SQL SERVER BASICS

California Creative Solutions

SQL Server

These are the basic topics focusing around the beginning aspects of using SQL Server. Topics are kept general as to apply to most RDBMS' similar to SQL Server

SQL Server Basics

SQL Server Languages – DML, DDL, DQL, DCL

DML – Data Manipulation Language – Insert, Update, Delete, Truncate*

Insert – Add data to an existing table

```
Insert into TableA values (1,'Hello','World')
```

Update – Change the value in a row

```
Update TableA
Set Column3 = 'Univierse'
Where Column3 = 'World'
```

Delete – Remove rows of data from a table

```
Delete
From TableA
Where Column1 = 1
```

Truncate – Remove ALL rows of data from a table (May be considered DDL by some)

```
Truncate Table
TableA
```

DDL – Data Definition Language – Create, Alter, Drop

Create – Create or define an object in SQL Server

```
Create Table TableA
(Column1 int, Column2 varchar(20), Column3 varchar(20))
```

Alter – Modify or change the table structure and columns

```
Alter Table TableA
Add Column4 nvarchar(50)

Alter Table TableA
Drop Column Column4

Alter Table TableA
Alter Column Column3 varchar(200)
```

Drop – Remove an object from the database

```
Drop table
TableA

Drop database
Adventureworks2012
```

DQL – Data Query Language – Select, From, Where, Group By, Having, Order By

Select – Choose which columns of data to display

Select Column1, Column2, Column3

From – Specify the table where data is being retrieved

From dbo.TableA

Where – Provide a condition to filter the result set

Where Column1 = 1

Group By – Combine matching data points within a column

Group By Column1

Having – Provide a condition to filter aggregated columns

Having Sum(Column1) =< 100

Order By – Choose which columns to sort the data by

Order By Column3 Desc

DCL – Data Control Language – Grant, Revoke, Deny

Grant – Give permissions to someone to perform an action on an object

Grant Select
On dbo.TableA
To Jane

Revoke – Take back a given permission from a user

Revoke Select
On dbo.TableA
From Jane

Deny – Prevent someone from doing an operation on an object

Deny Select
On dbo.TableA
To Jane

SQL Server Constraints

Key Constraints – Primary Key, Foreign Key, Unique (Key)

Primary Key – Also uses Not Null and Unique, this specifies a column(s) to be used in identifying each individual row of data. It will create a Unique Clustered Index as a result.

```
Alter Table TableA
Add Primary Key (Column1)
```

Foreign Key – Used to specify that the chosen column will be referencing another column set of data in table.

```
Alter Table TableB
Add Foreign Key (Column1)
References TableA(Column1)
```

Unique Key – Used to specify that a column(s) should be indexed and sorted for faster retrieval of data. Used on columns that are commonly queried. Creates Unique non clustered index.

```
Alter Table TableA
Add Unique (Column2)
```

Column Constraints – Not Null, Check, Default

Check – Used to limit what values will be valid within the column.

```
Alter Table TableA
Add Check (Column1 <= 100)
```

Not Null – Specify if Null values are allowed within a column or not

```
Alter Table TableA
Alter Column Column1 int Not Null
```

Default – Will insert a default value if there is no value given

```
Alter Table TableA
Alter Column Column1 Default 999999
```

Dropping a Constraint

All constraints except Null and Not Null can be removed. A column must always be either Null or Not Null

```
Alter Table TableA
Drop Constraint Constraint_Name
```

```
Alter Table TableA
Drop Primary Key
```

Joining Tables in SQL

Tables are often divided into smaller pieces through the process of Normalization. This helps to limit data to only what is needed at any given time. The less data you pull, the faster the process. Sometimes though, we need more information and must combine tables together. This is often done using Joins.

Joins – Inner, Right Outer, Left Outer, Full Outer

Inner Join – This will combine rows from multiple tables, where the column values must match

```
Select *
From TableA
Inner join TableB
On TableA.Column1 = TableB.Column1
```

Left Outer Join – This will display the matching values between the two tables, just like inner join, but ALSO display the non-matching values from the first table

```
Select *
From TableA
Left Outer join TableB
On TableA.Column1 = TableB.Column1
```

Right Outer Join – This will display the matching values between the two tables, just like inner join, but ALSO display the non-matching values from the second table

