Федеральное агентство связи

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

**Межрегиональный** **учебный** **центр**

**переподготовки** **специалистов**

ЛАБОРАТОРНАЯ РАБОТА №3

По дисциплине «Базы данных»

Выполнил: Иванов Иван Иванович

Студент гр.ИИ-661, МЦПС

Приняла: Любовь Филипповна Данилова

Ст. преподаватель кафедры ММБП

Новосибирск 2018 г.

**Лабораторная** **работа** **3**

**Тема:** **Знакомство** **с** **СУБД** **Microsoft** **SQL** **Server** .

**Цель** **работы:** Ознакомится с интерфейсом СУБД, разработать БД.

**Задание** **без** **выбора** **вариантов,** **выполняется** **в** **полном** **объеме:**

1. Установить ПО, согласно инструкциям, представленным в лабораторной работе. Реализовать в СУБД **Microsoft** **SQL** **Server** пример № 1 «Учебная база данных». Реализовать запросы, представленные в конце файла.

По результатам выполнения ЛР, формируется файл БД с запросами, отчет по ЛР, в текстовом формате. В отчете необходимо описать основые этапы работы. Представить текст задачи, SQL запросы и результаты запросов ("Скрин" из среды Microsoft SQL Server).

**Знакомство** **со** **средой** **разработки.**

**Microsoft** **SQL** **Server** — система управления реляцион[ными б](https://ru.wikipedia.org/wiki/Sybase)азами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.

1. Для начала работы необходимо установить программный пакет, который можно скачать с официального сайта мыйкрософт. Версия продукта сервер 2017 Express Edition. После скчивания необходимо запустить файл и следовать рекомендациям установщика. По итогу установки на экране представится следующее диалоговое окно. Все настройки оставляем по умолчанию.

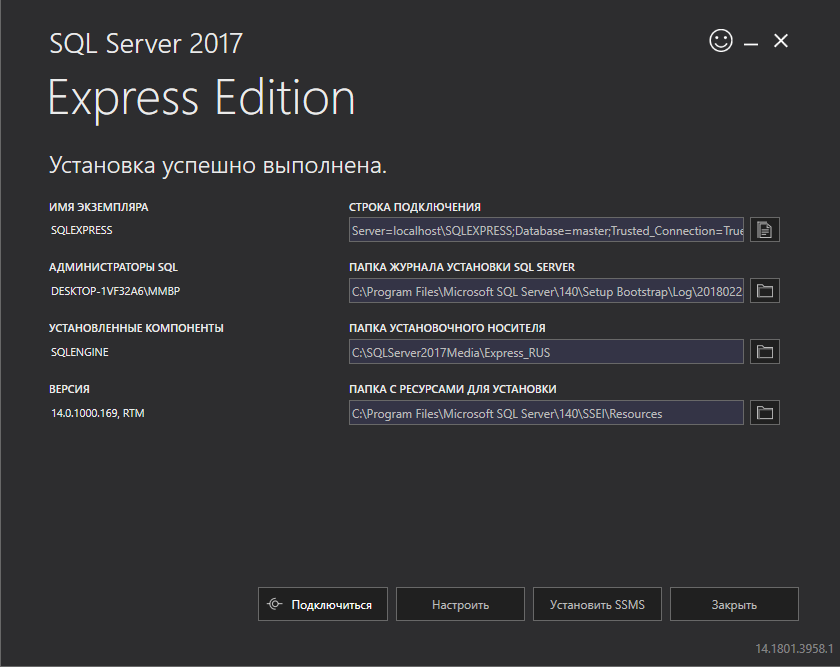


Рис. 1.

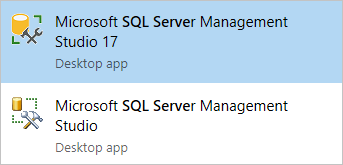
2. Для работы так же потребуется скачать пакет Microsoft SQL Server

|  |  |
| --- | --- |
| Management Studio Express. SSMS 17.x — последняя  поддерживает SQL Server 2 |  |
| версия *SQL* *Server* *Management* *Studio*, которая  017. |

Среда Microsoft SQL Server Management Studio — это бесплатная интегрированная среда для доступа, настройки, управления, администрирования и разработки всех компонентов SQL Server, а также объединение простых в использовании графических средств и редакторов с богатыми сценарными возможностями для доступа к SQL Server разработчиков и администраторов с любым уровнем подготовки.

