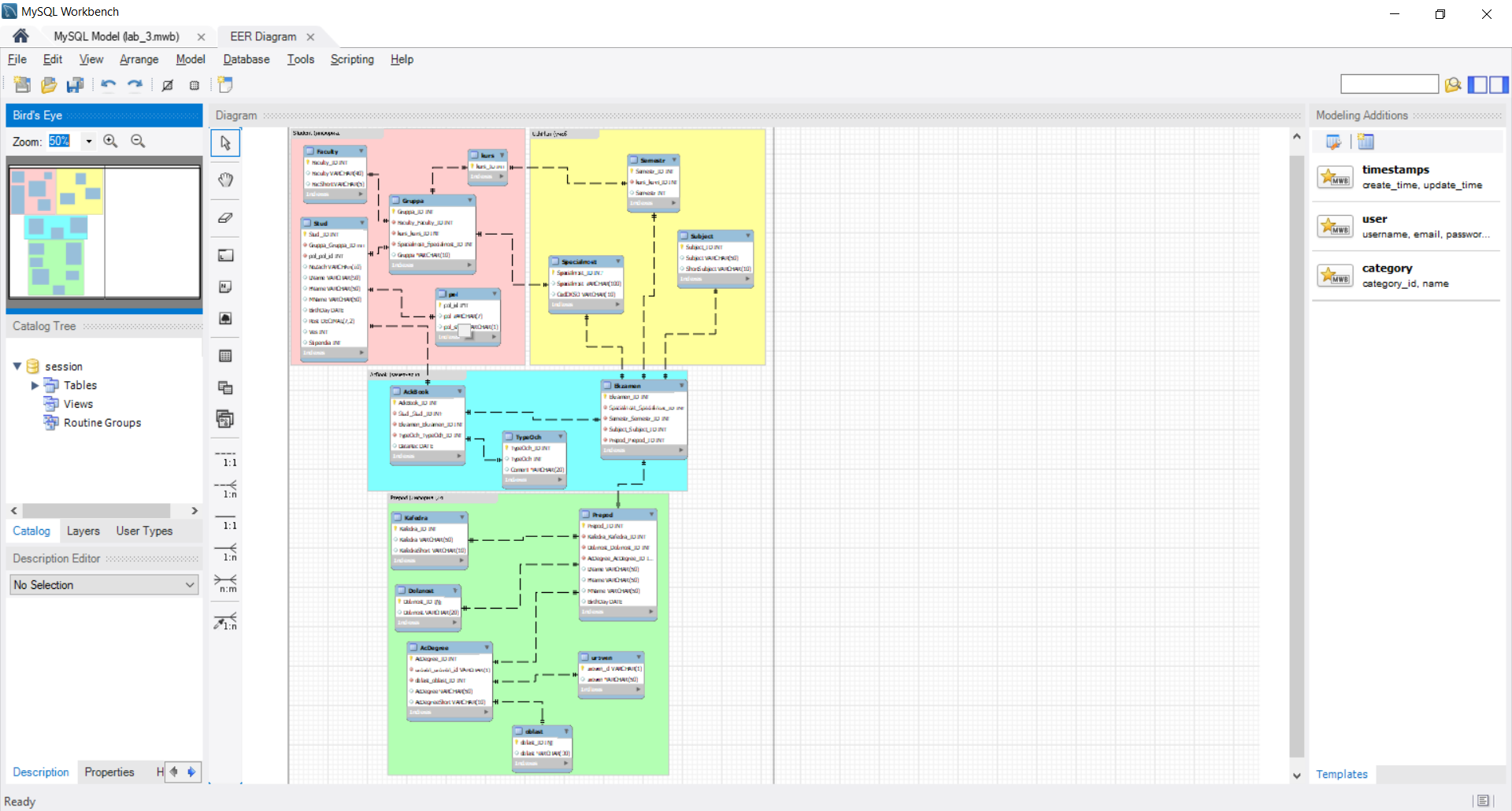


*Рис. 2. Таблицы, которые были созданы в окне модели.*

1. **Создание ER-диаграммы из таблиц в БД.**

Далее была добавлена ER-диаграмма (выбрано поле Add Diagram). Путем перетаскивания таблиц из левой колонки в свободную область были переброшены все созданные таблицы, после чего были сгруппированы, как в концептуальной модели.

После этого каждая из таблиц была соединена между собой соответствующим концептуальной модели образом при помощи связей 1:1.



*Рис. 3. Созданная ER-диаграмма.*

1. **Заполнение таблиц данными.**

Заполнение БД происходило при помощи запросов с использованием скрипта, который предложен в методических указаниях к работе. Перед заполнением скрипта БД нужно было записать на виртуальный сервер. Подробное руководство было также описано в методических указаниях. Скрипт не до конца верен в столбцах, связанных с идентификаторами, но после некоторых исправлений полностью компилируется.

