Retrieving Columns from a Table or View

- Use SELECT with column list to display columns
- Use FROM to specify a source table or view
  - Specify both schema and table names
- Delimit names if necessary

SQL Server only

```
[Sales Order Details]
```

- End all statements with a semicolon

```
;
```

Keyword	Expression
SELECT	**<select list>**
FROM	**<table source>**

```
SELECT companyname, country
FROM Sales.Customers;
```

Retrieving Columns from a Table or View

SELECT Syntax

```
SELECT ...
FROM <table> AS <alias>;
```

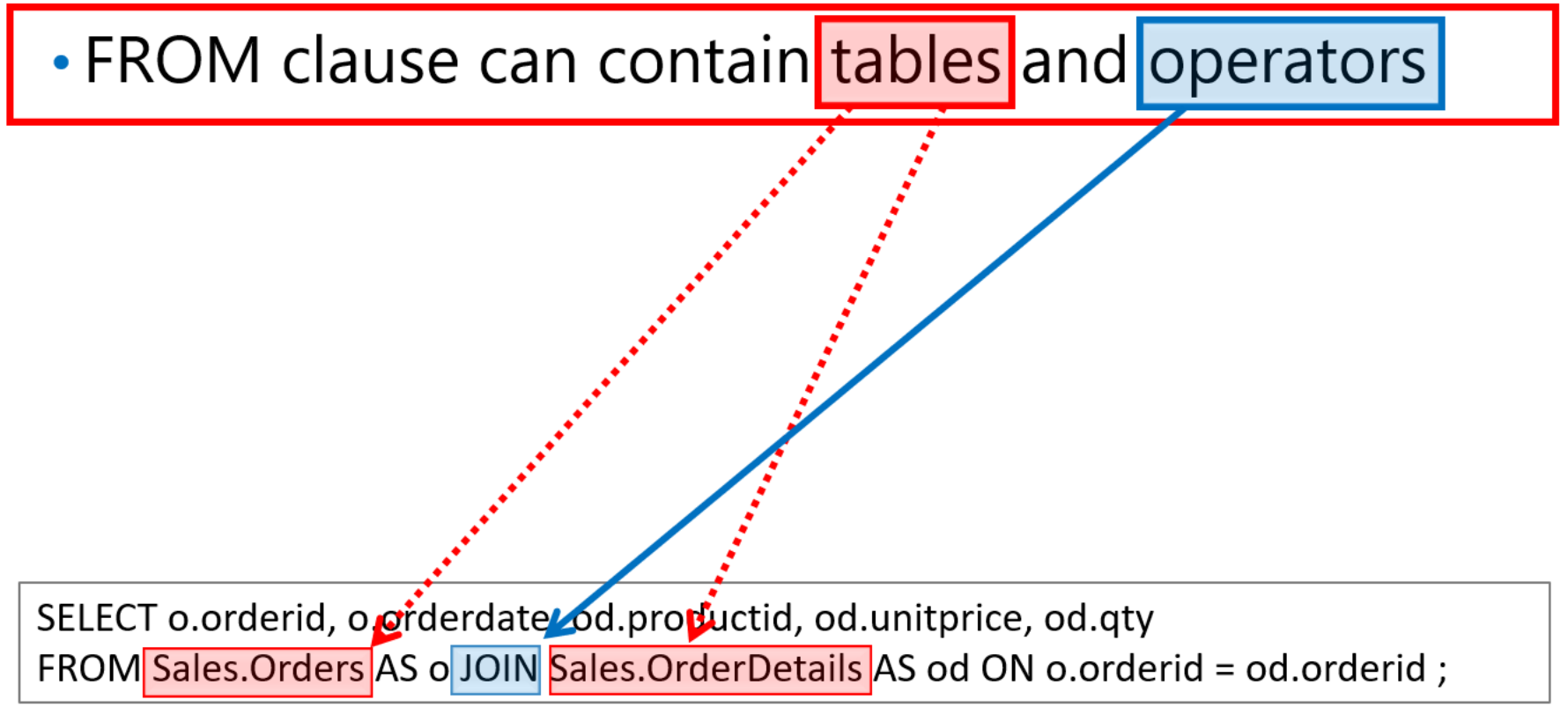
```
SELECT o.orderid, o.orderdate, od.productid, od.unitprice, od.qty
FROM Sales.Orders AS o
        JOIN Sales.OrderDetails AS od ON o.orderid = od.orderid ;
```

With alias → **Improved readibility**

Without alias

```
SELECT Sales.Orders.orderid, Sales.Orders.orderdate,
                    Sales.OrderDetails.productid,Sales.OrderDetails.unitprice,
                    Sales.OrderDetails.qty
FROM Sales.Orders
        JOIN Sales.OrderDetails ON Sales.Orders.orderid = Sales.OrderDetails.orderid;
```

- FROM clause determines source tables to be used in SELECT statement
- FROM clause can contain tables and operators



