МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

|  |
| --- |
| Институт информационных технологий |
| наименование института (факультета) |
| Кафедра математического и программного обеспечения ЭВМ |
| наименование кафедры  Проектирование баз данных |

наименование дисциплины в соответствии с учебным планом

ЛАБОРАТОРНАЯ РАБОТА №5

«СУБД MySQL»

|  |  |
| --- | --- |
| Исполнитель |  |
| студент | 1ПИб-02-3оп-22 |
|  | группа |
|  | Маркелов С. А. |
|  | Фамилия, имя, отчество |
| Руководитель | Селяничев О. Л. |
|  | Ф.И.О. преподавателя |
| Оценка |  |
| Подпись |  |

2025 год

Первые задания выполняются в консольном клиенте MySQL Command Line Client. Запускаем его (рис. 1).

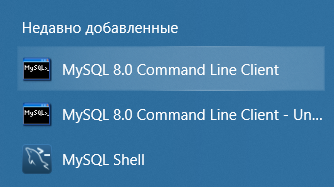


Рис. 1. Запуск MySQL 8.0 Command Line Client

1. Создайте базу данных.

Для создания БД вводим в командную строку SQL-запрос (рис. 2):

*CREATE DATABASE PI33;*

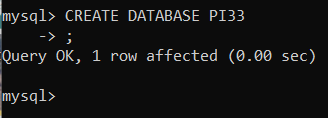


Рис. 2. Создание БД

Выбираем созданную БД для дальнейшего использования с помощью SQL-команды (рис. 3):

*USE PI33*;

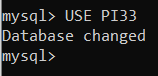


Рис. 3. Выбор БД

1. Создайте таблицу базы данных с полями фамилии, года рождения, оценок по математике, информатике, иностранному языку.

Создаем структуру таблицы с помощью команды (рис. 4):

*CREATE TABLE STUDENTS (id integer auto\_increment primary key, surname varchar(30), birth date, math integer, inf integer, eng integer);*

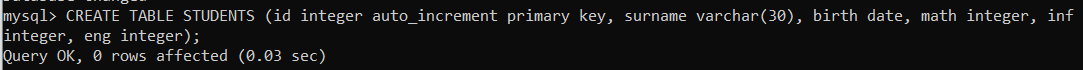


Рис. 4. Создание структуры таблицы

1. Внесите в таблицу 3 записи.

Внесем в таблицу 3 записи с помощью команды (рис. 5):

*INSERT INTO STUDENTS (surname, birth, math, inf, eng) values*

*(‘Маркелов’, ‘2004-07-26’, 5, 5, 5),*

*(‘Подтелков’, ‘2004-06-01’, 5, 5, 5),*

*(‘Беляков’, ‘2004-07-26’, 4, 5, 5);*

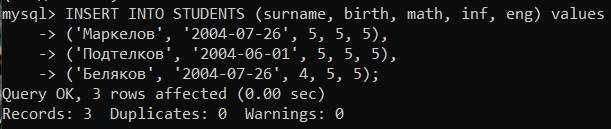


Рис. 5. Добавление записей

Выведем получившуюся таблицу с помощью команды (рис. 6):

*SELECT \* FROM STUDENTS;*

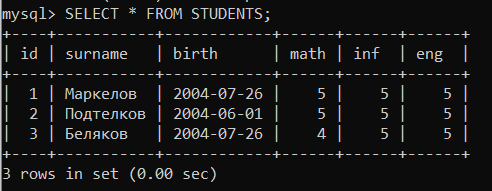


Рис. 6. Вывод таблицы

Дальнейшная работа будет производиться в графическом клиенте MySQL Workbench. Запускаем его (рис. 7).

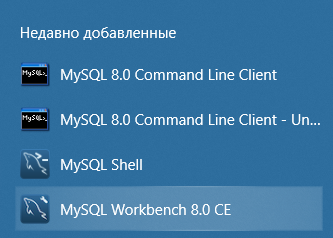


Рис. 7. Запуск MySQL Workbench 8.0 CE

1. Внесите изменения в структуру таблицы – создайте поле, содержащее пол студента.

Добавляем новое поле. Для этого в области редактора запросов вводим команду (рис. 8):

*ALTER TABLE STUDENTS ADD GENDER BIT;*

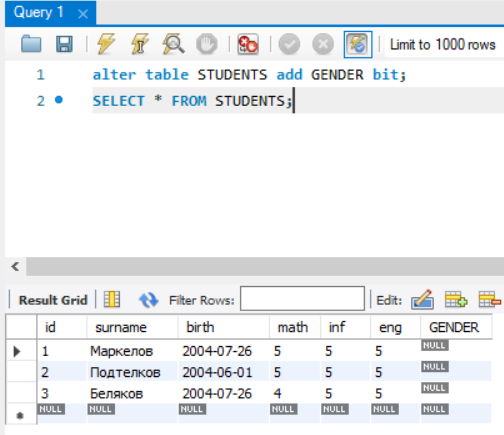


Рис. 8. Добавление поля

1. Дополните данными таблицу, указав пол каждого студента

Для поля GENDER был выбран булевый тип данных BIT. Условимся, что 1 – это мужчина, а 0 – это женщина.