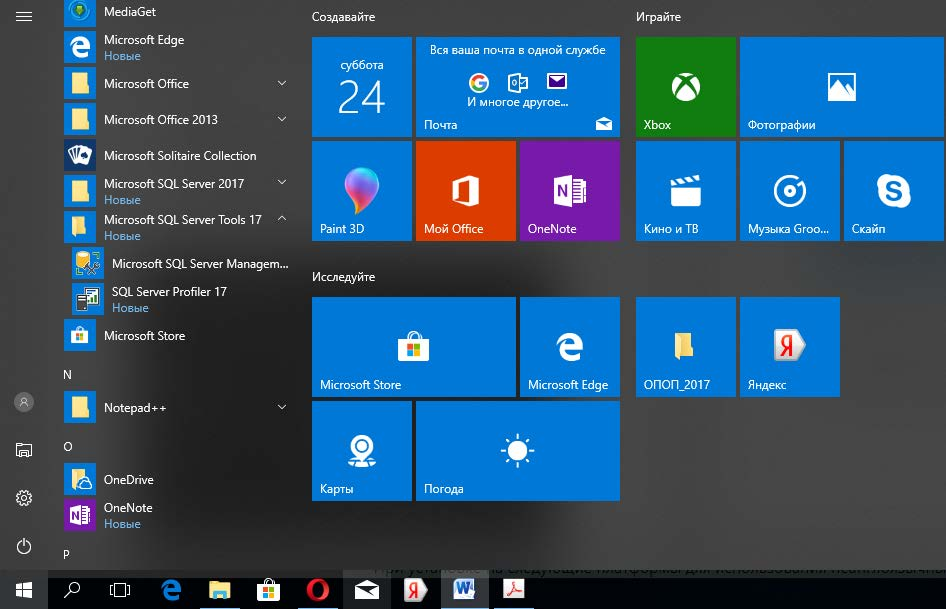
При установке SSMS 17.x не обновляются и не заменяются версии SSMS 16.x или более ранние. Среда SSMS 17.x устанавливается параллельно с предыдущими версиями, и обе версии остаются доступными для использования. Если на компьютере есть несколько параллельных установок SSMS, всегда проверяйте, правильную ли версию вы запускаете. Последняя версия называется *Microsoft* *SQL* *Server* *Management* *Studio* *17* и имеет новый значок:

Рис. 2.



3. После установки пакета SSMS 17.x в окне Пуск-Программы операционной системы необходимо выбрать соответствующий значок и запустить.

Рис. 3



4. После запуска появляется окно, но котором необходимо нажать кнопку «Соединить».

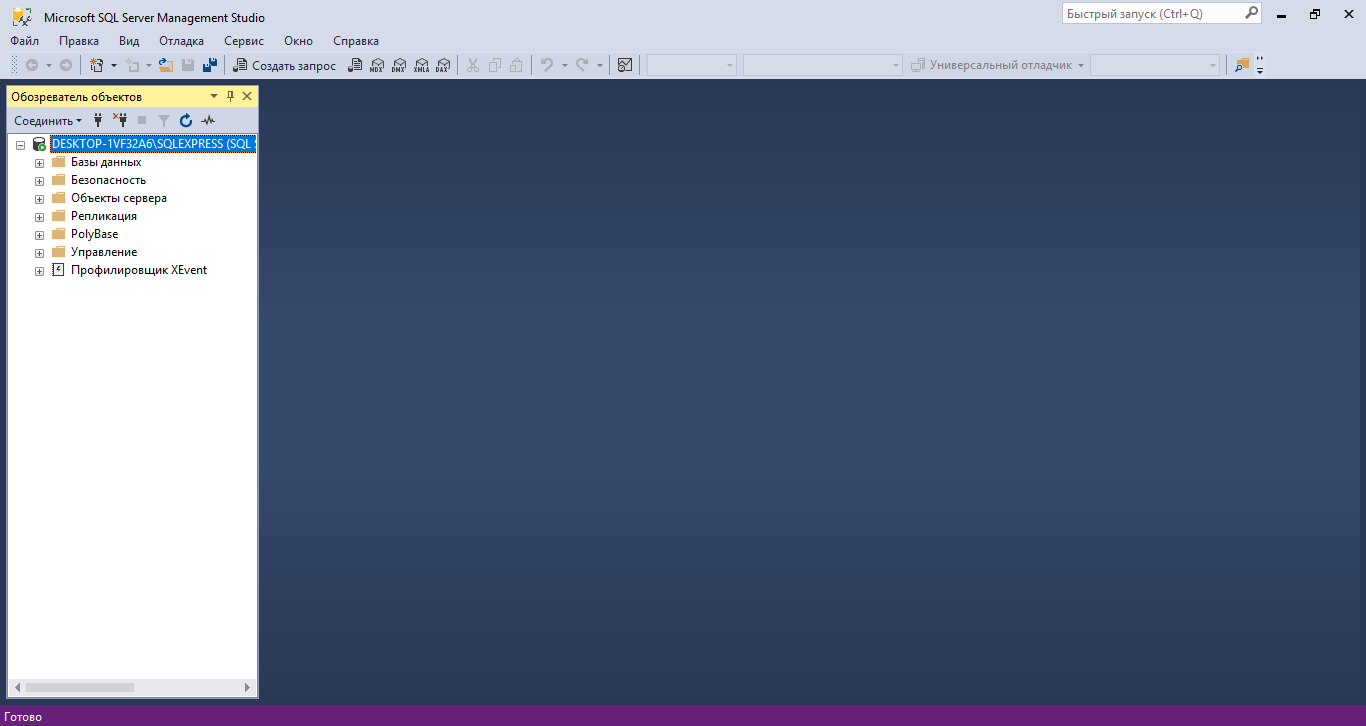
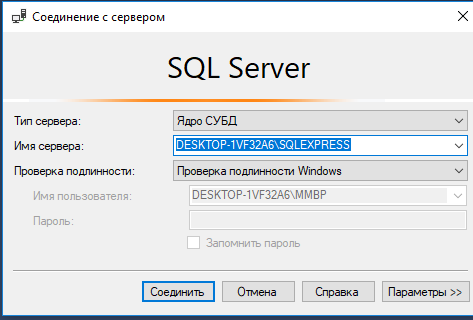


