Министерство Образования и Науки Республики Казахстан

Карагандинский Технический Университет имени Абылкаса Сагинова

Кафедра ИВС

***Лабораторная работа №3***

**Дисциплина:** Основы баз данных

# Тема: Создание отношений и диаграммы базы данных. Ввод данных

**Принял:**

*Нұртай М.Д.*

*(оценка)*

*(фамилия, инициалы)*

*(подпись) (дата)*

**Выполнил:**

*Новиков Д.Д.*

*(фамилия, инициалы)*

*гр. Ис-20-2*

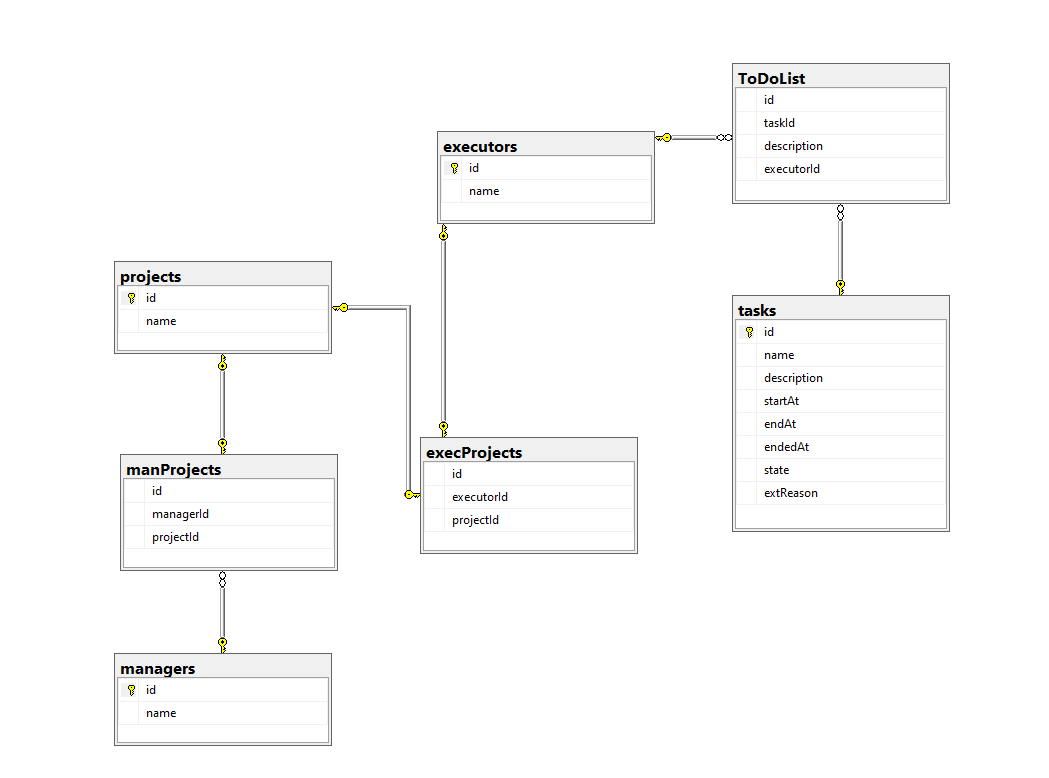
Караганда 2022

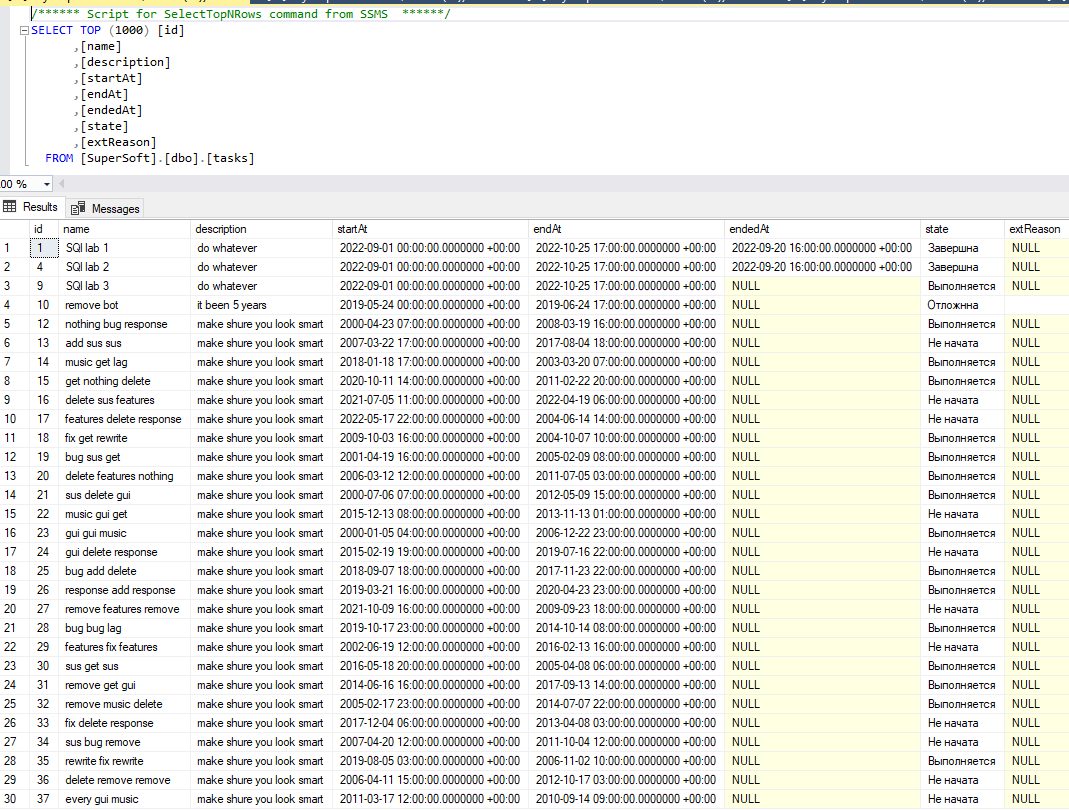
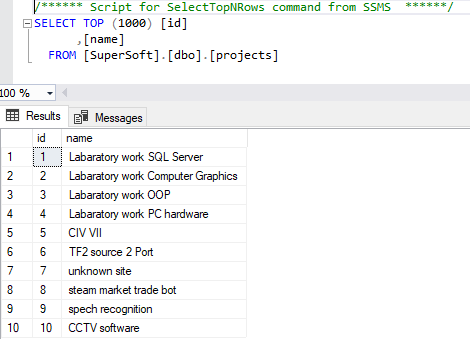
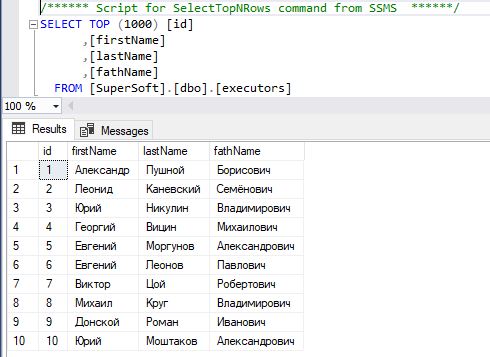
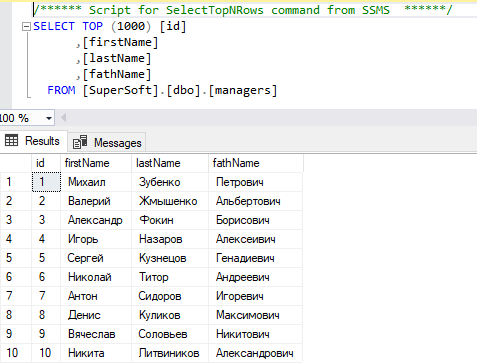
**Цель работы:** освоить способы создания отношений и диаграммы отношений базы данных средствами СУБД MS SQL Server 2012

