



SQL

Базовый уровень

DDL операции

Создание таблицы

```
CREATE TABLE [schema_name .] table_name (  
    column_name <data_type>  
        [ NULL | NOT NULL ]  
        [ CONSTRAINT constraint_name ]  
        [ DEFAULT constant_expression ] |  
        [ IDENTITY],
```

```
    ...  
)
```

```
[ CONSTRAINT constraint_name ] {  
    { PRIMARY KEY | UNIQUE }  
    [ FOREIGN KEY ] REFERENCES ref_tab_nme ( ref_column ) ]  
    [ ON DELETE { NO ACTION | CASCADE | SET NULL | SET DEFAULT } ]  
    [ ON UPDATE { NO ACTION | CASCADE | SET NULL | SET DEFAULT } ]  
    CHECK [ NOT FOR REPLICATION ] ( logical_expression )  
}
```



CONSTRAINT

- Необязательное ключевое слово, указывающее на начало определения ограничения PRIMARY KEY, NOT NULL, UNIQUE, FOREIGN KEY или CHECK.



NULL | NOT NULL

- Определяет, допустимы ли для столбца значения NULL.



PRIMARY KEY

- Ограничение, которое обеспечивает целостность сущностей для указанного столбца (столбцов) с помощью уникального индекса. Можно создать только одно ограничение PRIMARY KEY для таблицы.



UNIQUE

- Ограничение, которое обеспечивает целостность сущностей для указанного столбца (столбцов) с помощью уникального индекса. В таблице может быть несколько ограничений UNIQUE.

FOREIGN KEY REFERENCES

- Ограничение, которое обеспечивает ссылочную целостность данных в столбце (столбцах)
- NO ACTION - формируется ошибка, выполняется откат операции удаления строки из родительской таблицы.
- CASCADE - если из родительской таблицы удаляется строка, соответствующие ей строки удаляются и из ссылающейся таблицы.
- SET NULL - все значения, составляющие внешний ключ, при удалении соответствующей строки родительской таблицы устанавливаются в NULL.
- SET DEFAULT - все значения, составляющие внешний ключ, при удалении соответствующей строки родительской таблицы устанавливаются в значение по умолчанию.

--создание таблицы

```
CREATE TABLE HumanResources.TestDepartment (
```

```
    DepartmentID smallint IDENTITY(1,1) NOT NULL,
```

```
    Name varchar(200) NOT NULL,
```

```
    GroupName  varchar(200) NULL,
```

```
    ModifiedDate datetime NOT NULL DEFAULT (getdate())
```

```
    CONSTRAINT PK_TestDepartment_DepartmentID  
        PRIMARY KEY (DepartmentID)
```

```
)
```

Автоинкремент

По умолчанию –
текущая дата

Первичный ключ

Временные таблицы

- Можно создавать локальные и глобальные временные таблицы.
 - Локальные временные таблицы видимы только во время текущей сессии, объявляются как `#table_name`
 - глобальные - во всех сессиях, объявляются как `##table_name`
- Временные таблицы автоматически удаляются при выходе за пределы области определения, если не удалять их явно с помощью инструкции DROP TABLE.

Временные таблицы

--ЛОКАЛЬНАЯ

```
CREATE TABLE #MyTempTable (ID INT PRIMARY KEY);  
INSERT INTO #MyTempTable VALUES (1);  
select * from #MyTempTable;
```



Будет работать только
в этой сессии

--глобальная

```
CREATE TABLE ##MyGTempTable (ID INT PRIMARY KEY);  
INSERT INTO ##MyGTempTable VALUES (1);  
select * from ##MyGTempTable;
```


Работа с переменными

Объявление переменных в языке

- Инструкция DECLARE инициализирует переменную следующим образом:
 - Назначение имени. Первым символом имени должен быть одиночный символ @.
 - Назначение длины и типа данных, определяемого системой или пользователем.
 - Присваивает созданной переменной значение NULL.

-- *объявление переменной*

```
DECLARE @MyCounter int;
```

--*объявление нескольких переменных*

```
DECLARE @LastName nvarchar(30),  
        @FirstName nvarchar(20),  
        @StateProvince nchar(2);
```

Присвоение значения

- Чтобы изменить значение переменной можно использовать:

- инструкцию SET

SET @FirstNameVariable = 'Amy';

- указав её в списке выбора инструкции SELECT.

