EIE 3112
Structural Query Language (SQL)
Part 2

T. Connolly and C. Begg, "*Database Systems: A Practical Approach to Design, Implementation, and Management*," 6th Edition, Chapter 6&7, Pearson, 2015. (5th Edition is also fine)

You Will Learn

- Joining Tables
- Summarizing Tables
- Views of Databases

SELECT from Multiple Tables

- Most databases have many tables
- Combine tables using the join operator
- Specify matching condition
  - Can be any comparison but usually =
  - Primary key = foreign key (most common join condition)

SELECT from Multiple Tables

Query:

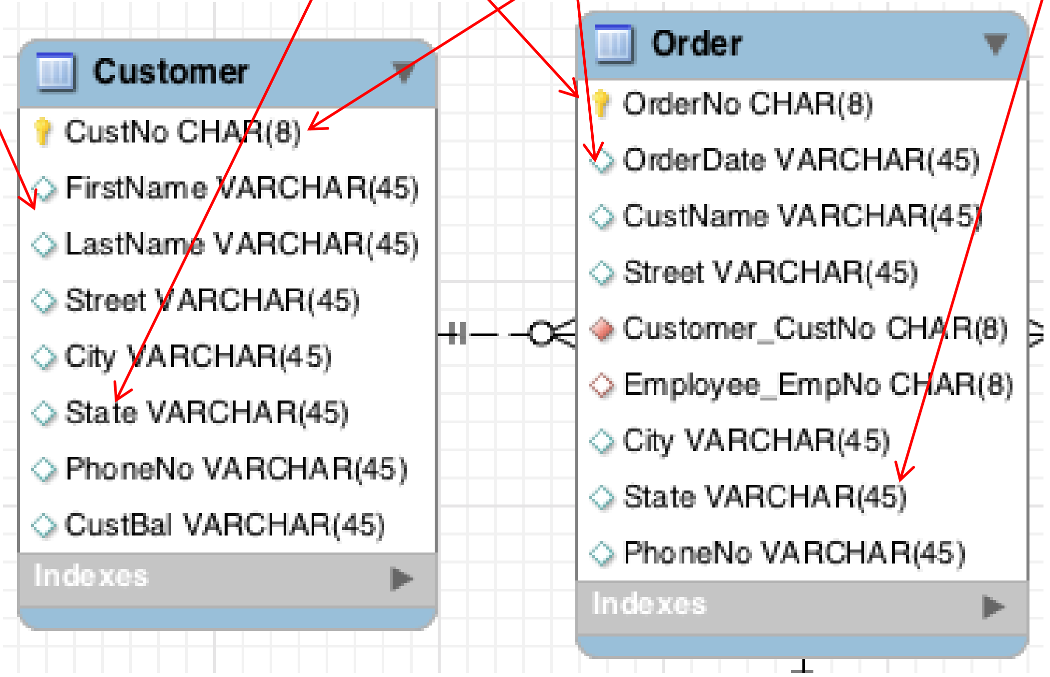
List the order number, the order date, the customer number, the customer name, the customer’s state, and the shipping state in which the customer’s state differs from the shipping’s state.

SELECT from Multiple Tables

Data Items Involves:

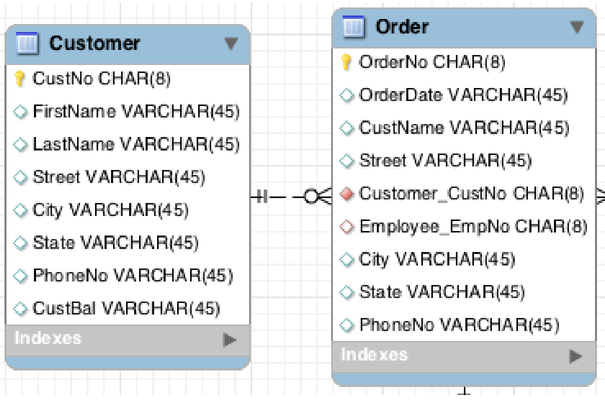
List the *order number*, the *order date*, the *customer number*, the *customer name*, the *customer's state*, and the *shipping state* in which the *customer's state* differs from the *shipping's state*.