Рис. 4.



5. Далее открывается среда разработки.

Рис. 5.

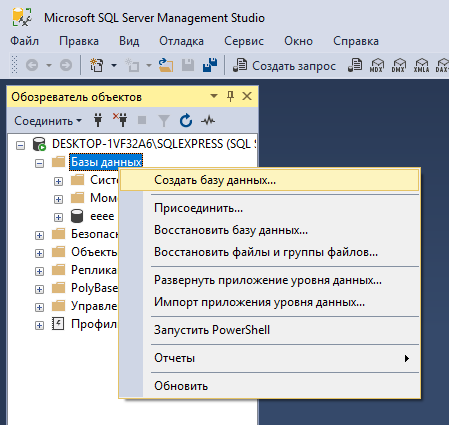
6. Для создания Базы данных, необходимо навести курсор в Обозревателе объектов на элемент «Базы данных», нажать правой кнопкой мыши и выбрать «Создать БД»

Рис. 6.

7. Далее задаем имя БД. Имя не должно иметь пробелов. И выбираем «ОК»

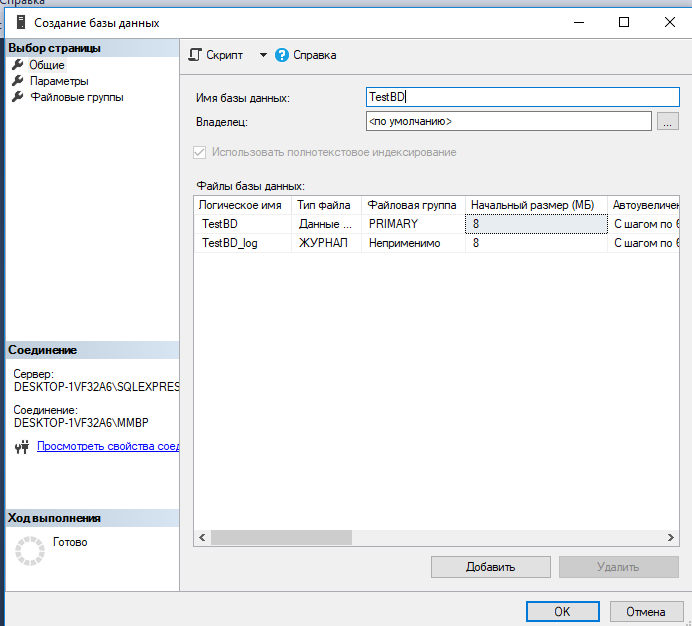


Рис. 7.

