

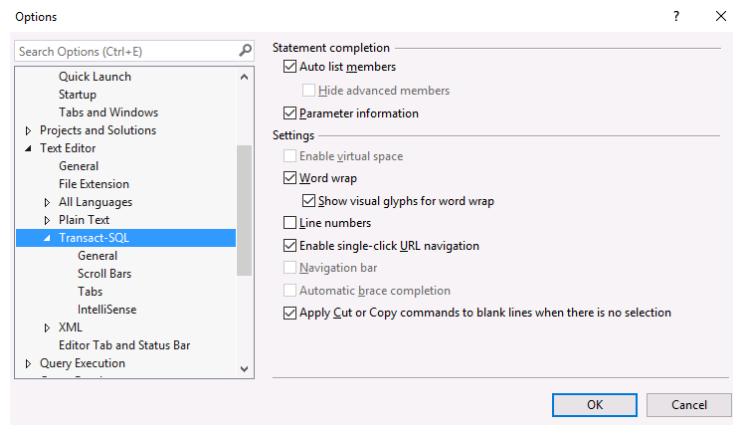
# Joseph SQL Notes

T-SQL (Transact-SQL)

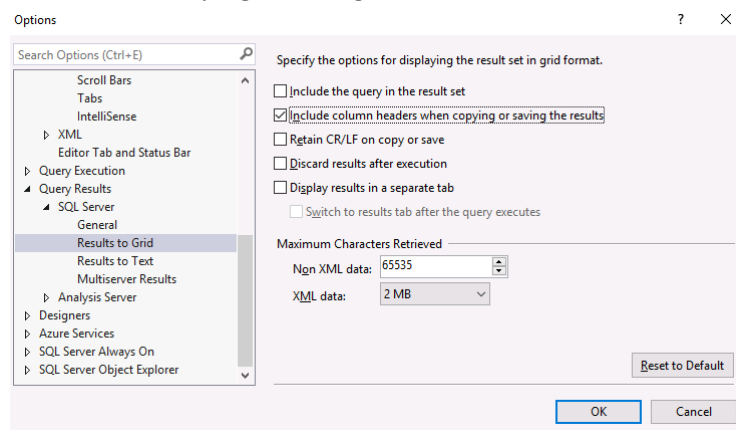
DML – Data manipulation languages

## Settings for Microsoft SQL Server Management Studio

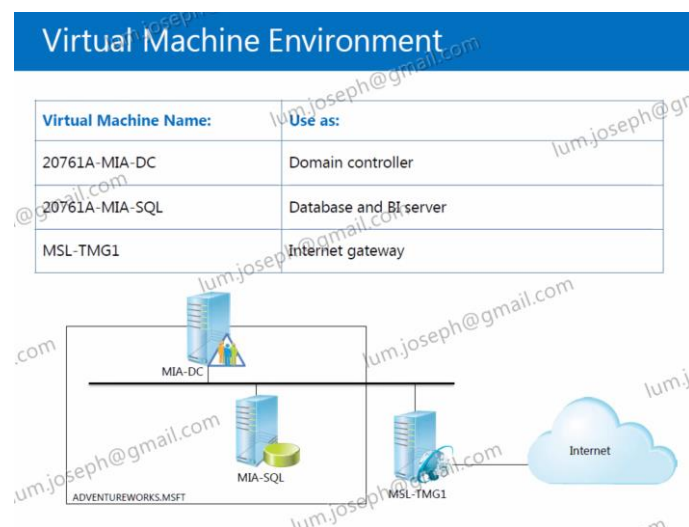
### 1. Enable Word Wrap

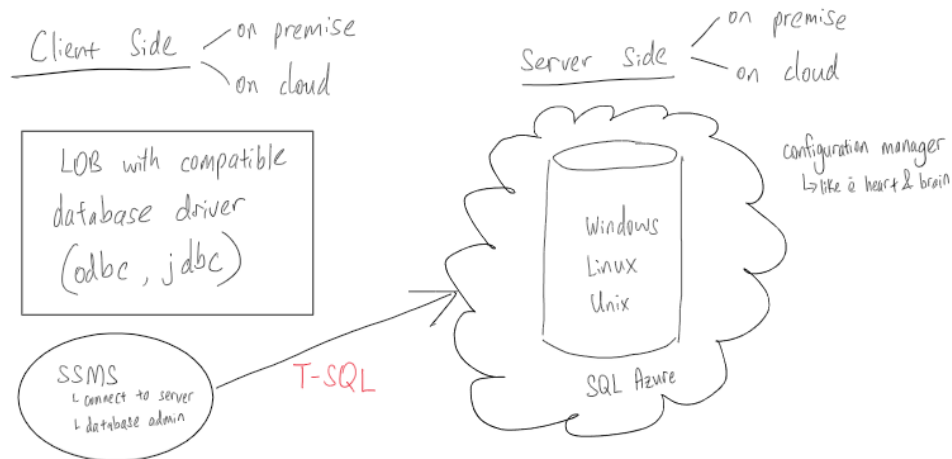


### 2. Include column headers when copying or saving the results



Using SQL on your local machine



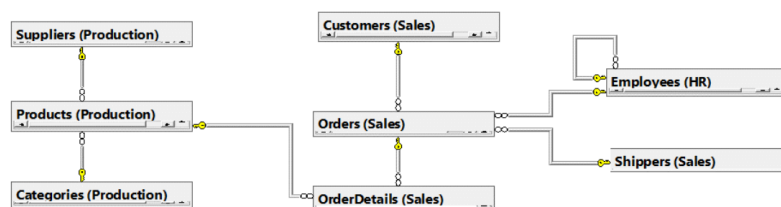


<b>DML</b>	Data manipulation language
<b>DDL</b>	Data definition Language <ul style="list-style-type: none"> <li>Object building</li> </ul>
<b>DCL</b>	Data control language <ul style="list-style-type: none"> <li>Access control for respective users</li> </ul>

## Within SQL database

- Tables (store and modify data)**
- Views (read only analytics)**
- Stored Procedure, Functions**
  - don't hold data
  - do programming like analytics, updating data, modifying data
- .net objects**
  - Added in 2008
  - Programming language: C#, VB

### • TSQL Database



There are 2 types of SQL database

- OLTP** – Online transaction processing
  - Usually, every item transaction has its individual table e.g. Employee
- OLAP** – Online analytical processing
  - Read only solutions – do not edit OLAP

## Using SQL within Command Prompt

### 1. Type in `sqlcmd /?`

```
Command Prompt
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Student>sqlcmd /?
Microsoft (R) SQL Server Command Line Tool
Version 14.0.1000.160 NT
Copyright (C) 2017 Microsoft Corporation. All rights reserved.

usage: Sqlcmd [-U login id] [-P password]
[-S server] [-H hostname] [-E trusted connection]
[-N Encrypt Connection] [-C Trust Server Certificate]
[-d use database name] [-l login timeout] [-t query timeout]
[-h headers] [-s colseparator] [-w screen width]
[-a packetsize] [-e echo input] [-I Enable Quoted Identifiers]
[-c cmdend] [-L[c] list servers[clean output]]
[-q "cmdline query"] [-Q "cmdline query" and exit]
[-m errorlevel] [-V severitylevel] [-W remove trailing spaces]
[-u unicode output] [-r[0|1] msgs to stderr]