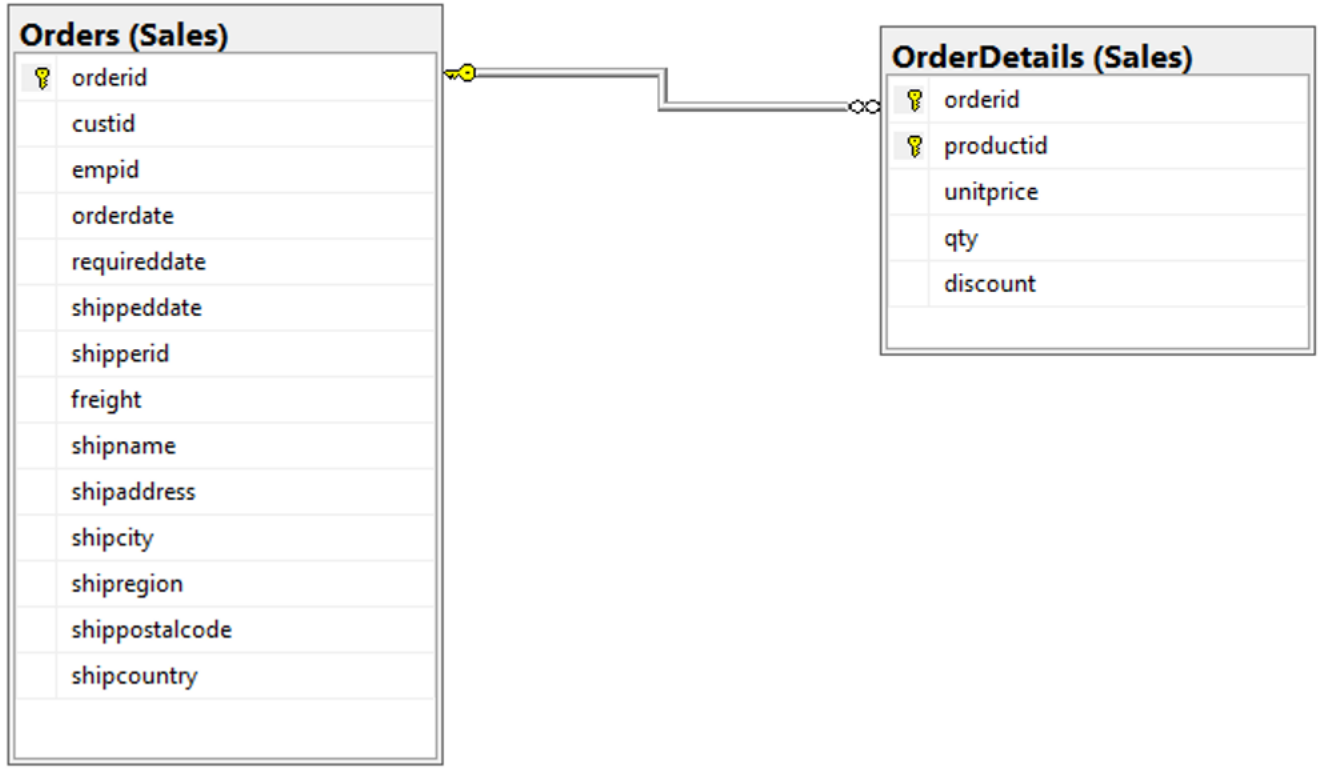
```
SELECT *
FROM Sales.Orders ;
```

```
SELECT *
FROM Sales.OrderDetails ;
```

JOIN

orderid	custid	empid	orderdate
10248	85	5	2006-07-04 00:00:00.000
10249	79	6	2006-07-05 00:00:00.000
10250	34	4	2006-07-08 00:00:00.000
10251	84	3	2006-07-08 00:00:00.000
10252	76	4	2006-07-09 00:00:00.000
10253	34	3	2006-07-10 00:00:00.000
10254	14	5	2006-07-11 00:00:00.000
10255	68	9	2006-07-12 00:00:00.000
10256	88	3	2006-07-15 00:00:00.000

orderid	productid	unitprice	qty	discount
10248	11	14,00	12	0.000
10248	42	9,80	10	0.000
10248	72	34,80	5	0.000
10249	14	18,60	9	0.000
10249	51	42,40	40	0.000
10250	41	7,70	10	0.000
10250	51	42,40	35	0.150
10250	65	16,80	15	0.150
10251	22	16,80	6	0.050
10251	57	15,60	15	0.050
10251	65	16,80	20	0.000
10252	20	64,80	40	0.050


Orders (Sales)
orderid
custid
empid
orderdate
requireddate
shippeddate
shipperid
freight
shipname
shipaddress
shipcity
shipregion
shippostalcode
shipcountry
OrderDetails (Sales)
orderid
productid
unitprice
qty
discount

```
CREATE TABLE Sales.Orders
(
  orderid        INT           NOT NULL IDENTITY,
  custid         INT           NULL,
  empid          INT           NOT NULL,
  orderdate      DATETIME      NOT NULL,
  requireddate   DATETIME      NOT NULL,
  shippeddate    DATETIME      NULL,
  shipperid      INT           NOT NULL,
  freight        MONEY         NOT NULL
    CONSTRAINT DFT_Orders_freight DEFAULT(0),
  shipname       NVARCHAR(40) NOT NULL,
  shipaddress    NVARCHAR(60) NOT NULL,
  shipcity       NVARCHAR(15) NOT NULL,
  shipregion     NVARCHAR(15) NULL,
  shippostalcode NVARCHAR(10) NULL,
  shipcountry    NVARCHAR(15) NOT NULL,
  CONSTRAINT PK_Orders PRIMARY KEY(orderid),
  CONSTRAINT FK_Orders_Customers FOREIGN KEY(custid)
    REFERENCES Sales.Customers(custid),
  CONSTRAINT FK_Orders_Employees FOREIGN KEY(empid)
    REFERENCES HR.Employees(empid),
  CONSTRAINT FK_Orders_Shippers FOREIGN
KEY(shipperid)
    REFERENCES Sales.Shippers(shipperid)
);
```

```
CREATE TABLE Sales.OrderDetails
(
  orderid   INT           NOT NULL,
  productid INT           NOT NULL,
  unitprice MONEY         NOT NULL
    CONSTRAINT DFT_OrderDetails_unitprice
DEFAULT(0),
  qty       SMALLINT      NOT NULL
    CONSTRAINT DFT_OrderDetails_qty DEFAULT(1),
  discount  NUMERIC(4, 3) NOT NULL
    CONSTRAINT DFT_OrderDetails_discount DEFAULT(0),
  CONSTRAINT PK_OrderDetails PRIMARY KEY(orderid,
productid),
  CONSTRAINT FK_OrderDetails_Orders FOREIGN
KEY(orderid)
    REFERENCES Sales.Orders(orderid),
  CONSTRAINT FK_OrderDetails_Products FOREIGN
KEY(productid)
    REFERENCES Production.Products(productid),
  CONSTRAINT CHK_discount  CHECK (discount BETWEEN 0
AND 1),
  CONSTRAINT CHK_qty  CHECK (qty > 0),
  CONSTRAINT CHK_unitprice CHECK (unitprice >= 0)
);
```