8. Для того что бы созданная БД отобразилась в Обозревателе объектов нажимаем на значок «Обновить»

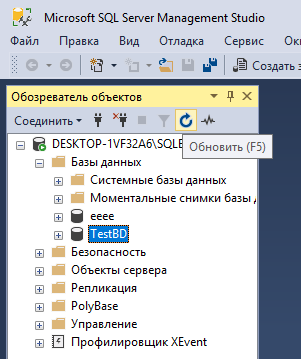


Рис. 8.

9. Следующей задачей после создания базы данных является создание всех необходимых таблиц. Подобно созданию базы данных, таблицы в ней также можно создать либо с помощью языка Transact-SQL, либо посредством обозревателя объектов. Как и в случае с созданием базы данных, здесь мы рассмотрим создание таблиц только с помощью обозревателя объектов.

10.Чтобы создать таблицу базы данных с помощью обозревателя объектов, разверните в нем узел (Базы данных), а потом узел требуемой базы данных, щелкните правой кнопкой папку Tables (Таблицы) и в открывшемся контекстном меню выберите пункт (Таблица….). В верхней части с правой стороны окна средства Management Studio откроется окно для создания столбцов новой таблицы. Введите имена столбцов таблицы, их типы данных и разрешение значений NULL для каждого столбца, как это показано в правой верхней панели на рисунках.

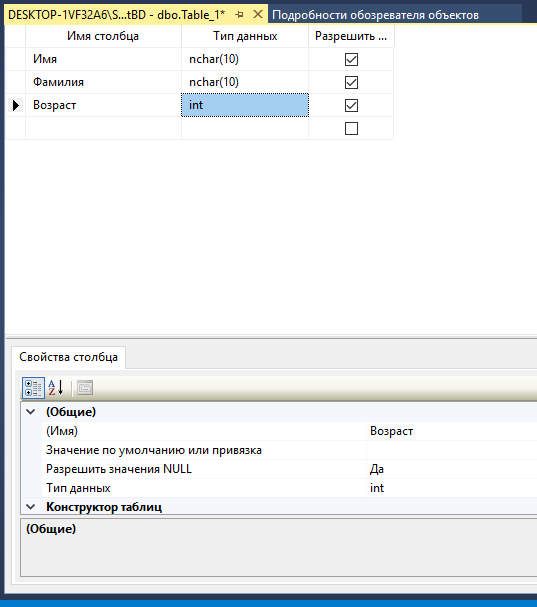
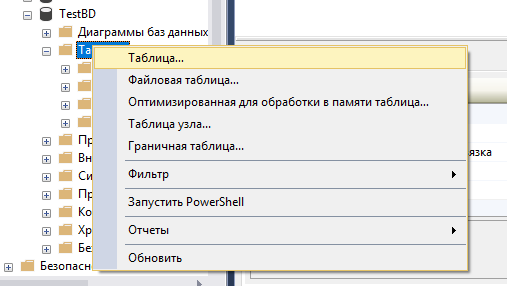
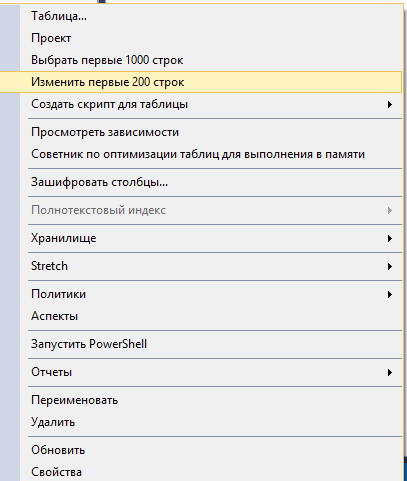


Рис. 9.

Рис. 10.

11.Чтобы выбрать для столбца один из поддерживаемых системой типов данных, в столбце (Тип данных) выберите, поле, в результате в открывшемся раскрывающемся списке выберите требуемый тип данных для столбца.

Тип данных существующего столбца можно изменить на вкладке Column Properties (Свойства столбца) Для одних типов данных, таких как char, требуется указать длину в строке Length (Длина), а для других, таких как decimal, на вкладке (Свойства столбца) требуется указать масштаб и точность в соответствующих строках (Масштаб) и (Точность). Для некоторых других, таких как int, не требуется указывать ни одно из этих свойств. (Недействительные значения для конкретного типа данных выделены затененным шрифтом в списке всех возможных свойств столбца.)

Чтобы разрешить значения NULL для данного столбца, следует установить для него соответствующий флажок поля. Также, если для столбца требуется значение по умолчанию, его следует ввести в строку (Значение по умолчанию или привязка) панели (Свойства столбца). Далее сохраняем и задаем имя таблицы.