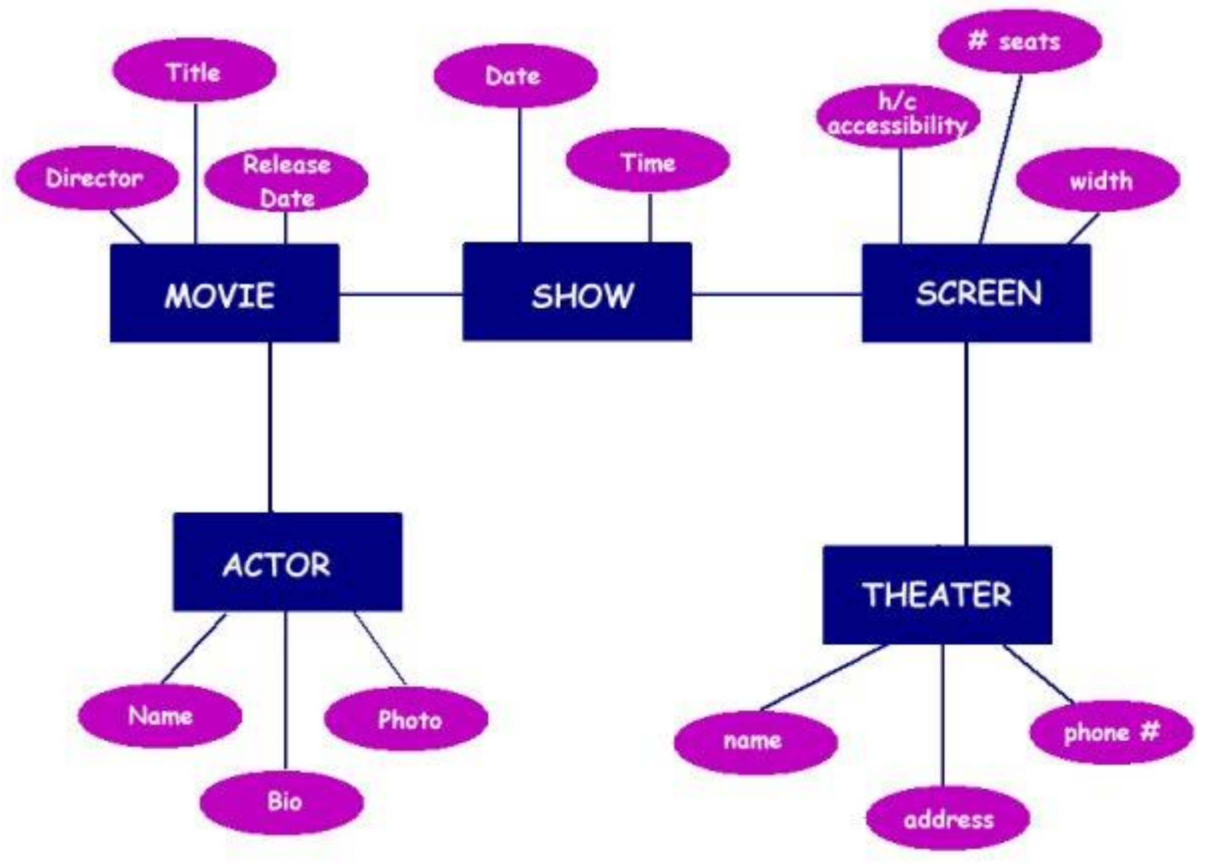
```
Select *
From TableA
Right Outer join TableB
On TableA.Column1 = TableB.Column1
```

Full Outer Join – This join will display all values from both tables, whether they match or do not match

```
Select *
From TableA
Full Outer join TableB
On TableA.Column1 = TableB.Column1
```

Data Modeling in SQL Server

Data modeling is essential in the creation of any database. Here, you'll design a blue print to the overall design of the entire database. Each individual table, column, and relationship would be displayed. To do this we use Entity Relationship Diagrams like below.