На данный момент в таблице содержатся только мужчины. Чтобы запрос выглядел более наглядно, добавим еще одну запись, где пол будет женский (рис. 9):

*INSERT INTO STUDENTS (surname, birth, math, inf, eng) values*

*(‘Фатькина’, ‘2003-12-23’, 4, 4, 4);*

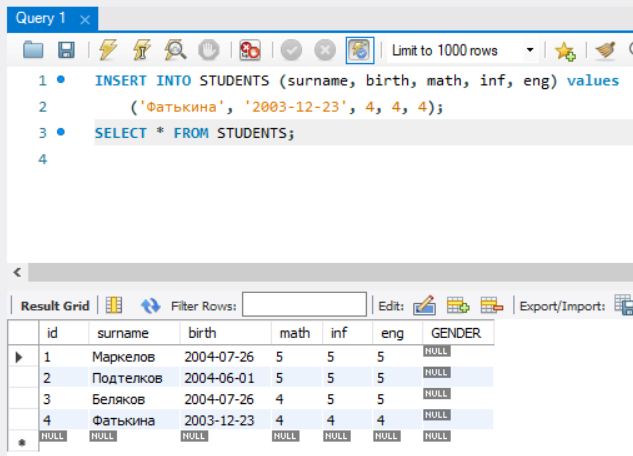


Рис. 9. Добавление записи

Заполняем поле GENDER с помощью команды (рис. 10):

*UPDATE STUDENTS*

*SET GENDER = (surname = ‘Маркелов’ OR surname = ‘Подтелков’ OR surname = ‘Беляков’);*

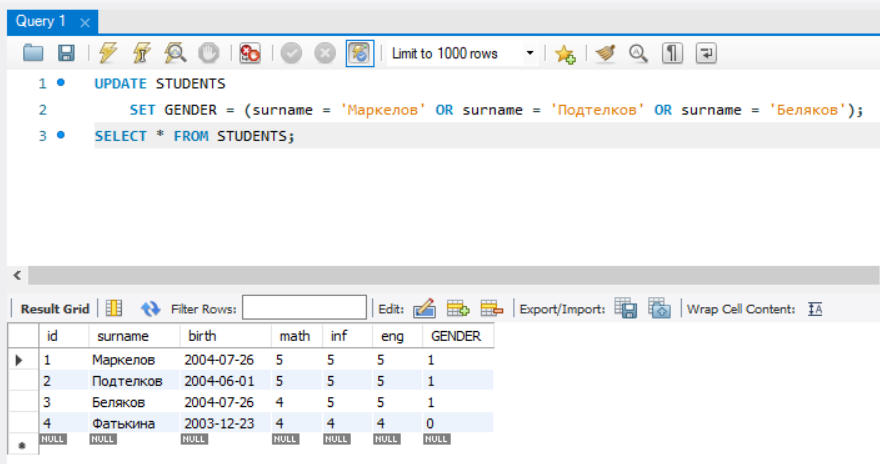


Рис. 10. Заполнение поля GENDER

1. Создайте таблицу с указанными выше полями, которая будет содержать сведения о студентах параллельной группы. Выполните это задание способом, отличным от способа, используемого в задании 2.

Для создания новой таблицы копируем структуру из уже существующей с помощью команды (рис. 11):

*CREATE TABLE PARALLEL LIKE STUDENTS;*



Рис. 11. Копирование структуры таблицы

1. Введите во вторую таблицу 3 записи.

Вводим записи с помощью команды (рис. 12):

*INSERT INTO PARALLEL (surname, birth, math, inf, eng, GENDER) values*

*(‘Иванов’, ‘2001-01-01’, 5, 5, 3, 1),*

*(‘Петрова’, ‘2002-02-02’, 5, 3, 5, 0),*

*(‘Смирнов’, ‘2003-03-03’, 3, 5, 4, 1);*

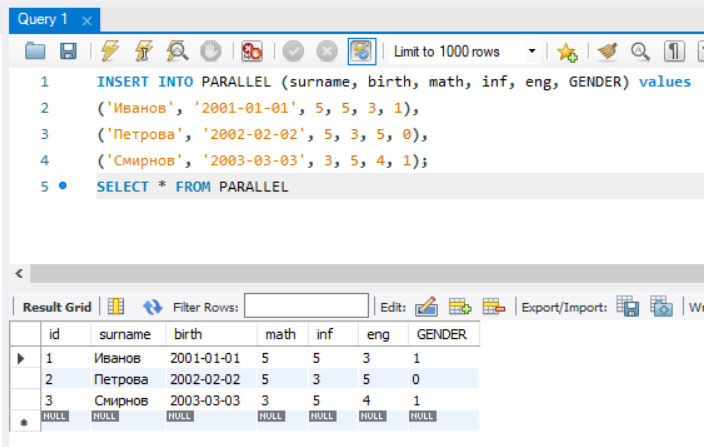


Рис. 12. Внесение записей

1. Выполните сортировку записей первой таблицы в алфавитном порядке по полю фамилий.

