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  Базы данных |

наименование дисциплины в соответствии с учебным планом

ЛАБОРАТОРНАЯ РАБОТА №5

«Язык структурированных запросов SQL»

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

2024 год

Цель: изучить команды описания данных (DDL), команды манипулирования данными (DML) и команды запросов данных (DQL) языка SQL.

SQL (Structured Query Language) – это язык структурированных запросов, используемый для взаимодействия с базами данных.

SQL = DDL + DML + DQL, где

* DDL (Data Definition Language) – язык описания данных
* DML (Data Manipulation Language) – язык манипуляции данными
* DQL (Data Query Language) – язык запросов данных

1. Создайте таблицу базы данных с полями, содержание которых – фамилия, год рождения, оценки по математике, информатике, иностранному языку.

Для начала создадим файл базы данных формата .mdb. Формат .accdb не подойдет, т. к. он не поддерживается программой DTM SQL Editor.

Для подключения файла с БД в программе DTM SQL Editor верхней строке меню нажимаем на кнопку «Connect». Далее нажимаем на кнопку «Browse» и выбираем нужный файл (рис. 1).

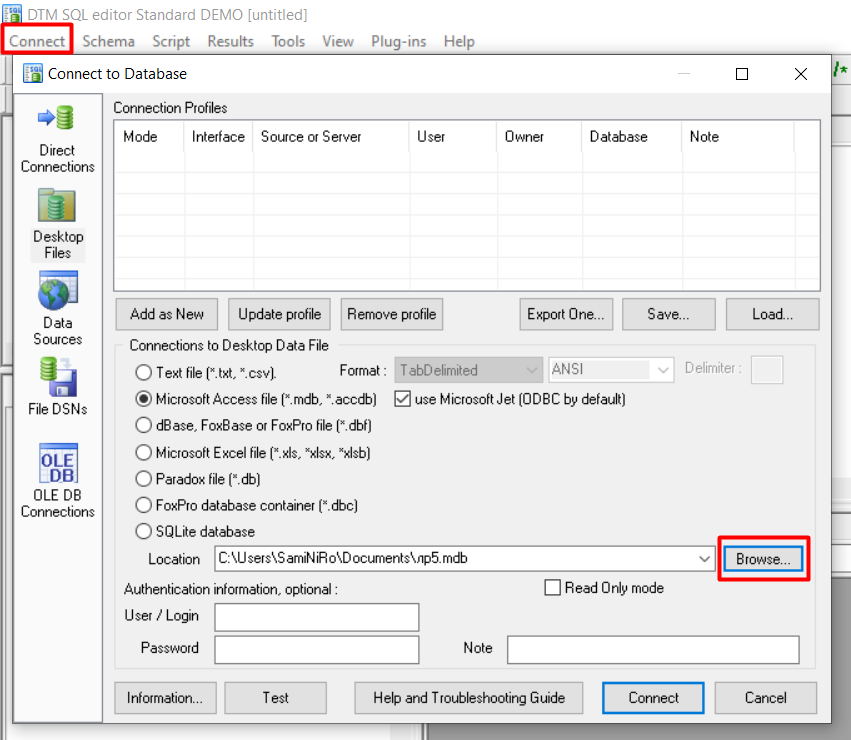


Рис. 1. Подключение файла БД

Теперь переходим к созданию таблицы БД. Для этого в языке SQL существует команда CREATE, относящаяся к категории DDL. Она имеет следующий вид:

CREATE TABLE имя\_табл (

Имя\_поля1 тд,

Имя\_поля2 тд,

…

);

Вводим в области редактора запросов следующую команду (рис. 2):

CREATE TABLE ПИ\_33 (

Фамилия CHAR(25),

ДР DATETIME,

Мат INT,

ИнЯз INT,

Инф INT

);

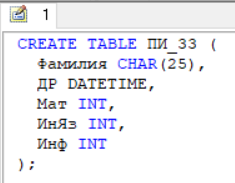


Рис. 2. Создание структуры БД с помощью команды CREATE

Выведем получившуюся таблицу с помощью команды SELECT. Это оператор в SQL, относящийся к DQL и предназначенный для извлечения данных из базы данных. Он позволяет выбрать конкретные столбцы или все данные из таблицы, с возможностью применения фильтрации, сортировки, группировки и других операций.