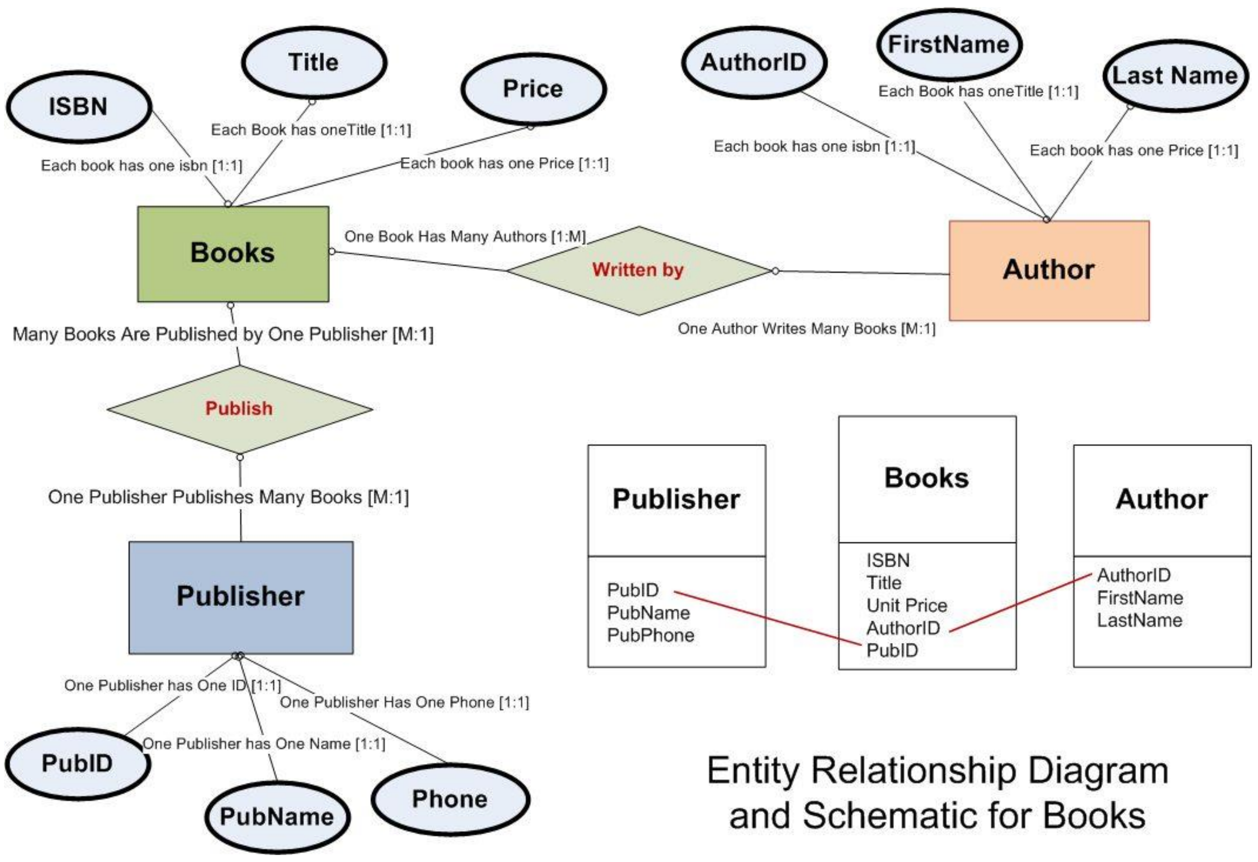


In this ER-Diagram, each table is known as an Entity. They are represented with the Rectangles

The columns that would be found in those tables are displayed as attributes, with ovals

This diagram is in a simple form, and displaying only the Entities and Attributes. There are no direct references to the type of relationships that are being used.

Relationships are how entities relate or connect their data to one another. Relationships will often be displayed using a diamond shape, as seen in the next figure.


ISBN
Title
Price
Each book has one isbn [1:1]
Each Book has oneTitle [1:1]
Each book has one Price [1:1]
AuthorID
FirstName
Last Name
Each book has one isbn [1:1]
Each Book has oneTitle [1:1]
Each book has one Price [1:1]
Books
One Book Has Many Authors [1:M]
Written by
Author
One Author Writes Many Books [M:1]
Many Books Are Published by One Publisher [M:1]
Publish
One Publisher Publishes Many Books [M:1]
Publisher
One Publisher has One ID [1:1]
One Publisher Has One Phone [1:1]
One Publisher has One Name [1:1]
PubID
PubName
Phone
Publisher
PubID
PubName
PubPhone
Books
ISBN
Title
Unit Price
AuthorID
PubID
Author
AuthorID
FirstName
LastName
Entity Relationship Diagram and Schematic for Books

In this more detailed ER-Diagram, we can see the relationships, their description, and how many instances for each entity.

With the descriptions, you see them in Diamond shapes, showing what action is being taken within the relationship. This just helps to understand the nature of the relation and how the data connects

In small annotations, you'll see [1:1], [1:M], and sometimes [M:N]. These are the instances or cardinality of how many times an entity is participating in the relationship. Most commonly you'll have either one-to-one, one-to-many, or many-to-many.

1 Employee → Assigned → 1 Parking Spot

1 Teacher → Teaches → Many Classes

Many Students → Enroll In → Many Courses