МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Череповецкий государственный университет»

**Лабораторная работа № 1**

**«Программа управления БД DTM SQL editor»**

**Выполнил:**

студент гр. 1ИВТпб-01-31оп

Климов А.Г.  
**Проверил:**  преподаватель

Селяничев О.Л.  
Отметка о зачете:

Череповец

2017 год

**Задания**

Часть I.

1. Создайте пустую БД Access.

2. Введите для нее alias через ODBC.

3. Ознакомьтесь с интерфейсом программы, изучите назначение элементов изображения.

Часть II.

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

2. Внесите в таблицу 5 записей.

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

4. Дополните данными таблицу, указав пол каждого студента.

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.

6. Внесите во вторую таблицу 5 записей.

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

8. Сформируйте запросы на выборку:

а) студентов с указанием фамилии и даты рождения;

б) студентов-отличников по математике;

в) студентов-отличников по всем предметам;

г) студентов, чей возраст старше 20 лет.

9. Сформируйте параметрический запрос:

а) студентов с фамилией «Иванов»;

б) студентов Ивановых, имеющих «5» по математике.

**Выполнение заданий**

Часть I.

1. Создаём пустую БД в Access (рис. 1).

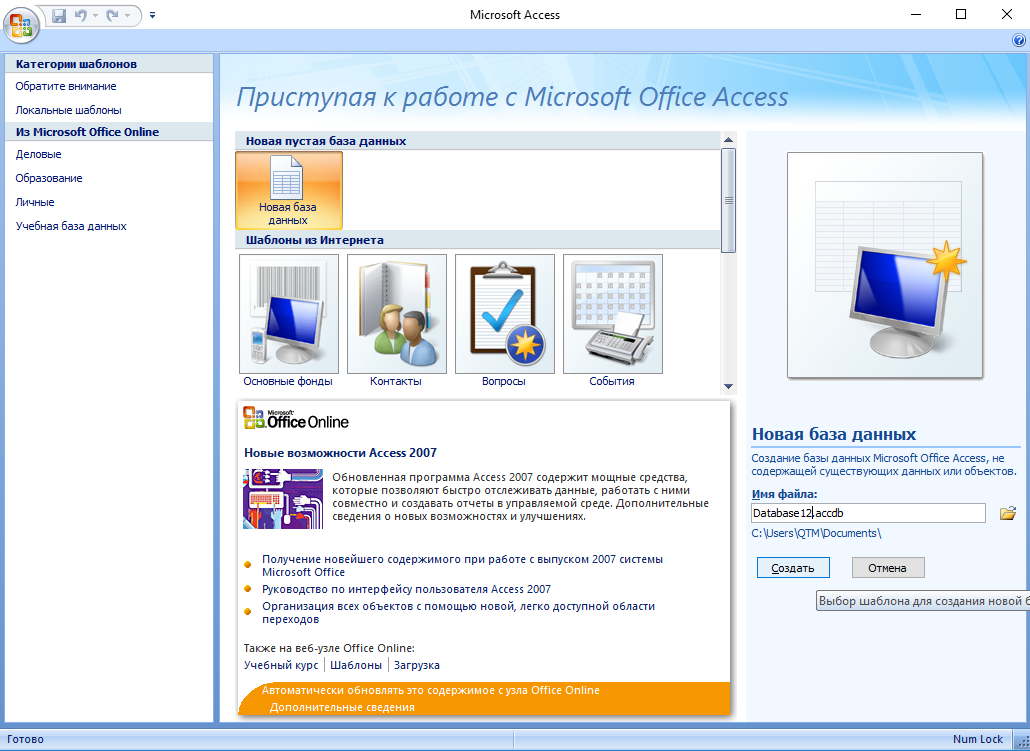


Рис. 1. Создание БД

DTM SQL editor – универсальный редактор SQL сценариев и набор инструментов для работы с базами данных. Поддерживает интерфейсы ODBC, OLE DB, IDAPI и OCI, что обеспечивает доступ к большинству современных баз. Работает с несколькими соединениями с разными базами одновременно. Содержит: простой построитель SQL сценариев, инструменты для просмотра и редактирования схемы базы (со свойствами объектов), поддержку Visual Source Safe. Обеспечивает использование макросов, библиотек сценариев, снимков схемы, историй выполнения и т.д. Гибкий графический интерфейс с многоуровневым Undo/Redo, выделением синтаксических конструкций цветами.

2. Главное окно программы представлено на рис. 2. Оно включает структуру БД, поле для ввода SQL сценариев, параметры выбранного элемента БД и представления. В верхней части находится панель инструментов.

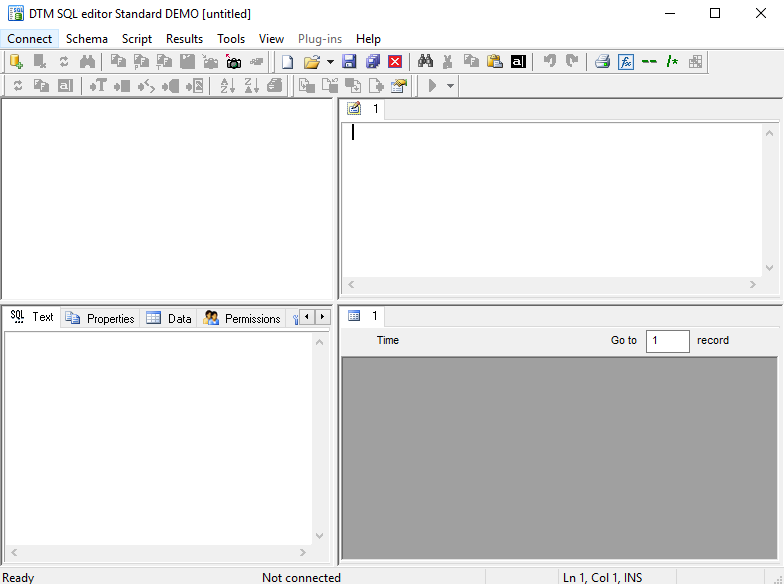


Рис. 2. Главное окно программы

Чтобы начать работу с БД, необходимо к ней подключиться. Для этого на панели инструментов необходимо перейти в раздел “Connect” и из выпадающего списка выбрать “Connect to database…” (рис. 3).

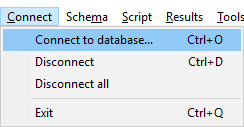


Рис. 3. Подключение к БД

В появившемся окне выбираем тип подключаемой БД (рис. 4). В нашем случае это “Microsoft Access file (\*.mdb, \*.accdb)” и при помощи кнопки “Browse” открываем файл БД.

