

Извличане на данни от множество таблици

Temporary tables (временни таблици)

- ▶ Local temporary table – имената им започват със знака #
- ▶ Global temporary table – имената им започват със знака ##
- ▶ Различия от постоянните таблици в БД:
 - Място на съхранение – съхраняват се в системната база от данни tempdb
 - Време на съществуване
 - Автоматично премахване

Пример:Извличане на данни за клиентите от London и съхраняването им в локална временна таблица (local temporary table)

```
USE Northwind
```

```
GO
```

```
/* създаване на локална временна таблица */
```

```
SELECT *
```

```
    INTO #Customers_London
```

```
    FROM dbo.Customers
```

```
        WHERE City LIKE '%London%'
```

Локалната временна таблицата се създава в БД tempdb и съществува до затваряне на конекцията на потребителя; тя е достъпна само в текущата конекция на текущия потребител.

Пример:Извличане на данни за клиентите от London и съхраняването им в глобална временна таблица (global temporary table)

```
USE Northwind
```

```
GO
```

```
/* създаване на глобална временна таблица */
```

```
SELECT *
```

```
    INTO ##Customers_London
```

```
    FROM dbo.Customers
```

```
    WHERE City LIKE '%London%'
```

Глобалната временна таблицата се създава в БД tempdb и съществува до затваряне на всички конекции със сървъра; тя е достъпна от всички отворени конекции от други потребители.

Извличане на данни от множество таблици в SQL

- ▶ Комбиниране на данни от множество таблици – чрез **съединяване** (клауза **Join** в оператор **Select**)
- ▶ Комбиниране на данни от множество резултатни набори – чрез **обединяване, разлика и сечение** на резултатните набори, получени от **Select**

Комбиниране на данни от множество таблици чрез Join

- ▶ Въведение в съединяването (Join)
- ▶ Използване на Inner Joins
- ▶ Използване на Outer Joins
- ▶ Съединяване на таблица сама със себе си
- ▶ Използване на Cross Joins
- ▶ Съединяване на повече от две таблици

Пример на SELECT с клауза Join

- Извеждане данни за клиентите (код, име) и направените от тях поръчки (номер и дата):

```
USE Northwind
SELECT C.CustomerId, CompanyName,
       OrderId, OrderDate
FROM Customers C INNER JOIN Orders OD
ON C.CustomerId = OD.CustomerId
GO
```

Частичен синтаксис на SELECT с клауза Join

▶ **SELECT** [ALL | **DISTINCT**]
[**TOP** *expression* [**PERCENT**]] <select_list>
FROM <table source> [[**AS**] table_alias]
<join type>**JOIN**
 <table source> [**AS** table_alias]
ON <search_condition>
[**WHERE** *search_condition*]
[**ORDER BY** *order_expression*
 [**ASC** | **DESC**][, ...]]

SELECTJOIN.....ON

- **<select_list>** – списък за избор (определя колоните, които ще бъдат селектирани в резултатния набор)
- Ключовата дума **JOIN** специфицира таблиците, които ще се съединяват и как ще се съединяват
- Ключовата дума **ON** задава условието на съединяването

SELECTJOIN.....ON

- ▶ Условието за съединяване на таблиците се задава въз основа на
 - общи колони от таблиците
 - първични и външни ключове
- Ако първичният ключ е съставен, в ON клаузата се задават условия за съединяване на всяка негова част, свързани с AND

Използване псевдоними (Aliases) за имената на таблиците

- ▶ Псевдонимът се задава след името на таблицата с ключовата дума AS или директно
 - Products AS P или Products P
- ▶ Псевдонимите се използват при задаване на уточнените имена на колоните от таблиците
 - P.ProductID равнозначно на Products.ProductId
- ▶ Изполването на псевдонимите повишава читаемостта на скриптовите и опростява писането на комплексни заявки

Типове съединяване (join type)