Выполняем сортировку с помощью команды (рис. 13):

*SELECT \* FROM STUDENTS*

*ORDER BY SURNAME;*

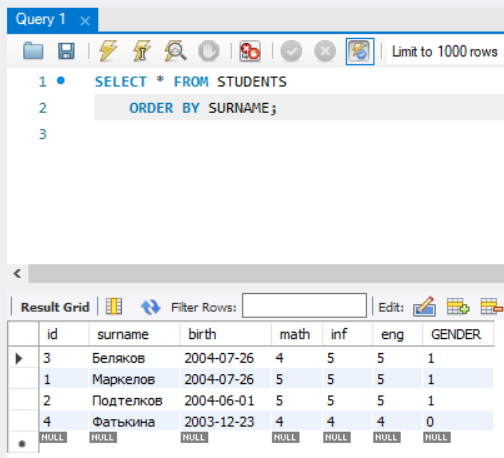


Рис. 13. Сортировка записей

Сортировка выполнена верно, фамилии расположены по алфавиту – от Б до Ф.

1. Сформируйте запросы на выборку:

а) фамилий и даты рождения студентов;

Для данного запроса используем команду (рис. 14):

*SELECT surname, birth FROM STUDENTS;*

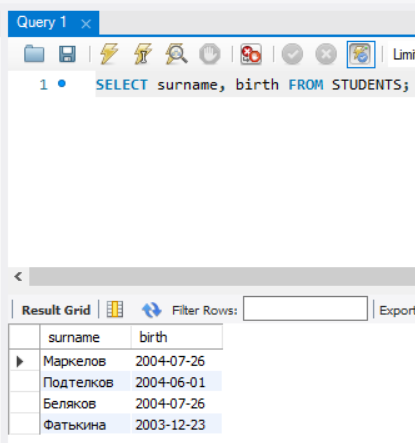


Рис. 14. Запрос на фамилии и даты рождения

Запрос выполнен верно, программа вывела только фамилии студентов и даты их рождения. Поля с оценками и полом исчезли с экрана.

б) отличников по математике

Для данного запроса используем команду (рис. 15):

*SELECT \* FROM STUDENTS*

*WHERE math = 5;*

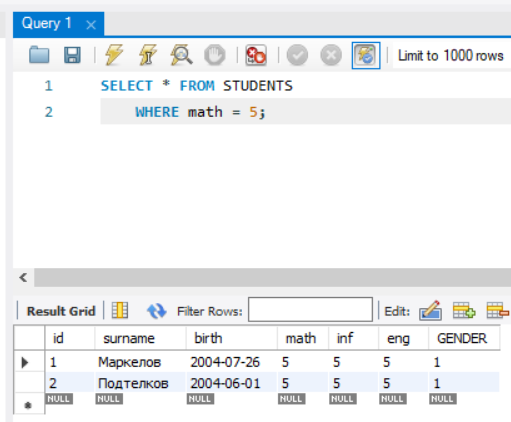


Рис. 15. Запрос отличников по математике

Запрос выполнен верно, программа вывела только записи студентов, у которых по математике стоит оценка 5. У остальных студентов за этот предмет оценка 4, поэтому они не были выведены.

в) отличников по всем предметам

Для данного запроса используем команду (рис. 16):

*SELECT \* FROM STUDENTS*

*WHERE math = 5 AND inf = 5 AND eng = 5;*

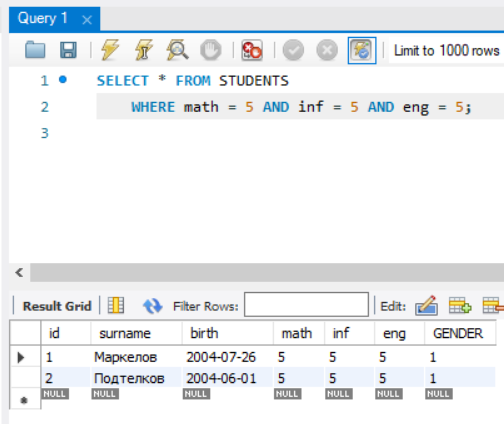


Рис. 16. Запрос отличников по всем предметам

Запрос выполнен верно, программа вывела только записи студентов, у которых по всем предметам стоит оценка 5. У остальных студентов имеется хотя бы одна оценка ниже 5, поэтому они не были выведены.

г) студентов, возраст которых 20 лет и старше

На данный момент всем студентам, указанным в таблице, есть 20 лет. Для наглядности запроса добавим запись еще одного студента, которому нет 20 лет, с помощью запроса (рис. 17):

*INSERT INTO STUDENTS (surname, birth, math, inf, eng) values*

*(‘Бурмистров’, ‘2005-12-23’, 4, 4, 4);*

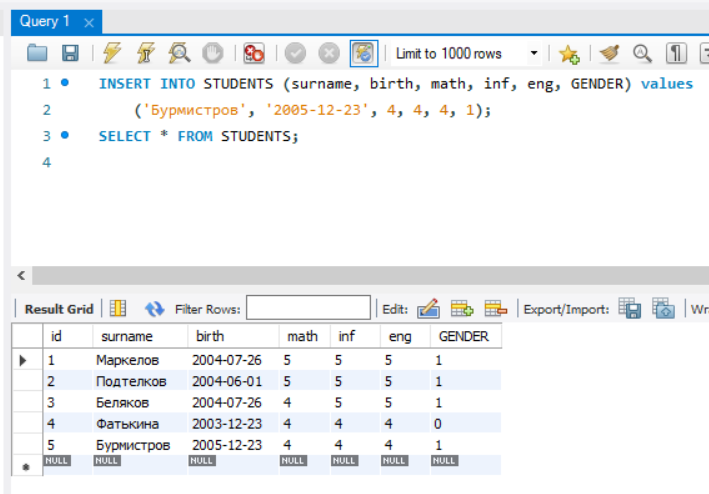


Рис. 17. Добавление записи

Для данного запроса используем команду (рис. 18):

