**Лабораторная работа № 2**

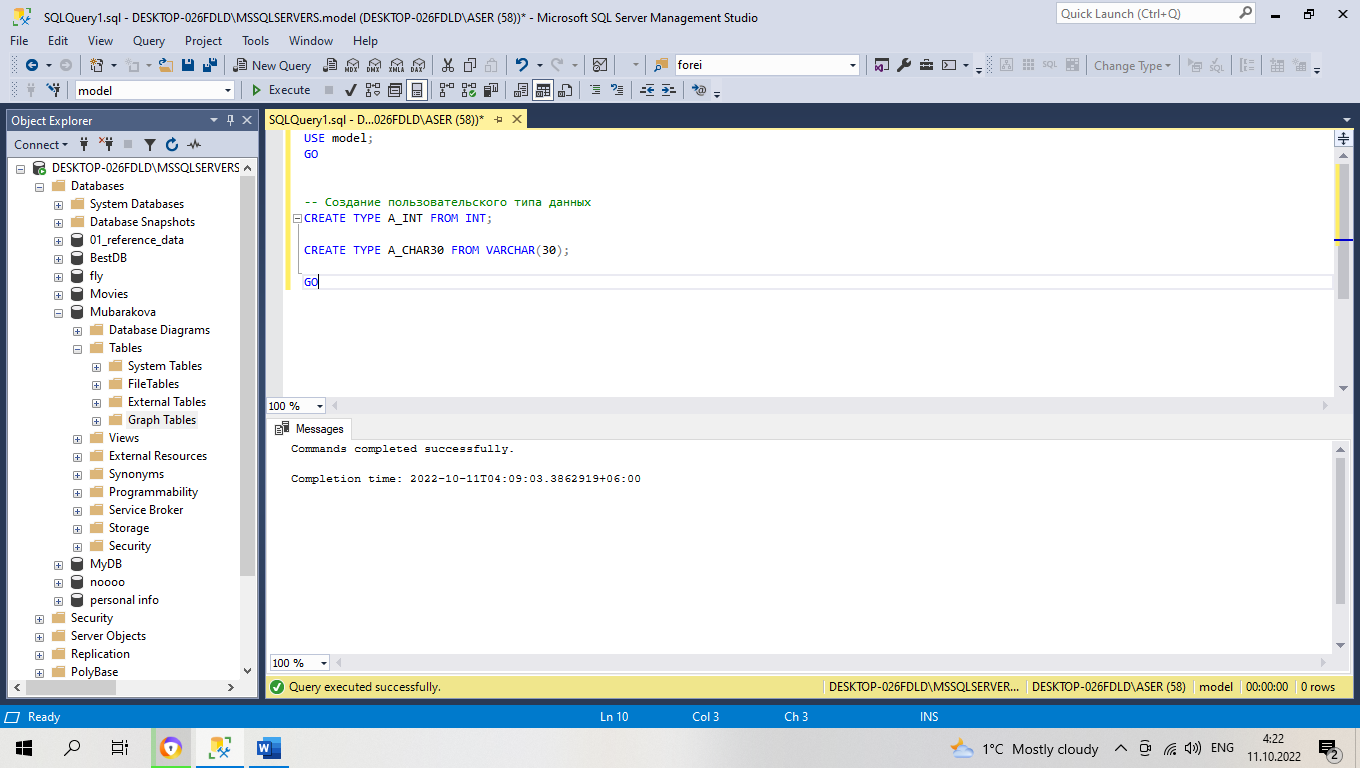
**Тема**: "Создание таблиц базы данных"

**Цель работы**: усвоить способы создания таблиц, пользовательских типов данных, умолчаний, ограничений и индексов базы данных средствами СУБД MS SQL Server 2012.

