|  |
| --- |
| Карагандинский технический университет имени Абылкаса Сагинова  Кафедра ИВС  **Лабораторная работа №2**  По дисциплине: "Системы управления базами данных"  *(наименование дисциплины)*  Тема: Создание таблиц базы данных    **Преподаватель:**  Нуртай М.Д.  *(фамилия, инициалы)*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(подпись) (дата)*  **Выполнил:**  ст.гр. ITM-20-1 Щербаков И.С.  *(фамилия, инициалы)*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(подпись) (дата)*  Караганды 2022 |

**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.

**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

**Результат выполнения индивидуального задания:**

Скрипт создания таблицы Roles:

use StudScherbakov

CREATE TABLE Roles

(

Id INT primary key,

RoleName NVARCHAR (20),

)

CREATE TABLE Users

(

Id INT primary key,

FirstName NVARCHAR(20),

LastName NVARCHAR(20),

RoleId INT unique, foreign key(RoleId) references Roles(Id),

)

insert into Roles (Id,RoleName) values (1, 'Admin');

insert into Roles (Id,RoleName) values (2, 'User');

insert into Roles (Id,RoleName) values (3, 'QA');

insert into Users (Id,FirstName, LastName, RoleId) values (1, 'Anton', 'Vladimirov', 2);

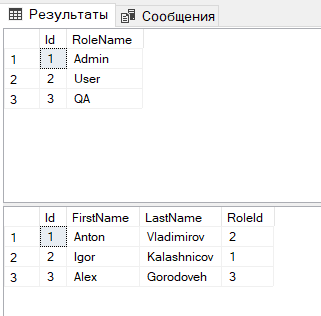
insert into Users (Id,FirstName, LastName, RoleId) values (2, 'Igor', 'Kalashnicov', 1);

insert into Users (Id,FirstName, LastName, RoleId) values (3, 'Alex', 'Gorodoveh', 3);

select \* from Roles

select\* from Users

Результат работы кода



**Ответы на контрольные вопросы:**

1. **Перечень поддерживаемых типов данных?**

***СНAR*** – фиксированные текстовые строки до 2000 байт. Значение типа *CHAR* дополняется до указанной длины пробелами.

***VARCHAR 2*** — текстовые строки переменной длины до 4000 байт.

***NUMBER*** — числовые данные.

***DECIMAL*** — числовые данные

***DATE*** — используется для хранения дат.

***RAW*** — используется для хранения двоичных данных до 2000 байт.

***LONG*** — используется для хранения текстовых данных длиной до 2 ГБ

***LONG RAW*** — используется для хранения двоичных данных до 2 ГБ

***ROWID*** — используется для хранения идентификаторов **ROWID** базы данных **Oracle** в специальном формате (адреса строк таблицы).

***BLOB*** — сохраняется до 4 ГБ двоичных данных. Данные этого типа хранятся вне таблицы, а в таблице **Oracle** находятся лишь указатели на объекты

***CLOB***, ***NCLOB*** — сохраняется до 4 ГБ текстовых данных. *NCLOB* – это тип данных *NLS* большой фиксированной длины (*NLS* означает *National Language Set* – набор для национальных языков – и используется для работы в **Oracle** на языках, отличных от английского. В английском для хранения одного символа нужен 1 байт, а в некоторых языках мира с наборами больших символов (японском, китайском, корейском), языках, где текст читается справа налево (арабский, иврит) для хранения одного символа требуется несколько байт). Данные этого типа хранятся вне таблицы, а в таблице находятся лишь указатели на объекты.

***BFILE*** — сохраняется до 4 ГБ неструктурированных данных, причем в файлах операционной системы (внешние файлы).

1. **Пользовательский тип данных. Назначение и способы создания?**

В SQL Server 2005 (9.x) представлены определяемые пользователем типы данных (UDT). Пользовательские типы расширяют систему типов SQL путем разрешения хранения объектов и пользовательских структур данных в базе данных SQL Server. Определяемые пользователем типы могут содержать несколько типов данных, и их поведение может отличаться от традиционных псевдонимов типов данных, которые состоят из одного системного типа данных SQL Server. Определяемые пользователем типы определяются с помощью любого языка, поддерживаемого средой .NET CLR, который создает поддающийся проверке код. Сюда входят языки Microsoft Visual C# и Visual Basic .NET. Данные представляются в виде полей и свойств класса или структуры .NET, а поведения определяются методами класса или структуры.

Пользовательские типы (UDT) можно применять в качестве идентификатора столбцов таблицы, как переменную в пакете Transact-SQL или как аргумент функции Transact-SQL либо хранимой процедуры.

1. **Способы определения таблиц?**

Подход к определению схемы в SQL/89 состоит в том, что все таблицы с одним идентификатором полномочий создаются (определяются) путем выполнения одного оператора определения схемы. При этом в стандарте не определяется способ выполнения оператора определения схемы: должен ли он выполняться только в интерактивном режиме или может быть встроен в программу, написанную на традиционном языке программирования.

1. **Назначение ограничений?**

SQL-ограничения используются для указания правил данных таблицы, для ограничения типа данных, которые могут входить в таблицу. Это обеспечивает точность и надежность данных в таблице. Если между ограничением и действием данных существует какое-либо нарушение, действие прерывается.

1. **Перечень поддерживаемых умолчаний, ограничений, правил?**

Умолчания, ограничения и правила *— это* необязательные атрибуты, которые мож­но определять по колонкам и таблицам базы данных. У*молчания*(зна­чения по умолчанию) — это значения, которые заносятся в определенную колонку. *Ограничения*(constraints) используются как способ идентифицирования допустимых значений для колонки (чтобы откло­нять недопустимые значения), а также как средство обеспечения целостности дан­ных в таблицах базы данных и между связанными таблицами. Ограничение только по одной колонке называется ограничением на значение (колонки): оно ограничивает значения только этой колонки. Ограничение, которое влияет на несколько колонок, называется ссылочным ограничением: в этом случае комбинация значений для колонок, указанных в данном ограничении, должна отвечать требованиям этого ограни­чения. Имеется пять типов ограничений: NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY и CHECK.

1. **Способы создания ограничений?**

В обозревателе объектов разверните таблицу, в которую необходимо добавить проверочное ограничение, щелкните правой кнопкой мыши пункт Ограничения и выберите команду Создать ограничение.

1. **Назначение и классификация индексов?**

Словно указатель в грамотно составленной книге, индекс помогает быстро получить доступ к строкам требуемых данных в таблице, соответствующих запросу. Таким образом, их использование позволяет ускорить выполнение требуемого запроса.

B-Tree. Семейство B-Tree индексов — это наиболее часто используемый тип индексов, организованных как сбалансированное дерево, упорядоченных ключей.

Пространственные индексы

HASH.

Bitmap.

Reverse index.

Inverted index.

Partial index.

Function-based index.

1. **Способы создания индексов?**

Для того, чтобы добавить индекс, нам необходимо использовать команду CREATE INDEX, что позволит нам указать имя индекса и определить таблицу и колонку или индекс колонки и определить используется ли индекс по возрастанию или по убыванию. Индекса также могут быть уникальными, так же, как и констрейнт UNIQUE.

1. **Что представляют собой Null-значение?**

NULL в СУБД — специальное значение (псевдо значение), которое может быть записано в поле таблицы базы данных (БД). 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.