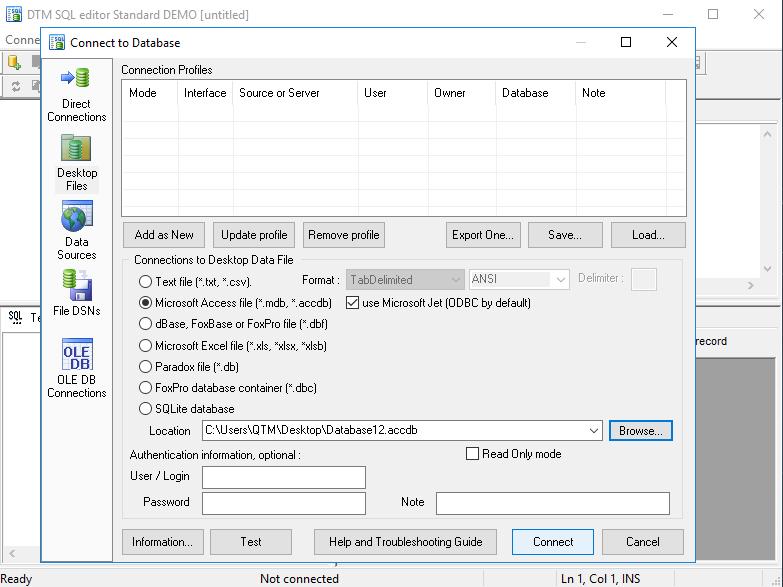


Рис. 4. Подключение к БД

Результат подключения к БД представлен на рис. 5.

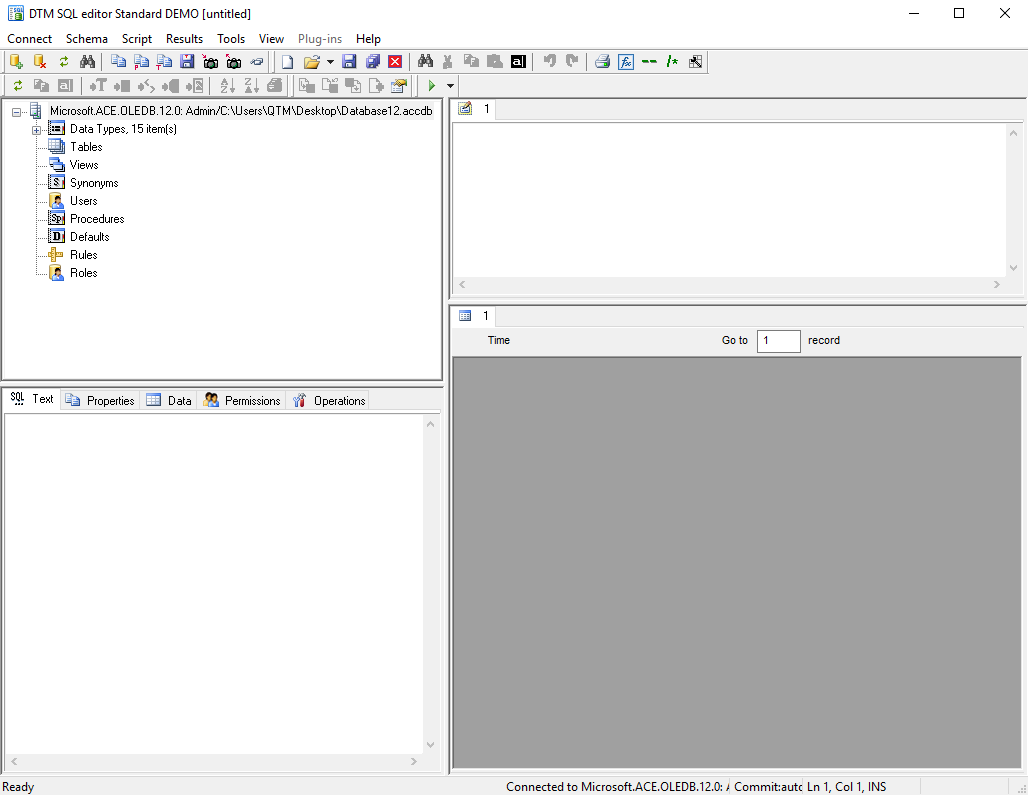


Рис. 5. Подключение к БД

Часть II.

1. Создаём таблицу базы данных с полями, содержание которых - фамилия, год рождения, оценки по математике, информатике, иностранному языку. Команда выглядит следующим образом:

**CREATE TABLE person (surname varchar(15), year1 date, math int, languag int, inf int);**

Ввод команды происходит в правой верхней части программы. Пример введённой команды рис. 6.

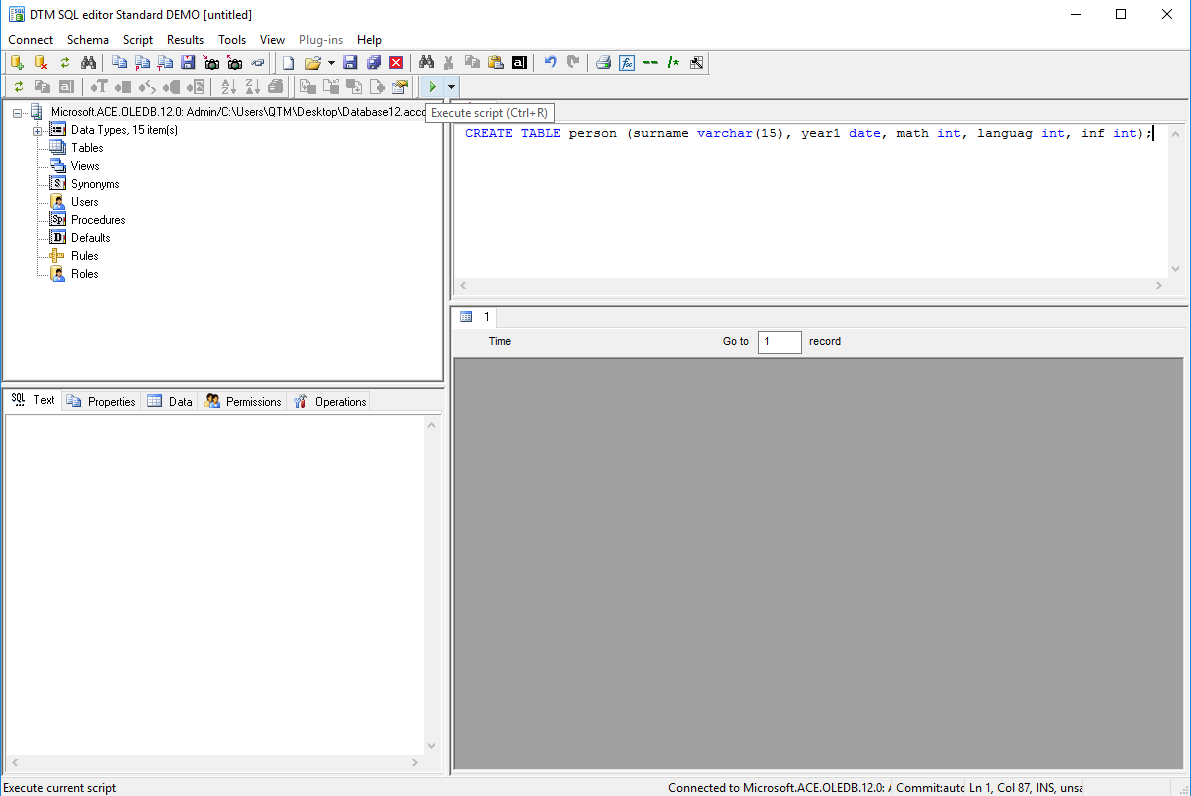


Рис. 6. Ввод команды

Выполнение команды происходит при нажатии на кнопку “Execute script” на панели инструментов или комбинацией клавиш “Ctrl+R”. Результат выполнения команды представлен на рис. 7 и рис. 8.