12.Для заполнения таблиц необходимо нажать правую кнопку мыши и выбрать– Изменить первые 200 строк, далее откроются поля для заполнения в них ввести необходимые данные.

Рис. 11.

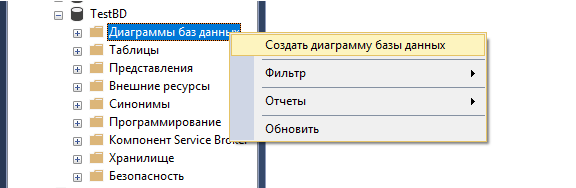
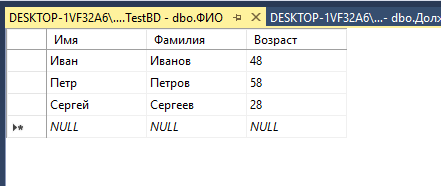


Рис. 12.

13.Создав все таблицы базы данных, можно использовать еще одну возможность среды SQL Server Management Studio, чтобы отобразить диаграмму типа "сущность — отошение" — диаграмму (ER) (entity-relationship) этой базы данных. (Процесс преобразования таблиц базы данных в диаграмму "сущность — отношение" (ER) называется обратным проектированием.)

Чтобы создать диаграмму "сущность — отношение" (ER) для базы данных, щелкните правой кнопкой ее подпапку (Диаграммы баз данных) и в контекстном меню выберите пункт (Создать диаграмму базы данных).

Рис. 13.

Откроется диалоговое окно (Добавление таблицы), в котором нужно выбрать таблицы для добавления в диаграмму. Добавив все необходимые таблицы, нажмите кнопку (Закрыть), и мастер создаст диаграмму, подобную показанной на рис.

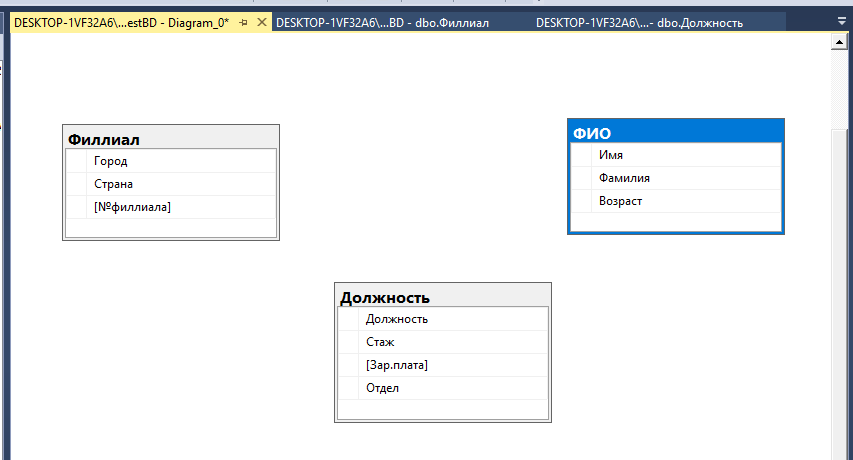


Рис. 14.

На рис. показана только промежуточная, а не конечная, диаграмма "сущность — отношение" (ER) базы данных, поскольку, хотя на ней и показаны все таблицы с их столбцами (и соответствующими первичными ключами), на ней все же отсутствуют отношения между таблицами. Отношение между двумя таблицами основывается на первичном ключе одной из таблиц и возможным соответствующим столбцом (или столбцами) другой таблицы.

14.Чтобы создать эти три отношения, диаграмму "сущность — отношение" (ER) нужно реконструировать, указав для каждой таблицы столбцы, которые соответствуют ключевым столбцам других таблиц. Такой столбец называется внешним ключом (foreign key). Чтобы увидеть, как это делается, определим столбец Фамилия таблицы Должность, как внешний ключ таблицы Фамилия. Для этого выполним следующие действия:

В созданной диаграмме щелкните правой кнопкой графическое представление таблицы ФИО и в контекстном меню выберите пункт (Отношения). В открывшемся диалоговом окне (Связи по внешнему ключу) нажмите кнопку (Добавить).