1

- ▶ Вътрешно съединяване (съединяване по равенство)

INNER JOIN

- ▶ Външно съединяване **OUTER JOIN**

- Ляво външно съединяване

LEFT OUTER JOIN

- Дясно външно съединяване

RIGHT OUTER JOIN

- Пълно външно съединяване

FULL OUTER JOIN

Типове съединяване (join type)

2

- ▶ Картезианско (Декартово) съединяване
CROSS JOIN

Всеки ред от едната таблица се съединява с всеки ред от другата таблица

- ▶ Съединяване на таблицата сама със себе си (self join)

Изисква таблицата да съдържа колона, която съответства на нейния първичен ключ или на алтернативен ключ.

Използване на Inner Join

1

- ▶ **Вътрешното съединяване INNER JOIN** комбинира редовете от таблиците, като сравнява стойностите в колоните, които са общи за двете таблици и връща само тези редове, за които е изпълнено зададеното условие
- ▶ **INNER JOIN** е тип на съединяване по подразбиране и може да се задава само с ключовата дума **JOIN**

Използване на Inner Join

2

- ▶ Колоните в списъка за избор и в условието за съединяване се задават с уточнени имена, ако има опасност от двусмислие
- ▶ В **условието за съединяване не** се препоръчва да се включват колони, които съдържат неопределени стойности (**null values**), тъй като те не са равни на нито една друга стойност
- ▶ Чрез WHERE могат да се ограничат редовете, които се включват в резултата

Пример: Извеждане на данни за стоките и техните поръчки (SQLQuery_join_Products.sql)

USE Northwind

GO

```
SELECT P.ProductID, ProductName  
, OrderID, Quantity, O.UnitPrice  
, Quantity*O.UnitPrice AS Total  
FROM Products P  
INNER JOIN [Order Details]O  
ON P.ProductID = O.ProductID
```


Какъв е резултатът, извеждан от SELECT?

USE Northwind

GO

```
SELECT O.OrderID, P.ProductID ,  
P.ProductName, O.Quantity, O.UnitPrice  
,O.Quantity*O.UnitPrice AS Total  
FROM Products P JOIN [Order Details] O  
ON P.ProductID = O.ProductID  
WHERE ProductName like '%Chocolade%'  
ORDER BY O.OrderID
```

Използване на OUTER JOIN

► OUTER JOIN

- Комбинираща редовете от двете таблици, които отговарят на зададеното чрез ON условие
- Включва и всички редове от едната от тях (лявата или дясната таблица), за които липсват съответни редове в другата таблица
- Редовете, които не срещат редове в другата таблица съдържат стойности NULL в колоните от нея

Използване на OUTER JOIN

2

- Таблицата се определя като лява или дясна зависимост от мястото, което заема по отношение на ключовата дума JOIN
- Ляво външно съединяване
LEFT OUTER JOIN
от лявата таблица се вземат всички редове и се комбинират с редове от дясната таблица, само тези от тях, за които е изпълнено условието

Използване на OUTER JOIN

- Дясно външно съединяване
RIGHT OUTER JOIN
от дясната таблица се вземат всички редове и се комбинират с редове от лявата таблица само тези от тях, за които е изпълнено условието
- **OUTER** може да се изпусне, т. е. да се зададе само **LEFT JOIN** или **RIGHT JOIN**
- **OUTER JOIN** може да се използва само за съединяване на две таблици

Данни за поръчките на стоките

SELECT

P.ProductID, P.ProductName,
OD.OrderID, Quantity , OD.UnitPrice

FROM Products P

INNER JOIN [Order Details] OD

ON P.ProductID = OD.ProductID

Данни за всички стоки, независимо дали са поръчани

```
SELECT P.ProductID,P.ProductName  
      ,OD.OrderID, Quantity , OD.UnitPrice  
FROM Products AS P  
      LEFT JOIN [Order Details] OD  
      ON P.ProductID = OD.ProductID
```

Лявото външно съединяване на таблиците извлича всички редове от таблицата Products, а от Order Details се присъединяват данни само за стоките, за които има поръчки.