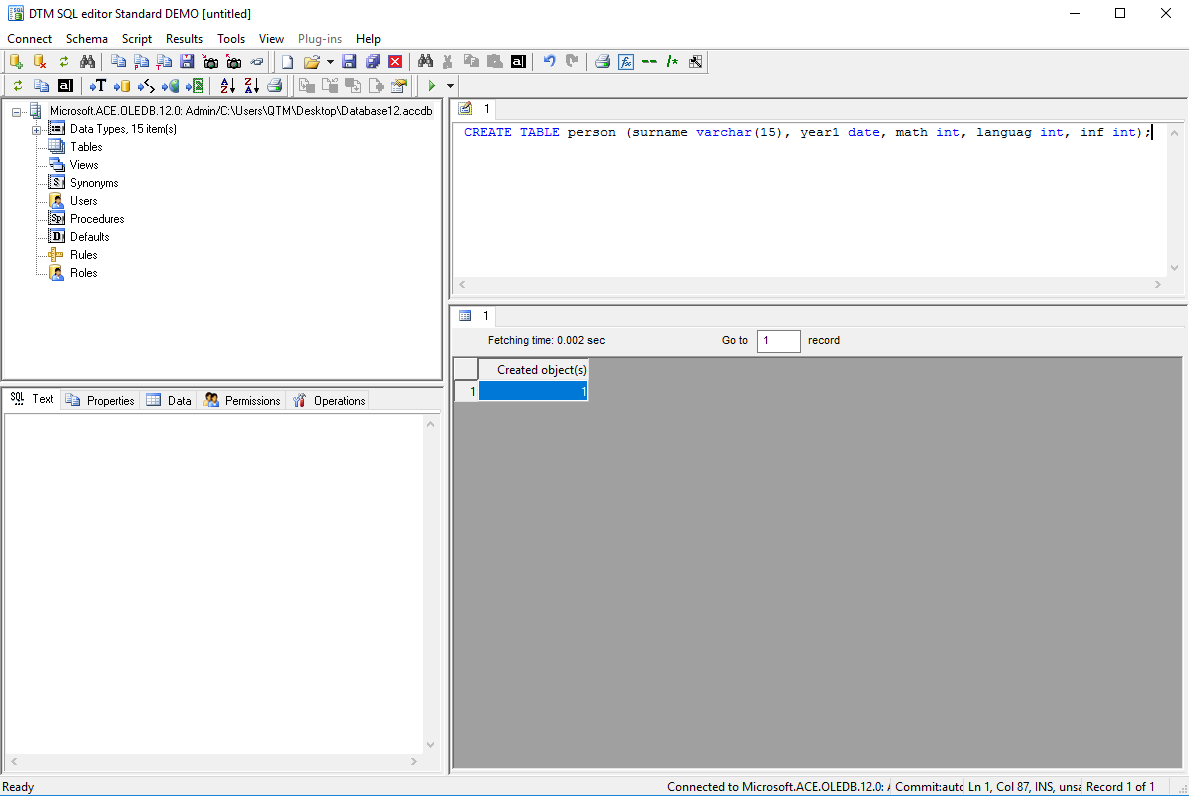


Рис. 7. Результат выполнения команды

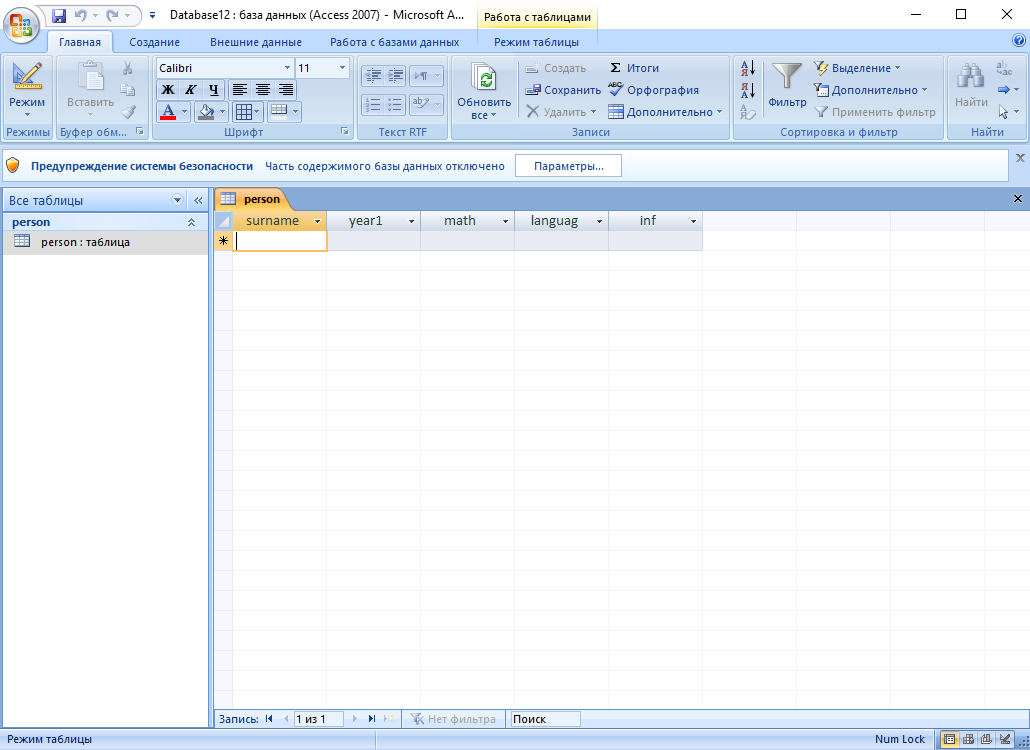


Рис. 8. Результат выполнения команды

2. Вносим в таблицу 5 записей при помощи последовательности команд:

INSERT INTO person(surname,year1,math,languag,inf) VALUES ('Ivanov','1990-12-17','5','4','5');

INSERT INTO person(surname,year1,math,languag,inf) VALUES (' Staskevich','1997-12-17','3','3','3');

INSERT INTO person(surname,year1,math,languag,inf) VALUES (' Pepeonkov','1998-01-01','3','3','3');

INSERT INTO person(surname,year1,math,languag,inf) VALUES (' Osipov','1997-12-12','4','4','4');

INSERT INTO person(surname,year1,math,languag,inf) VALUES (' Rodina','1997-07-04','3','3','4');

Результат выполнения представлен на рис. 9 и рис. 10.