FROM Sales.Orders AS o JOIN Sales.OrderDetails AS od ON

ON o.orderid = od.orderid

orderid	custid	empid	orderdate	requireddate	shippeddate	shipperid	freight	shipname	shipaddress	shipcity	shipregion	shippostalcode	shipcountry	orderid	productid	unitprice	qty	discount
10248	85	5	2006-07-04 00:00:00.000	2006-08-01 00:00:00.000	2006-07-16 00:00:00.000	3	32,38	Ship to 85-B	6789 rue de l'Abbaye	Reims	NULL	10345	France	10248	11	14,00	12	0.000
10248	85	5	2006-07-04 00:00:00.000	2006-08-01 00:00:00.000	2006-07-16 00:00:00.000	3	32,38	Ship to 85-B	6789 rue de l'Abbaye	Reims	NULL	10345	France	10248	42	9,80	10	0.000
10248	85	5	2006-07-04 00:00:00.000	2006-08-01 00:00:00.000	2006-07-16 00:00:00.000	3	32,38	Ship to 85-B	6789 rue de l'Abbaye	Reims	NULL	10345	France	10248	72	34,80	5	0.000
10249	79	6	2006-07-05 00:00:00.000	2006-08-16 00:00:00.000	2006-07-10 00:00:00.000	1	11,61	Ship to 79-C	Luisenstr. 9012	Münster	NULL	10328	Germany	10249	14	18,60	9	0.000
10249	79	6	2006-07-05 00:00:00.000	2006-08-16 00:00:00.000	2006-07-10 00:00:00.000	1	11,61	Ship to 79-C	Luisenstr. 9012	Münster	NULL	10328	Germany	10249	51	42,40	40	0.000
10250	34	4	2006-07-08 00:00:00.000	2006-08-05 00:00:00.000	2006-07-12 00:00:00.000	2	65,83	Destination SCQXA	Rua do Paço, 7890	Rio de Janeiro	RJ	10195	Brazil	10250	41	7,70	10	0.000
10250	34	4	2006-07-08 00:00:00.000	2006-08-05 00:00:00.000	2006-07-12 00:00:00.000	2	65,83	Destination SCQXA	Rua do Paço, 7890	Rio de Janeiro	RJ	10195	Brazil	10250	51	42,40	35	0.150
10250	34	4	2006-07-08 00:00:00.000	2006-08-05 00:00:00.000	2006-07-12 00:00:00.000	2	65,83	Destination SCQXA	Rua do Paço, 7890	Rio de Janeiro	RJ	10195	Brazil	10250	65	16,80	15	0.150
10251	84	3	2006-07-08 00:00:00.000	2006-08-05 00:00:00.000	2006-07-15 00:00:00.000	1	41,34	Ship to 84-A	3456, rue du Commerce	Lyon	NULL	10342	France	10251	22	16,80	6	0.050
10251	84	3	2006-07-08 00:00:00.000	2006-08-05 00:00:00.000	2006-07-15 00:00:00.000	1	41,34	Ship to 84-A	3456, rue du Commerce	Lyon	NULL	10342	France	10251	57	15,60	15	0.050
10251	84	3	2006-07-08 00:00:00.000	2006-08-05 00:00:00.000	2006-07-15 00:00:00.000	1	41,34	Ship to 84-A	3456, rue du Commerce	Lyon	NULL	10342	France	10251	65	16,80	20	0.000
10252	76	4	2006-07-09 00:00:00.000	2006-08-06 00:00:00.000	2006-07-11 00:00:00.000	2	51,30	Ship to 76-B	Boulevard Tirou, 9012	Charleroi	NULL	10318	Belgium	10252	20	64,80	40	0.050
10252	76	4	2006-07-09 00:00:00.000	2006-08-06 00:00:00.000	2006-07-11 00:00:00.000	2	51,30	Ship to 76-B	Boulevard Tirou, 9012	Charleroi	NULL	10318	Belgium	10252	33	2,00	25	0.050
10252	76	4	2006-07-09 00:00:00.000	2006-08-06 00:00:00.000	2006-07-11 00:00:00.000	2	51,30	Ship to 76-B	Boulevard Tirou, 9012	Charleroi	NULL	10318	Belgium	10252	60	27,20	40	0.000
10253	34	3	2006-07-10 00:00:00.000	2006-07-24 00:00:00.000	2006-07-16 00:00:00.000	2	58,17	Destination JPAIY	Rua do Paço, 8901	Rio de Janeiro	RJ	10196	Brazil	10253	31	10,00	20	0.000
10253	34	3	2006-07-10 00:00:00.000	2006-07-24 00:00:00.000	2006-07-16 00:00:00.000	2	58,17	Destination JPAIY	Rua do Paço, 8901	Rio de Janeiro	RJ	10196	Brazil	10253	39	14,40	42	0.000
10253	34	3	2006-07-10 00:00:00.000	2006-07-24 00:00:00.000	2006-07-16 00:00:00.000	2	58,17	Destination JPAIY	Rua do Paço, 8901	Rio de Janeiro	RJ	10196	Brazil	10253	49	16,00	40	0.000
10254	14	5	2006-07-11 00:00:00.000	2006-08-08 00:00:00.000	2006-07-23 00:00:00.000	2	22,98	Destination YUJRD	Hauptstr. 1234	Bern	NULL	10139	Switzerland	10254	24	3,60	15	0.150
10254	14	5	2006-07-11 00:00:00.000	2006-08-08 00:00:00.000	2006-07-23 00:00:00.000	2	22,98	Destination YUJRD	Hauptstr. 1234	Bern	NULL	10139	Switzerland	10254	55	19,20	21	0.150
10254	14	5	2006-07-11 00:00:00.000	2006-08-08 00:00:00.000	2006-07-23 00:00:00.000	2	22,98	Destination YUJRD	Hauptstr. 1234	Bern	NULL	10139	Switzerland	10254	74	8,00	21	0.000
10255	68	9	2006-07-12 00:00:00.000	2006-08-09 00:00:00.000	2006-07-15 00:00:00.000	3	148,33	Ship to 68-A	Starenweg 6789	Genève	NULL	10294	Switzerland	10255	2	15,20	20	0.000
10255	68	9	2006-07-12 00:00:00.000	2006-08-09 00:00:00.000	2006-07-15 00:00:00.000	3	148,33	Ship to 68-A	Starenweg 6789	Genève	NULL	10294	Switzerland	10255	16	13,90	35	0.000
10255	68	9	2006-07-12 00:00:00.000	2006-08-09 00:00:00.000	2006-07-15 00:00:00.000	3	148,33	Ship to 68-A	Starenweg 6789	Genève	NULL	10294	Switzerland	10255	36	15,20	25	0.000
10255	68	9	2006-07-12 00:00:00.000	2006-08-09 00:00:00.000	2006-07-15 00:00:00.000	3	148,33	Ship to 68-A	Starenweg 6789	Genève	NULL	10294	Switzerland	10255	59	44,00	30	0.000
10256	88	3	2006-07-15 00:00:00.000	2006-08-12 00:00:00.000	2006-07-17 00:00:00.000	2	13,97	Ship to 88-B	Rua do Mercado, 5678	Resende	SP	10354	Brazil	10256	53	26,20	15	0.000
10256	88	3	2006-07-15 00:00:00.000	2006-08-12 00:00:00.000	2006-07-17 00:00:00.000	2	13,97	Ship to 88-B	Rua do Mercado, 5678	Resende	SP	10354	Brazil	10256	77	10,40	12	0.000
10257	35	4	2006-07-16 00:00:00.000	2006-08-13 00:00:00.000	2006-07-22 00:00:00.000	3	81,91	Destination JYDLM	Carrera1234 con Ave. Carlos Soublette #8-35	San Cristóbal	Táchira	10199	Venezuela	10257	27	35,10	25	0.000
10257	35	4	2006-07-16 00:00:00.000	2006-08-13 00:00:00.000	2006-07-22 00:00:00.000	3	81,91	Destination JYDLM	Carrera1234 con Ave. Carlos Soublette #8-35	San Cristóbal	Táchira	10199	Venezuela	10257	39	14,40	6	0.000
10257	35	4	2006-07-16 00:00:00.000	2006-08-13 00:00:00.000	2006-07-22 00:00:00.000	3	81,91	Destination JYDLM	Carrera1234 con Ave. Carlos Soublette #8-35	San Cristóbal	Táchira	10199	Venezuela	10257	77	10,40	15	0.000
10258	20	1	2006-07-17 00:00:00.000	2006-08-14 00:00:00.000	2006-07-23 00:00:00.000	1	140,51	Destination RVDMF	Kirchgasse 9012	Graz	NULL	10157	Austria	10258	2	15,20	50	0.200
10258	20	1	2006-07-17 00:00:00.000	2006-08-14 00:00:00.000	2006-07-23 00:00:00.000	1	140,51	Destination RVDMF	Kirchgasse 9012	Graz	NULL	10157	Austria	10258	5	17,00	65	0.200
10258	20	1	2006-07-17 00:00:00.000	2006-08-14 00:00:00.000	2006-07-23 00:00:00.000	1	140,51	Destination RVDMF	Kirchgasse 9012	Graz	NULL	10157	Austria	10258	32	25,60	6	0.200
10259	13	4	2006-07-18 00:00:00.000	2006-08-15 00:00:00.000	2006-07-25 00:00:00.000	3	3,25	Destination LGGCH	Sierras de Granada 9012	México D.F.	NULL	10137	Mexico	10259	21	8,00	10	0.000
10259	13	4	2006-07-18 00:00:00.000	2006-08-15 00:00:00.000	2006-07-25 00:00:00.000	3	3,25	Destination LGGCH	Sierras de Granada 9012	México D.F.	NULL	10137	Mexico	10259	37	20,80	1	0.000

