

Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5

# DATA DEFINITION LANGUAGE OPERATORS (DDL) IN STRUCTURED QUERY LANGUAGE

- 1. Introduction to Introduction to SQL
- 1.1 Data Definition Language (DDL)
- 1.2 Data Manipulation Language (DML)
- 1.3 Data Query Language (DQL)
- 2. Introduction to SQL Server
- 2.1 SQL Server Management Studio
- 2.2 Create a new Database
- $2.3 \ {\rm The \ most \ important \ } {\rm Queries \ in \ SQL}$
- 3. CREATE TABLE
  - 3.1 Database Modelling
  - 3.2 Create Tables using the Designer Tools
  - 3.3 SQL Constraints
    - 3.3.1 PRIMARY KEY
    - 3.3.2 FOREIGN KEY
    - 3.3.3 NOT NULL / Required Columns
    - 3.3.4 UNIQUE
    - 3.3.5 CHECK
    - 3.3.6 DEFAULT
    - 3.3.7 AUTO INCREMENT or IDENTITY
- 4. The INSERT INTO
- 5. The UPDATE
- 6. The DELETE
- 7. The SELECT
- 8. Summary



Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5

# Introduction to 'SQL' in Database Design and Management

SQL (Structured Query Language) is a database computer language designed for managing data in relational database management systems (RDBMS). SQL, is a standardized computer language that was originally developed by IBM for querying, altering and defining relational databases, using declarative statements.

# SQL – Structured Query language

A Database Computer Language designed for Managing Data in Relational Database Management Systems (RDBMS)

Query Examples:

insert into STUDENT (Name , Number, SchoolId) values ('John Smith', '100005', 1)
select SchoolId, Name from SCHOOL
select \* from SCHOOL where SchoolId > 100
update STUDENT set Name='John Wayne' where StudentId=2
delete from STUDENT where SchoolId=3

We have 4 different Query Types: INSERT, SELECT, UPDATE and DELETE





Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5

## What can SQL do?

- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database
- SQL can create new databases
- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database
- SQL can set permissions on tables, procedures, and views

Even if SQL is a standard, many of the database systems that exist today implement their own version of the SQL language. In this document we will use the Microsoft SQL Server as an example.



Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5

There are lots of different database systems, or DBMS – Database Management Systems, such as:

- Microsoft SQL Server
- o Enterprise, Developer versions, etc.
- o Express version is free of charge
- Oracle
- MySQL (Oracle, previously Sun Microsystems) MySQL can be used free of charge

(open source license), Web sites that use MySQL: YouTube, Wikipedia, Facebook

- Microsoft Access
- IBM DB2
- Sybase





Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5

# **SQL Process**

When you are executing an SQL command for any RDBMS, the system determines the best way to carry out your request and SQL engine figures out how to interpret the task. There are various components included in this process.

These components are :

- Query Dispatcher
- Optimization Engines
- Classic Query Engine
- SQL Query Engine, etc.

A classic query engine handles all the non-SQL queries, but a SQL query engine won't handle logical files.

Following is a simple diagram showing the SQL Architecture:





# SQL Commands

The standard SQL commands to interact with relational databases are CREATE, SELECT, INSERT, UPDATE, DELETE and DROP. These commands can be classified into the following groups based on their nature:

# 1.1 Data Definition Language (DDL):

The Data Definition Language (DDL) manages table and index structure. The most basic

items of DDL are the CREATE, ALTER, RENAME and DROP statements:

• CREATE creates an object (a table, for example) in the database.

• DROP deletes an object in the database, usually irretrievably.

• ALTER modifies the structure an existing object in various ways-for example, adding

a column to an existing table.

# 1.2 Data Manipulation Language (DML):

The Data Manipulation Language (DML) is the subset of SQL used to add, update and delete data.

The acronym CRUD refers to all of the major functions that need to be implemented in a

relational database application to consider it complete. Each letter in the acronym can be

mapped to a standard SQL statement:

Operation	SQL	Description
Create	INSERT INTO	inserts new data into a database
Read (Retrieve)	SELECT	extracts data from a database
Update	UPDATE	updates data in a database
Delete (Destroy)	DELETE	deletes data from a database

# 1.3 DQL (Data Query Language):

# • SELECT



Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5

## Introduction to SQL Server

Microsoft is the vendor of SQL Server. We have different editions of SQL Server, where SQL Server Express is free to download and use.