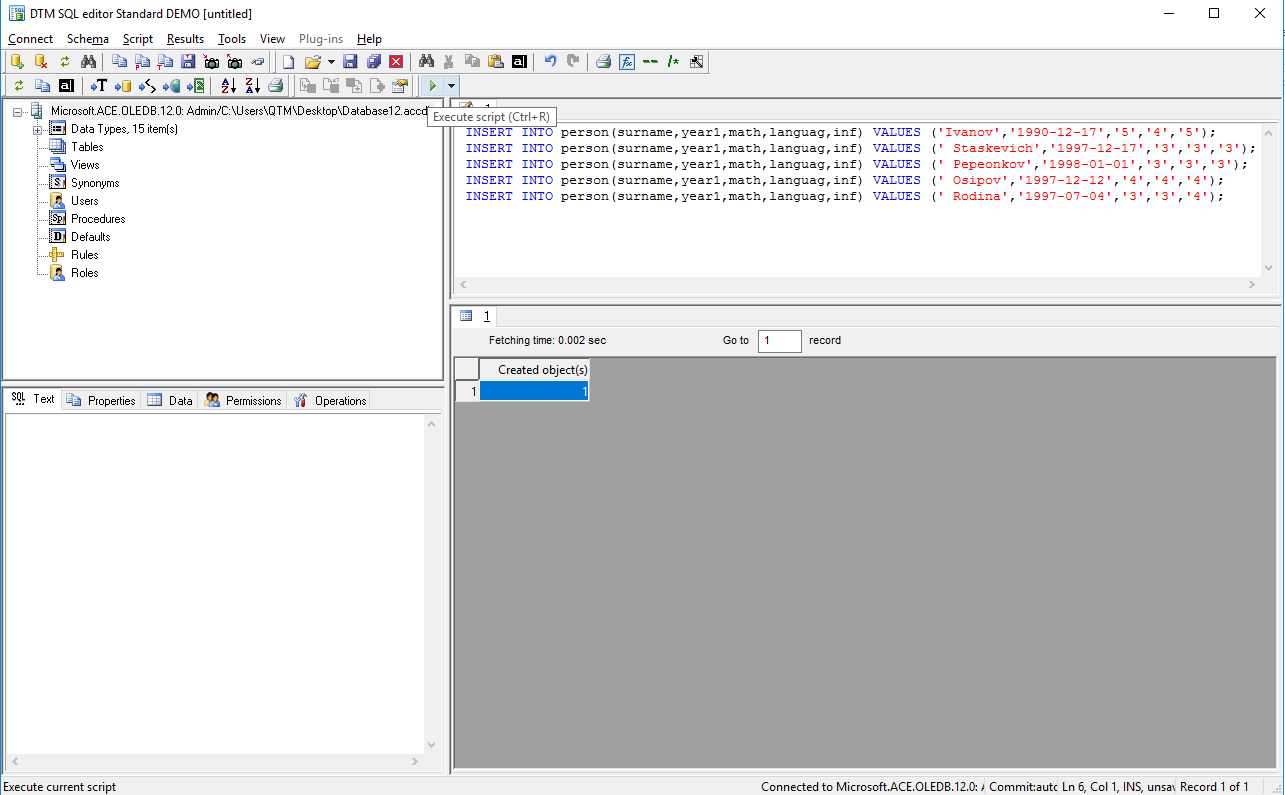


Рис. 9. Результат выполнения последовательности команд

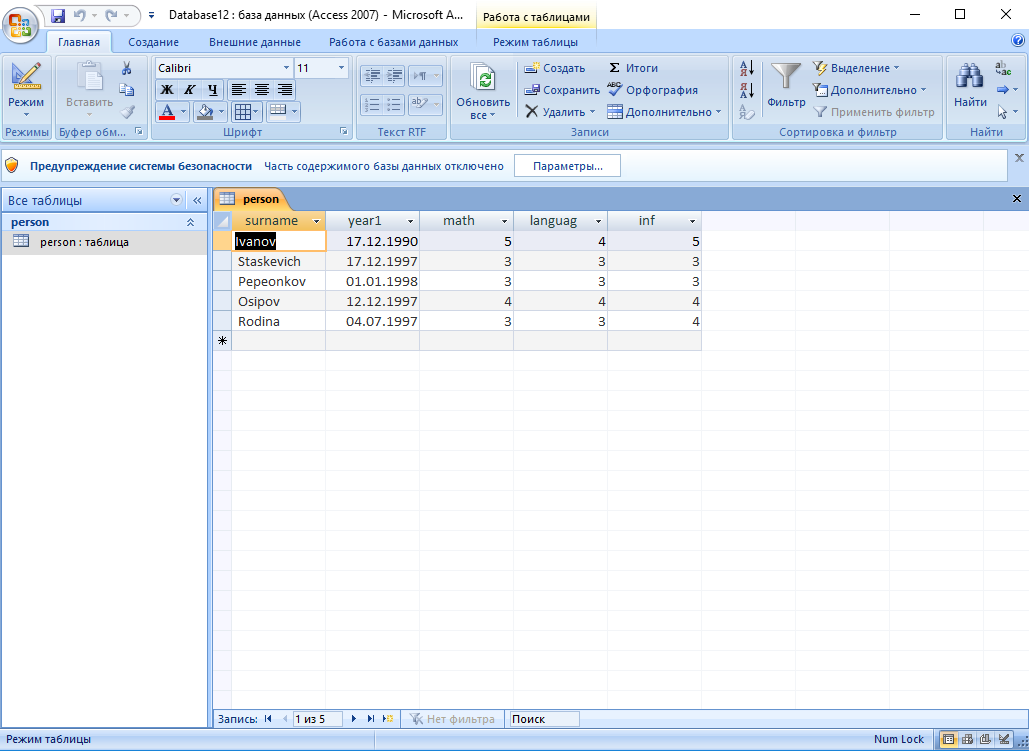


Рис. 10. Результат выполнения последовательности команд

3. Вносим изменения в структуру таблицы - создаём поле, которое будет содержать пол студента:

ALTER TABLE person add sex char(1);

Результат выполнения представлен на рис. 11.

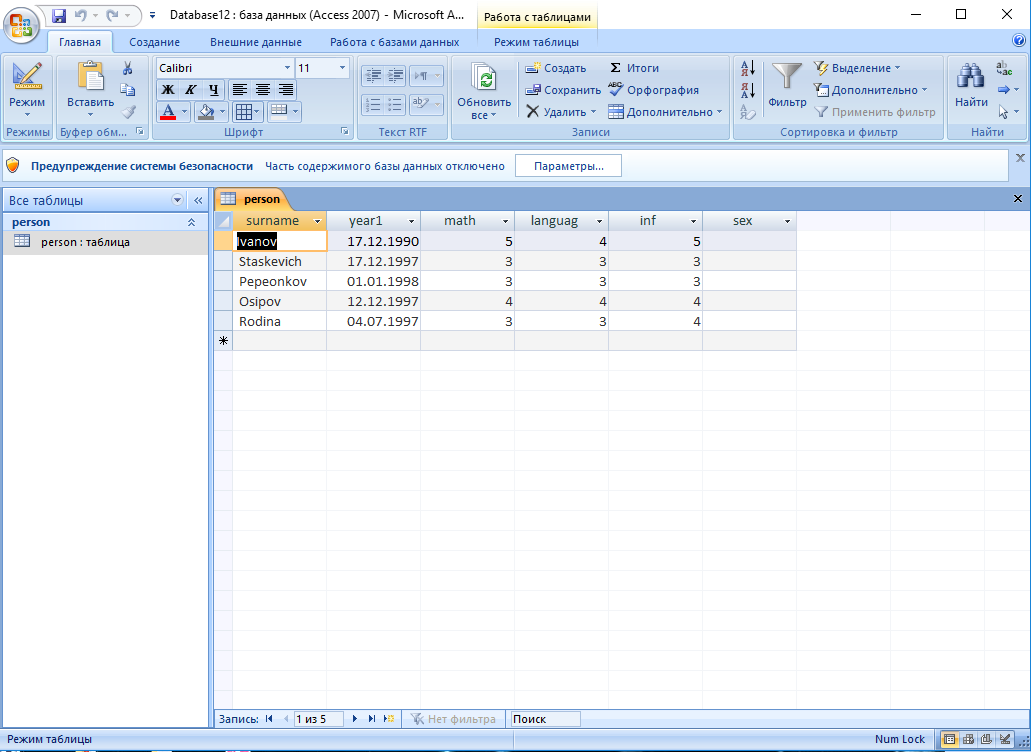


Рис. 11. Добавление нового поля

4. Дополняем данными таблицу, указав пол каждого студента:

Update person set sex = 'M';

Update person set sex = 'F' where surname='Rodina';

Результат выполнения представлен на рис. 12.