FROM Sales.Orders AS o JOIN Sales.OrderDetails AS od ON

ON o.orderid = od.orderid

orderid	custid	empid	orderdate	orderid	productid	unitprice	qty	discount
10248	85	5	2006-07-04	10248	11	14,00	12	0.000
10248	85	5	2006-07-04	10248	42	9,80	10	0.000
10248	85	5	2006-07-04	10248	72	34,80	5	0.000
10249	79	6	2006-07-05	10249	14	18,60	9	0.000
10249	79	6	2006-07-05	10249	51	42,40	40	0.000
10250	34	4	2006-07-08	10250	41	7,70	10	0.000
10250	34	4	2006-07-08	10250	51	42,40	35	0.150
10250	34	4	2006-07-08	10250	65	16,80	15	0.150
10251	84	3	2006-07-08	10251	22	16,80	6	0.050
10251	84	3	2006-07-08	10251	57	15,60	15	0.050
10251	84	3	2006-07-08	10251	65	16,80	20	0.000

- FROM clause determines source tables to be used in SELECT statement
- FROM clause can contain tables and operators
- Result set of FROM clause is virtual table
  - Subsequent logical operations in SELECT statement consume this virtual table


```
SELECT o.orderid, o.orderdate, od.productid, od.unitprice, od.qty
FROM Sales.Orders AS o JOIN Sales.OrderDetails AS od ON o.orderid = od.orderid ;
```

FROM Sales.Orders AS o JOIN Sales.OrderDetails AS od ON

ON o.orderid = od.orderid

orderid	custid	empid	orderdate	orderid	productid	unitprice	qty	discount
10248	85	5	2006-07-04	10248	11	14,00	12	0.000
10248	85	5	2006-07-04	10248	42	9,80	10	0.000
10248	85	5	2006-07-04	10248	72	34,80	5	0.000
10249	79	6	2006-07-05	10249	14	18,60	9	0.000
10249	79	6	2006-07-05	10249	51	42,40	40	0.000
10250	34	4	2006-07-08	10250	41	7,70	10	0.000

```
SELECT o.orderid, o.orderdate, od.productid, od.unitprice, od.qty
FROM Sales.Orders AS o JOIN Sales.OrderDetails AS od ON o.orderid = od.orderid ;
```