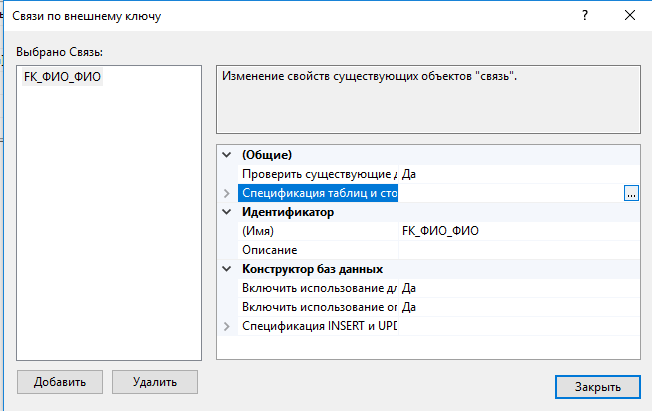
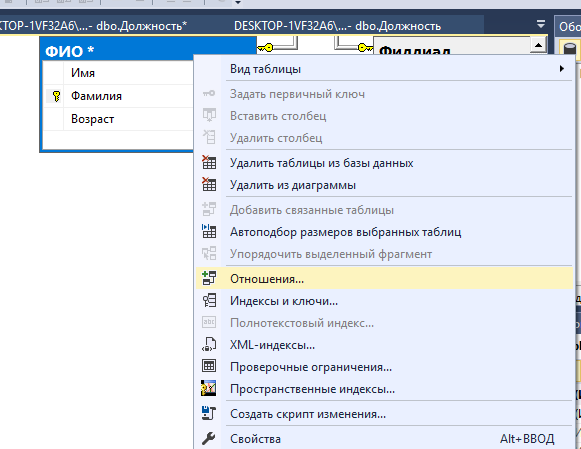


Рис. 15.

Рис. 16.

В правой панели диалогового окна расширьте первый столбец, выберите в нем строку (Спецификация таблиц и столбцов) и нажмите кнопку с троеточием во втором столбце этой строки.

В открывшемся диалоговом окне (Таблицы и столбцы) в раскрывающемся списке (Таблица первичного ключа) выберите таблицу Должность с соответствующим первичным ключом – Должность.

Выберите для этой таблицы столбец Фамилия в качестве первичного ключа и этот же столбец для таблицы ФИО в качестве внешнего ключа, после чего нажмите кнопку ОK, чтобы закрыть окно (Таблицы и столбцы). Нажмите кнопку (Закрыть), чтобы закрыть окно (Связи по внешнему ключу).

15.Среда SQL Server Management Studio предоставляет завершенное средство для создания всех типов запросов. С ее помощью можно создавать, сохранять, загружать и редактировать запросы. Кроме этого, над запросами можно работать без подключения к какому-либо серверу. Этот инструмент также предоставляет возможность разрабатывать запросы для разных проектов.

Предоставляется возможность работать с запросами как посредством редактора запросов, так и с помощью обозревателя решений. В этом разделе рассматриваются оба эти инструмента. Кроме этих двух компонентов среды SQL Server Management Studio мы рассмотрим отладку SQL-кода, используя встроенный отладчик.

16.Редактор запросов

Чтобы открыть панель редактора запросов (Редактор запросов), на панели инструментов среды SQL Server Management Studio нажмите кнопку (Создать запрос).

Редактор запросов можно использовать для выполнения следующих задач:  создания и выполнения инструкций языка Transact-SQL;

 сохранения созданных инструкций языка Transact-SQL в файл;

создание и анализирование планов выполнения общих запросов;

графического иллюстрирования плана выполнения выбранного запроса.

Редактор запросов содержит встроенный текстовый редактор и панель инструментов с набором кнопок для разных действий. Главное окно редактора запросов разделено по горизонтали на панель запросов (вверху) и панель результатов (внизу). Инструкции Transact-SQL (т. е. запросы) для исполнения вводятся в верхнюю панель, а результаты обработки системой этих запросов отображаются в нижней панели. На рис. показан пример ввода запроса в редактор запросов и результатов выполнения этого запроса.

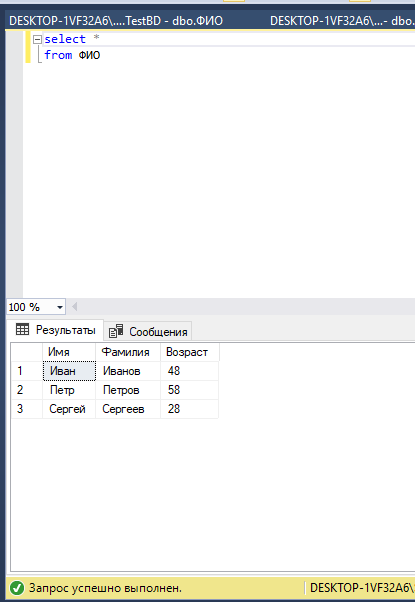


Рис.17

**1. Учебная база данных**

В приводимых в пособии примерах построения SQL-запросов и контрольных упражнениях используется база данных, состоящая из следующих таблиц.

