Linking Multiple Tables

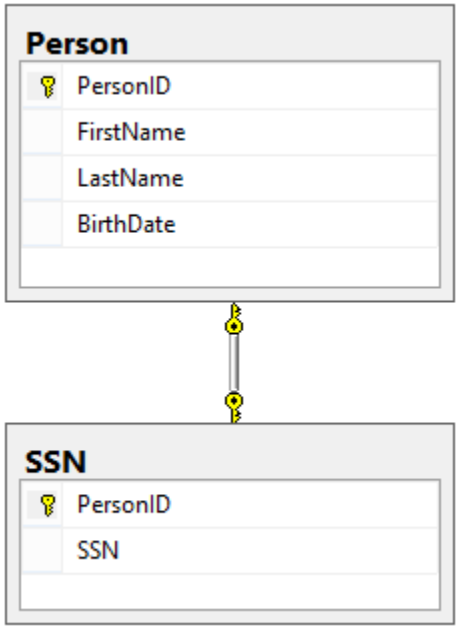
- Relational Database
- One to One
- One to Many
- Many to Many
- Entity Relationship Diagram
- Table Alias
- Relations in the WHERE clause
- INNER JOIN
- LEFT JOIN
- RIGHT JOIN
- FULL JOIN
- CROSS JOIN
- AND and OR keywords
- Multiple Joins
- Self Join

Relational Database

- Definition: A relational database is a collection of data items organized as a set of formally-described tables from which data can be accessed or reassembled in many different ways without having to reorganize the database tables.
- The tables in a relational database are “related” to one another through their primary and foreign keys
- There are three relation types possible between tables:
  - One to One
  - One to Many
  - Many to Many

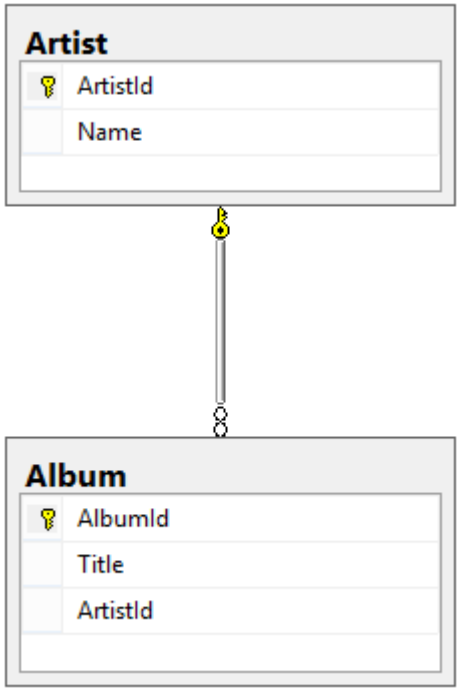
One to One Relationship

- A single record in the parent table can only relate to a single record in the child table
- For example a person can only have one Social Security Number



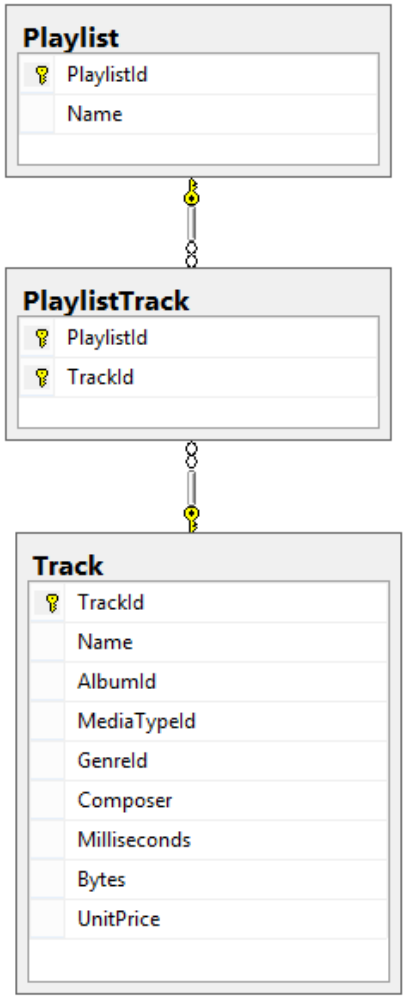
One to Many Relationship

- A single record in the parent table can relate to multiple records in the child table, but a child record can only have one parent record
- For example a single artist can have multiple albums