FROM Sales.Orders AS o JOIN Sales.OrderDetails AS od ON

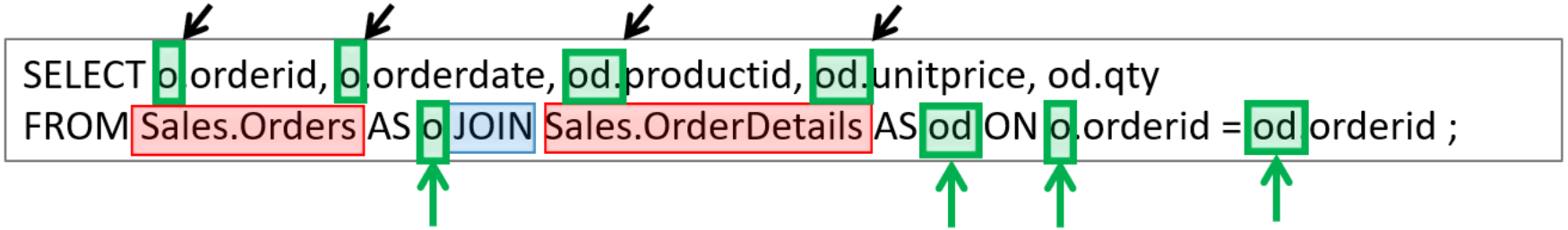
ON o.orderid = od.orderid

```
SELECT o.orderid, o.orderdate, od.productid, od.unitprice, od.qty
```

orderid	orderdate	productid	unitprice	qty
10248	2006-07-04 00:00:00.000	11	14,00	12
10248	2006-07-04 00:00:00.000	42	9,80	10
10248	2006-07-04 00:00:00.000	72	34,80	5
10249	2006-07-05 00:00:00.000	14	18,60	9
10249	2006-07-05 00:00:00.000	51	42,40	40
10250	2006-07-08 00:00:00.000	41	7,70	10
10250	2006-07-08 00:00:00.000	51	42,40	35
10250	2006-07-08 00:00:00.000	65	16,80	15
10251	2006-07-08 00:00:00.000	22	16,80	6
10251	2006-07-08 00:00:00.000	57	15,60	15
10251	2006-07-08 00:00:00.000	65	16,80	20

- FROM clause determines source tables to be used in SELECT statement
- FROM clause can contain tables and operators
- Result set of FROM clause is virtual table
  - Subsequent logical operations in SELECT statement consume this virtual table
- FROM clause can establish table aliases

→ for use by subsequent phases of query ←



- ANSI SQL-92
  - Tables joined by JOIN operator in FROM Clause

```
SELECT ...
FROM   Table1 JOIN Table2
       ON <on_predicate>
```

- ANSI SQL-89
  - Tables joined by commas in FROM Clause
    - Not recommended: accidental Cartesian products!

```
SELECT ...
FROM   Table1, Table2
WHERE  <where_predicate>
```

- 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

Name
Davis
Funk
King

Product
Alice Mutton
Crab Meat
Ipoh Coffee

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

- Join types in FROM clauses specify the operations performed on the virtual table:

Join Type	Description
Cross	Combines all rows in both tables (creates Cartesian product)
Inner	Starts with Cartesian product; applies filter to match rows between tables based on predicate
Outer	Starts with Cartesian product; all rows from designated table preserved, matching rows from other table retrieved. Additional NULLs inserted as placeholders

Exercises joins, keys

	employee_id	first_name	last_name	department_id
1	610001	James	Alexander	1
2	610002	David	Thompson	1
3	610003	Frances	Drake	1
4	610004	Alexandria	Link	2
5	610005	Peter	Link	2
6	610007	David	Cruze	NULL

"First" table = referencing table = child

	department_id	supervisor_fn	supervisor_ln
1	1	Jane	Horton
2	2	Mitch	Simmons
3	3	Paul	Franklin

"Second" table = lookup table = referenced table = parent

```
CREATE TABLE dbo.Departments
(           department_id int NOT NULL,
            supervisor_fn varchar(50),
            supervisor_ln varchar(50),
            CONSTRAINT PK_department_id PRIMARY KEY (department_id)
);
```

```
CREATE TABLE dbo.Employees
(           employee_id int NOT NULL,
            first_name varchar(50),
            last_name varchar(50),
            department_id int,
            CONSTRAINT PK_employee_id PRIMARY KEY (employee_id),
            CONSTRAINT FK_Employees_Departments FOREIGN KEY (department_id)
                        REFERENCES dbo.Departments (department_id)
);
```

Using the CROSS JOIN Clause

- Return all rows from one table along with all rows from the other table. WHERE conditions should always be included.

```
SELECT E.first_name, E.last_name, D.supervisor_fn, D.supervisor_ln
FROM Employees E
        CROSS JOIN Departments D
```

	department_id	supervisor_fn	supervisor_ln
1	1	Jane	Horton
2	2	Mitch	Simmons
3	3	Paul	Franklin

	employee_id	first_name	last_name	department_id
1	610001	James	Alexander	1
2	610002	David	Thompson	1
3	610003	Frances	Drake	1
4	610004	Alexandria	Link	2
5	610005	Peter	Link	2
6	610007	David	Cruze	NULL

	first_name	last_name	supervisor_fn	supervisor_ln
1	James	Alexander	Jane	Horton
2	David	Thompson	Jane	Horton
3	Frances	Drake	Jane	Horton
4	Alexandria	Link	Jane	Horton
5	Peter	Link	Jane	Horton
6	David	Cruze	Jane	Horton
7	James	Alexander	Mitch	Simmons
8	David	Thompson	Mitch	Simmons
9	Frances	Drake	Mitch	Simmons
10	Alexandria	Link	Mitch	Simmons
11	Peter	Link	Mitch	Simmons
12	David	Cruze	Mitch	Simmons
13	James	Alexander	Paul	Franklin
14	David	Thompson	Paul	Franklin
15	Frances	Drake	Paul	Franklin
16	Alexandria	Link	Paul	Franklin
17	Peter	Link	Paul	Franklin
18	David	Cruze	Paul	Franklin