Стоки, които не са поръчани

- ▶ В резултата за стоките, които не са поръчвани, в колоните OrderID (номер на поръчката) и количество (Quantity) липсват стойности.

```
SELECT P.ProductID, ProductName  
,OD.OrderID, Quantity, OD.UnitPrice  
FROM Products AS P  
LEFT JOIN [Order Details] OD  
ON P.ProductID = OD.ProductID  
WHERE ((OD.OrderID) Is Null)
```

Данни за всички стоки, независимо дали са поръчани с използване на **RIGHT JOIN**

```
SELECT P.ProductID, ProductName  
        , OrderID, Quantity, OD.UnitPrice  
FROM [Order Details] OD  
        RIGHT JOIN Products AS P  
        ON P.ProductID = OD.ProductID
```


Пълно външно съединение

FULL OUTER JOIN

- ▶ В резултата се включват всички редове от двете таблици A и B, като се комбинират тези от тях, за които е изпълнено условието за съединяване
- ▶ Ако ред от таблицата A не среща съответен ред в таблица B, полетата от таблицата B, които се извеждат в резултатния набор приемат стойности Null
- ▶ Ако ред от таблицата B не среща съответен ред в таблица A, полетата от таблицата A, които се извеждат в резултатния набор приемат стойности Null

Съединяване на таблицата сама със себе си (SELF JOIN)

- ▶ Изискване: таблицата трябва да съдържа колона (FK), която е обща с друга колона от нея (PK или алтернативен ключ)
- ▶ Особености:
 - В JOIN се съединяват две копия на таблицата, за които се задават различни псевдоними
 - Всеки ред от таблицата се среща и със себе си и в резултата могат да се получат редове с дублирано съдържание; за да се отстранят се използва клауза WHERE

Microsoft SQL Server Management Studio

File Edit View Query Project Debug Tools Window Community Help

New Query | [Icons] | Change Type | [Icons] | Northwind | Execute | [Icons]

SQLQuery2.sql - M...ia-PC\Maria (53)* | SQLQuery1.sql - M...ia-PC\Maria (52)*

```
SELECT EmployeeID, FirstName  
       , LastName  
       , Title, ReportsTo  
FROM Employees
```

Results | Messages

	EmployeeID	FirstName	LastName	Title	ReportsTo
1	1	Nancy	Davolio	Sales Representative	2
2	2	Andrew	Fuller	Vice President, Sales	NULL
3	3	Janet	Leverling	Sales Representative	2
4	4	Margaret	Peacock	Sales Representative	2
5	5	Steven	Buchanan	Sales Manager	2
6	6	Michael	Suyama	Sales Representative	5
7	7	Robert	King	Sales Representative	5
8	8	Laura	Callahan	Inside Sales Coordinator	2
9	9	Anne	Dodsworth	Sales Representative	5

Query executed succes... | MARIA-PC (10.0 RTM) | Maria-PC\Maria (53) | Northwind | 00:00:00 | 9 rows

Ready | Ln 1 | Col 1 | INS

Пример: Справка за кодовете и имената на служителите и имената и длъжностите на техните преки ръководители

```
SELECT E1.EmployeeID, E1.FirstName  
      , E1.LastName  
      , (E2.FirstName + ' ' + E2.LastName)  
                                AS Supervisor  
      , E2.Title AS Supervisor_title  
FROM Employees E1  
      INNER JOIN Employees AS E2  
      ON E1.[ReportsTo] = E2.EmployeeID  
ORDER BY 4
```

- ▶ За ръководителите от най-високо ниво не се извеждат данни.