Таблица 1.1. STUDENT (Студент)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STUDENT\_ID | SURNAME | NAME | STIPEND | KURS | CITY | BIRTHDAY | UNIV\_ID |
| 1 3 6 10 12  265 32 654 276 55  … … | Иванов Петров  Сидоров  Кузнецов Зайцева  Павлов  Котов  Лукин Петров  Белкин  … … | Ив ан Петр  Вадим  Борис  Ольга  Андрей Павел  Артем  Антон Вадим  … … | 150 200 150 0 250 0 150 200 200 250  … … | 1 3 4 2 2 3 5 3 4 5  … … | Орел  Курск Москва  Брянск Липецк  Воронеж  Белгород Воронеж NULL Воронеж  … … | 3/12/1982 1/12/1980 7/06/1979 8/12/1981 1/05/1981 5/11/1979 **NULL** 1/12/1981 5/08/1981 7/01/1980  … … | 10 10 22 10 10 10 14 10 22 10  … … |

STUDENT\_ID– числовой код, идентифицирующий студента, SURNAME– фамилия студента,

NAME– имя студента,

STIPEND– стипендия, которую получает студент, KURS– курс, на котором учится студент,

CITY– город, в котором живет студент, BIRTHDAY– дата рождения студента,

UNIV\_ID– числовой код, идентифицирующий университет, в котором учится студент.

Таблица 1.2. LECTURER (Преподаватель)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LECTURER\_ID | SURNAME | NAME | CITY | UNIV\_ID |
| 24 46 74 108 276 328  … … | Колесников  Никонов Лагутин Струков  Николаев Сорокин  … … | Борис  Иван  Павел  Николай Виктор  Андрей  … … | Воронеж Воронеж Москва Москва Воронеж  Орел  … … | 10 10 22 22 10 10  … … |

LECTURER\_ID – числовой код, идентифицирующий преподавателя,

SURNAME– фамилия преподавателя, NAME– имя преподавателя,

CITY– город, в котором живет преподаватель,

UNIV\_ID– идентификатор университета, в котором работает преподаватель.

Таблица 1.3. SUBJECT (Предмет обучения)

|  |  |  |
| --- | --- | --- |
| SUBJ\_ID | SUBJ\_NAME HOUR | SEMESTER |
| 10 22 43 56 94 73  … … | Информатика 56 Физика 34 Математика 56 История 34 Английский 56 Физкультура 34  … … … … | 1 1 2 4 3 5  … … |

SUBJ\_ID– идентификатор предмета обучения, SUBJ\_NAME– наименование предмета обучения, HOUR– количество часов, отводимых на изучение SEMESTER– семестр, в котором изучается данный

предмета, предмет.

Таблица 1.4. UNIVERSITY (Университеты)

|  |  |  |  |
| --- | --- | --- | --- |
| UNIV\_ID | UNIV\_NAME | RATING | CITY |
| 22 10 11 32 14 15 18  … … | МГУ  ВГУ  НГУ  РГУ  БГУ  ТГУ  ВГМА  … … | 606 296 345 416 326 368 327  … … | Москва  Воронеж  Новосибирск Ростов  Белгород  Томск  Воронеж  … … |

UNIV\_ID– идентификатор университета, UNIV\_NAME– название университета, RATING– рейтинг университета,

CITY– город, в котором расположен университет.

Таблица 1.5. EXAM\_MARKS (Экзаменационные оценки)

|  |  |  |
| --- | --- | --- |
| EXAM\_ID | STUDENT\_ID | SUBJ\_ID MARK EXAM\_DATE |
| 145 34 75 238 639 43  … … | 12 32 55 12 55 6  … … | 10 5 12/01/2000 10 4 23/01/2000 10 5 05/01/2000 22 3 17/06/1999 22 **NULL** 22/06/1999 22 4 18/01/2000  … … … … … … |

EXAM\_ID– идентификатор экзамена, STUDENT\_ID– идентификатор студента, SUBJ\_ID– идентификатор предмета обучения, MARK– экзаменационная оценка,

EXAM\_DATE– дата экзамена.