[-i inputfile] [-o outputfile] [-z new password]
[-f <codepage> | i:<codepage>[,o:<codepage>]] [-Z new password and exit]
[-k[1|2] remove[replace] control characters]
[-y variable length type display width]
[-Y fixed length type display width]
[-p[1] print statistics[colon format]]
[-R use client regional setting]
[-K application intent]
[-M multisubnet failover]
[-b On error batch abort]
[-v var = "value"...] [-A dedicated admin connection]
[-X[1] disable commands, startup script, environment variables [and exit]]
[-x disable variable substitution]
[-j Print raw error messages]
[-g enable column encryption]
[-G use Azure Active Directory for authentication]
[-? show syntax summary]

C:\Users\Student>
```

### 2. Type `sqlcmd`

- Type your SQL script then end the script with **“go”**

```
C:\Users\Student>sqlcmd
1> use corporatedb
2> select * from employeeprofile
3> go
Changed database context to 'CorporateDB'.
-----
employeid  employeename  employeeetype  salary
-----
1001 Michael Shepard  Regular  20000.0000
1002 Eva Longoria    Temporary  20000.0000
(2 rows affected)
1>
```

### 3. To end the script or close

- Type **“exit”** then press Enter
- Click the X to close the entire command prompt

```
C:\Users\Student>sqlcmd
1> use corporatedb
2> select * from employeeprofile
3> go
Changed database context to 'CorporateDB'.
employeid  employeename
-----
1001 Michael Shepard
1002 Eva Longoria
(2 rows affected)
1> exit

C:\Users\Student>
```

Language used by Microsoft is **SQL-99 standard**

Object	Parent	Description
Solution	-	A solution is a conceptual container for projects. Solutions have a <b>.ssmssl</b> extension, and are always displayed at the top of the hierarchy.
Project	Solution	Projects contain queries (T-SQL scripts), database connection metadata, and other miscellaneous files. You can file any number of projects within a solution. Projects have a <b>.ssmssqlproj</b> extension.
Script	Project	T-SQL script files with a <b>.sql</b> extension are the basic files used to work with SQL Server.