Команда SELECT для вывода всей таблицы будет иметь вид

SELECT \* FROM имя\_табл;

где \* обозначает, что нужно вернуть все столбцы, содержащиеся в таблице. В нашем случае команда будет иметь вид:

SELECT \* FROM ПИ\_33;

Результатом команды будет следующая таблица, на данный момент без записей (рис. 3).



Рис. 3. Таблица БД после создания

2. Внесите в таблицу 5 записей.

Для внесения записей в SQL используется команда INSERT, относящаяся к категории DML. Она имеет вид:

INSERT INTO имя\_табл (поле1, поле2, ...)

VALUES

(знач1\_1, знач2\_1, ...),

(знач1\_2, знач2\_2, ...),

…

Внесем в таблицу 5 записей о студентах группы (рис. 4):

INSERT INTO ПИ\_33 (Фамилия, ДР, Мат, ИнЯз, Инф)

VALUES

('Маркелов', '26.07.2004', 5, 5, 5),

('Подтелков', '01.06.2004', 5, 5, 5),

('Бурмистров', '02.06.2005', 4, 5, 5),

('Беляков', '04.04.2004', 4, 5, 5),

('Фатькина', '23.12.2003', 5, 5, 4)

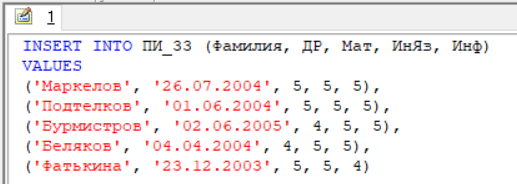


Рис. 4. Внесение записей в БД с помощью команды INSERT

Выведем получившуюся таблицу с помощью команды SELECT \* FROM ПИ\_33 (рис. 5):

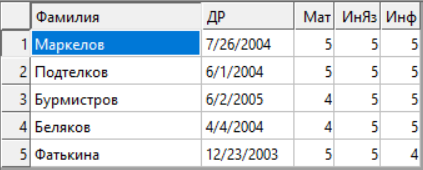


Рис. 5. Таблица после внесения записей

3. Внесите изменения в структуру таблицы – создайте поле, которое будет содержать пол студента.

Для добавления нового поля воспользуемся командой ALTER. Команда ALTER в SQL относится к DML и используется для изменения структуры существующей таблицы или других объектов базы данных после их создания. С помощью этой команды можно добавлять, удалять или изменять поля в таблицах, а также менять другие характеристики объектов базы данных.

Команда ALTER для добавления нового поля будет иметь вид:

ALTER TABLE имя\_табл

ADD имя\_поля тд;

Для добавления поля «Пол» в нашей БД о студентах введем команду (рис. 6):

ALTER TABLE ПИ\_33

ADD Пол char(2)

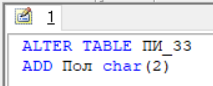


Рис. 6. Добавление поля «Пол» с помощью команды ALTER

Выведем таблицу с новым полем с помощью команды SELECT \* FROM ПИ\_33 (рис. 7). На данный момент у всех записей поле «Пол» имеет значение <NULL>.

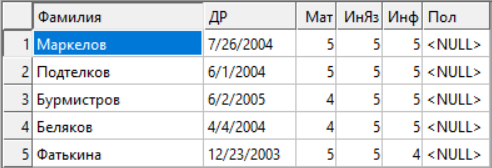


Рис. 7. Таблица с добавленным полем «Пол»

4. Дополните данными таблицу, указав пол каждого студента.