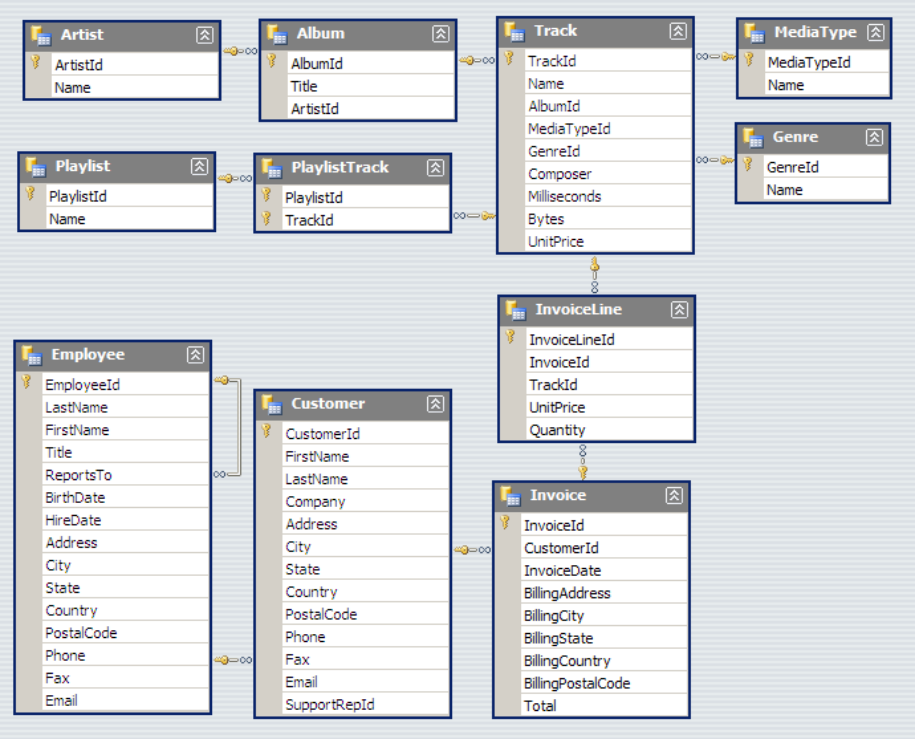
Many to Many Relationship

- Multiple records in the parent table can relate to multiple records in the child table
- To represent a many to many relationship, a linking table needs to be created that references the primary keys of both tables
- For example a playlist can have many tracks, and a track can be in many playlists
- The PlayListTrack table is a linking table. It contains the primary keys for both the Playlist and Track tables



Entity Relationship Diagrams

- An ERD shows how tables relate to one another
- The tables are connected via their primary and foreign keys
- There are applications that allow you to create ERDs like Microsoft Visio (I prefer 2010 to 2013)
- SQL Server Management Studio can also display relationships explicitly defined in the database



Multiple Tables in the FROM Clause

- If you want to pull data from multiple tables then they must all be included in the FROM clause
- The JOIN keyword is used to define relationships between tables
- When working with data from multiple tables it is good practice to prefix the columns with their table name or alias

```
SELECT
    Customer.FirstName
    ,Customer.LastName
    ,Invoice.Total
    ,Invoice.InvoiceDate
FROM Customer
JOIN Invoice
    ON Invoice.CustomerId = Customer.CustomerId
WHERE Customer.LastName = 'Philips'
```

Results | Messages

	FirstName	LastName	Total	InvoiceDate
1	Mark	Philips	8.91	2009-01-06 00:00:00.000
2	Mark	Philips	1.98	2010-08-13 00:00:00.000
3	Mark	Philips	3.96	2010-11-15 00:00:00.000
4	Mark	Philips	5.94	2011-02-17 00:00:00.000
5	Mark	Philips	0.99	2011-10-08 00:00:00.000
6	Mark	Philips	1.98	2013-03-31 00:00:00.000
7	Mark	Philips	13.86	2013-05-11 00:00:00.000

The JOIN Keyword

- The JOIN keyword can only be used in the FROM clause
- It is used to connect tables to one another
- Each JOIN keyword requires an associated ON keyword
- The ON keyword defines which columns to use when relating tables to each other
- There are 5 types of joins
  - INNER JOIN
  - LEFT OUTER JOIN
  - RIGHT OUTER JOIN
  - FULL OUTER JOIN
  - CROSS JOIN

```
FROM Employee
INNER JOIN Customer
ON Customer.SupportRepId =
Employee.EmployeeId
```

```
FROM Employee
LEFT OUTER JOIN Customer
ON Customer.SupportRepId =
Employee.EmployeeId
```