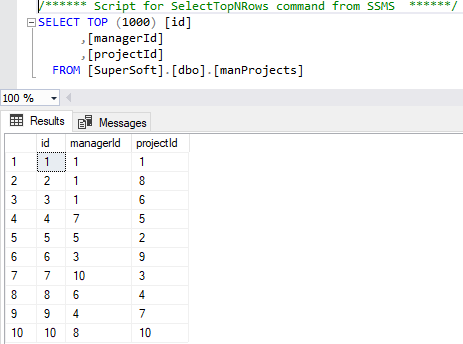
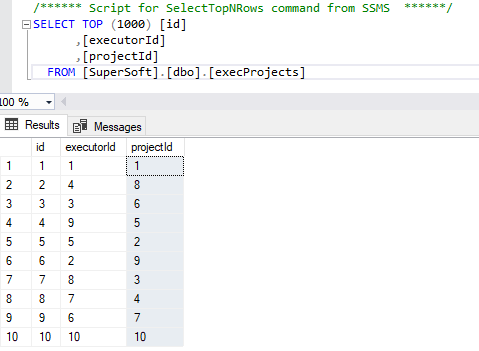
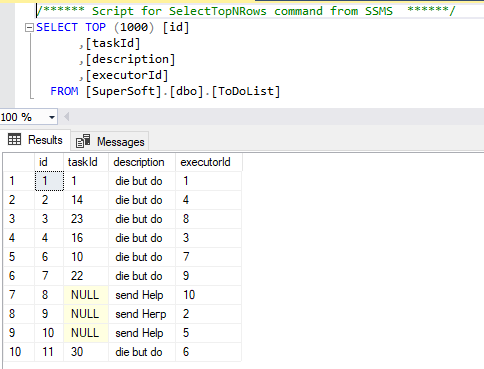
**Задачи:**

1. Построить отношения между таблицами вашей базы данных
2. Построить диаграмму вашей базы данных
3. Ввести в ваши таблицы тестовые наборы данных (в родительские таблицы – не менее 10 записей, в дочерние – не менее 30). При необходимости откорректируйте созданные в лабораторной работе № 2 ограничения на вводимые данные

**Результаты выполнения индивидуального задания**

1. Были построены отношения между таблицами и создана диаграмма отношений
2. Таблицы были наполнены данными





**Контрольные вопросы**

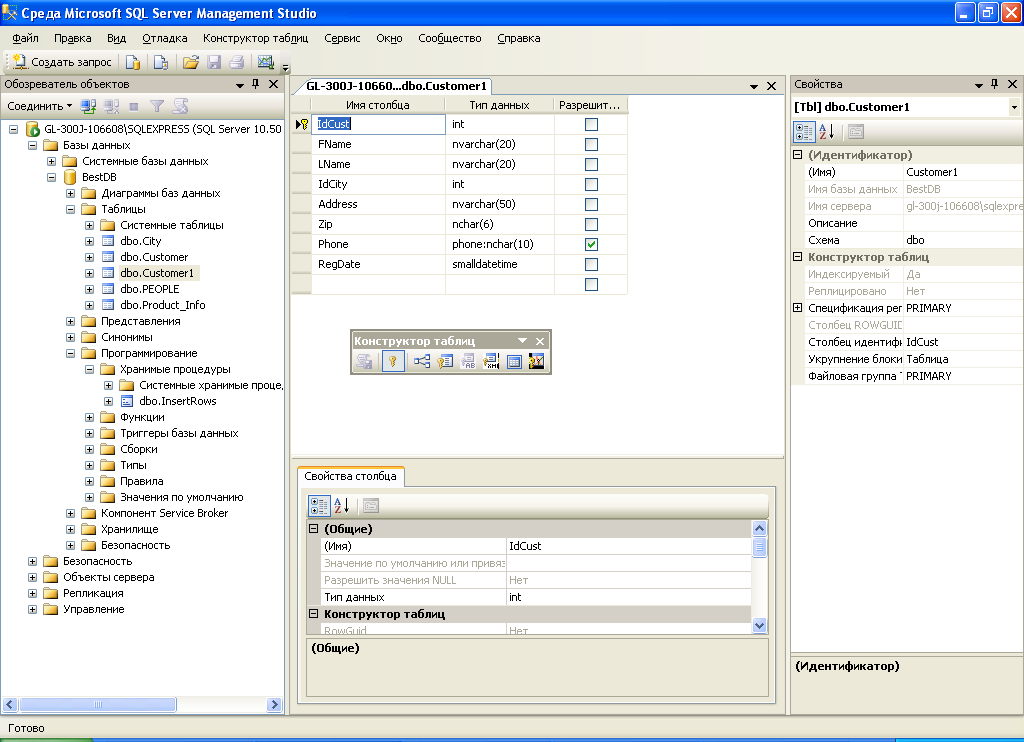
1. Назначение механизма отношений. Виды отношений?