Tables Involves:



Joining Two Tables

```
SELECT OrderNo, OrderDate, CustNo, `Order`.CustName,
    Customer.State AS 'Customer State',
    Order.State AS 'Shipping State'
FROM
    Customer INNER JOIN (
        `Order`
    )
    ON Customer.CustNo=Order.Customer_CustNo
```



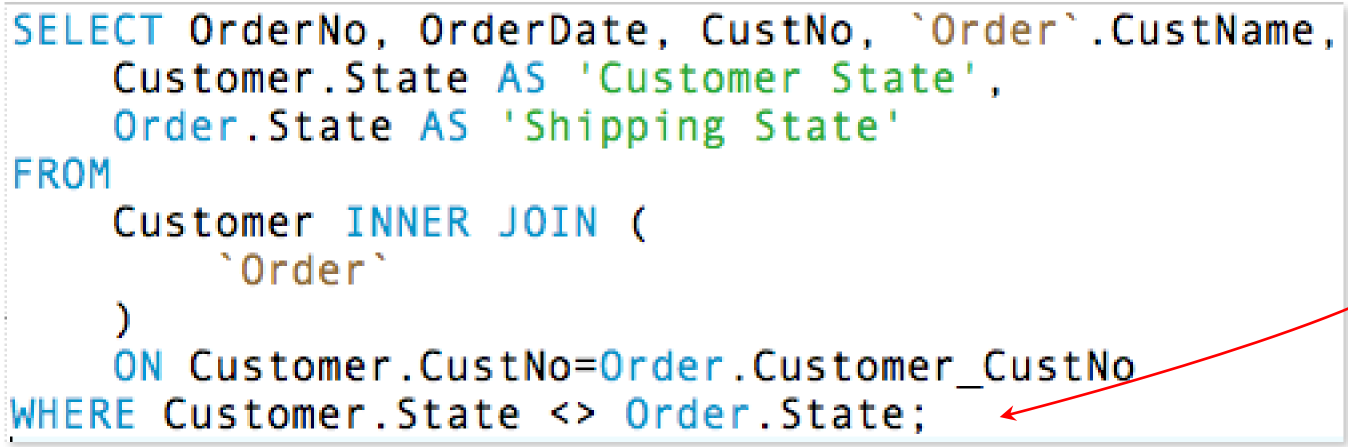
OrderNo	OrderDate	CustNo	CustName	Customer State	Shipping State
O1116324	1/23/2007	C0954327	Sheri Gordon	CO	CO
O2334661	1/14/2007	C0954327	Mrs. Ruth Gor...	CO	WA
O3331222	1/13/2007	C1010398	Jim Glussman	CO	CO
O1111111	10/1/2007	C1234567	Man-Wai Mak	HK	HK
O1234567	5/25/2013	C1234567	Man-Wai Mak	HK	HK
O2233457	1/12/2007	C2388597	Beth Taylor	WA	WA
O4714645	1/11/2007	C2388597	Beth Taylor	WA	WA
O5511365	1/22/2007	C3340959	Betty White	WA	WA
O7989497	1/16/2007	C3499503	Bob Mann	WA	WA

Joining Two Tables

Conditions of Results:

List the order number, the order date, the customer number, the customer name, the customer's state, and the shipping state *in which the customer's state differs from the shipping's state*.

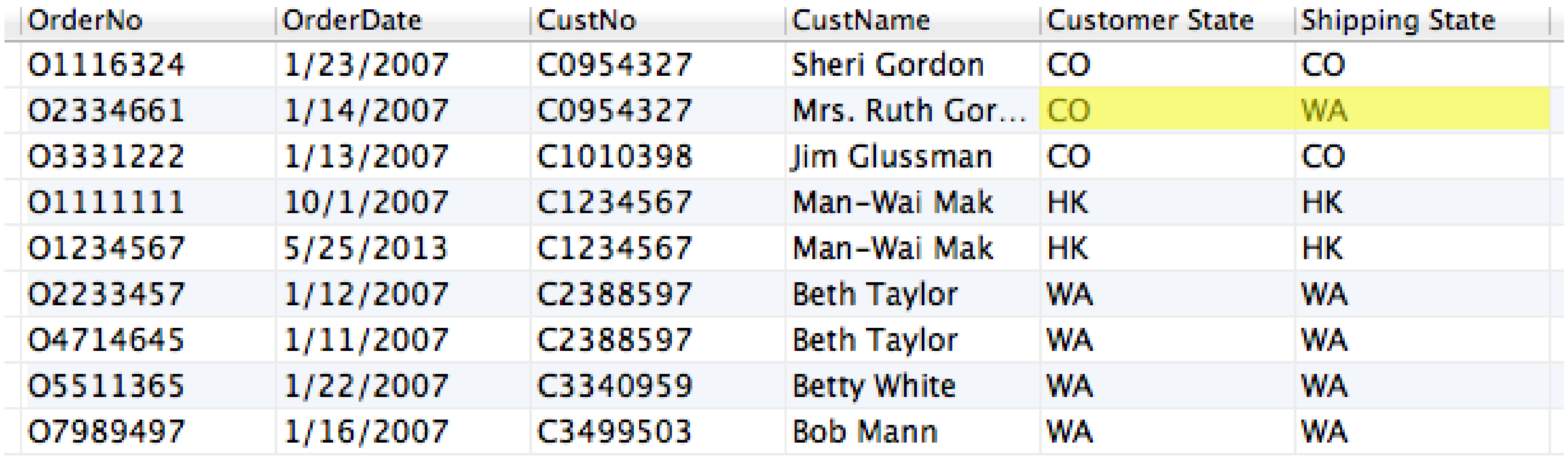
Solution:



```
SELECT OrderNo, OrderDate, CustNo, `Order`.CustName,
    Customer.State AS 'Customer State',
    Order.State AS 'Shipping State'
FROM
    Customer INNER JOIN (
        `Order`
    )
    ON Customer.CustNo=Order.Customer_CustNo
WHERE Customer.State <> Order.State;
```