Для изменения существующих записей в таблице используется команда UPDATE. Она относится к категории DML и имеет следующий вид:

UPDATE имя\_табл

SET имя\_поля1 = знач1, имя\_поля2 = знач2, ...

WHERE условие;

Дополним записи нашей таблицы информацией о поле студентов (рис. 8):

UPDATE ПИ\_33

SET Пол = 'М'

WHERE Фамилия IN ('Маркелов', 'Подтелков', 'Бурмистров', 'Беляков')

UPDATE ПИ\_33

SET Пол = 'Ж'

WHERE Фамилия = 'Фатькина'

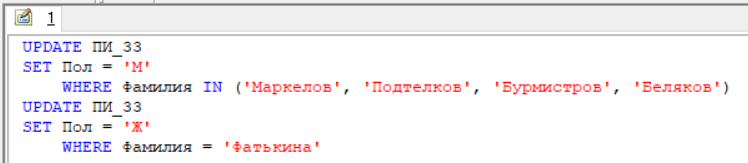


Рис. 8. Дополнение записей с помощью команды UPDATE

Выведем таблицу с заполненным полем «Пол» с помощью команды SELECT \* FROM ПИ\_33 (рис. 9).

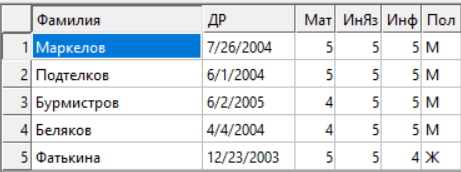


Рис. 9. Таблица БД с заполненным полем «Пол»

5. Создайте таблицу с теми же полями – она будет содержать сведения о студентах параллельной группы. Решите это задание способом, отличным от того, каким было выполнено задание 1.

Для решения данного задания скопируем структуру таблицы ПИ\_33 в новую таблицу ПИ\_31 с помощью команды SELECT. Команда SELECT при копировании структуры таблицы будет иметь следующий синтаксис:

SELECT \* INTO нов\_табл

FROM стар\_табл

WHERE FALSE*;*

Скопируем структуру таблицы (рис. 10):

SELECT \* INTO ПИ\_31

FROM ПИ\_33

WHERE FALSE;

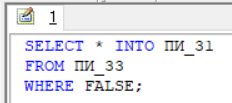


Рис. 10. Копирование структуры таблицы с помощью команды SELECT

Выведем новую таблицу, пока без записей, с помощью команды SELECT \* FROM ПИ\_31 (рис. 11).



Рис. 11. Новая таблица ПИ\_31 со структурой, скопированной с таблицы ПИ\_33

6. Внесите во вторую таблицу 5 записей.

Аналогично заданию 2, вносим записи с помощью команды INSERT (рис. 12).

INSERT INTO ПИ\_31 (Фамилия, ДР, Мат, ИнЯз, Инф, Пол)

VALUES

('Иванов', '01.01.2001', 2, 2, 2, 'М'),

('Петрова', '02.02.2002', 3, 3, 3, 'Ж'),

('Смирнов', '03.03.2003', 4, 4, 4, 'М'),

('Сидорова', '04.04.2004', 2, 3, 2, 'Ж'),

('Васильев', '05.05.2005', 4, 4, 5, 'М')

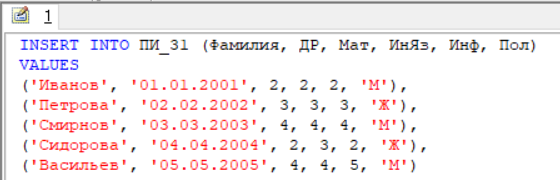


Рис. 12. Добавление записей в таблицу ПИ\_31

Выведем получившуюся таблицу ПИ\_31 с помощью команды SELECT \* FROM ПИ\_31 (рис. 13).