Таблица 1.6. SUBJ\_LECT (Учебные дисциплины преподавателей)

|  |  |
| --- | --- |
| LECTURER \_ID | SUBJ\_ID |
| 24 46 74 108 276 328  … … | 24 46 74 108 276 328  … … |

LECTURER\_ID – идентификатор преподавателя, SUBJ\_ID– идентификатор предмета обучения.

ВОПРОСЫ

1. Какие поля приведенных таблиц являются первичными ключами?

2. Какие данные хранятся в столбце 2 в таблице “Предмет обучения”?

3. Как по-другому называется строка? Столбец?

4. Почему мы не можем запросить для просмотра первые пять строк?

**Перечень** **запросов.**

1. Напишите запрос, позволяющий получить из таблицы EXAM\_MARKS значения столбца MARK ( экзаменационная оценка) для всех студентов , исключив из списка повторение одинаковых строк.

2. Напишите запрос, который выполняет вывод списка фамилий студентов, обучающихся на третьем и более старших курсах. 7. Напишите запрос, выбирающий данные о фамилии, имени и номере курса для студентов, получающих стипендию больше 140.

3. Напишите запрос, выполняющий вывод находящихся в таблице EXAM\_MARKS номеров предметов обучения, экзамены по которым сдавались между 10 и 20 января 1999 года.

4. Напишите запрос, выбирающий данные обо всех предметах обучения , экзамены по которым сданы студентами, имеющими идентификаторы 12 и 32.

5. Напишите запрос, выбирающий сведения о студентах, у которых имена начинаются на буквы ‘И’ или ‘С’.

6. Напишите запрос для выбора из таблицы EXAM\_MARKS записей , для которых отсутствуют значения оценок (поле MARK).

7. Напишите запрос, который выполняет выборку для каждого студента значения его идентификатора и максимальной из полученных им оценок. 13. Напишите запрос, выполняющий вывод фамилии первого в алфавитном порядке (по фамилии) студента, фамилия которого начинается на букву “И”.

8. Напишите запрос, который выполняет вывод для каждого предмета обучения наименование предмета и максимальное значение номера семестра, в котором этот предмет преподается.

9. Напишите запрос, который выполняет вывод данных для каждого конкретного дня сдачи экзамена о количестве студентов, сдававших экзамен в этот день .

10. Напишите запрос для получения среднего балла для каждого курса по каждому предмету.

11. Предположим, что стипендия всем студентам увеличена на 20%. Напишите запрос к таблице STUDENT, выполняющий вывод номера студента, фамилию студента и величину увеличенной стипендии. Выходные данные упорядочить : а) по значению последнего столбца (величине стипендии); б) в алфавитном порядке фамилий студентов.

12. Напишите запрос, который по таблице EXAM\_MARKS позволяет найти а) максимальные и б) минимальные оценки каждого студента и выводит их вместе с идентификатором студента.

13. Напишите запрос, который выполняет выборку имен всех студентов, имеющих по предмету c идентификатором 102 балл ниже общего среднего балла.

14. Напишите запрос, выполняющий вывод количества предметов, по которым экзаменовался каждый студент, сдававший более 20-ти предметов.

15. Напишите команду SELECT, использующую связанные подзапросы и выполняющую вывод имен и идентификаторов студентов, у которых стипендия совпадает с максимальным значением стипендии для города, в котором живет студент.

16. Создайте объединение двух запросов, которые выдают значения полей UNIV\_NAME, CITY, RATING для всех университетов. Те из них, у которых рейтинг равен или выше 300, должны иметь комментарий ‘Высокий ’ , все остальные – ‘Низкий ’ .

17. Напишите команду, которая выдает список фамилий студентов, с комментарием ‘успевает’ у студентов, имеющих все положительные оценки, комментарием ‘не успевает’ для сдававших экзамены, но имеющих хотя бы одну неудовлетворительную оценку, и комментарием ‘не сдавал’ – для всех остальных. В выводимом результате фамилии студентов упорядочить по алфавиту.

18. Напишите запрос, который выполняет вывод данных о фамилиях , сдававших экзамены студентов, вместе с идентификаторами каждого сданного ими предмета обучения .

19. Напишите запрос, который выполняет выборку значений фамилии всех студентов с указанием для студентов, сдававших экзамены, идентификаторов сданных ими предметов обучения .

20. Напишите запрос, который выполняет вывод данных о фамилиях студентов, сдававших экзамены, вместе с наименованиями каждого сданного ими предмета обучения .