Отношение (связи) в базе данных на SQL-сервере – это логическая связь между двумя таблицами. При установлении отношения между таблицами, мы информируем SQL-сервер, что первичный ключ одной таблицы связан с внешним ключом другой. Таким образом, необходимо иметь по одному ключу в каждой таблице. Отношения можно использовать и для того, чтобы накладывать ограничения целостности на вводимые данные. Данные в двух таблицах должны зависеть друг от друга и установив между этими таблицами отношение можно быть уверенными в том, что никакая SQL-команда не сможет нарушить условия этой зависимости. Ссылочная целостность основывается на идеи, что есть две таблицы, которые содержат повторяющуюся информацию, и что эти повторяющиеся элементы должны неукоснительно соответствовать друг другу. Например, если у нас есть первичный ключ в одной таблице, и внешний ключ в другой (при этом они содержат значения, которые в точности соответствуют друг другу), то очень важно, чтобы эти значения либо не менялись вообще, либо менялись одновременно и одинаково.

В базах данных существует три вида отношений: "один к одному", "один ко многим" и "многие ко многим". Отношения в реляционных базах данных чаще всего реализуются связкой "внешний ключ/первичный ключ", реже – связкой "внешний ключ/уникальный ключ". Отношение между двумя таблицами вида "многие ко многим" реализуется добавлением третьей связующей таблицы и двумя связками "внешний ключ/первичный ключ".

1. Способы создания отношений

Применение диалоговых средств Management Studio

1. В окне «Обозреватель объектов», найдите узел вашей базы данных и разверните его. Найдите и выделите таблицу (нужно выбрать таблицу которая будет содержать внешний ключ), затем щелкните на ней правой кнопкой мыши. Выберите команду «Проект», чтобы перейти в конструктор таблиц.
2. В конструкторе таблиц найдите на панели инструментов кнопку  «Отношения» и щелкните на ней. Того же результата можно достичь, щелкнув правой кнопкой мыши и выбрав в контекстном меню команду «Отношения…».
3. Откроется окно «Связи по внешнему ключу», в нем отображаются все отношения данной таблицы.
4. Для того, чтобы создать новое отношение щелкните на кнопке «Добавить».
5. Далее необходимо выбрать параметры отношения. В поле «Спецификация таблиц и столбцов» щелкните на кнопке с многоточием, появится окно «Таблицы и столбцы». Здесь в поле списка «Таблица первичного ключа» по умолчанию выбрана таблица, для которой создается отношение. Разверните список и выберите имя нужной вам таблицы. В качестве «Таблица внешнего ключа» выбрано имя таблицы, для которой создается отношение. При щелчке на «Добавить» было сгенерировано имя отношения, которое теперь изменилось в соответствии с выбранными таблицами.
6. Теперь перейдите к раскрывающемуся списку ниже списка с именем главной таблицы (Таблица первичного ключа), где перечислены все поля выбранной таблицы. Выберите столбец из первой таблицы.
7. В списке ниже списка «Таблица внешнего ключа» выберите среди столбцов второй таблицы столбец для внешнего ключа. Если бы для определения отношения требовалось использовать составной внешний ключ, то есть ключ из нескольких столбцов, то здесь следовало бы, перемещаясь, каждый раз на строку ниже, выбирать по очереди все необходимые столбцы. После этого нажмите кнопку «Ок», окно «Таблицы и столбцы» закроется.
8. Щелкните на кнопке «Закрыть» окна «Связи по внешнему ключу», чтобы продолжить работу. Закройте окно Конструктора, все внесенные изменения будут физически записаны в базу данных

Применение операторов Transact-SQL

Создать отношение или ограничение внешнего ключа в новой таблице возможно посредством команды CREATE TABLE, а в уже существующей таблице это делается с помощью команды ALTER TABLE.

1. Ссылочная целостность данных. Способы ее поддержания?

Данные в двух таблицах должны зависеть друг от друга и установив между этими таблицами отношение можно быть уверенными в том, что никакая SQL-команда не сможет нарушить условия этой зависимости. Ссылочная целостность основывается на идеи, что есть две таблицы, которые содержат повторяющуюся информацию, и что эти повторяющиеся элементы должны неукоснительно соответствовать друг другу. Например, если у нас есть первичный ключ в одной таблице, и внешний ключ в другой (при этом они содержат значения, которые в точности соответствуют друг другу), то очень важно, чтобы эти значения либо не менялись вообще, либо менялись одновременно и одинаково.

1. Способы отображения зависимостей между таблицами БД

Когда созданы отношения (связи) между таблицами, база данных достигла той точки, когда данные в одной таблице начинают зависеть от данных в другой таблице. SQL Server дает возможность увидеть, зависит ли некая таблица (или любой другой объект БД) от других или нет.