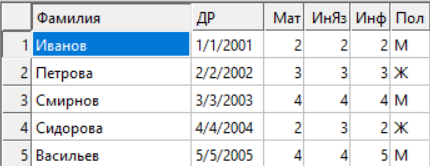


Рис. 13. Таблица ПИ\_31 с внесенными записями

7. Осуществите сортировку записей первой таблицы в алфавитном порядке поля фамилий.

Осуществлять сортировку будем также с помощью команды SELECT. Синтаксис команды при сортировке имеет вид:

SELECT \* FROM имя\_табл

ORDER BY имя\_поля1 ASC/DESC, имя\_поля2 ASC/DESC, …

где ASC – сортировка по возрастанию, DESC – по убыванию.

Осуществим сортировку по возрастанию (т. е. по алфавиту) по полю «Фамлиия» (рис. 14):

SELECT \* FROM ПИ\_33

ORDER BY Фамилия ASC

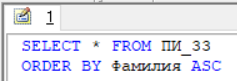


Рис. 14. Сортировка по возрастанию с помощью команды SELECT

В результате получилась следующая таблица – все фамилии идут по алфавиту, от Б до Ф (рис. 15).

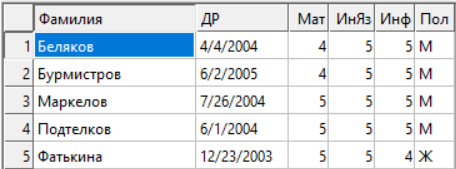


Рис. 15. Отсортированная по алфавиту таблица

8. Сформируйте запросы на выборку:

а) студентов с указанием фамилии и даты рождения;

Для этого используем уже привычную команду SELECT, которую неоднократно использовали для выведения всей таблицы. Но теперь вместо «\*» пишем имена полей, которые необходимо вывести. Синтаксис команды SELECT будет иметь вид:

SELECT поле1, поле2, … FROM имя\_табл

В нашем случае команда будет иметь вид (рис. 16):

SELECT Фамилия, ДР FROM ПИ\_33

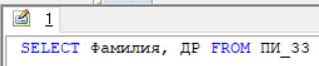


Рис. 16. Запрос на вывод фамилий и дат рождений с помощью команды SELECT

В результате получили таблицу только с указанием фамилий и дат рождения (рис. 17). Поля с оценками с экрана исчезли.

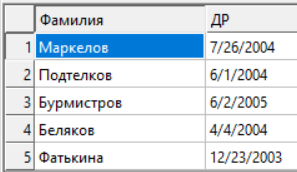


Рис. 17. Результат запроса по фамилиям и датам рождений

б) студентов-отличников по математике;

Для данного запроса дополним команду из пункта а условием:

SELECT поле1, поле2, … FROM имя\_табл

WHERE полеN = знач

В нашем случае команда будет иметь вид (рис. 18):

SELECT Фамилия, Мат FROM ПИ\_33

WHERE Мат = 5

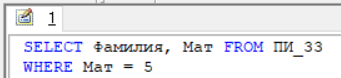


Рис. 18. Запрос отличников по математике с помощью команды SELECT

В результате в таблице остались только студенты, имеющие 5 по математике (рис. 19). У Белякова и Бурмистрова в поле «Мат» стоит оценка 4, поэтому они не были выведены.

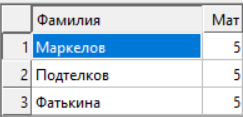


Рис. 19. Результат запроса отличников по математике

в) студентов-отличников по всем предметам;

Дополним запрос из пункта б несколькими условиями:

SELECT поле1, поле2, … FROM имя\_табл

WHERE поле1 = знач1 AND поле2 = знач2 AND …

В нашем случае запрос будет иметь вид (рис. 20):

SELECT Фамилия, Мат, ИнЯз, Инф FROM ПИ\_33

WHERE Мат = 5 AND ИнЯз = 5 AND Инф = 5

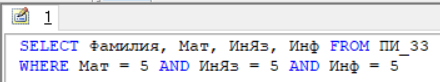


Рис. 20. Запрос отличников по всем предметам с помощью команды SELECT

В результате запроса в таблице остались только Маркелов и Подтелков – у этих студентов стоят оценки 5 во всех 3 полях (рис. 21). Остальные студенты имеют оценку ниже 5 в хотя бы одном из полей, поэтому на экран они выведены не были.