1. **Содержание отчета (в электронном виде):**
2. Название и цель работы
3. Индивидуальное задание
4. Скрипты на создание пользовательского типа данных, таблиц БД, индексов в соответствии с индивидуальным заданием
5. Ответы на контрольные вопросы

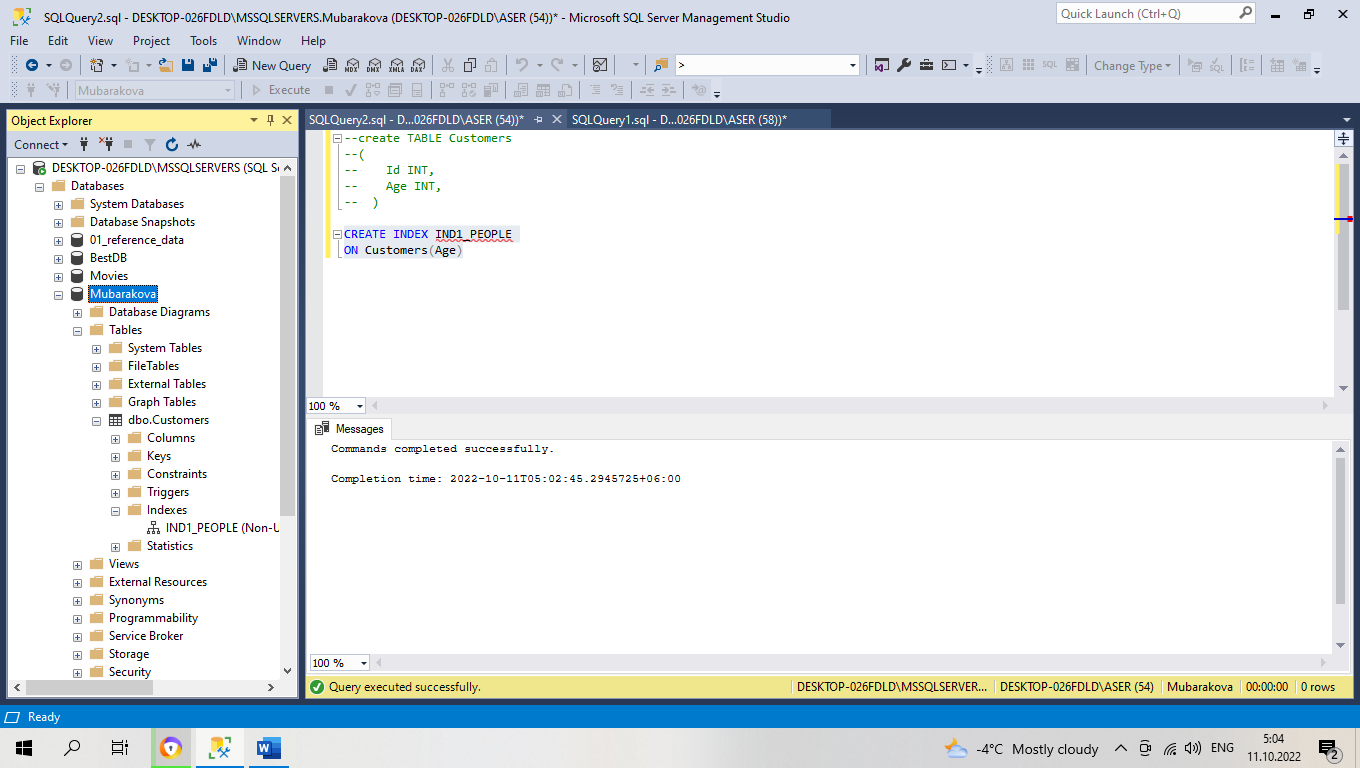
**Индивидуальное задание**

1. Создать пользовательский тип данных
2. Определить таблицы базы данных в соответствии с требованиями индивидуального варианта задания на разработку курсового проекта (при создании таблиц определите в них первичные и внешние ключи и при необходимости используйте значения по умолчанию, ограничения и правила и созданный вами в п. 1 пользовательский тип данных). ***Ввод данных в таблицы не осуществлять***!
3. Создать индексы для таблиц базы данных



CREATE INDEX IND1\_PEOPLE

ON Customers(Age)



**Контрольные вопросы**

1. Перечень поддерживаемых типов данных?