Documentation by Microsoft - **Transact-SQL Reference (Database Engine)**

- <https://docs.microsoft.com/en-us/sql/t-sql/language-reference?view=sql-server-ver15>
- Delimiters for table name / column names, must use them when there is spacing
  - [Database name].[ table name]
  - "Database name"." table name"

A database is contained within a single instance, cannot be stored across multiple instances.

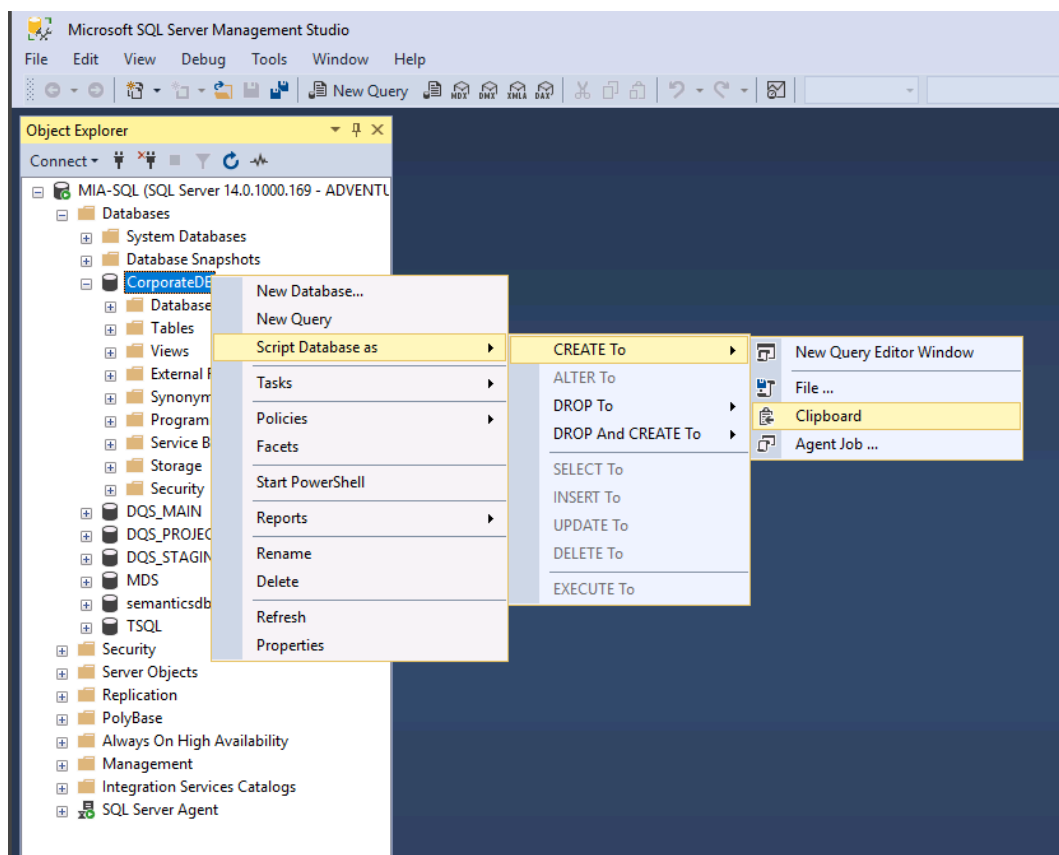
SQL server management studio (SSMS) solution allows you to organise SQL Scripts so that you can manage large collections of files. **Projects** can contain scripts, connection strings and other settings. **Solutions** are collections of projects.