1. **Запросы к БД и результаты их выполнения.**

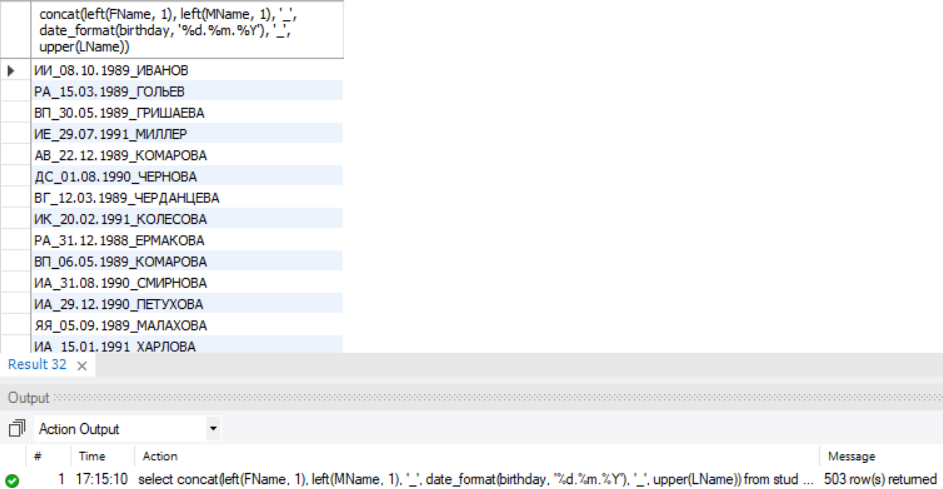
После заполнения БД к ней был применен ряд запросов.

1. Вывести фамилию, имя и дату рождения студентов, которые родились в интервал от 01.01.1983 до 31.12.1992 **в формате «ИО\_дата рождения\_ФАМИЛИЯ».**

select concat(left(FName, 1), left(MName, 1), '\_',

date\_format(birthday, '%d.%m.%Y'), '\_', upper(LName))

from stud where date(BirthDay)>='1983-01-01' and date(BirthDay)<='1991-12-31'



*Рис. 4. Результат запроса.*

1. Вывести большими буквами фамилии, месяц рождения студентов групп 302, 375 и 386.

select upper(LName), date\_format(birthday, '%m')

from stud

where Gruppa\_Gruppa\_ID=302 or Gruppa\_Gruppa\_ID=375 or Gruppa\_Gruppa\_ID=386



*Рис. 5. Результат запроса.*

1. Вывести номер зачетки, фамилию студентов групп 303 и 305 отсортировав список по фамилии.

select NoZach, LName

from stud where Gruppa\_Gruppa\_ID=303 or Gruppa\_Gruppa\_ID=305 order by LName



*Рис. 6. Результат запроса.*

1. Вывести сведения о преподавателях, отсортировав список по имени, фамилии, отчеству.

select \* from prepod order by LName, FName, MName

Изображение выглядит как стол

Автоматически созданное описание

*Рис. 7. Результат запроса.*

1. Вывести количество студентов, минимальную и максимальную дату рождения.

select count(stud\_id), min(birthday), max(birthday) from stud

Изображение выглядит как текст

Автоматически созданное описание

*Рис. 8. Результат запроса.*

1. Вывести список групп и количество студентов в них.

 *Рис. 9. Результат запроса.*

1. Вывести список групп (и количество студентов в них), где количество студентов меньше 20.

select distinct Gruppa\_Gruppa\_ID, count(Gruppa\_Gruppa\_ID)

from stud group by Gruppa\_Gruppa\_ID having count(Gruppa\_Gruppa\_ID)<20

Изображение выглядит как стол

Автоматически созданное описание

*Рис. 10. Результат запроса.*

1. **Индивидуальные запросы к БД и результаты их выполнения.**

После выполнения основной части работы к БД были применены индивидуальные запросы согласно варианту.

1. Выведите количество студентов, носящих одно отчество и определите максимальный рост среди представителей, носящих то или иное отчество. Назовите самое «низкое» отчество.

«отчество - <ОТЧЕСТВО>», «носит <кол-во студентов> человек», «<максимальный рост>»

select distinct concat('отчество - ', MName),

concat('носит ', count(MName), ' человек'), max(Rost)

from stud group by MName

Изображение выглядит как стол

Автоматически созданное описание

*Рис. 11. Результат запроса.*

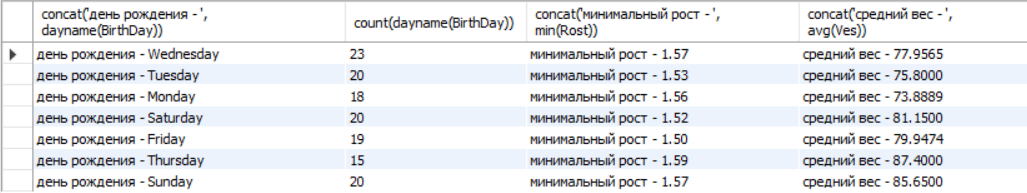
1. Выведите количество студентов с весом ниже 55 и выше 80 кг родившихся в тот или иной день недели, определите их минимальный рост и средний вес. Назовите самый рослый день недели.

select distinct

concat('день рождения - ', dayname(BirthDay)), count(dayname(BirthDay)),

concat('минимальный рост - ', min(Rost)), concat('средний вес - ', avg(Ves))

from stud where Ves<55 or Ves>80 group by dayname(BirthDay)



*Рис. 12. Результат запроса.*

1. Выведите распределение преподавателей по кафедре и году рождения.

(select distinct

concat('название кафедры - ', kafedra),

concat('кол-во преподавателей - ', count(Kafedra\_Kafedra\_ID))

from kafedra, prepod

where kafedra.Kafedra\_ID=prepod.Kafedra\_Kafedra\_ID group by Kafedra)

union

(select distinct

concat('год рождения - ', date\_format(BirthDay, '%Y')),

concat('кол-во преподавателей - ', count(Kafedra\_Kafedra\_ID))

from prepod group by date\_format(BirthDay, '%Y'))

Изображение выглядит как стол

Автоматически созданное описание

*Рис. 13. Результат запроса.*

**Вывод:** в ходе лабораторной работы была проведена работа с агрегатными функциями, были реализованы все предложенные запросы к созданной БД, в т. ч. индивидуальные; в работе с индивидуальными запросами использовались однотабличные запросы и агрегатные функции.