|  |  |  |
| --- | --- | --- |
| Тип данных | Описание | Сколько места занимает |
| bit | Целый тип данных, которые могут иметь значения 1, 0 или *NULL.* Колонки типа *bit* не могут иметь индексов.  Логический тип данных. Множество допустимых операций – это три логические побитовые операции: отрицание (~), дизъюнкция (|) и конъюнкция (&). Никакие арифметические операции для этого типа данных невозможны. | 1 байт для таблиц с 8-битными колонками |
| tinyint | Целочисленные данные в диапазоне от 0 до 255. | 1 байт |
| smallint | Целочисленные данные от -215 (-32768) до 215-1 (32767) | 2 байта |
| integer или int | Целочисленные данные от -231 (-2 147 483 648) до 231 -1 (2 147 483 647) | 4 байта |
| bigint | Восьмибайтное целое число от –263 (–9 223 372 036 854 775 808)  до 263-1 (9 223 372 036 854 775 807) | 8 байт |
| decimal (n[, m])  или  numeric (n[, m]) | Числа фиксированной точности и фиксированного масштаба. (Тип данных *numeric* является синонимом типа данных *decimal.)* При задании этого типа данных можно указать точность (параметр n) – общее количество десятичных цифр, включающее количество знаков целой и дробной части, – и масштаб (параметр m) – количество дробных цифр после десятичной точки. Можно вообще не указывать ни точности, ни масштаба, тогда им присваивается значение по умолчанию. Можно указать точность, не указывая масштаба, масштабу в этом случае также присваивается значение по умолчанию.  Точность может быть указана в диапазоне от 1 до 38. Если точность явно не задана, то по умолчанию устанавливается значение 18.  Масштаб может принимать значение в диапазоне от 0 до установленного (явно или неявно) значения точности. По умолчанию ему присваивается значение 0.  Максимальный диапазон от –1038+1 до 1038-1 | от 5 до 17 байт,  в зависимости  от точности |
| smallmoney | Данные для денежных величин от -214 748.3648 до 2 14 748.3647 (922 337 203 685 477.5807), с точностью до одной десятитысячной доли от денежной единицы | 4 байта |
| money | Данные для денежных величин от -263 (-922 337 203 685 477.5808) до 263-1 (922 337 203 685 477.5807), с точностью до одной десятитысячной доли от денежной единицы | 8 байт |
| float(n) | Числовые данные с плавающей точностью, которые могут находиться в диапазоне от -1.79x10 +308 до 1.79xl0+308. Параметр n определяет количество битов, применяемых для хранения мантиссы числа с плавающей точкой и может иметь значение от 1 до 53. Если  значение не указано, то принимается 53.  Если значение задано в диапазоне от 1 до 24, то для хранения числа отводится 4 байта. Количество значащих цифр будет 7.  При задании диапазона от 25 до 53 числу отводится 8 байтов. Количество значащих цифр – 15. | от 4 до 8 байт,  в зависимости  от точности |
| real | Числовые данные с плавающей точностью, которые могут находиться в диапазоне от -3.40x10+38 до 3.40х10+38. Синонимом для типа *real* является *float(24)* | 4 байта |
| char(n) | Символьные данные (не в кодировке Unicode) фиксированной длины, имеющие длину *n* символов, где *n* может принимать значение от 1 до 8000. | *n* байт (каждый символ занимает один байт.) |
| varchar (n) | Данные переменной длины не в кодировке Unicode, длиной в *п* символов, где *п* может принимать значение от 1 до 8000 | Факт. длина введенных данных |
| text | Применяется для символьных данных переменной длины не в кодировке Unicode, длиной более 8000 байт. Это - указатель на местоположение данных, которые хранятся отдельно от данных таблицы. Данный тип в ближайшее время будет удален из системы. | 16 байт для указателя |
| nchar (n) | Символьные данные в кодировке Unicode фиксированной длины, имеющие длину *п* символов, где *п* может принимать значение от 1 до 4000. Кодировка Unicode применяют по 2 байта на один символ данных и поддерживает все символы, имеющиеся в мире | 2 байта помно-жить на кол-во введенных символов |
| nvarchar | Данные в кодировке Unicode переменной длины, длиной до *n* символов, где *n* может принимать значение от 1 до 4000. Помните, что для одного символа в кодировке Unicode потребуется 2 байта; кодировка Unicode поддерживает все символы, имеющиеся в мире | 2 байта помножить на количество введенных символов |
| ntext | Данные в кодировке Unicode переменной длины, максимальной длиной до 230-1 (1 073 741 823) символов. Элемент данных, хранящийся в колонке типа *ntext,* является указателем на местоположение самих данных. Эти данные хранятся отдельно от данных в таблицах. Данный тип в ближайшее время будет удален из системы. | 16 байт на указатель и по 2 байта на каждый из символов,  введенных данных |
| datetime | Объединяет дату и время. Дата от 1 января 1753 года до 31 декабря 9999 года. Время в диапазоне от 00:00:00 до 23:59:59.997, с точностью 3.33 миллисекунды | 8 байт |