Using the INNER JOIN Clause

- To retrieve a list of employees by their ID numbers and match each employee with the ID of his or her current department supervisor.

	employee_id	first_name	last_name	department_id
1	610001	James	Alexander	1
2	610002	David	Thompson	1
3	610003	Frances	Drake	1
4	610004	Alexandria	Link	2
5	610005	Peter	Link	2
6	610007	David	Cruze	NULL

	department_id	supervisor_fn	supervisor_ln
1	1	Jane	Horton
2	2	Mitch	Simmons
3	3	Paul	Franklin

```
SELECT E.first_name, E.last_name, D.supervisor_fn, D.supervisor_ln
FROM Employees E
        INNER JOIN Departments D
        ON E.departments_id = D.departments_id
```

	first_name	last_name	supervisor_fn	supervisor_ln
1	James	Alexander	Jane	Horton
2	David	Thompson	Jane	Horton
3	Frances	Drake	Jane	Horton
4	Alexandria	Link	Mitch	Simmons
5	Peter	Link	Mitch	Simmons

David Cruz does not appear in the output list of employees matched with department supervisors. Supervisor Paul Franklin does not appear.

Using the LEFT OUTER JOIN Clause

- Include all records in the left table along with the matching records of the right table AND any nonmatching records.

	employee_id	first_name	last_name	department_id
1	610001	James	Alexander	1
2	610002	David	Thompson	1
3	610003	Frances	Drake	1
4	610004	Alexandria	Link	2
5	610005	Peter	Link	2
6	610007	David	Cruze	NULL

	department_id	supervisor_fn	supervisor_ln
1	1	Jane	Horton
2	2	Mitch	Simmons
3	3	Paul	Franklin

```
SELECT E.first_name, E.last_name, D.supervisor_fn, D.supervisor_ln
FROM Employees E
        LEFT OUTER JOIN Departments D
        ON E.departments_id = D.departments_id
```

	first_name	last_name	supervisor_fn	supervisor_ln
1	James	Alexander	Jane	Horton
2	David	Thompson	Jane	Horton
3	Frances	Drake	Jane	Horton
4	Alexandria	Link	Mitch	Simmons
5	Peter	Link	Mitch	Simmons
6	David	Cruze	NULL	NULL

David Cruz is included; he is not assigned to any department supervisor and thus his name shows a NULL value in the list.

Using the RIGHT OUTER JOIN Clause

- Include all records in the right table along with the matching records of the left table AND any nonmatching records.

	employee_id	first_name	last_name	department_id
1	610001	James	Alexander	1
2	610002	David	Thompson	1
3	610003	Frances	Drake	1
4	610004	Alexandria	Link	2
5	610005	Peter	Link	2
6	610007	David	Cruze	NULL

	department_id	supervisor_fn	supervisor_ln
1	1	Jane	Horton
2	2	Mitch	Simmons
3	3	Paul	Franklin

```
SELECT E.first_name, E.last_name, D.supervisor_fn, D.supervisor_ln
FROM Employees E
        RIGHT OUTER JOIN Departments D
        ON E.departments_id = D.departments_id
```

	first_name	last_name	supervisor_fn	supervisor_ln
1	James	Alexander	Jane	Horton
2	David	Thompson	Jane	Horton
3	Frances	Drake	Jane	Horton
4	Alexandria	Link	Mitch	Simmons
5	Peter	Link	Mitch	Simmons
6	NULL	NULL	Paul	Franklin

Paul Franklin is included; there are not employees assigned to him and thus his name shows a NULL value in the list.

Using the FULL OUTER JOIN Clause

- Include all matching records, and all nonmatching records of the left table and all nonmatching records of the right table.

	employee_id	first_name	last_name	department_id
1	610001	James	Alexander	1
2	610002	David	Thompson	1
3	610003	Frances	Drake	1
4	610004	Alexandria	Link	2
5	610005	Peter	Link	2
6	610007	David	Cruze	NULL

	department_id	supervisor_fn	supervisor_ln
1	1	Jane	Horton
2	2	Mitch	Simmons
3	3	Paul	Franklin

```
SELECT E.first_name, E.last_name, D.supervisor_fn, D.supervisor_ln
FROM Employees E
        FULL OUTER JOIN Departments D
        ON E.departments_id = D.departments_id
```

	first_name	last_name	supervisor_fn	supervisor_ln
1	James	Alexander	Jane	Horton
2	David	Thompson	Jane	Horton
3	Frances	Drake	Jane	Horton
4	Alexandria	Link	Mitch	Simmons
5	Peter	Link	Mitch	Simmons
6	David	Cruze	NULL	NULL
7	NULL	NULL	Paul	Franklin

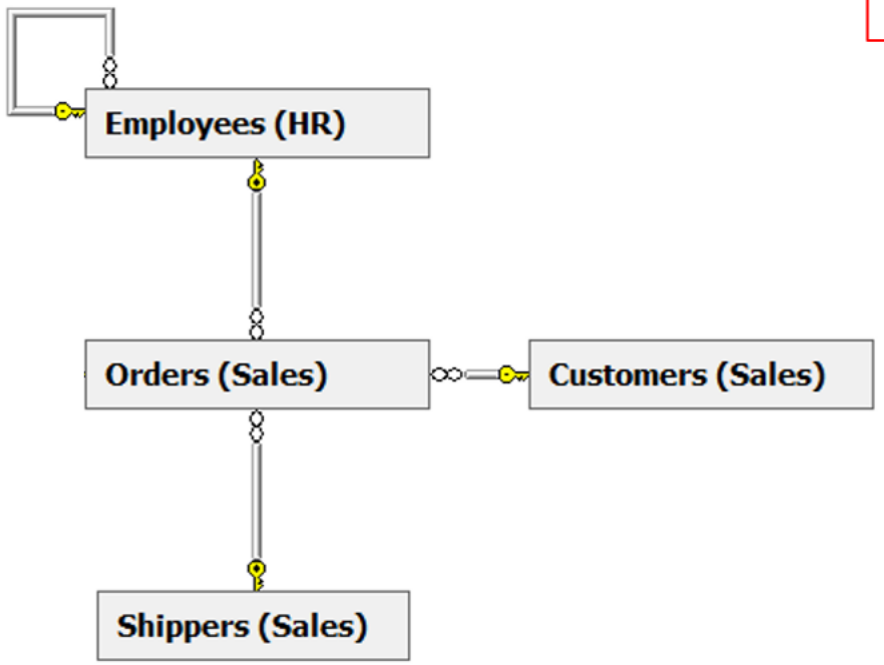
David Cruze and Paul Franklin are included.