JOIN Sample Tables

- The tables Album_temp and Track_temp are used in the upcoming JOIN slides
- The tables are related on the AlbumId column
- There is a missing matching record on each table

```
SELECT
     AlbumId
     ,Title
FROM Album_temp
```

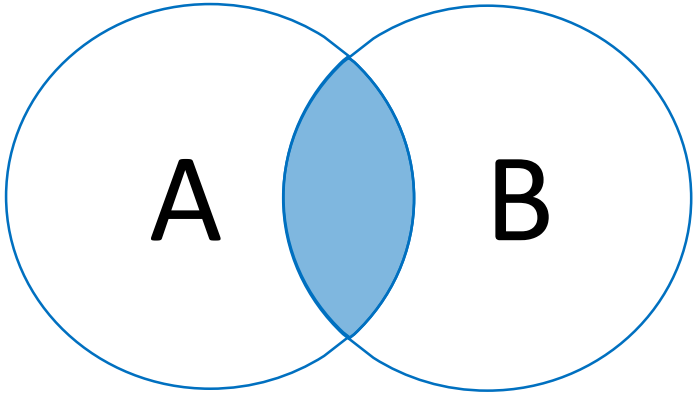
AlbumId	Title
1	For Those About To Rock We Salute You
3	Restless and Wild
4	Let There Be Rock

```
SELECT
     TrackId
     ,Name
     ,AlbumId
FROM Track_temp
ORDER BY AlbumId
```

TrackId	Name	AlbumId
8	Inject The Venom	1
2	Balls to the Wall	2
3	Fast As a Shark	3

INNER JOIN

- An INNER JOIN only returns values that have a record in both table A and table B
- The “INNER” key word is optional when using an inner join



INNER JOIN Example

- Joining the Album_temp and Track_temp tables
- Only rows returned are those where the AlbumID exists in both tables

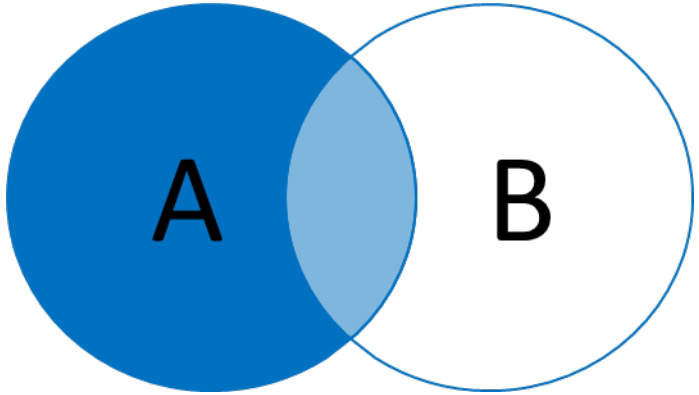
```
SELECT
    Album_temp.AlbumId
    ,Album_temp.Title
    ,Track_temp.TrackId
    ,Track_temp.Name
FROM Album_temp
JOIN Track_temp
    ON Album_temp.AlbumId = Track_temp.AlbumId
ORDER BY Album_temp.AlbumId
```

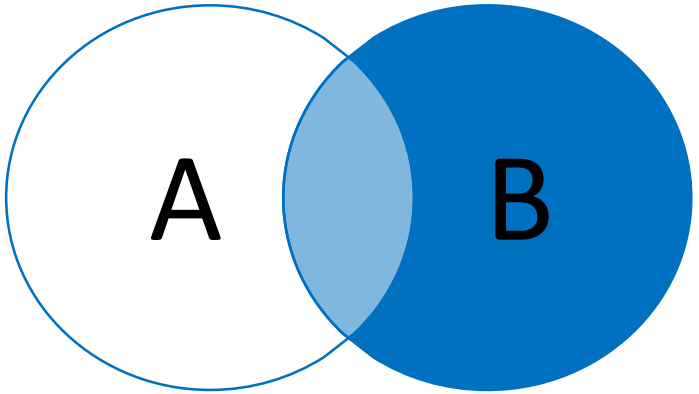
Results | Messages

	AlbumId	Title	TrackId	Name
1	1	For Those About To Rock We Salute You	8	Inject The Venom
2	3	Restless and Wild	3	Fast As a Shark