*SELECT \* FROM STUDENTS*

*WHERE birth < ‘2005-04-25’;*

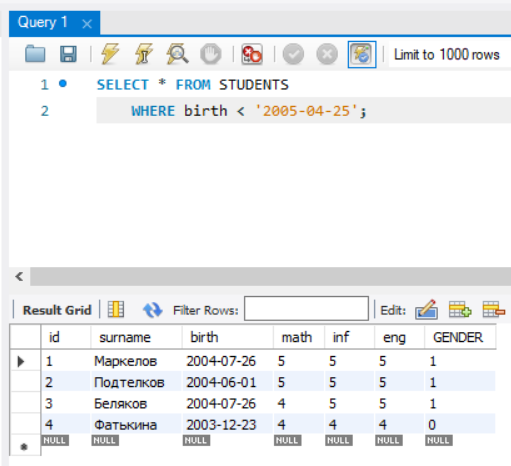


Рис. 18. Запрос студентов старше 20 лет

Запрос выполнен верно. По состоянию на 25.04.2025 20 лет исполнилось всем студентам, родившимся после 25.04.2005. Студент Бурмистров родился 23.12.2005, т. е. на момент выполнения ЛР ему было 19 лет, поэтому он не был выведен.

1. Постройте ERD

Для построения ER-диаграммы на панели вкладок выбираем «Database» («База данных»), в выпадающем меню выбираем пункт «Reverse Engineer» («Обратное проектирование») (рис. 19). Это позволит нам создать логическую модель БД.

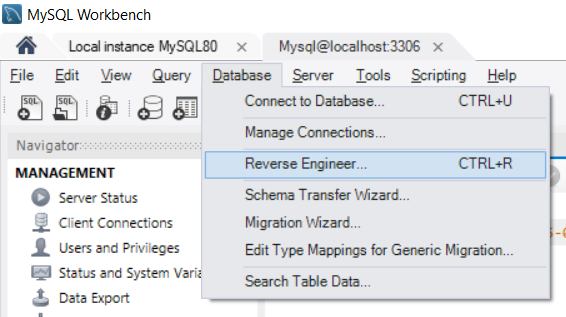


Рис. 19. Меню «Database»

Откроется окно «Reverse Engineer Database» («Обратное проектирование базы данных») (рис. 20-21). В нем будет предложено ввести параметры проектирования. Оставим их без изменения.

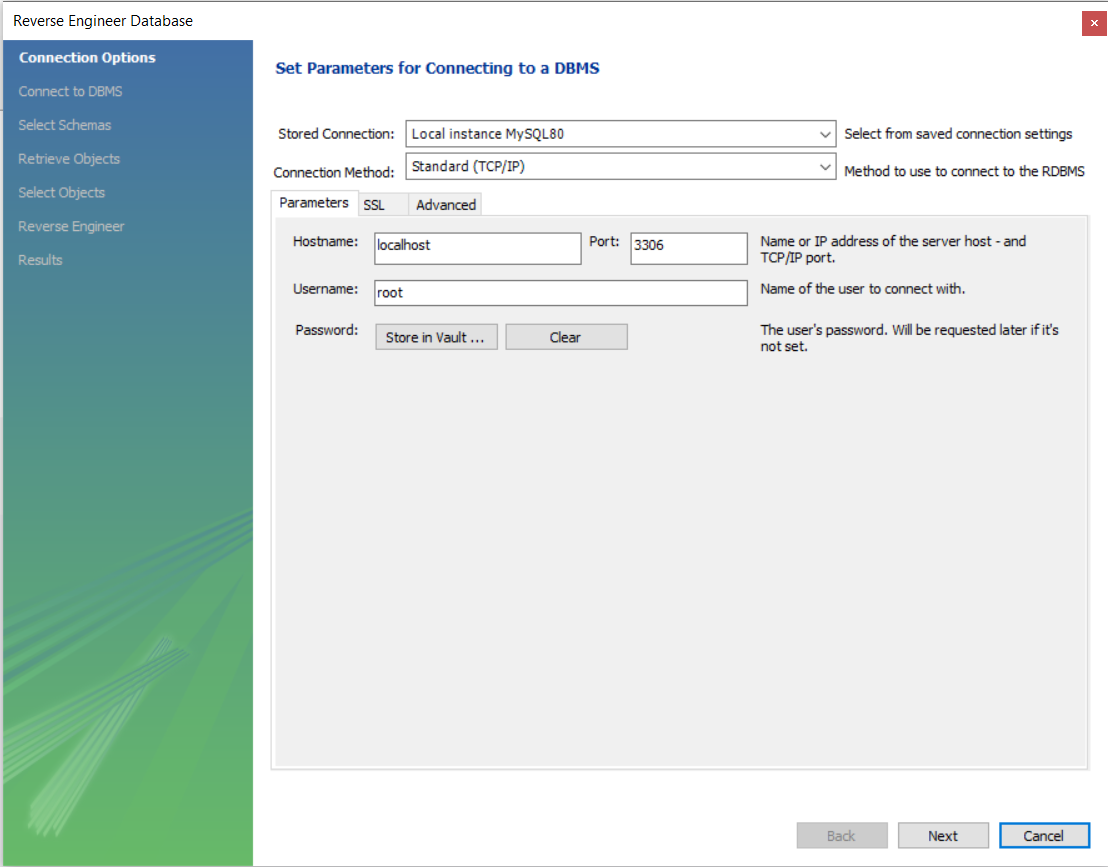


Рис. 20. Окно «Reverse Engineer Database»