Elements:	Predicates and Operators:
Predicates	ALL, ANY, BETWEEN, IN, LIKE, OR, SOME
Comparison Operators	=, >, <, >=, <=, <>, !=, !>, !<
Logical Operators	AND, OR, NOT
Arithmetic Operators	*, /, %, +, -, ,
Concatenation	+

Order of Evaluation	Operator
1	~ (Bitwise NOT)
2	/, *, % (Division, Multiply, Modulo)
3	+, -, &, ^,   (Positive/Add/Concatenate, Negative/Subtract, Bitwise AND, Bitwise Exclusive OR, Bitwise OR)
4	=, >, <, <=, >=, !=, !<, !> (Comparisons)
5	NOT
6	AND
7	ALL, ANY, BETWEEN, IN, LIKE, OR, SOME
8	= (Assignment)

## Create database

- Copy the SQL script to clipboard for reference
- Save the generated script in a separate query file. Can run script in future.
- Can be applied to editing / creating *database, tables, dataset*



## Types of JOINS

### 1. INNER JOIN

- a. Only shows entries found in both tables.

### 2. Outer Joins

#### a. LEFT JOIN

#### b. RIGHT JOIN

### 3. Cross Join (Cartesian Join)

- a. Get all the records from both tables. Includes all possible combinations.

- Characteristics of a Cartesian product

- Output or intermediate result of FROM clause
- Combine all possible combinations of two sets
- In T-SQL queries, usually not desired
  - Special case: table of numbers

Name	Product		Name	Product
Davis	Alice Mutton	=	Davis	Alice Mutton
Davis	Crab Meat		Davis	Crab Meat
Davis	Ipoh Coffee		Davis	Ipoh Coffee
Funk	Alice Mutton		Funk	Alice Mutton
Funk	Crab Meat		Funk	Crab Meat
Funk	Ipoh Coffee		Funk	Ipoh Coffee
King	Alice Mutton		King	Alice Mutton
King	Crab Meat		King	Crab Meat
King	Ipoh Coffee		King	Ipoh Coffee

### 4. SUB JOIN

Normal Join statement	Subquery Within the IN Clause
<pre>SELECT p.name, p.cost FROM product p JOIN sale s ON p.id=s.product_id WHERE s.price=2000;</pre>	<pre>SELECT name, cost FROM product WHERE id IN (SELECT product_id FROM sale);</pre>

## NULL Handling

- use function **ISNULL(<column\_name>, replacement\_value)**
- when concat multiple columns, as long as 1 of the columns is NULL → entire concatenation will be NULL

## Concat function

- Use **CONCAT(<column\_name>, '&', '&', <column\_name2>, '&', '&', <column\_name3>)**

## Data Types

Data types will also determine the amount of memory consumption, measures in number of bytes.

Example for numbers

- Data types – bigint / bit / int

Nvarchar compared to varchar

- Varchar → only can store alphabets. Will replace non-English character with question mark (?)
- Non-unicode varchar

Changing data type of entire column, use the following functions

- **CAST**( <column\_name> AS data\_type )
- **CONVERT**( data\_type, <column\_name>, style)
  - Note: style is the format type of the output
  - Documentation: <https://docs.microsoft.com/en-us/sql/t-sql/functions/cast-and-convert-transact-sql?view=sql-server-ver15>

Query	Results	Description
<b>SELECT</b> 1 + '2' AS result;	3	Auto implicit conversion from lower data type (varchar) to higher (int)
<b>SELECT</b> 1 + 'abc' AS result;	Msg Error	Cannot mix data types
<b>SELECT</b> <b>CAST</b> (1 AS <b>VARCHAR</b> (10)) + 'abc' AS result;	1abc	Same data type, therefore can add together

**Collation** is the format of the entire database

- Use **COLLATE** function
- **WHERE** <column\_name> **COLLATE** Latin1\_General\_CS\_AS = 'Goldberg';

## FORMAT function

The screenshot shows a SQL query in a query editor and its results. The query is:

```
DECLARE @m money = 120.595
SELECT @m AS unformatted_value,
       FORMAT(@m, 'C', 'zh-cn') AS zh_cn_currency,
       FORMAT(@m, 'C', 'en-us') AS en_us_currency,
       FORMAT(@m, 'C', 'de-de') AS de_de_currency;
```

Annotations in the image:

- Declared variable**: Points to the `DECLARE @m money = 120.595` line.
- SQL Function**: Points to the `FORMAT` function in the `SELECT` statement.
- Specify the type of currency**: Points to the format code `'C'` in the `FORMAT` function.
- Format**: Points to the locale codes `'zh-cn'`, `'en-us'`, and `'de-de'`.