| smalldatetime | Объединяет дату и время. Дата от 1 января 1900 года до 6 июня  2079 года. Время в диапазоне от 00:00:00 до 23:59:59, с точностью до одной минуты (точность меньше, чем у типа данных *datatime).* | 4 байта |
| date | Дата от 1 января 0001 года до 31 декабря 9999 года. Формат отображения: гггг-мм-дд, имеет 10 символов | 3 байта |
| time | Время с высокой точностью. От 00:00:00.0000000 до 23:59:99.9999999.  TIME[(<масштаб>)] Здесь масштаб – число от 0 до 7. Масштаб задает количество дробных знаков секунд или, как сказано в документации, количество сотен наносекунд. Если масштаб не указан, то предполагается 7. | 3, 4 или 5 байтов |
| datetimeoffset (<точность>, <масштаб>) | Дата от 1 января 0001 года до 31 декабря 9999 года. Время от 00:00:00.0000000 до 23:59:99.9999999. Учитывает часовой пояс. <точность> – целое число от 26 до 34. Задает общее количество знаков; <масштаб> – число от 0 до 7. Задает количество дробных знаков секунд. Если точность и масштаб не указаны, то предполагается (34, 7). Помимо даты и времени этот тип данных также содержит величину смещения часового пояса – от -14:00 до +14:00. | 8, 9, 10 байт |
| datetime2 | является расширенным вариантом типа данных DATETIME. Дата от 1 января 0001 года до 31 декабря 9999 года. Время от 00:00:00.0000000 до 23:59:99.9999999 | 6, 7, 8 байт |
| binary[(n)] | Двоичные данные фиксированной длины, состоящие из *n* байтов, где *n* может принимать значение от 1 до 8000. Тип *binary* следует применять, когда предполагается, что элементы данных, хранимые в колонке, будут иметь примерно одинаковый размер | *n*+4 байта |
| varbinary | Двоичные данные переменной длины, состоящие из *п* байтов, где *п* может принимать значение от 1 до 8000. Применяйте тип varbinary, если предполагаете, что элементы данных, хранимые в колонке, будут сильно отличаться по своим размера | Фактическая  длина введенных данных плюс 4 байта |
| image | Применяется для двоичных данных переменной длины, большей, чем 8000 байт (максимально до 231-1 байт). Элемент данных, хранящийся в колонке типа *image,* является указателем на местоположение данных типа *image.* Эти данные хранятся отдельно от данных в таблицах. Тип данных *image* будет удален из системы в какой-нибудь из следующих версий. Вместо него следует использовать тип данных varbinary(max). | 16 байт для указателя |
| geometry | Этот тип данных представляет геометрические объекты на двумерной плоской поверхности. Характеристиками объекта являются его координаты в плоской системе координат, уровень, мера и идентификатор пространственной ссылки (SRID, Spatial Reference Identifier). |  |
| geography | Тип данных geography позволяет описывать объекты на земной поверхности. |  |
| sql\_variant | Является типом данных, который может хранить данные различных типов. Это могут быть целочисленные, строковые типы данных, пространственные, тип данных XML. В столбце таблицы или в переменной с этим типом данных помимо самих данных хранятся также и метаданные – данные, описывающие характеристики хранимых данных. |  |
| hierarchyid | используется для представления иерархических данных, т. е. для представления данных в иерархической, древовидной структуре. Здесь можно описывать иерархические отношения между данными в виде отношения родитель-потомок. Иерархия является весьма естественным способом организации данных, описывающих структуру различных предметных областей человеческой деятельности. |  |
| uniqueidentifier | Хранит 16-байтной двоичное значение, являющееся глобальным уникальным идентификатором (GUID) | 16 байт |
| cursor | Тип данных CURSOR хранит указатель на набор данных – т. е. на множество строк, полученных из таблицы (таблиц) базы данных в результате выполнения запросак базе данных. Этот тип данных нельзя использовать при описании столбцов таблиц. Интересно, что само слово "cursor", по крайней мере в этом контексте, было образовано из выражения CURrent Set Of Records (текущий набор записей).  Не следует путать этот термин с тем курсором, который связан с текущим положением указателя мыши на экране или с текущим символом в текстовом окне. |  |
| table | Тип данных TABLE позволяет временно хранить в табличном виде результаты обработки данных. Эти данные впоследствии могут быть прочитаны с целью дальнейшей обработки. Тип данных TABLE может быть использован в хранимых процедурах, в функциях и в обычных пакетах (скриптах). Его нельзя использовать при описании столбцов таблиц. | Зависит от определения таблицы |
| xml | Тип данных XML используется для хранения данных в формате XML. Данные в этом элементе должны быть корректными данными формата XML. Этот тип данных появился в SQL Server версии 2005. |  |
| timestamp | Колонка с типом timestamp («метка времени») автоматически обновляется всякий раз при вставке или при изменении строки. Каждая таблица может иметь только одну колонку типа timestamp | 8 байт |