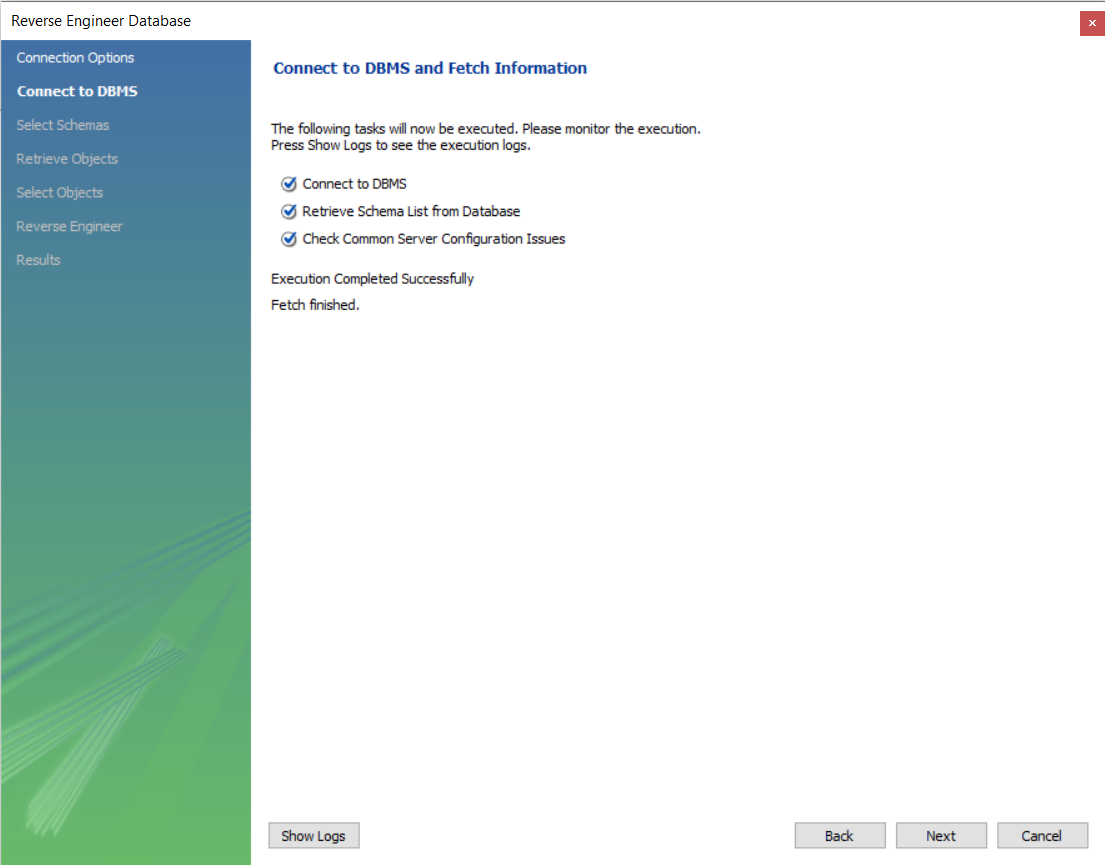


Рис. 21. Окно «Reverse Engineer Database»

Выбираем нашу БД PI33 (рис. 22).