Update person set sex = 'M' where surname=’Staskevich’ or surname=’Ivanov’ or surname=’Pepeonkov’ or surname=’Osipov’;

Update person set sex = 'M' where surname<>’Rodina’

Update person set sex = 'f' where surname='Rodina';

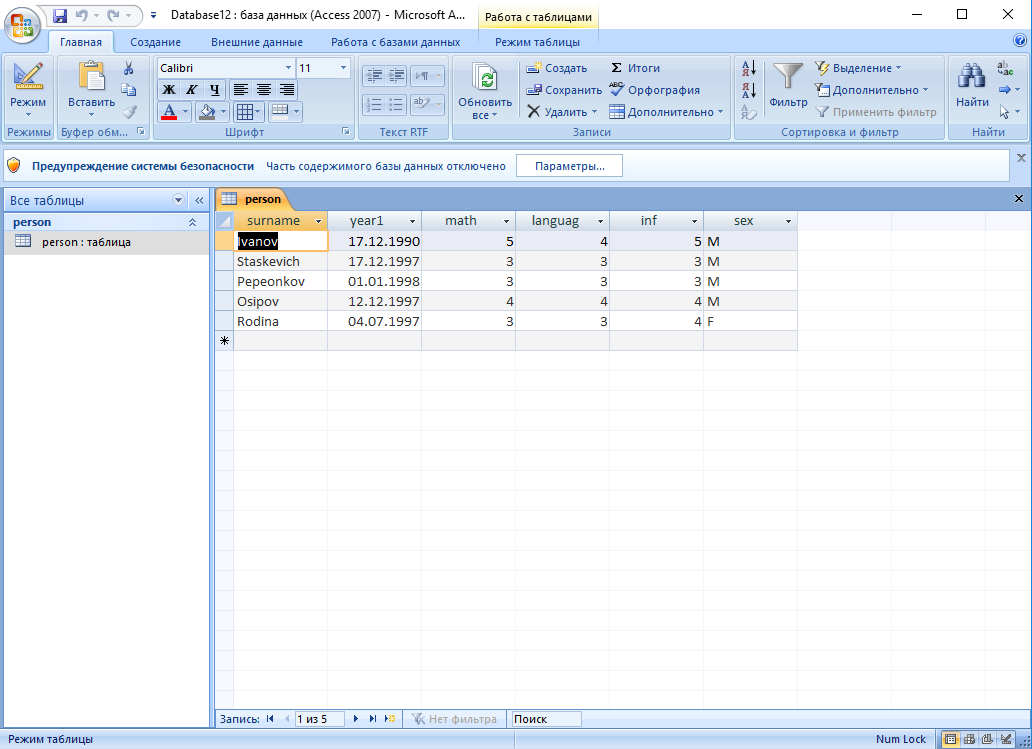


Рис. 12. Указываем пол каждого студента

5. Создаём таблицу с теми же полями, которая содержит сведения о студентах параллельной группы:

select \* into person1 from person where 0 = 1;

Результат выполнения представлен на рис. 13.

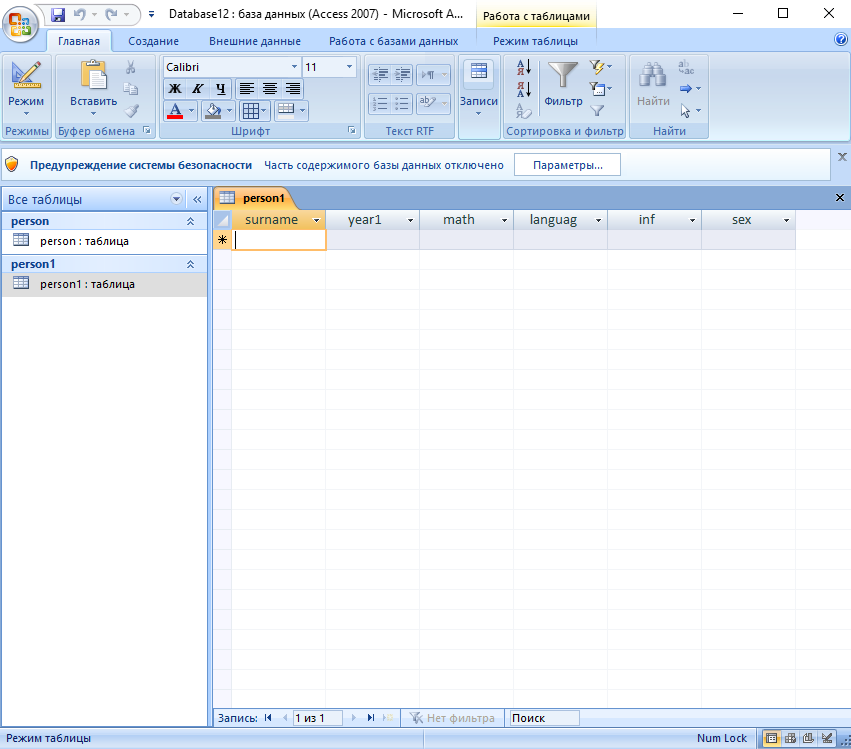


Рис. 13. Создание новой таблицы

6. Вносим во вторую таблицу 5 записей:

INSERT INTO person1(surname,year1,math,languag,inf,sex) VALUES ('Borisov','1997-2-6','3','4','3','M');

INSERT INTO person1(surname,year1,math,languag,inf,sex) VALUES ('Popov','1998-11-10','4','4','3','M');

INSERT INTO person1 VALUES ('Kapustin','1994-04-12','5','5','5','M');

INSERT INTO person1 VALUES ('Жуков','1996-11-8','4','5','5','M');

INSERT INTO person1 VALUES ('Псаки','1993-1-2','3','3','3','Ж');

Результат выполнения представлен на рис. 14.