1. Пользовательский тип данных. Назначение и способы создания?

Первый способ создания пользовательского типа данных в Management Studio в окне «Обозреватель объектов» щелкните правой кнопкой мыши по элементу «Определяемые пользователем типы данных» (который находится в папке «Программирование» | «Типы») и в контекстном меню выберите элемент «Создать определяемый пользователем тип данных…». Появится диалоговое окно.

Второй способ

USE model;

GO

-- Создание пользовательского типа данных

CREATE TYPE D\_INT FROM INT;

CREATE TYPE D\_CHAR30 FROM VARCHAR(30);

GO

1. Перечень поддерживаемых умолчаний, ограничений, правил?

Типы ограничений SQL

NOT NULL Constraint.

UNIQUE Constraint.

DEFAULT Constraint.

CHECK Constraint.

PRIMARY KEY Constraint.

FOREIGN KEY Constraint.

1. Способы создания ограничений?

CREATE TABLE Product\_Info

(Product\_ID smallint IDENTITY (1,1) NOT NULL,

Product\_Name char(20) NOT NULL,

Description char(30) NULL,

Price smallmoney NOT NULL,

Brand\_ID smallint NOT NULL,

CONSTRAINT CK\_Price CHECK (price>=1 AND price <=50000))

1. Назначение и классификация индексов?

Индексы в SQL — это отдельные таблицы поиска, которые используются поисковиком набора данных в Интернете для ускорения восстановления общей информации. Использование индекса в SQL заключается в быстром обнаружении данных в таблице набора данных без просмотра каждой ее строки.

Индексы используются для быстрого поиска данных без необходимости искать каждую строку в таблице базы данных при каждом доступе к таблице базы данных. Индексы могут быть созданы с использованием одного или нескольких столбцов таблицы базы данных, обеспечивая основу как для быстрого случайного поиска, так и для эффективного доступа к упорядоченным записям.

Индексы на основе выражений эффективно оценивают запросы с индексированным выражением.

Уникальные и неуникальные индексы. ...

Кластеризованные и некластеризованные индексы. ...

Разделенные и неразделенные индексы. ...

Двунаправленные индексы. ...

Индексы на основе выражений.

Кластеризованный индекс — это тип индекса, в котором записи таблицы физически переупорядочены в соответствии с индексом. Некластеризованный индекс — это особый тип индекса, в котором логический порядок индекса не соответствует физическому хранящемуся порядку строк на диске. Кластерные индексы сортируют и сохраняют строки данных в таблице или представлении на основе значений их ключей. Это столбцы, включенные в определение индекса. Для каждой таблицы может быть только один кластеризованный индекс, потому что сами строки данных могут храниться только в одном порядке.

1. Способы создания индексов?

create index index\_name  
on table\_name(column1, column2, .., columnN);

1. Что представляют собой Null-значение?

NULL соответствует понятию «пустое поле», то есть «поле, не содержащее никакого значения».

1. Типы целостности данных?

Каждый тип целостности данных – **доменный, табличный и ссылочный** обеспечиваются отдельным типом ограничений. Ограничения обеспечивают правильность данных введенных в поле, и какие отношения обеспечиваются между таблицами.