The results table shows the following data:

unformatted_value	zh_cn_currency	en_us_currency	de_de_currency
120.595	¥120.60	\$120.60	120,60 €

## Date Manipulation

See documentation for SQL date - <https://docs.microsoft.com/en-us/sql/t-sql/functions/date-and-time-data-types-and-functions-transact-sql?view=sql-server-ver15>

```
SELECT GETDATE() AS 'currentdatetime',
       CAST(GETDATE() AS DATE) AS 'currentdate',
       GETDATE() AS 'currenttime',
       YEAR(GETDATE()) AS 'currentyear',
       MONTH(GETDATE()) AS 'currentmonth',
       DAY(GETDATE()) AS 'currentday',
       DATEPART(week, GETDATE()) AS 'currentweeknumber',
       DATENAME(month, GETDATE()) AS 'currentmonthname';

-- instead of using GETDATE() you can use CURRENT_TIMESTAMP
```

currentdatetime	currentdate	currenttime	currentyear	currentmonth	currentday	currentweeknumber	currentmonthname
2021-09-11 23:50:15.590	2021-09-11	2021-09-11 23:50:15.590	2021	9	11	37	September

SQL function	Description
<b>SYSDATETIME</b>	<ul style="list-style-type: none"><li>• More precise than <b>CURRENT_TIMESTAMP</b></li><li>• Returns data type <b>datetime2</b></li></ul>
<b>CURRENT_TIMESTAMP</b>	<ul style="list-style-type: none"><li>• Returns data type <b>datetime</b></li></ul>

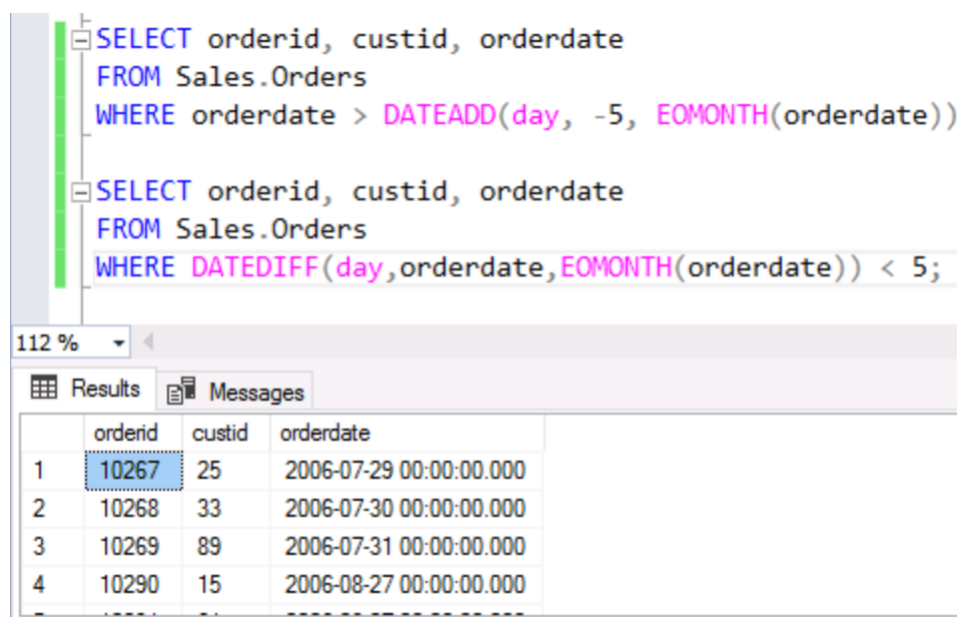
```
SELECT CAST('2015-12-11' as DATE) as 'Casted'
      , CONVERT(DATE, '2015-12-11') as 'Convert'
      , CONVERT(DATE, '12/11/2015', 101) AS somedate
      , SYSDATETIME() as 'systemdatetime'
      , DATEFROMPARTS(2015,12,11) as 'datefromparts'
```

Casted	Convert	somedate	systemdatetime	datefromparts
2015-12-11	2015-12-11	2015-12-11	2021-09-11 23:54:19.7689915	2015-12-11



## Getting the historical records

- example: last 5 days records
- There are 2 methods used – **DATEADD()** or **DATEDIFF()**