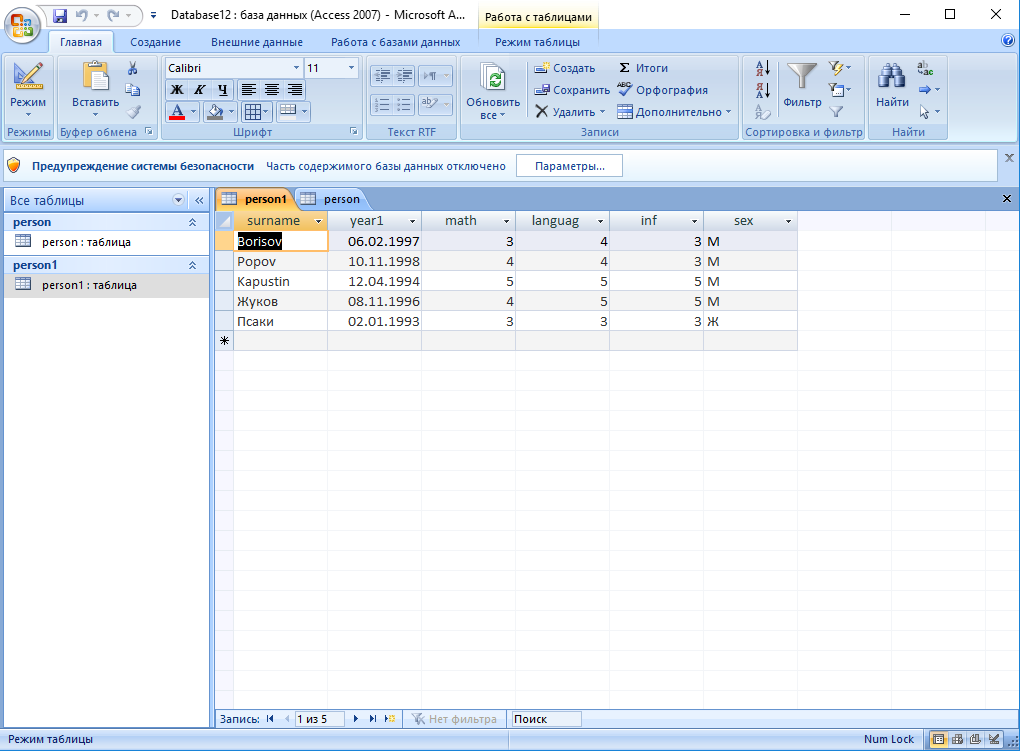


Рис. 14. Вносим записи во вторую таблицу

7. Осуществляем сортировку записей первой таблицы в алфавитном порядке поля фамилий:

SELECT \* FROM person ORDER BY surname;

Результат выполнения представлен на рис. 15.

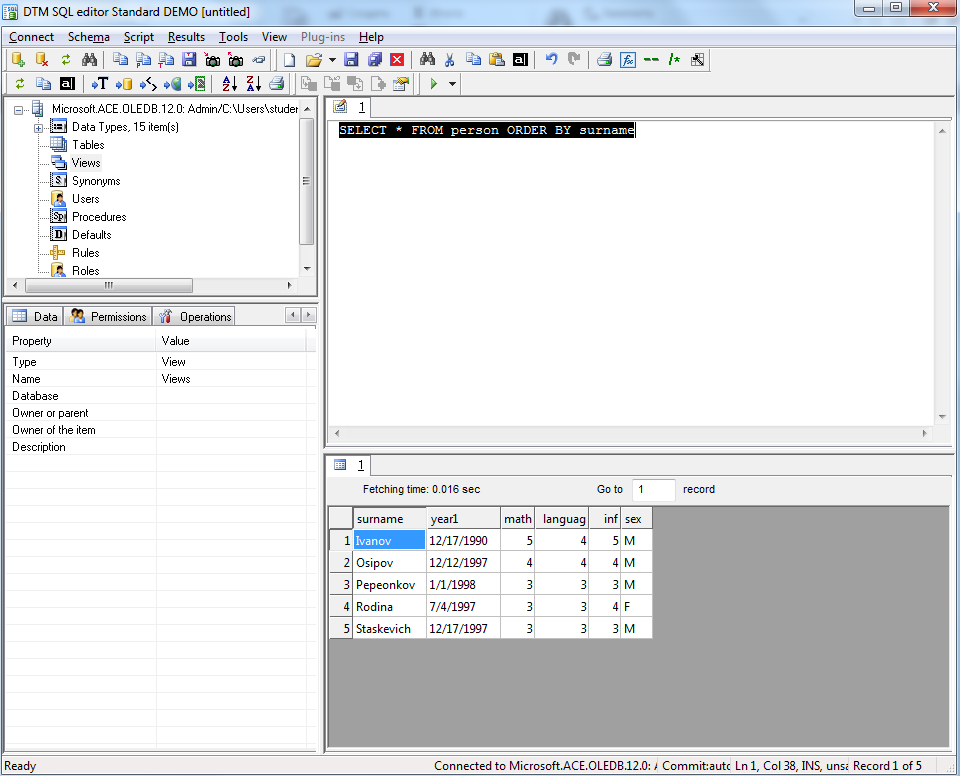


Рис. 15. Сортировка записей первой таблицы

8. Формируем запросы на выборку:

а) студентов с указанием фамилии и даты рождения:

SELECT surname,year1 FROM person;

Результат запроса (рис. 16).

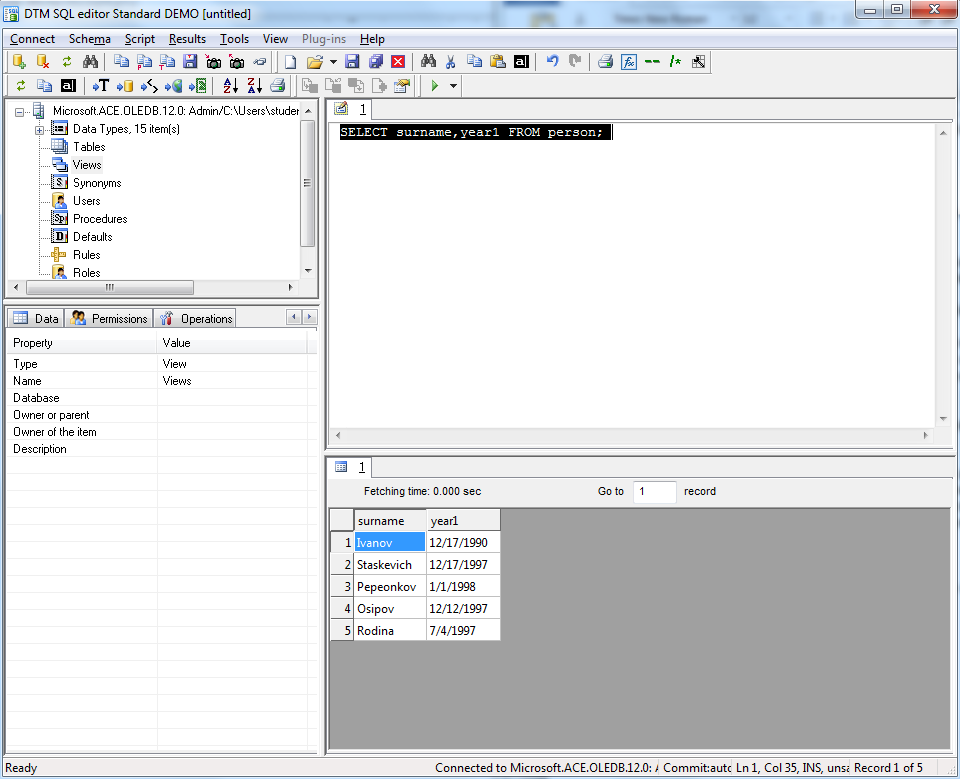


Рис. 16. Запрос студентов с указанием фамилии и даты рождения

б) студентов-отличников по математике:

SELECT \* FROM person where math=5;

Результат запроса на рис. 17.