1. В окне «Обозреватель объектов» выделите одну из таблиц вашей базы данных, щелкните на ее значке правой кнопкой мыши и выберите команду «Просмотреть зависимости».
2. В результате откроется диалоговое окно, в котором будут показаны все таблицы, зависящие от выбранной таблицы и те, от которых зависит она сама.

Отображение зависимостей можно получить при помощи диаграммы базы данных. Диаграмма базы данных в простейшей форме отображает таблицы (с перечислением атрибутов этих таблиц) и отношения между таблицами.

Диаграмма базы данных – это графическое представление таблиц БД с визуальным отображением связей между таблицами по внешним ключам. Создадим диаграмму базы данных:

1. В контекстном меню папки «Диаграммы базы данных» выберите команду «Создать диаграмму базы данных».
2. В диалоговом окне «Добавление таблиц» выберите все таблицы и нажмите на кнопку «Добавить».
3. Добавив таблицы, щелкните на кнопке «Закрыть» и вы увидите созданную диаграмму базы данных.

Используя диаграмму базы данных ограничения внешнего ключа можно создавать значительно быстрее: лишь перетаскивая поля из одной таблицы в другую. В качестве примера создадим внешний ключ в таблице Customer по полю IdCity для связи с таблицей City:

1. Выделите в таблице City поле IdCity и, не отпуская кнопку мыши, перетащите его на поле IdCity таблицы Customer.
2. В диалоговых окнах «Таблицы и столбцы» и «Связь по внешнему ключу» примите настройки по умолчанию.
3. Сохраните диаграмму базы данных под именем ILM.
4. Способы ввода данных в таблицы. Ограничения целостности при вводе данных

Для добавления в таблицу (или в представление) одной или нескольких строк используется оператор INSERT. Синтаксис команды:

INSERT INTO <имя таблицы> [(<имя поля>.,..)]

VALUES (<список выражений>) | <запрос>;

Под <запросом> подразумевается команда SELECT, результаты работы которой добавляются в указанную таблицу.

В предложении VALUES указываются выражения, порождающие значения атрибутов новой записи таблицы. Выражение может включать вызовы функций, определенных в данной СУБД, константы, знак операций конкатенации (объединения) строк (||) или знаки арифметических операций: –, +, \*, /. Типы значений выражений должны соответствовать типам полей таблицы. Строки и даты должны заключаться в одинарные кавычки. Формат даты должен соответствовать тому, который установлен в СУБД по умолчанию.

Если значения устанавливаются не для всех полей или порядок значений не соответствует тому порядку полей, который был установлен при создании таблицы, то после имени таблицы в скобках приводится список полей в соответствии со списком значений.

В тех случаях, когда при добавлении записи значение какого-либо поля неизвестно, его можно не устанавливать, пропустив это поле в списке полей или указав для него значение NULL (но только для тех полей, на которые не наложено ограничение целостности NOT NULL).

Если в списке полей отсутствует какое-либо поле таблицы, то ему будет присвоено значение NULL или значение по умолчанию (DEFAULT), если оно определено при создании таблицы.

Если в таблице есть столбец, который описывается с атрибутом IDENTITY. При добавлении новой строки в эту таблицу система автоматически формирует уникальное значение для такого столбца. По этой причине не нужно явно задавать для него никакого значения.

1. Удаление данных, удаление таблиц

Удаление данных (оператор DELETE)

Оператор DELETE позволяет удалить одну или более строк одной таблицы или представления. Синтаксис этой команды:

DELETE FROM <имя таблицы>

[ WHERE <условие> ];

Эта команда удаляет из указанной таблицы те записи, которые удовлетворяют заданному условию отбора (WHERE <условие>).

Внимание! Если не указывать условие выбора записей, то все записи таблицы будут удалены без предупреждения и без запроса на подтверждение!

Удаление строк таблицы (оператор TRUNCATE TABLE)

Если в операторе DELETE не задано условие в предложении WHERE, то этот оператор удаляет все строки таблицы. Удалить все строки таблицы можно при использовании оператора TRUNCATE TABLE. При этом удаление происходит много быстрее. Синтаксис этой команды:

TRUNCATE TABLE

[[<имя базы данных>.]<имя схемы>.]<имя таблицы>;

Здесь есть только одна неприятность. Если на таблицу, которую вы хотите очистить быстрым способом, есть ссылки внешних ключей других таблиц, то попытка удаления будет завершена с ошибкой. Удаление не пройдет, даже если подчиненная таблица не имеет вообще никаких строк.

Для удаления таблицы используется команда drop <имя таблицы>