Microsoft SQL Server Management Studio

File Edit View Query Project Debug Tools Window Community Help

New Query | Change Type | Northwind | Execute

SQLQuery2.sql - M...ia-PC\Maria (53)* | SQLQuery1.sql - M...ia-PC\Maria (52)*

```

SELECT E1.EmployeeID, E1.FirstName
      , E1.LastName
      , (E2.FirstName + ' ' + E2.LastName)
AS Supervisor
      , E2.Title AS Supervisor_title
FROM Employees E1 INNER JOIN Employees AS E2
ON E1.ReportsTo = E2.EmployeeID
ORDER BY 4

```

Results | Messages

	EmployeeID	FirstName	LastName	Supervisor	Supervisor_title
1	1	Nancy	Davolio	Andrew Fuller	Vice President, Sales
2	3	Janet	Leverling	Andrew Fuller	Vice President, Sales
3	4	Margaret	Peacock	Andrew Fuller	Vice President, Sales
4	5	Steven	Buchanan	Andrew Fuller	Vice President, Sales
5	8	Laura	Callahan	Andrew Fuller	Vice President, Sales
6	9	Anne	Dodsworth	Steven Buchanan	Sales Manager
7	6	Michael	Suyama	Steven Buchanan	Sales Manager
8	7	Robert	King	Steven Buchanan	Sales Manager

Query executed succes... | MARIA-PC (10.0 RTM) | Maria-PC\Maria (52) | Northwind | 00:00:00 | 8 rows

Ready | Ln 1 | Col 1 | INS

Пример: Справка за кодовете, имената и длъжностите на служителите–ръководители и имената и длъжностите на техните подчинени

USE Northwind

GO

```
SELECT (E1.FirstName + ' ' + E1.LastName)
        AS Supervisor, E1.Title
        , E2.EmployeeID, E2.FirstName
        , E2.LastName, E2.Title
FROM Employees E1
INNER JOIN Employees AS E2
ON E1.EmployeeID = E2.[ReportsTo]
```

Microsoft SQL Server Management Studio

File Edit View Query Project Debug Tools Window Community Help

New Query

Northwind

Execute

Object Explorer

Connect

- AdventureWorksLT2008
- Northwind
 - Database Diagrams
 - Tables
 - System Tables
 - dbo.Categories
 - dbo.CustomerCustom
 - dbo.CustomerDemog
 - dbo.Customers
 - dbo.Employees
 - Columns
 - EmployeeID (P
 - LastName (nva
 - FirstName (nva
 - Title (nvarchar
 - TitleOfCourtes
 - BirthDate (date
 - HireDate (date
 - Address (nvarchar
 - City (nvarchar
 - Region (nvarchar
 - PostalCode (nv
 - Country (nvarchar
 - HomePhone (n
 - Extension (nva
 - Photo (image,
 - Notes (ntext, n
 - ReportsTo (FK,
 - PhotoPath (nv
 - Keys
 - Constraints
 - Triggers

SQLQuery15.sql - ...PC\Kasheva (53))* SQLQuery14.sql - ...PC\Kasheva (54)* SQLQuery12.sql - ...PC\Kasheva (52)*

```

SELECT (E1.FirstName + ' ' + E1.LastName)
      AS Supervisor, E1.Title
      , E2.EmployeeID, E2.FirstName, E2.LastName , E2.Title
FROM Employees E1 INNER JOIN Employees AS E2
ON E1.EmployeeID = E2.[ReportsTo]
  
```

Results

	Supervisor	Title	EmployeeID	FirstName	LastName	Title
1	Andrew Fuller	Vice President, Sales	1	Nancy	Davolio	Sales Representative
2	Andrew Fuller	Vice President, Sales	3	Janet	Leverling	Sales Representative
3	Andrew Fuller	Vice President, Sales	4	Margaret	Peacock	Sales Representative
4	Andrew Fuller	Vice President, Sales	5	Steven	Buchanan	Sales Manager
5	Steven Buchanan	Sales Manager	6	Michael	Suyama	Sales Representative
6	Steven Buchanan	Sales Manager	7	Robert	King	Sales Representative
7	Andrew Fuller	Vice President, Sales	8	Laura	Callahan	Inside Sales Coordinator
8	Steven Buchanan	Sales Manager	9	Anne	Dodsworth	Sales Representative

Query executed successfully.