LEFT and RIGHT OUTER JOIN

- A LEFT OUTER JOIN returns all records from the table on left side of the join, and only records with a match on the right side of the join
- A RIGHT OUTER JOIN returns all records form the right side and only matching records from the left side
- The “OUTER” key word is optional when using an outer join
- LEFT JOINs are used much more in practice than RIGHT JOINs





LEFT JOIN Example

- All rows from the left table (Album_temp) are returned
- Only rows from the right table (Track_temp) with a matching AlbumId are returned

```
SELECT
    Album_temp.AlbumId
    ,Album_temp.Title
    ,Track_temp.TrackId
    ,Track_temp.Name
FROM Album_temp
LEFT JOIN Track_temp
    ON Album_temp.AlbumId = Track_temp.AlbumId
ORDER BY Album_temp.AlbumId
```

Results | Messages

	AlbumId	Title	TrackId	Name
1	1	For Those About To Rock We Salute You	8	Inject The Venom
2	3	Restless and Wild	3	Fast As a Shark
3	4	Let There Be Rock	NULL	NULL

RIGHT JOIN Example

- All rows from the right table (Track_temp) are returned
- Only rows from the left table (Album_temp) with a matching AlbumId are returned

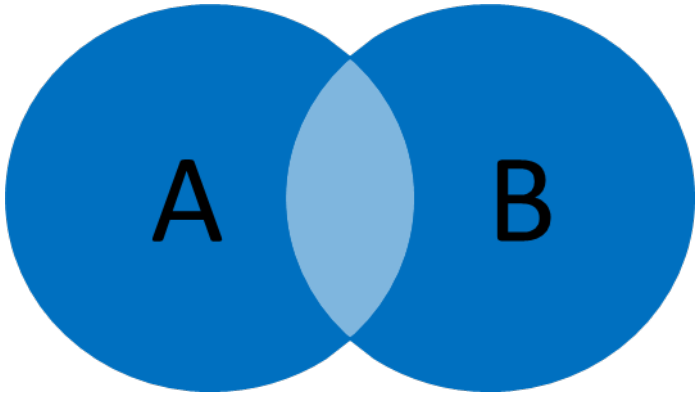
```
SELECT
    Album_temp.AlbumId
    ,Album_temp.Title
    ,Track_temp.TrackId
    ,Track_temp.Name
FROM Album_temp
RIGHT JOIN Track_temp
    ON Album_temp.AlbumId = Track_temp.AlbumId
ORDER BY Album_temp.AlbumId
```

Results | Messages

	AlbumId	Title	TrackId	Name
1	NULL	NULL	2	Balls to the Wall
2	1	For Those About To Rock We Salute You	8	Inject The Venom
3	3	Restless and Wild	3	Fast As a Shark

FULL OUTER JOIN

- A FULL OUTER JOIN returns all records from the table on left side of the join, and all records from the right side of the join
- The “OUTER” key word is optional when using a full outer join



FULL JOIN Example

- All rows from the left table (Album_temp) are returned
- All rows from the right table (Track_temp) are returned