SQL Server uses T-SQL (Transact-SQL). T-SQL is Microsoft's proprietary extension to SQL. TSQL is very similar to standard SQL, but in addition it supports some extra functionality, builtin functions, etc. T-SQL expands on the SQL standard to include procedural programming, local variables, various support functions for string processing, date processing, mathematics, etc.

SQL Server consists of a **Database Engine** and a **Management Studio** (and lots of other stuff which we will not mention here). The Database engine has no graphical interface - it is just a service running in the background of your computer (preferable on the server). The Management Studio is graphical tool for configuring and viewing the information in the database. It can be installed on the server or on the client (or both).



A Graphical User Interface to the database used for configuration and management of the database

7



Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5

# **SQL Server Management Studio**

SQL Server Management Studio is a GUI tool included with SQL Server for configuring, managing, and administering all components within Microsoft SQL Server. The tool includes both script editors and graphical tools that work with objects and features of the server. As mentioned earlier, version of SQL Server Management Studio is also available for SQL Server Express Edition, for which it is known as SQL Server Management Studio Express. A central feature of SQL Server Management Studio is the Object Explorer, which allows the user to browse, select, and act upon any of the objects within the server. It can be used to visually observe and analyze query plans and optimize the database performance, among others. SQL Server Management Studio can also be used to create a new database, alter any existing database schema by adding or modifying tables and indexes, or analyze performance. It includes the query windows which provide a GUI based interface to write and execute queries.

PCR2225/DDVELOPMENT ACL Server         PCR2225/DDVELOPMENT ACL         PCR225/DDVELOPMENT ACL <t< th=""><th>Vour SOI Server</th><th>ecute &gt; = &lt; 10 = 11 4 0 0 0</th><th>0 <u>7 2</u> # # K <u>}</u></th><th>- ×</th><th>Properties - 2 36</th></t<>	Vour SOI Server	ecute > = < 10 = 11 4 0 0 0	0 <u>7 2</u> # # K <u>}</u>	- ×	Properties - 2 36
Becunity     Becunity     B TEST     B WEATHERDATA	PCB225\DEVELOPMENT AQL Serve - Dotellarses Constituee Constitue	ASE Write your Schoold School United School	Address Phone PastCode Address Phone PastCode USA NULL NULL USA NULL NULL school Teordhem NULL NULL ol Osio NULL NULL	FinitAddman NULL NULL NULL NULL	Current connection parameter •  Current connection parameter •  Parame
a Generativ	B Security     B Security     B SECURITY     B SECURITY     B SECURITY	4 The result fr	om your Query		Seminn Trece SPID S2

When creating SQL commands and queries, the "Query Editor" (select "New Query" from the Toolbar) is used (shown in the figure above).



Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5

With SQL and the "Query Editor" we can do almost everything with code, but sometimes it is also a good idea to use the different Designer tools in SQL to help us do the work without coding (so much).

# 1.1 Create a new Database

It is quite simple to create a new database in Microsoft SQL Server. Just right-click on the "Databases" node and select "New Database..."



There are lots of settings you may set regarding your database, but the only information you must fill in is the name of your database:



Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5

🚪 New Database					
Select a page	🕵 Script 🝷 🚯	Help			
General					
Filegroups	Database name:				
_	Owner:		<default></default>		
	VUse full-text in	ndexing			
	Database files:				
	Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth
		Rows Data	PRIMARY	3	By 1 MB, unrestricted growth
	_log	Log	Not Applicable	1	By 10 percent, unrestricted gr
Caracter					
Connection					
Server: PC88235					
Connection:					
sa					
View connection properties					
Progress					
Ready	<	1111			<u>&gt;</u>
A <sup>40</sup> A				Add	Remove
					OK Cancel

You may also use the SQL language to create a new database, but sometimes it is easier to just use the built-in features in the Management Studio.



Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5

# 1.2 Queries

In order to make a new SQL query, select the "New Query" button from the Toolbar.