KASHEVA-PC (10.0 RTM) | Kasheva-PC\Kasheva (54) | Northwind | 00:00:00 | 8 rows

Ready

Ln 7 Col 1 Ch 1 INS

Справка за служителите с една и съща длъжност

```
SELECT E1.EmployeeID, E1.LastName  
      , Left(E1.Title,10) AS Title_1  
      , E2.EmployeeID, E2.LastName  
      , Left(E2.Title,10) AS Title_2  
FROM Employees AS E1 INNER JOIN  
                        Employees AS E2  
  ON E1.Title = E2.Title  
WHERE NOT E1.EmployeeID=E2.EmployeeID  
ORDER BY E1.EmployeeID, E2.EmployeeID
```


Microsoft SQL Server Management Studio

File Edit View Query Project Debug Tools Window Community Help

New Query

Northwind

Execute

Object Explorer

Connect

- AdventureWorksLT2008
- Northwind
 - Database Diagrams
 - Tables
 - System Tables
 - dbo.Categories
 - dbo.CustomerCustomerDemo
 - dbo.CustomerDemographics
 - dbo.Customers
 - dbo.Employees
 - Columns
 - EmployeeID (PK, int, not null)
 - LastName (nvarchar(20), not null)
 - FirstName (nvarchar(10), not null)
 - Title (nvarchar(30), null)
 - TitleOfCourtesy (nvarchar(25), null)
 - BirthDate (datetime, null)
 - HireDate (datetime, null)
 - Address (nvarchar(60), null)
 - City (nvarchar(15), null)
 - Region (nvarchar(15), null)
 - PostalCode (nvarchar(10), null)
 - Country (nvarchar(15), null)
 - HomePhone (nvarchar(24), null)
 - Extension (nvarchar(4), null)
 - Photo (image, null)
 - Notes (ntext, null)
 - ReportsTo (FK, int, null)
 - PhotoPath (nvarchar(255), null)
 - Keys
 - Constraints
 - Triggers

SQLQuery13.sql - ...PC\Kasheva (53)*

```

SELECT E1.EmployeeID, E1.LastName, Left(E1.Title,10) AS Title_1
, E2.EmployeeID, E2.LastName
, Left(E2.Title,10) AS Title_2
FROM Employees AS E1 INNER JOIN Employees AS E2
ON E1.Title = E2.Title
WHERE NOT E1.EmployeeID=E2.EmployeeID
ORDER BY E1.EmployeeID, E2.EmployeeID
  
```

Results

	EmployeeID	LastName	Title_1	EmployeeID	LastName	Title_2
1	1	Davolio	Sales Repr	3	Leverling	Sales Repr
2	1	Davolio	Sales Repr	4	Peacock	Sales Repr
3	1	Davolio	Sales Repr	6	Suyama	Sales Repr
4	1	Davolio	Sales Repr	7	King	Sales Repr
5	1	Davolio	Sales Repr	9	Dodsworth	Sales Repr
6	3	Leverling	Sales Repr	1	Davolio	Sales Repr
7	3	Leverling	Sales Repr	4	Peacock	Sales Repr
8	3	Leverling	Sales Repr	6	Suyama	Sales Repr
9	3	Leverling	Sales Repr	7	King	Sales Repr

Query executed successfully.