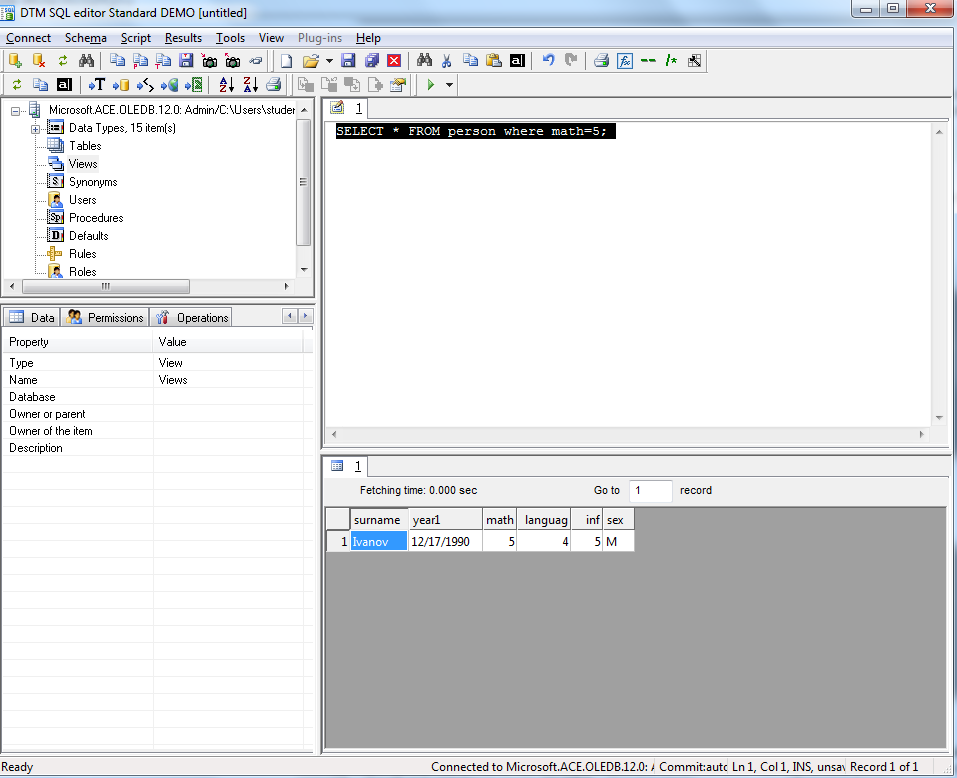


Рис. 17. Запрос студентов-отличников по математике

в) студентов-отличников по всем предметам:

SELECT \* FROM person where math=5 and languag=5 and inf=5;

Результат запроса на рис. 18.

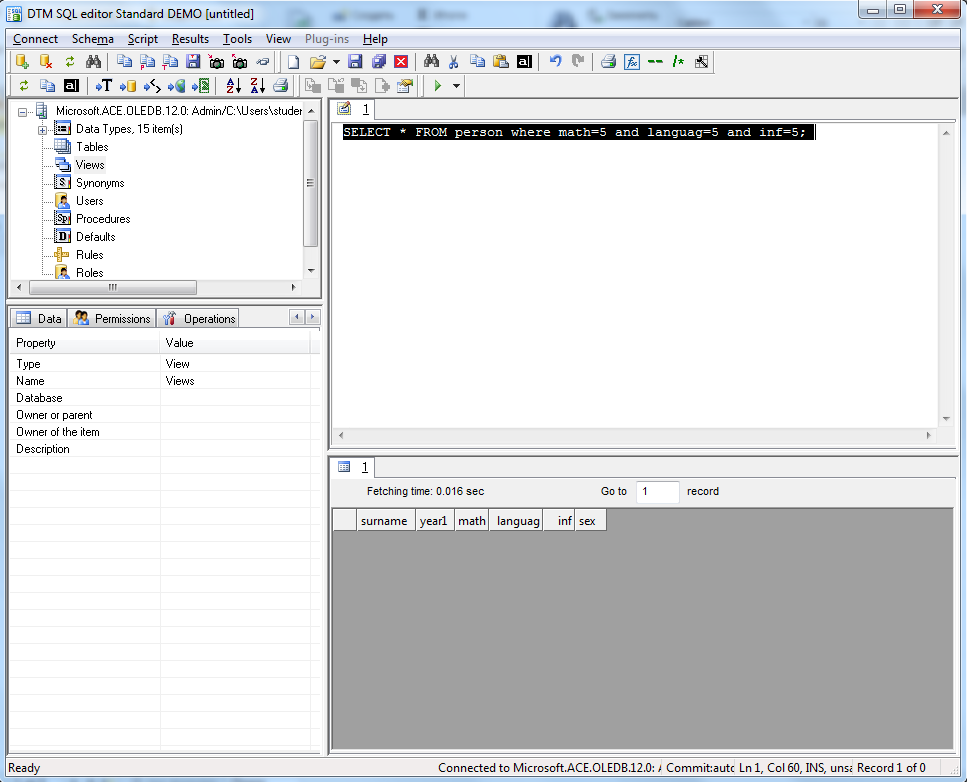


Рис. 18. Запрос студентов-отличников по всем предметам

г) студентов, чей возраст старше 20 лет:

SELECT \* FROM person where (DateDiff("yyyy",year1,DATE())>20);

Результат запроса на рис. 19.