Employees (HR)

	empid	lastname	firstname	title	titleofcourtesy	birthdate	hiredate	region	postalcode	country	phone	mgrid
1	1	Davis	Sara	CEO	Ms.	1958-12-08 00:00:00.000	2002-05-01 00:00:	WA	10003	USA	(206) 555-0101	NULL
2	2	Funk	Don	Vice President, Sales	Dr.	1962-02-19 00:00:00.000	2002-08-14 00:00:	WA	10001	USA	(206) 555-0100	1
3	3	Lew	Judy	Sales Manager	Ms.	1973-08-30 00:00:00.000	2002-04-01 00:00:	WA	10007	USA	(206) 555-0103	2
4	4	Peled	Yael	Sales Representative	Mrs.	1947-09-19 00:00:00.000	2003-05-03 00:00:	WA	10009	USA	(206) 555-0104	3
5	5	Buck	Sven	Sales Manager	Mr.	1965-03-04 00:00:00.000	2003-10-17 00:00:	NULL	10004	UK	(71) 234-5678	2
6	6	Suurs	Paul	Sales Representative	Mr.	1973-07-02 00:00:00.000	2003-10-17 00:00:	NULL	10005	UK	(71) 345-6789	5
7	7	King	Russell	Sales Representative	Mr.	1970-05-29 00:00:00.000	2004-01-02 00:00:	NULL	10002	UK	(71) 123-4567	5
8	8	Cameron	Maria	Sales Representative	Ms.	1968-01-09 00:00:00.000	2004-03-05 00:00:	WA	10006	USA	(206) 555-0102	3
9	9	Dolgopyatova	Zoya	Sales Representative	Ms.	1976-01-27 00:00:00.000	2004-11-15 00:00:	NULL	10008	UK	(71) 456-7890	5

PK

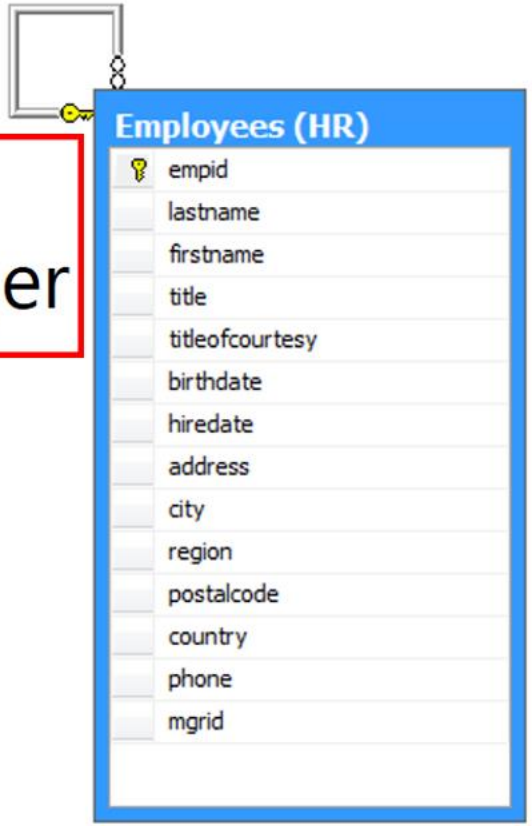
FK



- Why use self joins?
  - Compare rows in same table to each other
- Create two instances of same table in FROM clause
  - At least one alias required

empid	lastname	firstname	title	mgrid
1	Davis	Sara	CEO	NULL
2	Funk	Don	Vice President, Sales	1
3	Lew	Judy	Sales Manager	2
4	Peled	Yael	Sales Representative	3
5	Buck	Sven	Sales Manager	2
6	Suurs	Paul	Sales Representative	5
7	King	Russell	Sales Representative	5
8	Cameron	Maria	Sales Representative	3
9	Dolgopyatova	Zoya	Sales Representative	5

- Why use self joins?
  - Compare rows in same table to each other
- Create two instances of same table in FROM clause
  - At least one alias required
- Example: Return all employees and the name of the employee's manager



- Return all employees with ID of employee's manager when a manager exists (inner join):

```
SELECT  e.empid, e.lastname,
        e.title, e.mgrid, m.lastname
FROM   HR.Employees AS e
JOIN HR.Employees AS m
ON e.mgrid=m.empid;
```

HR.Employees AS e

empid	lastname	firstname	title	mgrid
1	Davis	Sara	CEO	NULL
2	Funk	Don	Vice President, Sales	1
3	Lew	Judy	Sales Manager	2
4	Peled	Yael	Sales Representative	3
5	Buck	Sven	Sales Manager	2
6	Suurs	Paul	Sales Representative	5
7	King	Russell	Sales Representative	5
8	Cameron	Maria	Sales Representative	3
9	Dolgopyatova	Zoya	Sales Representative	5

HR.Employees AS m

empid	lastname	firstname	title	mgrid
1	Davis	Sara	CEO	NULL
2	Funk	Don	Vice President, Sales	1
3	Lew	Judy	Sales Manager	2
4	Peled	Yael	Sales Representative	3
5	Buck	Sven	Sales Manager	2
6	Suurs	Paul	Sales Representative	5
7	King	Russell	Sales Representative	5
8	Cameron	Maria	Sales Representative	3
9	Dolgopyatova	Zoya	Sales Representative	5