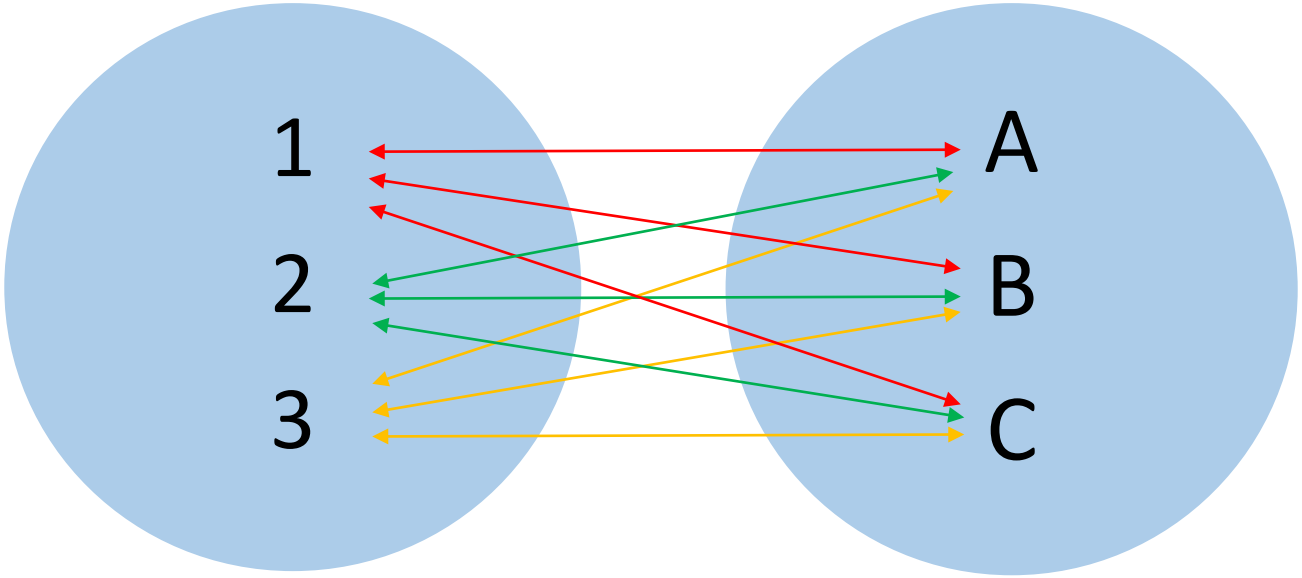
```
SELECT
    Album_temp.AlbumId
    ,Album_temp.Title
    ,Track_temp.TrackId
    ,Track_temp.Name
FROM Album_temp
FULL JOIN Track_temp
    ON Album_temp.AlbumId = Track_temp.AlbumId
ORDER BY Album_temp.AlbumId
```

Results | Messages

	AlbumId	Title	TrackId	Name
1	NULL	NULL	2	Balls to the Wall
2	1	For Those About To Rock We Salute You	8	Inject The Venom
3	3	Restless and Wild	3	Fast As a Shark
4	4	Let There Be Rock	NULL	NULL

CROSS JOIN (Cartesian Product)

- A CROSS JOIN matches every row in table A with every row in table B
- The ON keyword is not used with a CROSS JOIN as there is no selective matching done



CROSS JOIN Example

- Every row in the Album_temp table are matched with every row in the Track_temp table
- Each table has 3 rows so 3x3 returns 9 rows in the result set

```
SELECT
     Album_temp.AlbumId
     ,Album_temp.Title
     ,Track_temp.TrackId
     ,Track_temp.Name
FROM Album_temp
CROSS JOIN Track_temp
ORDER BY Album_temp.AlbumId
```

Results | Messages

	AlbumId	Title	TrackId	Name
1	1	For Those About To Rock We Salute You	2	Balls to the Wall
2	1	For Those About To Rock We Salute You	3	Fast As a Shark
3	1	For Those About To Rock We Salute You	8	Inject The Venom
4	3	Restless and Wild	2	Balls to the Wall
5	3	Restless and Wild	3	Fast As a Shark
6	3	Restless and Wild	8	Inject The Venom
7	4	Let There Be Rock	2	Balls to the Wall
8	4	Let There Be Rock	3	Fast As a Shark
9	4	Let There Be Rock	8	Inject The Venom

Table Aliases

- Table names can be aliased like column names
- The alias name can be substituted for the table name when identifying column objects
- The two select statements on this slide are identical. They return the same result set

```
SELECT
    Album_temp.AlbumId
    ,Album_temp.Title
    ,Track_temp.TrackId
    ,Track_temp.Name
FROM Album_temp
JOIN Track_temp
    ON Album_temp.AlbumId = Track_temp.AlbumId
ORDER BY Album_temp.AlbumId
```

```
SELECT
    A.AlbumId
    ,A.Title
    ,T.TrackId
    ,T.Name
FROM Album_temp AS A
JOIN Track_temp AS T
    ON A.AlbumId = T.AlbumId
ORDER BY A.AlbumId
```

Joining more than two tables

- Additional tables can be added to the query with additional JOIN clauses

```
SELECT
    A.Name AS ArtistName
    ,AB.Title AS AlbumName
    ,T.Name AS TrackName
FROM Artist A
JOIN Album AB
    ON AB.ArtistId = A.ArtistId
JOIN Track T
    ON T.AlbumId = AB.AlbumId
    AND T.Name LIKE 'The%'
WHERE A.Name = 'Iron Maiden'
ORDER BY
    AlbumName
    ,TrackName
```