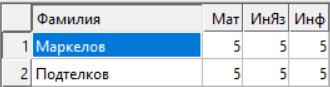


Рис. 21. Результат запроса отличников по всем предметам

г) студентов, чей возраст старше 20 лет.

Аналогично предыдущим пунктам, используем команду SELECT, но немного изменим условие – вместо «=» используем оператор «<» (рис. 22):

SELECT Фамилия, ДР

FROM ПИ\_33

WHERE ДР < DateAdd("yyyy", -20, Date());

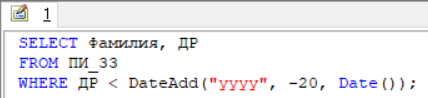


Рис. 22. Запрос студентов старше 20 лет с помощью команды SELECT

В результате запроса были выведены все студенты, кроме Бурмистрова (рис. 23). Этот студент родился 02.06.2005, т. е. по состоянию на январь 2025 года ему 19 лет. Все остальные студенты родились в 2003-2004 годах, им сейчас 20-21 лет, поэтому они были выведены.

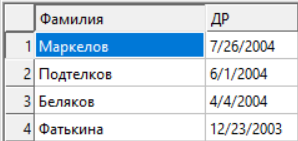


Рис. 23. Результат запроса студентов старше 20 лет

В условии была использована функция DateAdd.

DateAdd – это функция в Access, которая используется для добавления или вычитания определённого интервала времени (года, месяца, дня и т.д.) из заданной даты.

Функция имеет следующий синтаксис:

DateAdd (интервал, число, дата)

* интервал – указывает единицу времени, которую нужно добавить или вычесть. Это строка, заключённая в кавычки. В качестве интервала можно указать "yyyy" (год), "m" (месяц), "d" (день), "h" (час), "n" (минута), "s" (секунда).
* число – количество единиц времени, которое нужно добавить (положительное значение) или вычесть (отрицательное значение).
* дата – базовая дата, с которой нужно произвести операцию. Можно либо указать точную дату (например, #2025-01-10#), либо использовать встроенную функцию Date(), которая возвращает текущую системную дату.

В нашем случае интервал "yyyy" – год, число – -20, так как нам нужно вычесть 20 лет, а дата текущая. Если дата дня рождения окажется меньше результата выражения «текущая дата – 20 лет», значит условие верное, и такой студент будет выведен на экран.

9. Сформируйте параметрический запрос

В DTM SQL Editor не существует возможности создания параметрических запросов.

Убедимся в этом, попробовав создать параметрический запрос аналогично тому, как это делалось в Access – с использованием квадратных скобок «[ ]» (рис. 24):

SELECT Фамилия, ДР

FROM ПИ\_33

WHERE Фамилия = [];

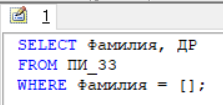


Рис. 24. Попытка создать параметрический запрос

В результате получаем ошибку – «Отсутствует значение для одного или нескольких требуемых параметров. 0x80040E10» (рис. 25).