The screenshot shows a SQL Server Enterprise Manager interface. Two SQL queries are visible in the query window, both selecting from the Sales.Orders table. The first query uses DATEADD and EOMONTH to find records from the last 5 days. The second query uses DATEDIFF and EOMONTH for the same purpose. Below the queries, the 'Results' tab is active, displaying a table with 4 rows of order data. The first row is highlighted with a blue selection box.

```
SELECT orderid, custid, orderdate
FROM Sales.Orders
WHERE orderdate > DATEADD(day, -5, EOMONTH(orderdate))

SELECT orderid, custid, orderdate
FROM Sales.Orders
WHERE DATEDIFF(day, orderdate, EOMONTH(orderdate)) < 5;
```

	orderid	custid	orderdate
1	10267	25	2006-07-29 00:00:00.000
2	10268	33	2006-07-30 00:00:00.000
3	10269	89	2006-07-31 00:00:00.000
4	10290	15	2006-08-27 00:00:00.000

## Replacing NULL values in dataset

You may view documentation for comparison between ISNULL and COALESCE -

<https://docs.microsoft.com/en-us/sql/t-sql/language-elements/coalesce-transact-sql?view=sql-server-ver15>

1. **ISNULL** ( check\_expression , replacement\_value )
2. **COALESCE** ( check\_expression , replacement\_value )

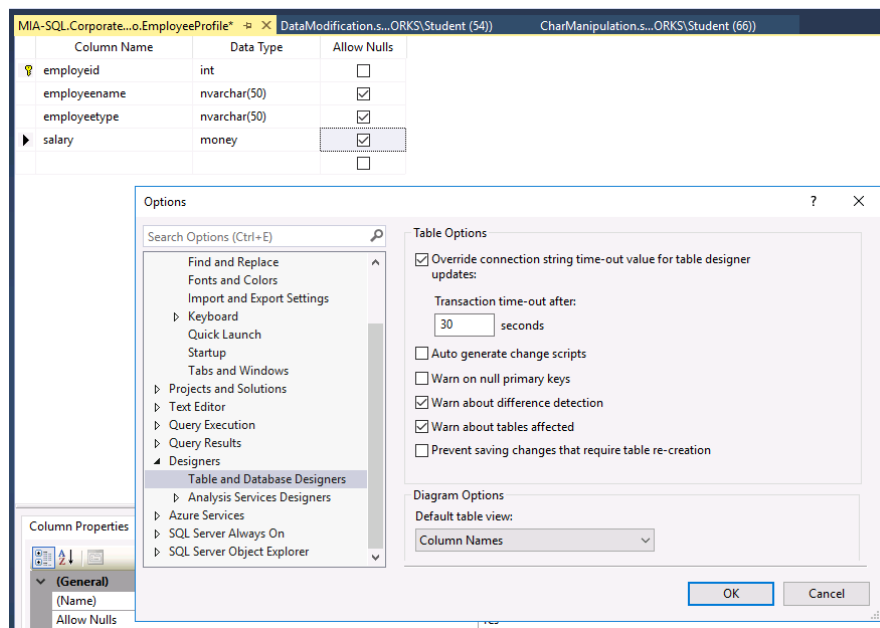
COALESCE is a standard ANSI SQL function and ISNULL is not. Should use the **COALESCE function**.

For COALESCE you can put multiple columns. It will display the replacement value in all the columns mentioned

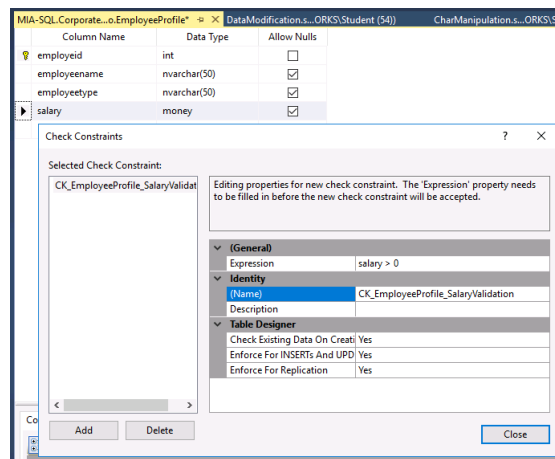
**COALESCE** ( shipregion, Col2, col3, 'No value' )

## Create constraints to database

- Click on Tools > Options > Designers > Table and Database Designers
- When you modify a table, such as adding constraints, you will drop the entire table and re-create it.
  - Clean your table. If you create a constraint that is not met by existing table, the new constraint will not be applied.
  - Change applied to entire database



Type in the **Expression** for the constraints – can be simple or very complex logic



## Functions in SQL

Function Category	Description
Scalar	Operate on a single row, return a single value
Grouped Aggregate	Take one or more input values, return a single summarizing value
Window	Operate on a window (set) of rows
Rowset	Return a virtual table that can be used in a T-SQL statement