Results | Messages

	ArtistName	AlbumName	TrackName
1	Iron Maiden	A Matter of Life and Death	The Legacy
2	Iron Maiden	A Matter of Life and Death	The Longest Day
3	Iron Maiden	A Matter of Life and Death	The Pilgrim
4	Iron Maiden	A Matter of Life and Death	The Reincarnation of Benjamii
5	Iron Maiden	A Matter of Life and Death	These Colours Don't Run
6	Iron Maiden	A Real Dead One	The Number Of The Beast
7	Iron Maiden	A Real Dead One	The Trooper
8	Iron Maiden	A Real Live One	The Clairvoyant
9	Iron Maiden	A Real Live One	The Evil That Men Do
10	Iron Maiden	Brave New World	The Fallen Angel

AND and OR keywords

- Use the AND and OR keywords after the ON keyword to further specify the relation between the tables in the JOIN clause
- AND and OR have the same syntax as in a WHERE clause

```
SELECT
    A.Name AS ArtistName
    ,AB.Title AS AlbumName
    ,T.Name AS TrackName
FROM Artist A
JOIN Album AB
    ON AB.ArtistId = A.ArtistId
LEFT JOIN Track T
    ON T.AlbumId = AB.AlbumId
    AND T.Name = 'The Longest Day'
WHERE A.Name = 'Iron Maiden'
```

Results | Messages

	ArtistName	AlbumName	TrackName
1	Iron Maiden	A Matter of Life and Death	The Longest Day
2	Iron Maiden	A Real Dead One	NULL
3	Iron Maiden	A Real Live One	NULL
4	Iron Maiden	Brave New World	NULL
5	Iron Maiden	Dance Of Death	NULL
6	Iron Maiden	Fear Of The Dark	NULL
7	Iron Maiden	Iron Maiden	NULL
8	Iron Maiden	Killers	NULL
9	Iron Maiden	Live After Death	NULL

Self Joins

- It is possible to join a table against itself
- The columns you join on should be related in some way
- In the example the ReportsTo column is the EmployeeId of an employee's manager
- It is mandatory to use table aliases when implementing self joins

```
SELECT
    E.LastName
    ,E.FirstName
    ,E.Title
    ,RT.LastName AS SupLastName
    ,RT.FirstName AS SupFirstName
    ,RT.Title AS SupTitle
FROM Employee E
LEFT JOIN Employee RT
    ON RT.EmployeeId = E.ReportsTo
```

Results | Messages

	LastName	FirstName	Title	SupLastName	SupFirstName	SupTitle
1	Adams	Andrew	General Manager	NULL	NULL	NULL
2	Edwards	Nancy	Sales Manager	Adams	Andrew	General Manager
3	Peacock	Jane	Sales Support Agent	Edwards	Nancy	Sales Manager
4	Park	Margaret	Sales Support Agent	Edwards	Nancy	Sales Manager
5	Johnson	Steve	Sales Support Agent	Edwards	Nancy	Sales Manager
6	Mitchell	Michael	IT Manager	Adams	Andrew	General Manager
7	King	Robert	IT Staff	Mitchell	Michael	IT Manager
8	Callahan	Laura	IT Staff	Mitchell	Michael	IT Manager

Linking tables using the WHERE clause

- You can join tables using only the FROM and WHERE clause
- Add tables to the FROM clause separated by a comma
- Add the linking logic to the WHERE clause
- This is not a good practice for two reasons
  - Only INNER JOINS are possible with this method
  - Merging filter and linking logic can make the WHERE clause hard to read

```
SELECT
    A.Name AS ArtistName
    ,AB.Title AS AlbumName
FROM Artist A
JOIN Album AB
    ON AB.ArtistId = A.ArtistId
WHERE A.Name = 'Iron Maiden'
    AND AB.Title LIKE 'The%'
```

Join Clause

```
SELECT
    A.Name AS ArtistName
    ,AB.Title AS AlbumName
FROM Artist A, Album AB
WHERE A.Name = 'Iron Maiden'
    AND AB.Title LIKE 'The%'
    AND AB.ArtistId = A.ArtistId
```

No Join Clause

Results | Messages

	ArtistName	AlbumName
1	Iron Maiden	The Number of The Beast
2	Iron Maiden	The X Factor

Summary

- Relational Database
- One to One
- One to Many
- Many to Many
- Entity Relationship Diagram
- Table Alias
- Relations in the WHERE clause
- INNER JOIN
- LEFT JOIN
- RIGHT JOIN
- FULL JOIN
- CROSS JOIN
- AND and OR keywords
- Multiple Joins
- Self Join