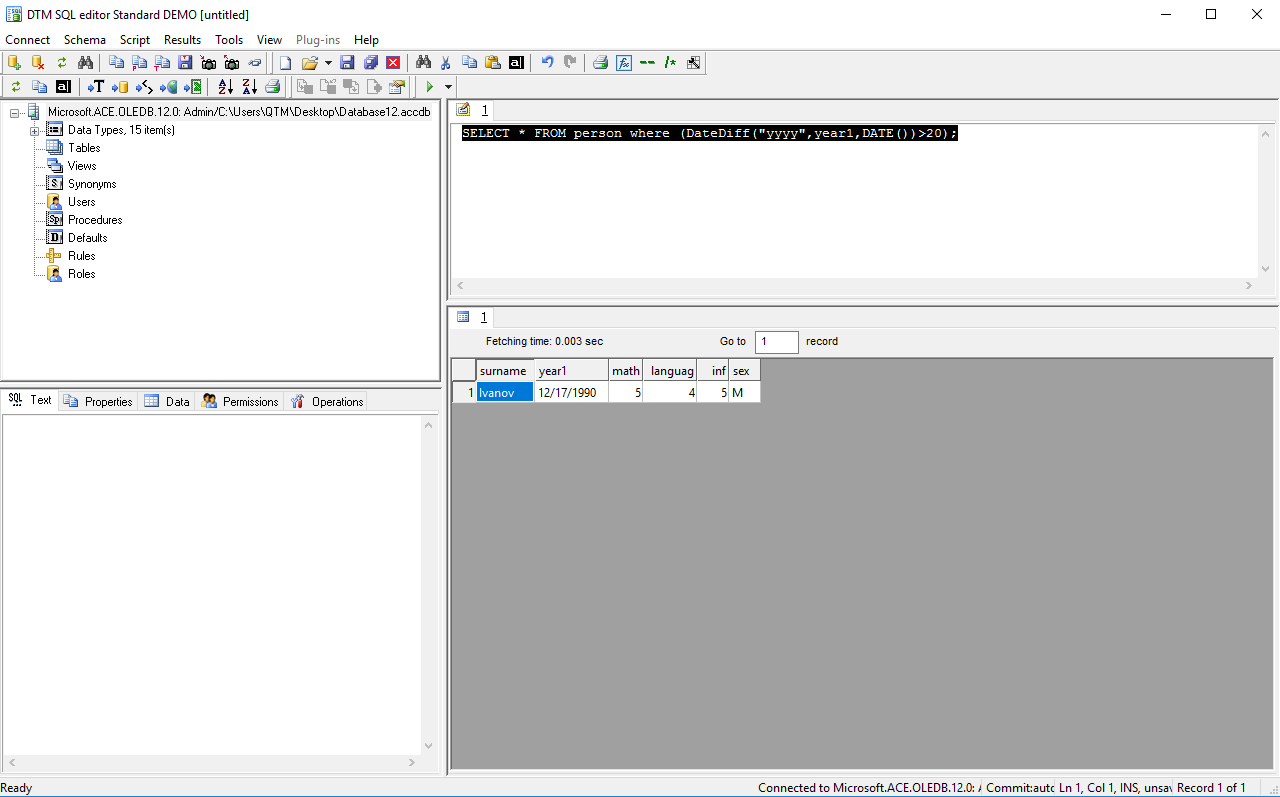


Рис. 19. студентов, чей возраст старше 20 лет

9. Сформируйте параметрический запрос:

а) студентов с фамилией «Иванов»:

SELECT \* FROM person WHERE surname = "Ivanov";

SELECT \* FROM person WHERE surname LIKE "%Ivanov%";

Результат запроса на рис. 20.

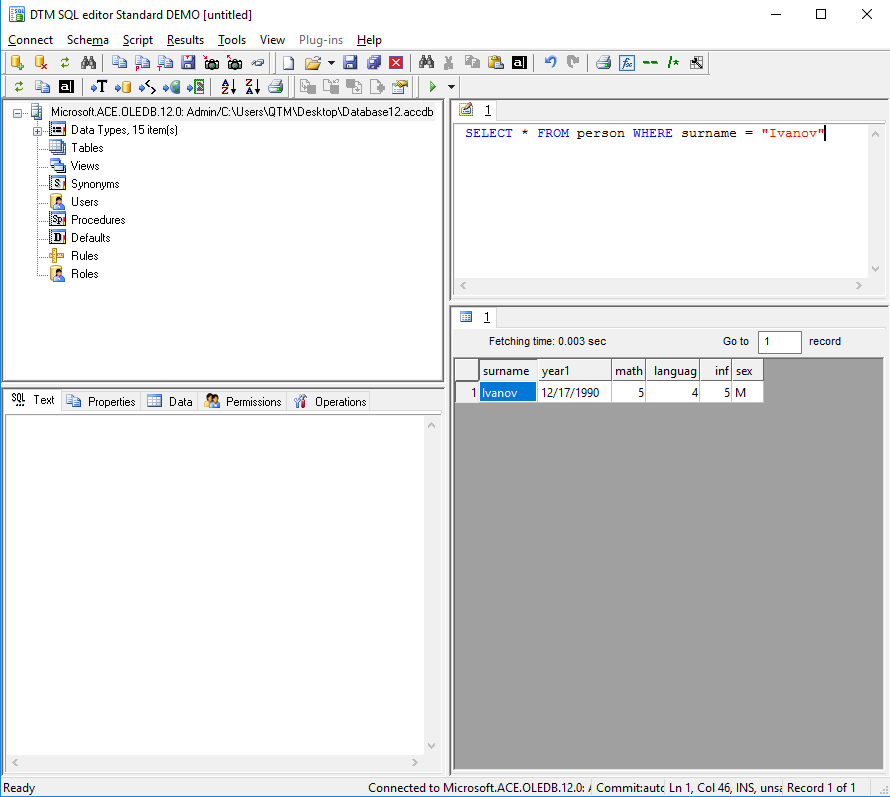


Рис. 20. Запрос студентов с фамилией “Иванов”

б) студентов Ивановых, имеющих «5» по математике:

SELECT \* FROM person WHERE surname = "Ivanov" and math=5;

Результат запроса на рис. 21.

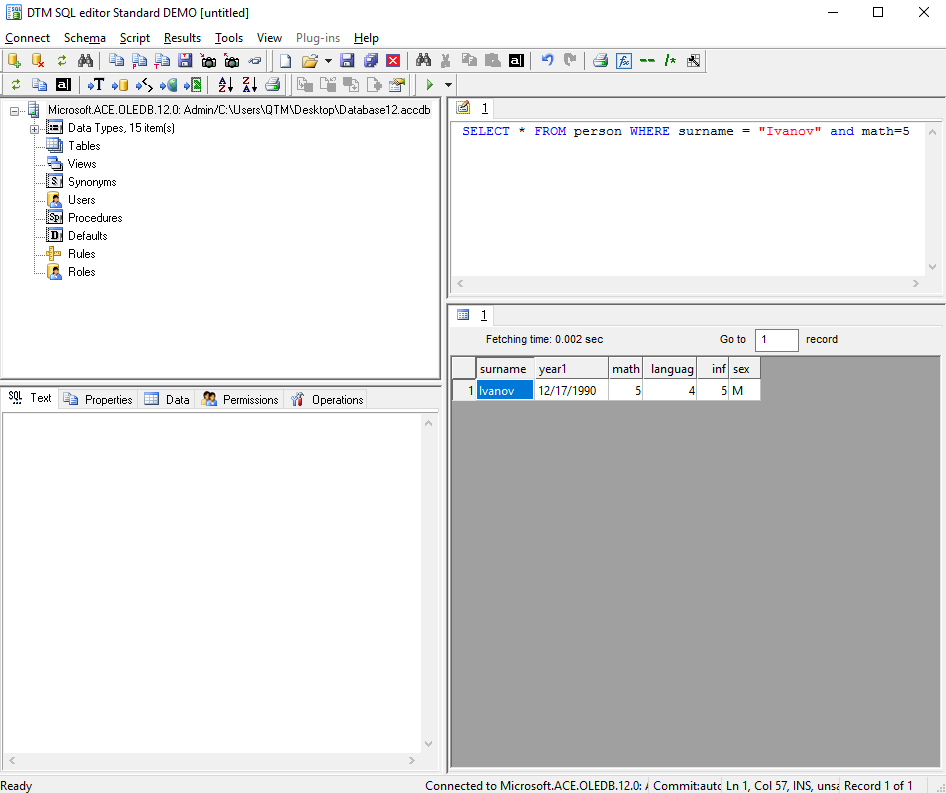


Рис. 21. Запрос Ивановых, имеющих «5» по математике

**Вывод:**

1. Ознакомился с программой управления БД “DTM SQL editor”.

2. Сформировал 2 таблицы со студентами разных групп.

3. Исследовал способы формирования запросов.