Se Microsoft SQL Server Management Studio										
Trans Query LING - I Desug Tools Window -	Connunity Help	0	003	19 ar	æ. M.,					
Celect Explorer + 9 X	SQLQuery1.sq	DNG (sa (53))*	Object Explo	rer Details		~				• X
Correct • 11 12 = 7 3	select ?	from CUSTONE	<b>.</b> P							
Book State State Control (Control (Contro) (Control (Control (Contro) (Contro) (Control (Contro) (Contro)						Yau write QL Code	e here			
	Contract (									
	Distant	Messages	THE SHEEP CONTRACTOR	Tethere	- Area Taller	- Addison	Disco 1	1.144		
	1	1000	Sinth	doba.	12	California	11111111	(3)	0	
	2 2	1001	Jackion	Smith	45	London	22222222	9	Your results w	
	3 3	1002	Johnsen	John	22	London	333333333	-	appear in this window	
	🔮 Query evecute	d successifully				1	09823510EVELO	484 (16.0 991)	sa (53) INVOICING	(0.00.00 3 mms
Reaty								Lint Celt	23 (5-23	245

Here we can write any kind of queries that is supported by the SQL language.



# **1.3 CREATE TABLE**

Before you start implementing your tables in the database, you should always spend some time design your tables properly using a design tool like, e.g., ERwin, Toad Data Modeler, PowerDesigner, Visio, etc. This is called Database Modeling.

# Database Design – ER Diagram

ER Diagram (Entity-Relationship Diagram)

- · Used for Design and Modeling of Databases.
- Specify Tables and <u>relationship</u> between them (Primary Keys and Foreign Keys)
   Table Name



Relational Database. In a relational database all the tables have one or more relation with each other using Primary Keys (PK) and Foreign Keys (FK). Note! You can only have one PK in a table, but you may have several FK's.

The CREATE TABLE statement is used to create a table in a database.

# Syntax:

```
CREATE TABLE table_name
(
  column_name1 data_type,
  column_name2 data_type,
  column_name3 data_type,
  ....
)
```



Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5

### The data type specifies what type of data the column can hold.

You have special data types for numbers, text dates, etc. Examples:

- Numbers: int, float
- Text/Stings: varchar(X) where X is the length of the string
- Dates: datetime
- etc.

### Example:

We want to create a table called "CUSTOMER" which has the following columns and data types:

	Column Name	Data Type	Allow Nulls
₽₿	CustomerId	int	
	CustomerNumber	int	
	LastName	varchar(50)	
	FirstName	varchar(50)	
	AreaCode	int	<ul> <li>Image: A set of the set of the</li></ul>
	Address	varchar(50)	<ul> <li>Image: A set of the set of the</li></ul>
	Phone	varchar(20)	<ul> <li>Image: A set of the set of the</li></ul>

```
CREATE TABLE CUSTOMER
(
CustomerId int IDENTITY(1,1) PRIMARY KEY,
CustomerNumber int NOT NULL UNIQUE,
LastName varchar(50) NOT NULL,
FirstName varchar(50) NOT NULL,
AreaCode int NULL,
Address varchar(50) NULL,
Phone varchar(50) NULL,
)
```



### **Best practice:**

When creating tables you should consider following these guidelines:

• Tables: Use upper case and singular form in table names - not plural, e.g.,

"STUDENT" (not students)

- Columns: Use Pascal notation, e.g., "StudentId"
- Primary Key:
  - If the table name is "COURSE", name the Primary Key column "CourseId", etc.
  - "Always" use Integer and Identity(1,1) for Primary Keys. Use UNIQUE constraint for other columns that needs to be unique, e.g. RoomNumber
- Specify Required Columns (NOT NULL) -

i.e., which columns that need to have data or not

- Standardize on few/these Data Types: int, float, varchar(x), datetime, bit
- Use English for table and column names
- Avoid abbreviations! (Use RoomNumber not RoomNo, RoomNr, ...)

# 2.1 Database Modelling

As mention in the beginning of the lecture, you should always start with database modelling before you start implementing the tables in a database system. Below we see a database model in created with ERwin.



Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5



With this tool we can transfer the database model as tables into different database systems, such as e.g., SQL Server. CA ERwin Data Modeler Community Edition is free with a 25 objects

limit. It has support for Oracle, SQL Server, MySQL, ODBC and Sybase.

Below we see the same tables inside the design tool in SQL Server.



Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5





Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5

# Microsoft SQL Server – Tips and Tricks



#### 2.2 Create Tables using the Designer Tools

Even if you can do "everything" using the SQL language, it is sometimes easier to do it in the designer tools in the Management Studio in SQL Server.