- Return all employees with ID of employee's manager when a manager exists (inner join):

```
SELECT  e.empid, e.lastname,
        e.title, e.mgrid, m.lastname
FROM    HR.Employees AS e
JOIN HR.Employees AS m
ON e.mgrid=m.empid;
```

HR.Employees AS e

empid	lastname	firstname	title	mgrid
1	Davis	Sara	CEO	NULL
2	Funk	Don	Vice President, Sales	1
3	Lew	Judy	Sales Manager	2
4	Peled	Yael	Sales Representative	3
5	Buck	Sven	Sales Manager	2
6	Suurs	Paul	Sales Representative	5
7	King	Russell	Sales Representative	5
8	Cameron	Maria	Sales Representative	3
9	Dolgopyatova	Zoya	Sales Representative	5

Davis
Funk
Lew
Funk
Buck
Buck
Lew
Buck

HR.Employees AS m

empid	lastname
1	Davis
2	Funk
3	Lew
4	Peled
5	Buck
6	Suurs
7	King
8	Cameron
9	Dolgopyatova

- Return all employees with ID of employee's manager when a manager exists (inner join):

```
SELECT  e.empid, e.lastname,
        e.title, e.mgrid, m.lastname
FROM    HR.Employees AS e
JOIN HR.Employees AS m
ON e.mgrid=m.empid;
```

HR.Employees AS e

empid	lastname	firstname	title	mgrid	
1	Davis	Sara	CEO	NULL	
2	Funk	Don	Vice President, Sales	1	Davis
3	Lew	Judy	Sales Manager	2	Funk
4	Peled	Yael	Sales Representative	3	Lew
5	Buck	Sven	Sales Manager	2	Funk
6	Suurs	Paul	Sales Representative	5	Buck
7	King	Russell	Sales Representative	5	Buck
8	Cameron	Maria	Sales Representative	3	Lew
9	Dolgopyatova	Zoya	Sales Representative	5	Buck

HR.Employees AS m

empid	lastname
1	Davis
2	Funk
3	Lew
4	Peled
5	Buck
6	Suurs
7	King
8	Cameron
9	Dolgopyatova

- Return all employees with ID of employee's manager when a manager exists (inner join):

```
SELECT  e.empid, e.lastname,
        e.title, e.mgrid, m.lastname
FROM    HR.Employees AS e
JOIN HR.Employees AS m
ON e.mgrid=m.empid;
```

HR.Employees AS e

empid	lastname	title	mgrid	lastname
2	Funk	Vice President, Sales	1	Davis
3	Lew	Sales Manager	2	Funk
4	Peled	Sales Representative	3	Lew
5	Buck	Sales Manager	2	Funk
6	Suurs	Sales Representative	5	Buck
7	King	Sales Representative	5	Buck
8	Cameron	Sales Representative	3	Lew
9	Dolgopyatova	Sales Representative	5	Buck

HR.Employees AS m

empid	lastname
1	Davis
2	Funk
3	Lew
4	Peled
5	Buck
6	Suurs
7	King
8	Cameron
9	Dolgopyatova

- Return all employees with ID of employee's manager when a manager exists (inner join):

```
SELECT  e.empid, e.lastname,
        e.title, e.mgrid, m.lastname
FROM    HR.Employees AS e
JOIN HR.Employees AS m
ON e.mgrid=m.empid;
```

HR.Employees AS e

empid	lastname	firstname	title	mgrid
1	Davis	Sara	CEO	NULL
2	Funk	Don	Vice President, Sales	1
3	Lew	Judy	Sales Manager	2
4	Peled	Yael	Sales Representative	3
5	Buck	Sven	Sales Manager	2
6	Suurs	Paul	Sales Representative	5
7	King	Russell	Sales Representative	5
8	Cameron	Maria	Sales Representative	3
9	Dolgopyatova	Zoya	Sales Representative	5

HR.Employees AS m

empid	lastname	firstname	title	mgrid
1	Davis	Sara	CEO	NULL
2	Funk	Don	Vice President, Sales	1
3	Lew	Judy	Sales Manager	2
4	Peled	Yael	Sales Representative	3
5	Buck	Sven	Sales Manager	2
6	Suurs	Paul	Sales Representative	5
7	King	Russell	Sales Representative	5
8	Cameron	Maria	Sales Representative	3
9	Dolgopyatova	Zoya	Sales Representative	5

- Return all employees with ID of employee's manager when a manager exists (inner join):

```
SELECT  e.empid, e.lastname,
        e.title, e.mgrid, m.lastname
FROM    HR.Employees AS e
JOIN HR.Employees AS m
ON e.mgrid=m.empid;
```

empid	lastname	title	mgrid	lastname
2	Funk	Vice President, Sales	1	Davis
3	Lew	Sales Manager	2	Funk
4	Peled	Sales Representative	3	Lew
5	Buck	Sales Manager	2	Funk
6	Suurs	Sales Representative	5	Buck
7	King	Sales Representative	5	Buck
8	Cameron	Sales Representative	3	Lew
9	Dolgopyatova	Sales Representative	5	Buck

- Return all employees with ID of employee's manager when a manager exists (inner join):

```
SELECT  e.empid, e.lastname,
        e.title, e.mgrid, m.lastname
FROM    HR.Employees AS e
JOIN HR.Employees AS m
ON e.mgrid=m.empid;
```

- Return all employees with ID of manager (outer join). This will return NULL for the CEO:

```
SELECT  e. empid, e.lastname,
          e.title, m.mgrid
FROM HR.Employees AS e
LEFT OUTER JOIN HR.Employees AS m
ON e.mgrid=m.empid;
```

empid	lastname	title	mgrid	lastname
1	Davis	CEO	NULL	NULL
2	Funk	Vice President, Sales	1	Davis
3	Lew	Sales Manager	2	Funk
4	Peled	Sales Representative	3	Lew
5	Buck	Sales Manager	2	Funk
6	Suurs	Sales Representative	5	Buck
7	King	Sales Representative	5	Buck
8	Cameron	Sales Representative	3	Lew
9	Dolgopyatova	Sales Representative	5	Buck