KASHEVA-PC (10.0 RTM) | Kasheva-PC\Kasheva (53) | Northwind | 00:00:00 | 30 rows

Ready

Ln 1 Col 1 INS

Декартово (картезианско) съединяване) CROSS JOIN

- ▶ CROSS JOIN извежда всяка комбинация от всички редове на двете таблици
- ▶ Не се изисква двете таблици да имат обща колона
- ▶ Не се задава клауза ON
- ▶ Броят на редовете в резултата е произведение от броя на редовете в двете таблици

Пример за CROSS JOIN

На основата на таблица с данни за футболните отбори от А група да се генерират комбинациите „Домакин–Гост“.

/*Създаване на таблица с данни за футболните отбори от А група*/

CREATE TABLE Teams

(TeamsName NCHAR(25)

, City NCHAR(20)

CONSTRAINT PKey **Primary Key**(City,TeamsName))

Генериране на комбинации „Домакин – Гост“

```
SELECT H.TeamsName AS Home  
, A.TeamsName AS Guest  
FROM Teams H CROSS JOIN Teams A  
WHERE H.TeamsName != A.TeamsName
```

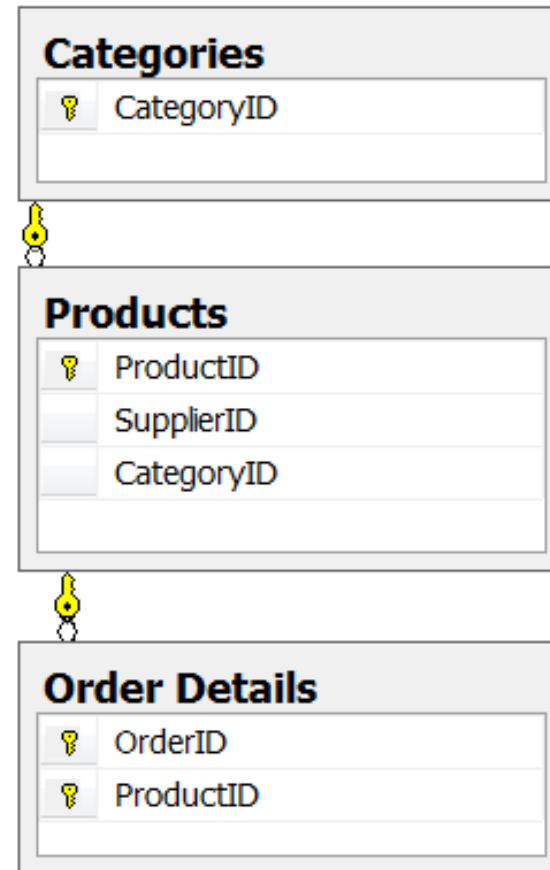
```
SELECT H.TeamsName AS Home  
, A.TeamsName AS Guest  
INTO Teams_Home_Guest  
FROM Teams H CROSS JOIN Teams A  
WHERE H.TeamsName != A.TeamsName
```

Съединяване на повече от 2 таблици

- ▶ В един SELECT могат да се съединяват произволен брой таблици
- ▶ Шаблони за съединения на няколко таблици:
 - JOIN верига
 - JOIN звезда – една централна таблица и няколко други таблици по върховете на звездата, които се свързват с редовете на централната таблица

Пример на JOIN верига

- ▶ Да се изведат данни за поръчките на стоки, които включват код и име на категорията стоки, код и име на стоката, номер на поръчката и стойност на поръчаното количество.



Пример на съединяване на верига от 3
таблици: извеждане данни за поръчките на
стоки и тяхната категория

USE Northwind

GO

SELECT C.CategoryID, CategoryName
,P.ProductID, ProductName,OD.OrderID
,OD.UnitPrice*Quantity AS Total

FROM **Categories C JOIN Products P**

ON C.CategoryID = P.CategoryID

JOIN [Order Details] OD

ON P.ProductID = OD.ProductID

Пример:Извеждане на данни за всички категории и всички продукти и за техните поръчки