Instead of creating a script you may as well easily use the designer for creating tables.

Step1: Select "New Table ...":



Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5



Step2: Next, the table designer pops up where you can add columns, data types, etc.

	Column Name	Data Type	Allow Nulls
۲Ÿ	CustomerId	int	
	CustomerNumber	int	
	LastName	varchar(50)	
	FirstName	varchar(50)	
	AreaCode	int	<ul> <li>Image: A set of the set of the</li></ul>
	Address	varchar(50)	<ul> <li>Image: A set of the set of the</li></ul>
	Phone	varchar(20)	<ul> <li>Image: A set of the set of the</li></ul>

In this designer we may also specify Column Names, Data Types, etc.

**Step 3:** Save the table by clicking the Save button.



Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5

# 2.3 SQL Constraints

Constraints are used to limit the type of data that can go into a table.

Constraints can be specified when a table is created (with the CREATE TABLE statement) or after the table is created (with the ALTER TABLE statement).

Here are the most important constraints:

- PRIMARY KEY
- NOT NULL
- UNIQUE
- FOREIGN KEY
- CHECK
- DEFAULT
- IDENTITY

In the sections below we will explain some of these in detail.

### 2.3.1 PRIMARY KEY

The PRIMARY KEY constraint uniquely identifies each record in a database table.

Primary keys must contain unique values. It is normal to just use running numbers, like 1, 2, 3, 4, 5, ... as values in Primary Key column. It is a good idea to let the system handle this for you by specifying that the Primary Key should be set to **identity(1,1)**. IDENTITY(1,1) means the first value will be 1 and then it will increment by 1.

Each table should have a primary key, and each table can have only ONE primary key.

If we take a closer look at the CUSTOMER table created earlier:

```
CREATE TABLE [CUSTOMER]
(
CustomerId int IDENTITY(1,1) PRIMARY KEY,
CustomerNumber int NOT NULL UNIQUE,
LastName varchar(50) NOT NULL,
FirstName varchar(50) NOT NULL,
AreaCode int NULL,
Address varchar(50) NULL,
Phone varchar(50) NULL,
)
```



Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5

As you see we use the "Primary Key" keyword to specify that a column should be the

### Primary Key.

	CustomerId	CustomerNum	ber   LastName   FirstName   AreaCode   Address	Phone
•	1	1000	Primary Keys must contain unique	1111111
	2	1001	numbers like this	2222222
	3 3	1002	Connison Conni 32 London	

# Setting Primary Keys in the Designer Tools:

If you use the Designer tools in SQL Server, you can easily set the primary Key in a table just by right-click and select "Set primary Key".

PC88235	5\DEVELOPEST - dbo.S0	CHOOL* 🗙 Object Exp	olorer Details
	Column Name	Data Type	Allow Nulls
Sch	oolId	int	
8	Set Primary Ke <u>y</u>		
	Insert Colu <u>m</u> n		
Ψ	Delete Colum <u>n</u>		<b>V</b>
s.	Relations <u>h</u> ips		<b>V</b>
1	Indexes/Keys		<b>V</b>
AB	<u>F</u> ulltext Index		<b>V</b>
Pani	XML Indexes		
	Check Constraints		
22	Spatial Indexes		
5	Generate Change <u>S</u> cript	t	
	P <u>r</u> operties	Alt+Enter	

The primary Key column will then have a small key  $\mathbf{P}$  front to illustrate that this column is a Primary Key.



Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5

# 2.3.2 FOREIGN KEY

A FOREIGN KEY in one table points to a PRIMARY KEY in another table.

Example:

	Column Name	Data Type	Allow Nulls
8	SchoolId	int	
	SchoolName	varchar(50)	
	Description	varchar(1000)	<ul> <li>Image: A set of the set of the</li></ul>
	Address	varchar(50)	<ul> <li>Image: A set of the set of the</li></ul>
	Phone	varchar(50)	<b>~</b>
	PostCode	varchar(50)	
	PostAddress	varchar(50)	<b>~</b>

CL	ASS		
	Column Name	Data Type	Allow Nulls
8	ClassId	int	
	SchoolId	int	
	ClassName	varchar(50)	
	Description	varchar(1000)	<b>~</b>

We will create a CREATE TABLE script for these tables:

#### SCHOOL:

```
CREATE TABLE SCHOOL
(
SchoolId int IDENTITY(1,1) PRIMARY KEY,
SchoolName varchar(50) NOT NULL UNIQUE,
Description varchar(1000) NULL,
Address varchar(50) NULL,
Phone varchar(50) NULL,
PostCode varchar(50) NULL,
PostAddress varchar(50) NULL,
)
```

#### CLASS:

```
CREATE TABLE CLASS
(
ClassId int IDENTITY(1,1) PRIMARY KEY,
SchoolId int NOT NULL FOREIGN KEY REFERENCES SCHOOL(SchoolId),
ClassName varchar(50) NOT NULL UNIQUE,
Description varchar(1000) NULL,
)
```



Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5

The FOREIGN KEY constraint is used to prevent actions that would destroy links between tables. The FOREIGN KEY constraint also prevents that invalid data from being inserted into the foreign key column, because it has to be one of the values contained in the table it points to.

# Setting Foreign Keys in the Designer Tools:

If you want to use the designer, right-click on the column that you want to be the Foreign Key and select "**Relationships...**":

PC	PC88235\DEVELOPMTEST - dbo.CLASS × Object Ex			Object Exp	olorer Details
		Column Name	Data	Туре	Allow Nulls
8	Clas	ssId	int		
▶	Sch	oolId	int		
	9	Set Primary Key			
	i i	Insert Column			<b>V</b>
	Ŧ	Delete Column			
	R	Relationships			
	1	Indexes/Keys			
	AB	Fulltext Index	Fulltext Index		
	<mark>ел</mark>	XML Indexes			
		Check Constraints			
	<u>2</u> X	Spatial Indexes			
	5	Generate Change Script			
	6	Properties	A	t+Enter	

The following window pops up (Foreign Key Relationships):



Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5

Foreign Key Relationships		? <mark>×</mark>			
Selected Relationship:					
FK_CLASS_CLASS*	Editing properties for new relationship. The 'Tables And Columns Specification' property needs to be filled in before the new relationship will be accepted.				
	⊿ (General)				
	Check Existing Data On Creati	Yes			
	Tables And Columns Specific				
	Foreign Key Base Table	CLASS			
	Foreign Key Columns	ClassId			
	Primary/Unique Key Base	CLASS			
	Primary/Unique Key Colu	ClassId			
	⊿ Identity				
	(Name)	FK_CLASS_CLASS			
	Description				
	Table Designer	-			
Add Delete		Close			

Click on the "Add" button and then click on the small "..." button. Then the following window

pops up (Tables and Columns):



Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5

Tables and Columns	? 💌
Relationship name:	
FK_CLASS_SCHOOL	
Primary key table: 🚺	Foreign key table: 🕗
SCHOOL	CLASS
SchoolId 👻	SchoolId
Select Primary Key Column	Select Foreign Key Column
	OK Cancel

Here you specify the primary Key Column in the Primary Key table and the Foreign Key Column in the Foreign Key table.

#### 2.3.3 NOT NULL / Required Columns

The NOT NULL constraint enforces a column to NOT accept NULL values. The NOT NULL constraint enforces a field to always contain a value. This means that you cannot insert a new record, or update a record without adding a value to this field. If we take a closer look at the CUSTOMER table created earlier:

```
CREATE TABLE [CUSTOMER]
(
CustomerId int IDENTITY(1,1) PRIMARY KEY,
CustomerNumber int NOT NULL UNIQUE,
LastName varchar(50) NOT NULL,
FirstName varchar(50) NOT NULL,
AreaCode int NULL,
Address varchar(50) NULL,
Phone varchar(50) NULL,
)
```



Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5

We see that "CustomerNumber", "LastName" and "FirstName" is set to "NOT NULL", this means these columns needs to contain data. While "AreaCode", "Address" and "Phone" may be left empty, i.e, they don't need to filled out.

Note! A primary key column cannot contain NULL values.

# Setting NULL/NOT NULL in the Designer Tools:

In the Table Designer you can easily set which columns that should allow NULL or not:

PC	88235\DEVELOPEST - dbo.S	CHOOL × Object Ex	plorer Details
	Column Name	Data Type	Allow Nulls
▶8	SchoolId	int	
	SchoolName	varchar(50)	
	Description	varchar(1000)	<b>V</b>
	Address	varchar(50)	<b>V</b>
	Phone	varchar(50)	<b>V</b>
	PostCode	varchar(50)	<b>V</b>
	PostAddress	varchar(50)	

#### **2.3.4 UNIQUE**

The **UNIQUE** constraint uniquely identifies each record in a database table. The UNIQUE and PRIMARY KEY constraints both provide a guarantee for uniqueness for a column or set of columns.