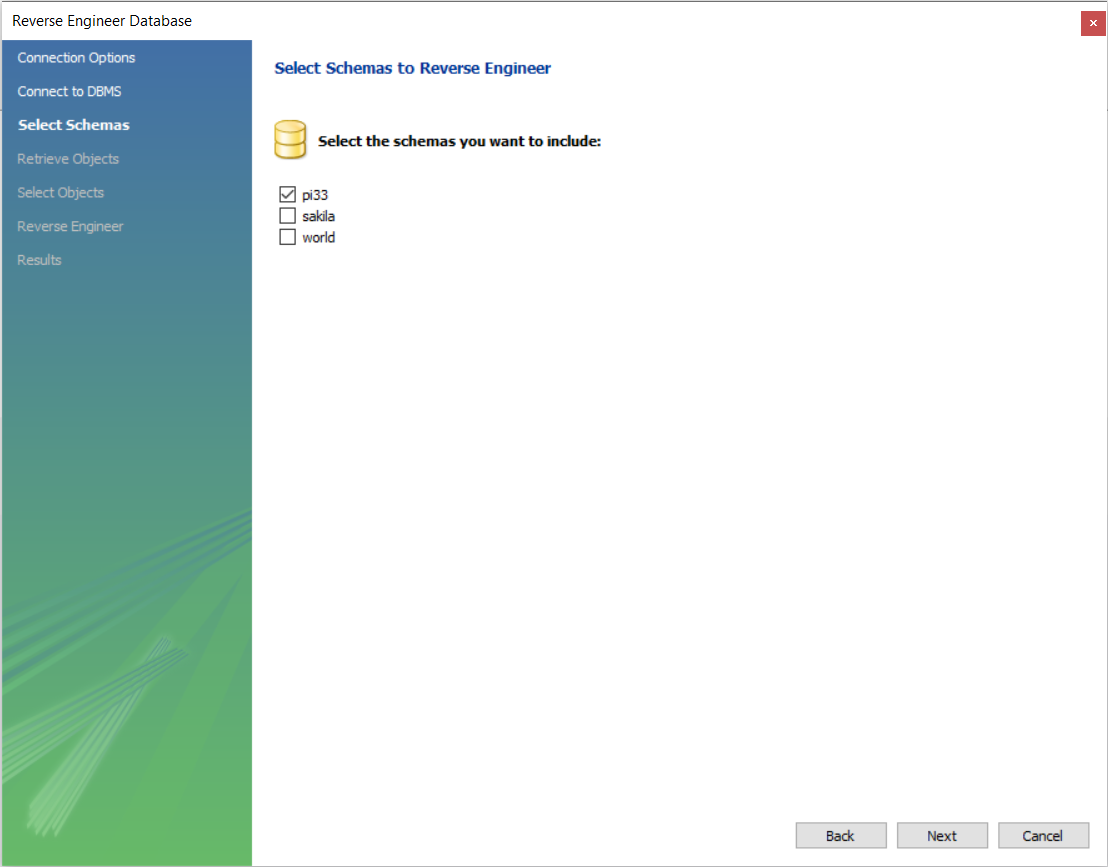


Рис. 22. Выбор БД

Далее нажимаем «Next» («Следующий») (рис. 23) и затем «Execute» (рис. 24).

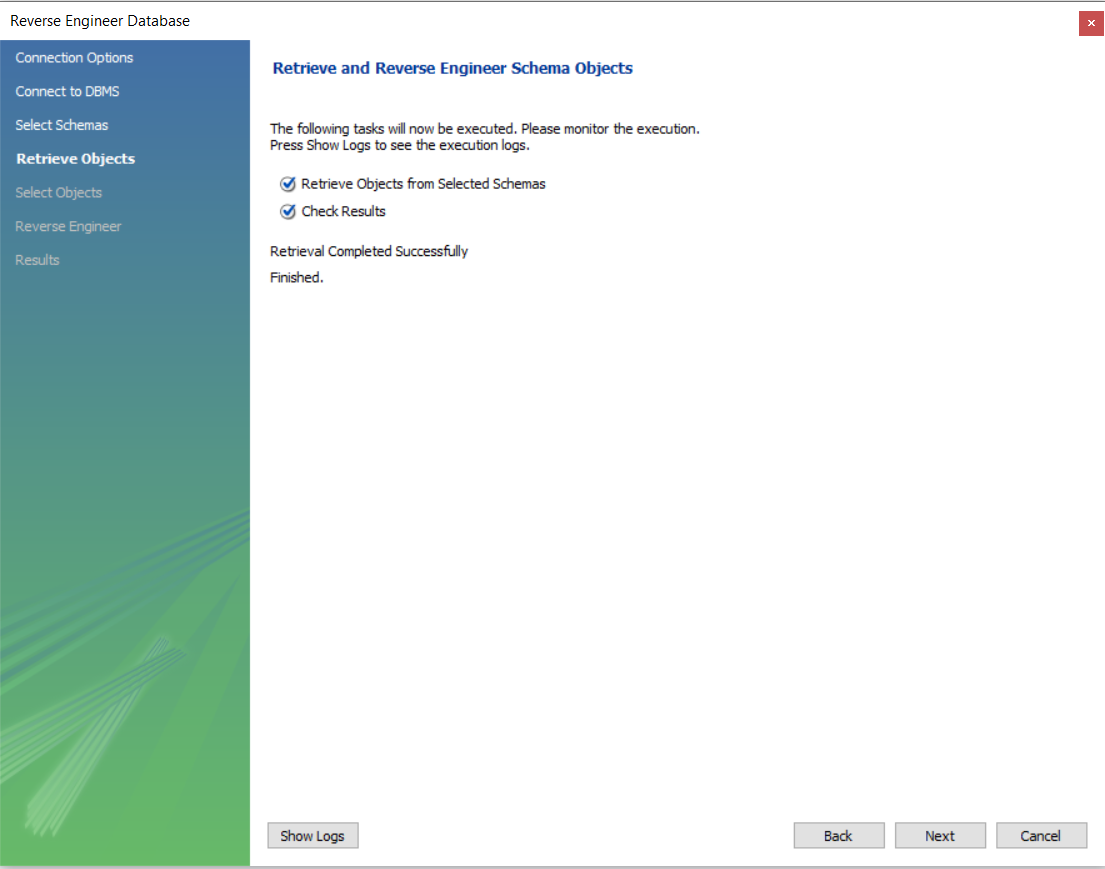


Рис. 23. Окно «Reverse Engineer Database»