USE Northwind

GO

SELECT C.CategoryID, CategoryName
,P.ProductID, ProductName
,OD.OrderID

,OD.UnitPrice*Quantity AS Total

FROM Categories C LEFT JOIN Products P

ON C.CategoryID = P.CategoryID

LEFT JOIN [Order Details] OD

ON P.ProductID = OD.ProductID

ORDER BY 1, 4

Пример: Извеждане на данни за всички продукти – категория, номера на поръчките и тяхната стойност

USE Northwind

GO

SELECT C.CategoryID, CategoryName

,P.ProductID, ProductName

,OD.OrderID

,OD.UnitPrice*Quantity AS Total

FROM Categories C JOIN Products P

ON C.CategoryID = P.CategoryID

LEFT JOIN [Order Details] OD

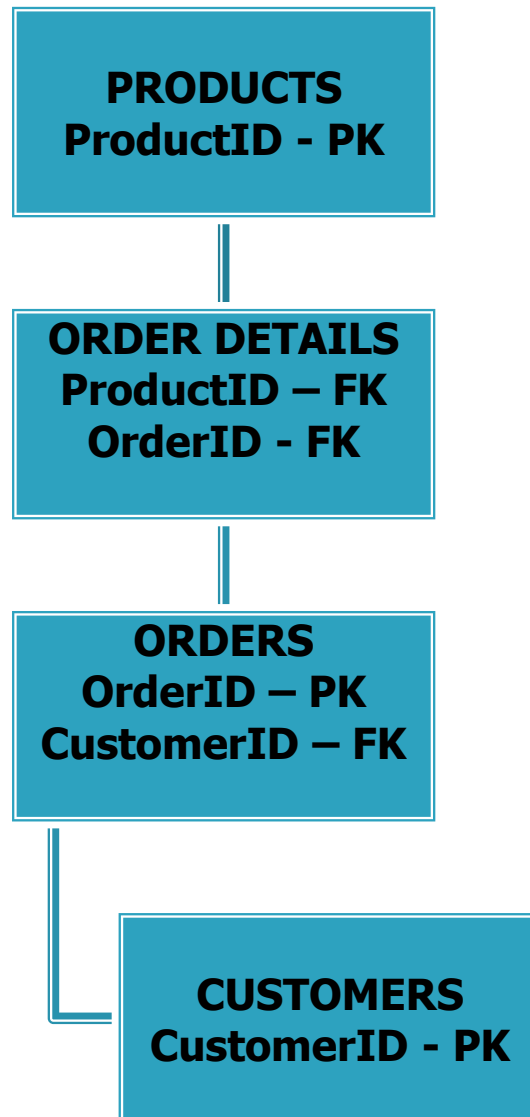
ON P.ProductID = OD.ProductID

ORDER BY 1, 4

Особеност при използване на външно съединение в JOIN верига

- ▶ Ако се използва LEFT JOIN за първата двойка във веригата, то този тип съединяване трябва да се зададе и за всяко от следващите съединявания във веригата.
- ▶ Ако се използва INNER JOIN за поредицата съединявания, то за последното съединяване от веригата може да се зададе външно съединяване (ляво или дясно зависи от случая).