A PRIMARY KEY constraint automatically has a UNIQUE constraint defined on it.

**Note!** You can have many UNIQUE constraints per table, but only one PRIMARY KEY constraint per table.

If we take a closer look at the CUSTOMER table created earlier:

```
CREATE TABLE [CUSTOMER]
(
CustomerId int IDENTITY(1,1) PRIMARY KEY,
CustomerNumber int NOT NULL UNIQUE,
LastName varchar(50) NOT NULL,
FirstName varchar(50) NOT NULL,
AreaCode int NULL,
Address varchar(50) NULL,
Phone varchar(50) NULL,
)
```



Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5

We see that the "CustomerNumber" is set to UNIQUE, meaning each customer must have a unique Customer Number. Example:

	CustomerId	CustomerNumber	LastName	FirstName	AreaCode	Address	Phone
1	1	1000	Smith	John	12	California	11111111
2	2	1001	Jackson	Smith	45	London	22222222
3	3	1002	Johnsen	John	32	London	33333333

# Setting UNIQUE in the Designer Tools:

If you want to use the designer, right-click on the column that you want to be UNIQUE and select "**Indexes/Keys...**":

PC	PC88235\DEVELOPEST - dbo.SCHOOL × Object Explorer Details				
	Column Name		Data Type	Allow Nulls	
8	SchoolId		int		
۲	SchoolName				
	Description	U,U	Set Primary Key		
	Address		Insert Colu <u>m</u> n		
	Phone	1	Delete Colum <u>n</u>		
	PostCode	3	Relations <u>h</u> ips		
	PostAddress	1	Indexes/Keys		
		AB	<u>F</u> ulltext Index		
		8ні	XML Indexes		
			Check Constraints		
		22	Spatial Indexes		
		-	Generate Change <u>S</u> cri	pt	
			P <u>r</u> operties	Alt+Enter	

Then click "Add" and then set the "Is Unique" property to "Yes":



Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5

Indexes/Keys				8 8		
Selected Primary/Unique Key or In	dex:					
IX_SCHOOL*	Ed	liting properties for new uniqu	ue key or index.			
PK_SCHOOL	2					
	1	(General)		-		
		Columns	SchoolId (ASC)			
		Is Unique	Yes	•		
		Туре	Yes	E		
	4	Identity	No			
		(Name)	IX_SCHOOL			
		Description				
	4	Table Designer				
		Create As Clustered	No			
	⊳	Data Space Specification	PRIMARY			
	▶ Fill Specification +					
Add Delete				Close		

# 2.3.5 CHECK

The CHECK constraint is used to limit the value range that can be placed in a column.

If you define a CHECK constraint on a single column it allows only certain values for this column.

If you define a CHECK constraint on a table, it can limit the values in certain columns based on values in other columns in the row.

#### Example:

```
CREATE TABLE [CUSTOMER]
(
CustomerId int IDENTITY(1,1) PRIMARY KEY,
CustomerNumber int NOT NULL UNIQUE CHECK(CustomerNumber>0),
LastName varchar(50) NOT NULL,
FirstName varchar(50) NOT NULL,
AreaCode int NULL,
Address varchar(50) NULL,
Phone varchar(50) NULL,
)
```

In this case, when we try to insert a Customer Number less than zero we will get an error message.



Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5

# Setting CHECK constraints in the Designer Tools:

If you want to use the designer, right-click on the column where you want to set the constraints and select "**Check Constraints...**":

PC88235\DEVELOP dbo.CUSTOMER × Object Explorer Details					
	Column Name		Data Type	Allow Nulls	
8	CustomerId	int	t		
	CustomerName	va	rchar(50)		
►	CustomerNumber	i	•		_
	Address	<b>%</b>	Set Primary Ke <u>y</u>		
	Phone	1	Insert Colu <u>m</u> n		
	PostCode	Ŧ	Delete Colum <u>n</u>		_
	PostAddress	R	Relations <u>h</u> ips		
	EMail	浬	Indexes/Keys		
		AB	<u>F</u> ulltext Index		
_		<b>F</b>	XML Indexes		
			Check Constraints.	•	
		22	Spatial Indexes		
		5	Generate Change S	cript	
		Ŀ.	P <u>r</u> operties	Alt+Enter	

Then click "Add" and then click "..." in order to open the Expression window:



Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5

Check Constraints Selected Check Constraint:				? 🔀
CK_CUSTOMER*		iting properties for new chec be filled in before the new cl	ck constraint. The 'Expression' pr heck constraint will be accepted.	roperty needs
	4	(General)		2
		Expression		
	4	Identity		
		(Name)	CK_CUSTOMER	
		Description		
	4	Table Designer		
		Check Existing Data On Cre	ati Yes	
		Enforce For INSERTs And U	PC Yes	
		Enforce For Replication	Yes	
Add Delete				Close

In the Expression window you can type in the expression you want to use:

Check Constraint Expression	? 💌
Expression:	
CustomerNumber>0	*
	-
	Cancel



Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5

# 2.3.6 DEFAULT

The DEFAULT constraint is used to insert a default value into a column.

The default value will be added to all new records, if no other value is specified.

Example:

```
CREATE TABLE [CUSTOMER]
(
CustomerId int IDENTITY(1,1) PRIMARY KEY,
CustomerNumber int NOT NULL UNIQUE,
LastName varchar(50) NOT NULL,
FirstName varchar(50) NOT NULL,
Country varchar(20) DEFAULT 'Iraq - AL Anbar',
AreaCode int NULL,
Address varchar(50) NULL,
Phone varchar(50) NULL,
)
```

# Setting DEFAULT values in the Designer Tools:

Select the column and go into the "Column Properties":

Column Properties	
2	
▲ (General)	
(Name)	Country
Allow Nulls	Yes
Data Type	varchar
Default Value or Binding	'Iraq-ALAnbar
Length	50



Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5

### 2.3.7 AUTO INCREMENT or IDENTITY

Very often we would like the value of the primary key field to be created automatically every

time a new record is inserted.

#### Example:

```
CREATE TABLE CUSTOMER
(
CustomerId int IDENTITY(1,1) PRIMARY KEY,
CustomerNumber int NOT NULL UNIQUE,
LastName varchar(50) NOT NULL,
FirstName varchar(50) NOT NULL,
AreaCode int NULL,
Address varchar(50) NULL,
Phone varchar(50) NULL,
)
```

As shown below, we use the IDENTITY() for this. IDENTITY(1,1) means the first value will be 1 and then it will increment by 1.

# Setting identity (1,1) in the Designer Tools:

We can use the designer tools to specify that a Primary Key should be an identity column

that is automatically generated by the system when we insert data in to the table.

Click on the column in the designer and go into the Column Properties window:

Column Properties	
21 3	
Data Type	int
Default Value or Binding	
Table Designer	
Collation	<database default=""></database>
Computed Column Specification	
Condensed Data Type	int
Description	
Deterministic	Yes
DTS-published	No
Full-text Specification	No
Has Non-SQL Server Subscriber	No
<ul> <li>Identity Specification</li> </ul>	Yes
(Is Identity)	Yes
Identity Increment	1
Identity Seed	1
Indexable	Yes
Is Columnset	No
Is Sparse	No -



Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5

## 2.3.8 ALTER TABLE

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table. To add a column in a table, use the following syntax:

#### ALTER TABLE table\_name ADD column\_name datatype

To delete a column in a table, use the following syntax (notice that some database systems don't allow deleting a column):

#### ALTER TABLE table\_name DROP COLUMN column\_name

To change the data type of a column in a table, use the following syntax:

#### ALTER TABLE table\_name ALTER COLUMN column\_name datatype



Lecturer Name: Shihab Hamad Khaleefah Academic Status: BhD. In Computer Science Qualification: - Lecturer Course Material: Database Management System Grade Level: Second Stage, Lec. 4-5

Summary

#### In this lecture,

The student's will be able to learn how to make the definition of database terminology from tables, queries, reports and forms. also will be able to create and manage databases, use data stored in the databases. The most important point is to understanding the data models of database, three levels of the database, the physical data structure, three factors of the data model with entire example illustrates the work of this function.