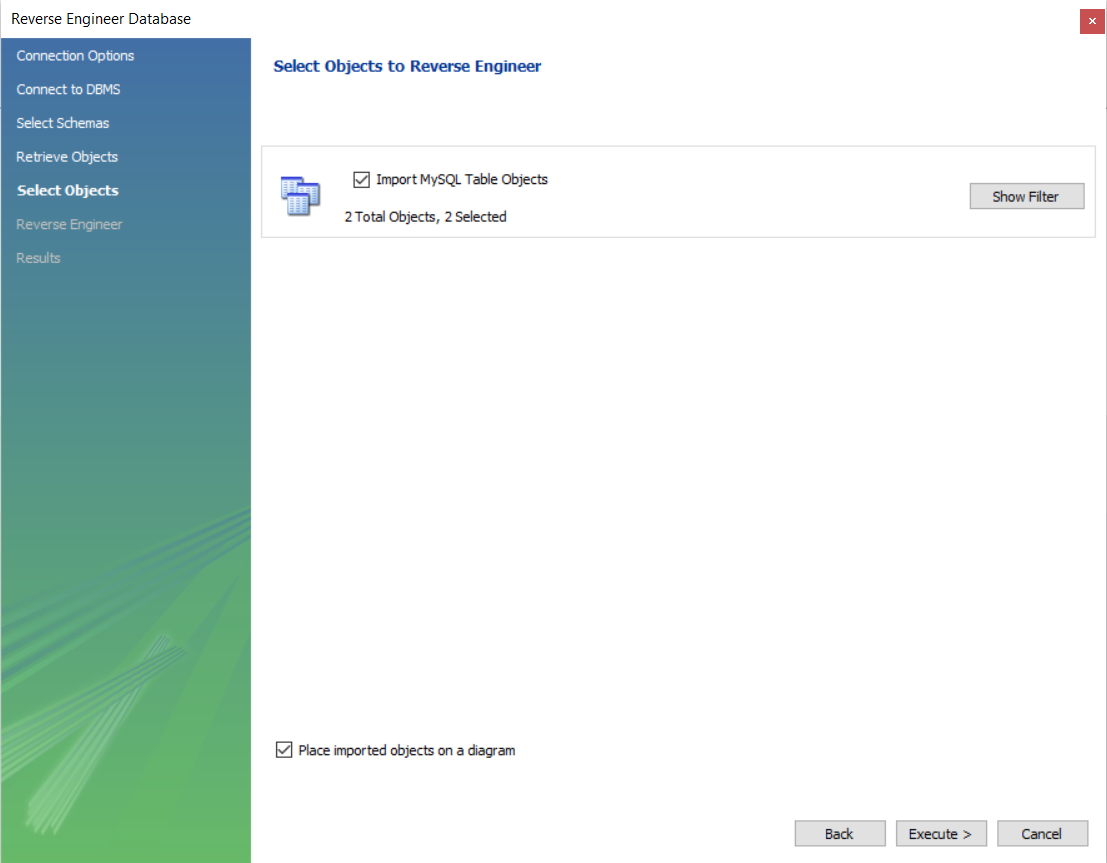


Рис. 24. Окно «Reverse Engineer Database»

Полученная логическая модель в виде ER-диаграммы представлена на рис. 25.

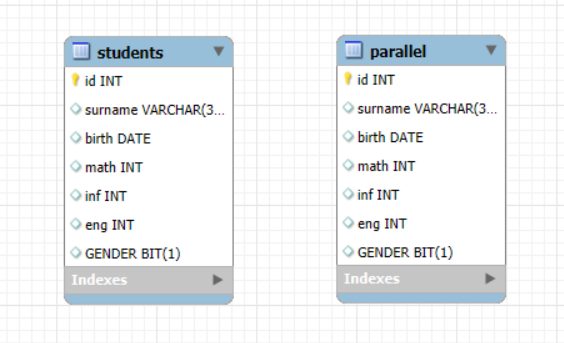


Рис. 25. ER-диаграмма логической модели

Свяжем данные сущности. Для этого выделим в отдельную сущность данные о студентах, как о физических лицах – фамилии и даты рождения.

Для создания новой сущности на панели инструментов слева нажимаем на кнопку «Place a New Table» («Установить новую таблицу») (рис. 26).



Рис. 26. Создание новой сущности

Дважды нажимаем по сущности. В нижней части программы откроется область настройки сущности (рис. 27). В строке «Table Name» («Имя таблицы») вводим название таблицы, в нашем случае «list of students» («Список студентов»). Ниже создаем атрибуты сущности:

* id\_students – ключевое поле;
* surname – фамилия;
* birth – дата рождения.