Пример на JOIN верига от 4 таблици

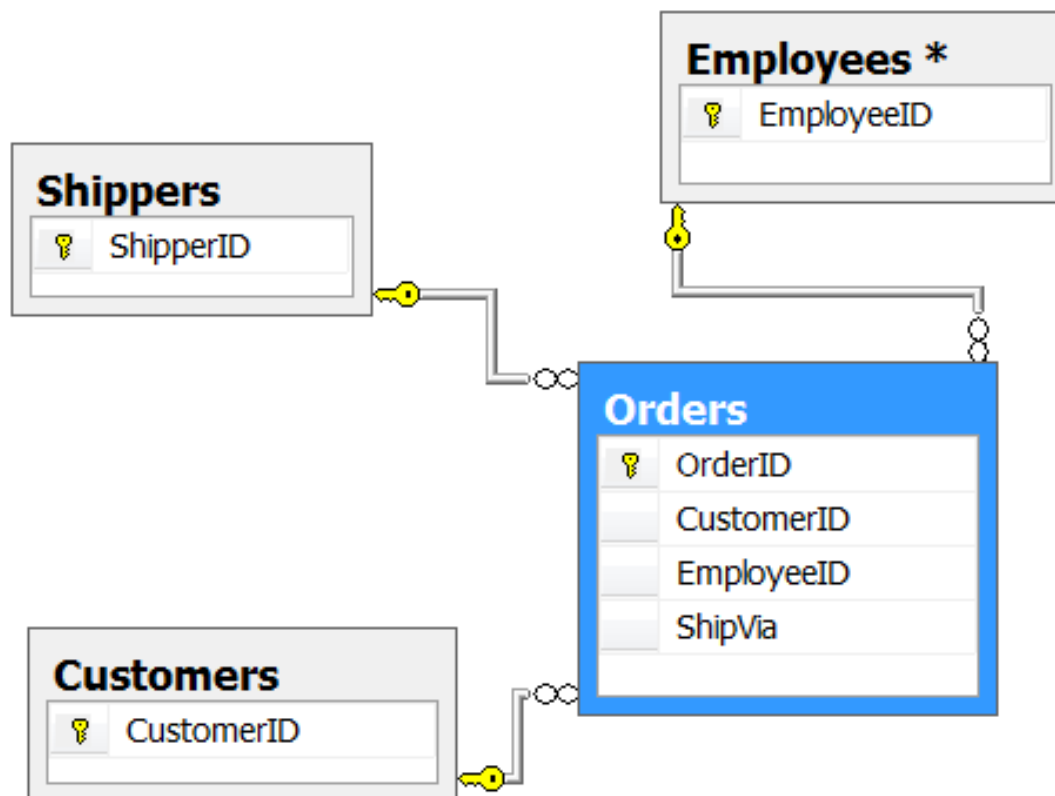


Пример на съединяване на 4 таблици

```
SELECT C.CompanyName, OD.ProductID  
  , P.ProductName, P.UnitPrice  
  , OD.Quantity, O.OrderID, O.OrderDate  
FROM Products AS P JOIN [Order Details] AS OD  
ON P.ProductID = OD.ProductID  
INNER JOIN Orders AS O  
ON OD.OrderID = O.OrderID  
INNER JOIN Customers AS C  
ON O.CustomerID = C.CustomerID  
ORDER BY C.CompanyName, O.OrderDate
```

Пример на JOIN звезда

- За всяка направена поръчка да се изведат номер и дата на поръчката, име на клиента, двете имена на служителя, приел поръчката и името на търговеца-превозвач.



```
USE Northwind
GO
SELECT O.OrderID, OrderDate, C.CompanyName
AS Customer
, FirstName+" "+LastName AS Employee_Name
, SH.CompanyName AS Shipper
FROM Orders O JOIN Employees E
ON O.EmployeeID = E.EmployeeID
JOIN Shippers SH
ON O.ShipVia = SH.ShipperID
JOIN Customers C
ON O.CustomerID = C.CustomerID
ORDER BY 3,1
```

Данни за всички клиенти, направените от тях поръчки
и търговците-превозвачи

(Използване на външно съединяване в JOIN звезда)

```
USE Northwind
```

```
GO
```

```
SELECT C.CompanyName AS Customer
```

```
, FirstName+" "+LastName AS Employee_Name
```

```
, SH.CompanyName AS Shipper, O.OrderID, OrderDate
```

```
FROM Orders O
```

```
JOIN Employees E
```

```
ON O.EmployeeID = E.EmployeeID
```

```
JOIN Shippers SH
```

```
ON O.ShipVia = SH.ShipperID
```

```
RIGHT JOIN Customers C
```

```
ON O.CustomerID = C.CustomerID
```

```
ORDER BY I
```