DECLARE @EmpIDVariable int;

SELECT *@EmpIDVariable = MAX(BusinessEntityID)*
FROM HumanResources.Employee;

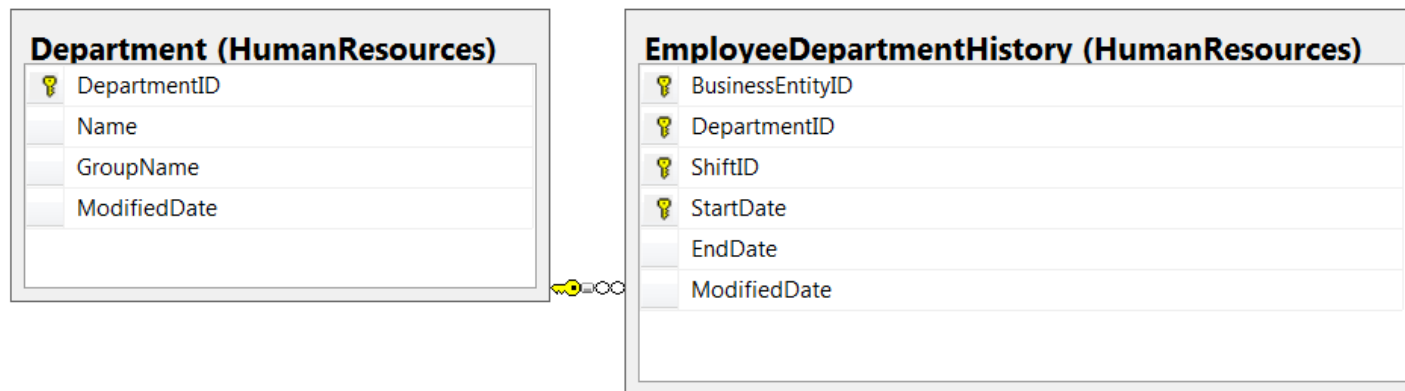
Использование переменных

```
DECLARE @find varchar(30);  
SET @find = 'Man%';
```

```
SELECT p.LastName,  
       p.FirstName,  
       ph.PhoneNumber  
FROM Person.Person p JOIN Person.PersonPhone ph  
     ON p.BusinessEntityID = ph.BusinessEntityID  
WHERE LastName LIKE @find;
```

Задание

- Найти отделы с максимальным числом сотрудников:
 - В переменную запомнить максимальный размер
 - Вывести отделы, размер которых равен переменной



Database access

Роли SQL Server

Имя роли уровня базы данных	Описание
db_owner	могут выполнять все действия по настройке и обслуживанию базы данных, а также удалять базу данных.
db_securityadmin	могут изменять членство в роли и управлять разрешениями. Добавление участников к этой роли может привести к непреднамеренному повышению прав доступа.
db_accessadmin	могут добавлять или удалять права удалённого доступа к базе данных для имён входа и групп Windows, а также имён входа SQL Server.
db_backupoperator	могут создавать резервные копии базы данных.
db_ddladmin	могут выполнять любые команды языка определения данных (DDL) в базе данных.
db_datawriter	могут добавлять, удалять или изменять данные во всех пользовательских таблицах.
db_datareader	могут считывать все данные из всех пользовательских таблиц.
db_denydatawriter	не могут добавлять, изменять или удалять данные в пользовательских таблицах базы данных.
db_denydatareader	не могут считывать данные из пользовательских таблиц базы данных.

Права пользователя

- Создание пользователя базы данных

--создаём нового пользователя Test

use master;

CREATE LOGIN Test

WITH PASSWORD = '1234\$NoPassW0rd';

USE AdventureWorks2012;

CREATE USER Test FOR LOGIN Test;

Управление разрешениями (permission)

- Разрешение доступа к объекту

GRANT <permission>

ON [OBJECT ::] object_name

TO Database_user

- Запрет доступа к объекту

DENY <permission>

ON [OBJECT ::] object_name

TO Database_user

Пример

--выдача прав

```
use AdventureWorks2012;  
GRANT select on  
OBJECT::HumanResources.Employee to Test;
```

--запрет права

```
USE AdventureWorks2012;  
DENY SELECT ON OBJECT::Person.Address TO Test;
```