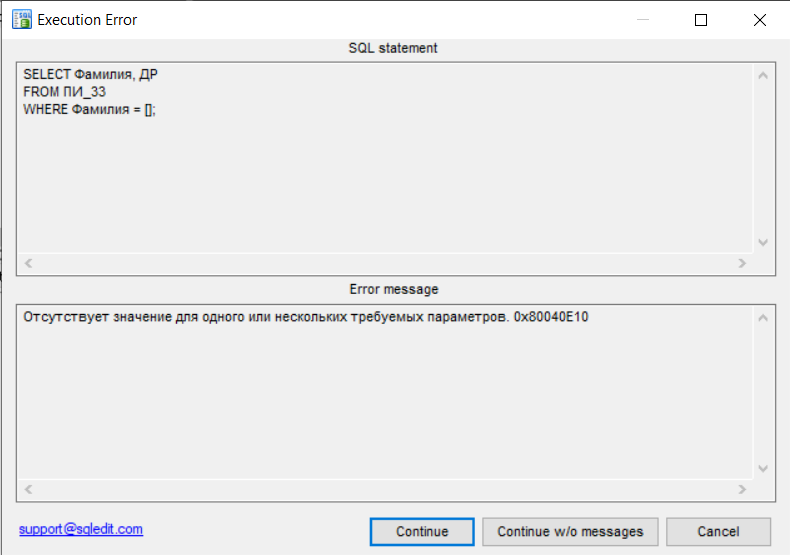


Рис. 25. Ошибка при попытке создания параметрического запроса

Выполнить данное задание мы можем средствами Access – также, как это делалось в лабораторной работе №1. Для этого откроем файл с нашей БД в Access (рис. 26).

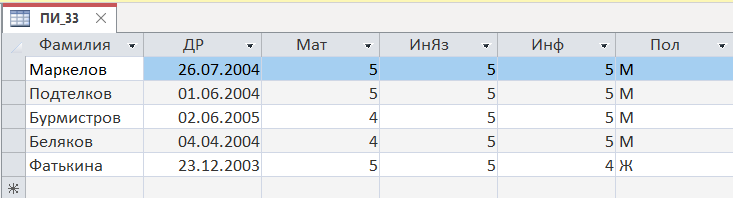


Рис. 26. БД, созданная в рамках данной работы, в Access

В разделе «Создание» открываем «Конструктор запросов» (рис. 27).

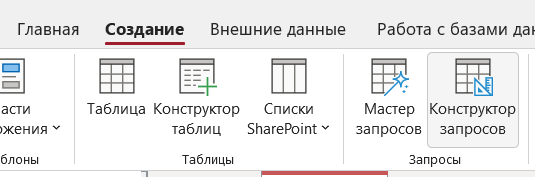


Рис. 27. Конструктор запросов

Выбираем таблицу ПИ33 и поля «Фамилия» и «ДР». В условии отбора по полю «Фамилия» вводим «[Введите фамилию]» (рис. 28).

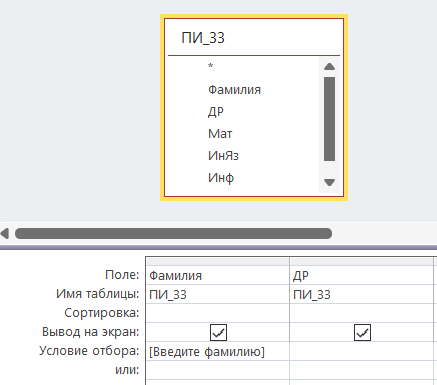


Рис. 28. Выбор параметров запроса

Выполняем запрос, нажав на кнопку «Выполнить» (рис. 29).

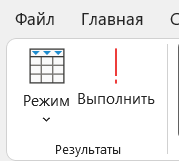


Рис. 29. Кнопка «Выполнить»

После выполнения запроса появится диалоговое окно, которое запросит ввести фамилию (рис. 30).

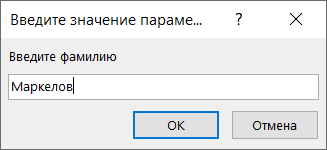


Рис. 30. Окно введения параметра

Результат запроса по фамилии «Маркелов» представлен на рис. 31.

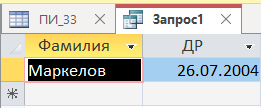


Рис. 31. Результат параметрического запроса