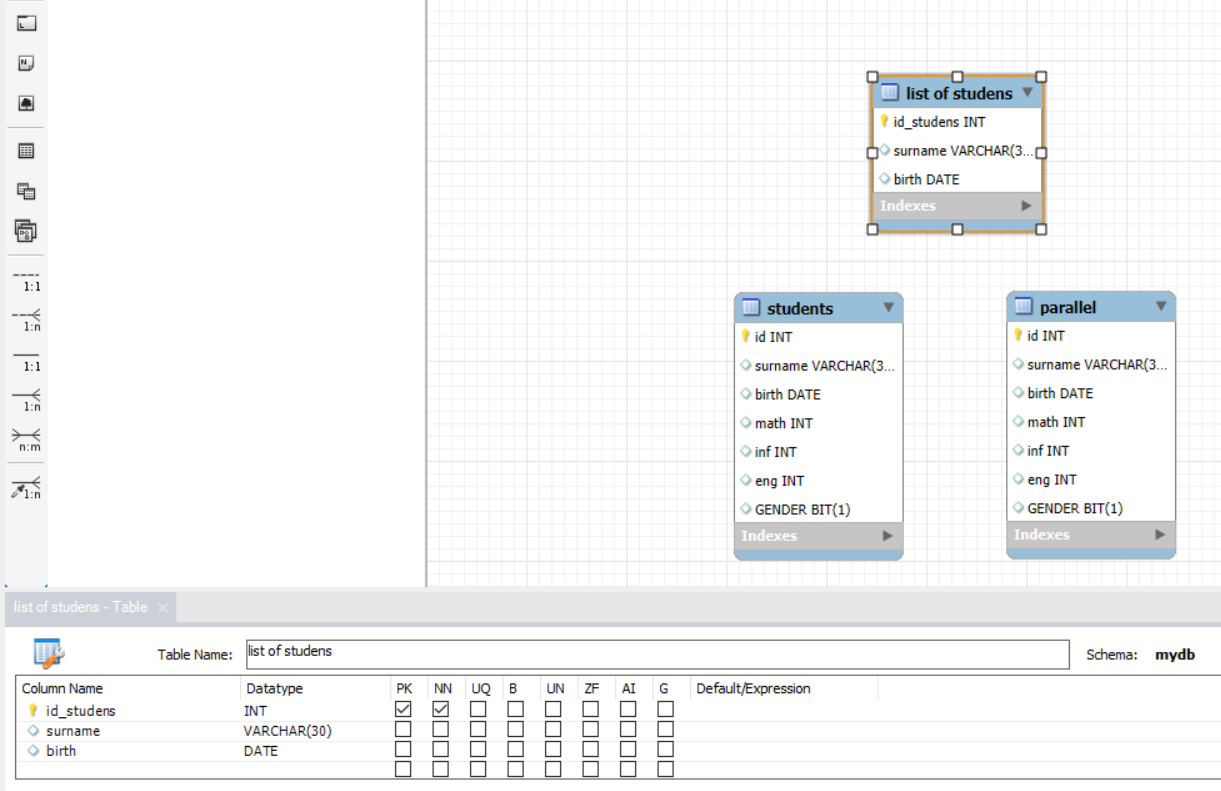


Рис. 27. Настройка сущности «list of students» и ее атрибутов

Аналогично отредактируем сущности «students» и «parallel». Удаляем поля «surname» и «birth» (рис. 28-29).

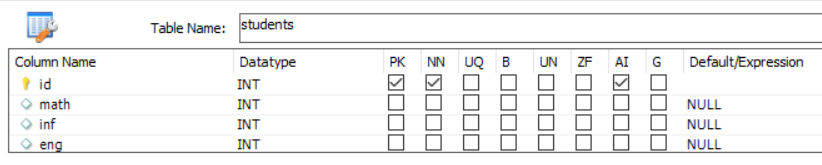


Рис. 28. Настройка сущности «students» и ее атрибутов

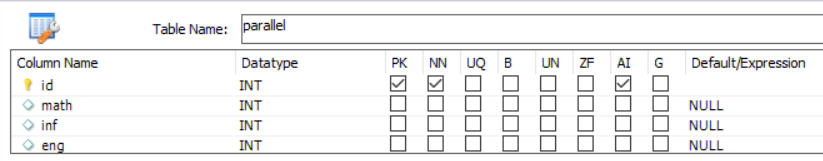


Рис. 29. Настройка сущности «parallel» и ее атрибутов

Для создания новой связей между сущностями на панели инструментов слева нажимаем на кнопку «Place a New 1:n Non-Identifying Relationship» («Установить новое неидентифицированное отношение 1:n») (рис. 30).

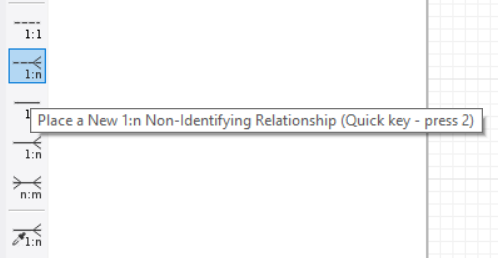


Рис. 30. Создание связи

После выбора типа связи нажимаем сначала на сущность «students», а затем на сущность «list of students». Аналогично, нажимаем сначала на сущность «parallel», а затем на сущность «list of students». Связи установлены (рис. 31).

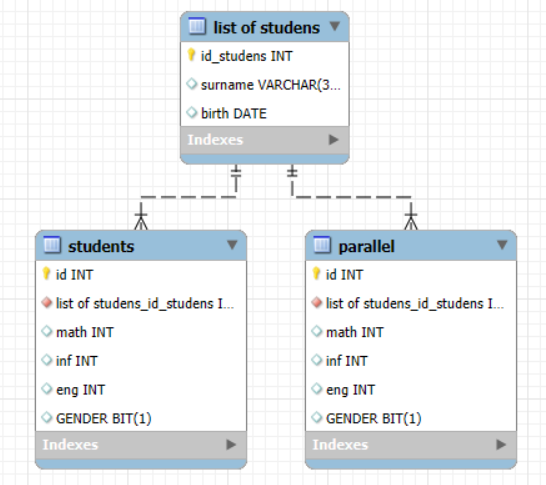


Рис. 31. ER-диаграмма логической модели