Joining Two Tables

Applying the Condition:

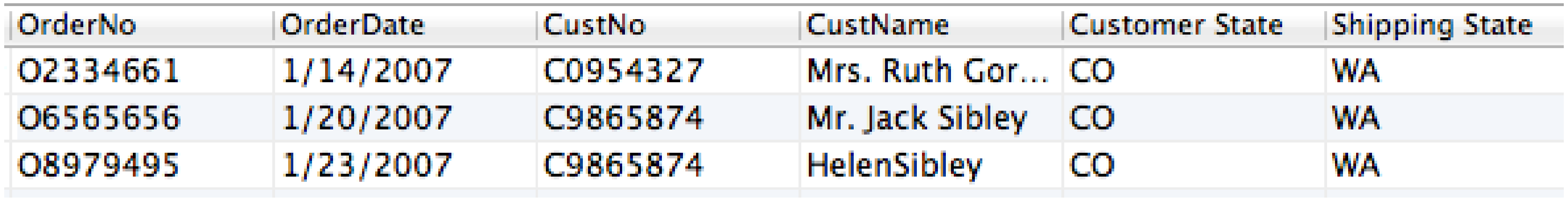


OrderNo	OrderDate	CustNo	CustName	Customer State	Shipping State
O1116324	1/23/2007	C0954327	Sheri Gordon	CO	CO
O2334661	1/14/2007	C0954327	Mrs. Ruth Gor...	CO	WA
O3331222	1/13/2007	C1010398	Jim Glussman	CO	CO
O1111111	10/1/2007	C1234567	Man-Wai Mak	HK	HK
O1234567	5/25/2013	C1234567	Man-Wai Mak	HK	HK
O2233457	1/12/2007	C2388597	Beth Taylor	WA	WA
O4714645	1/11/2007	C2388597	Beth Taylor	WA	WA
O5511365	1/22/2007	C3340959	Betty White	WA	WA
O7989497	1/16/2007	C3499503	Bob Mann	WA	WA

⋮

```
WHERE Customer.State <> Order.State;
```

Result:



OrderNo	OrderDate	CustNo	CustName	Customer State	Shipping State
O2334661	1/14/2007	C0954327	Mrs. Ruth Gor...	CO	WA
O6565656	1/20/2007	C9865874	Mr. Jack Sibley	CO	WA
O8979495	1/23/2007	C9865874	HelenSibley	CO	WA

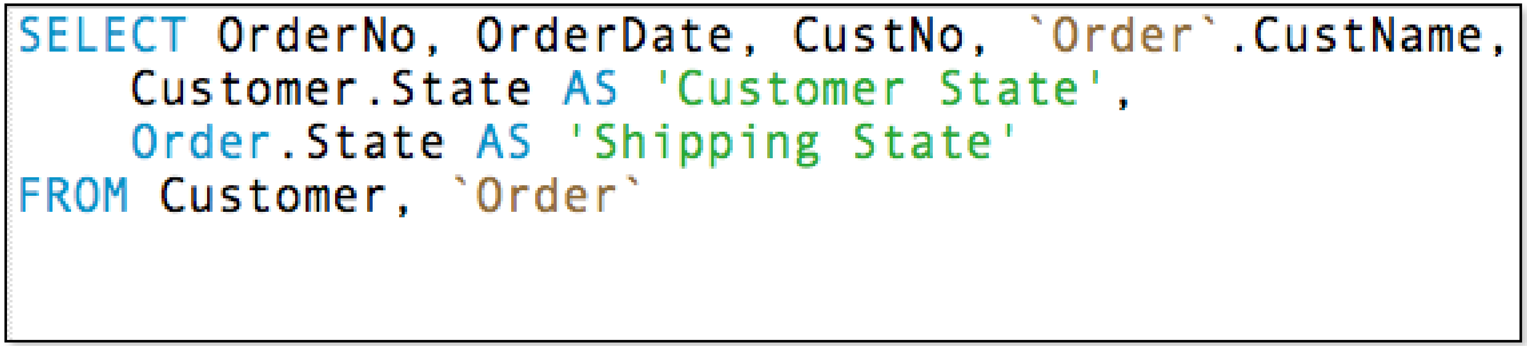
(Cartesian Product)

Joining Tables with Cross Products

Query:

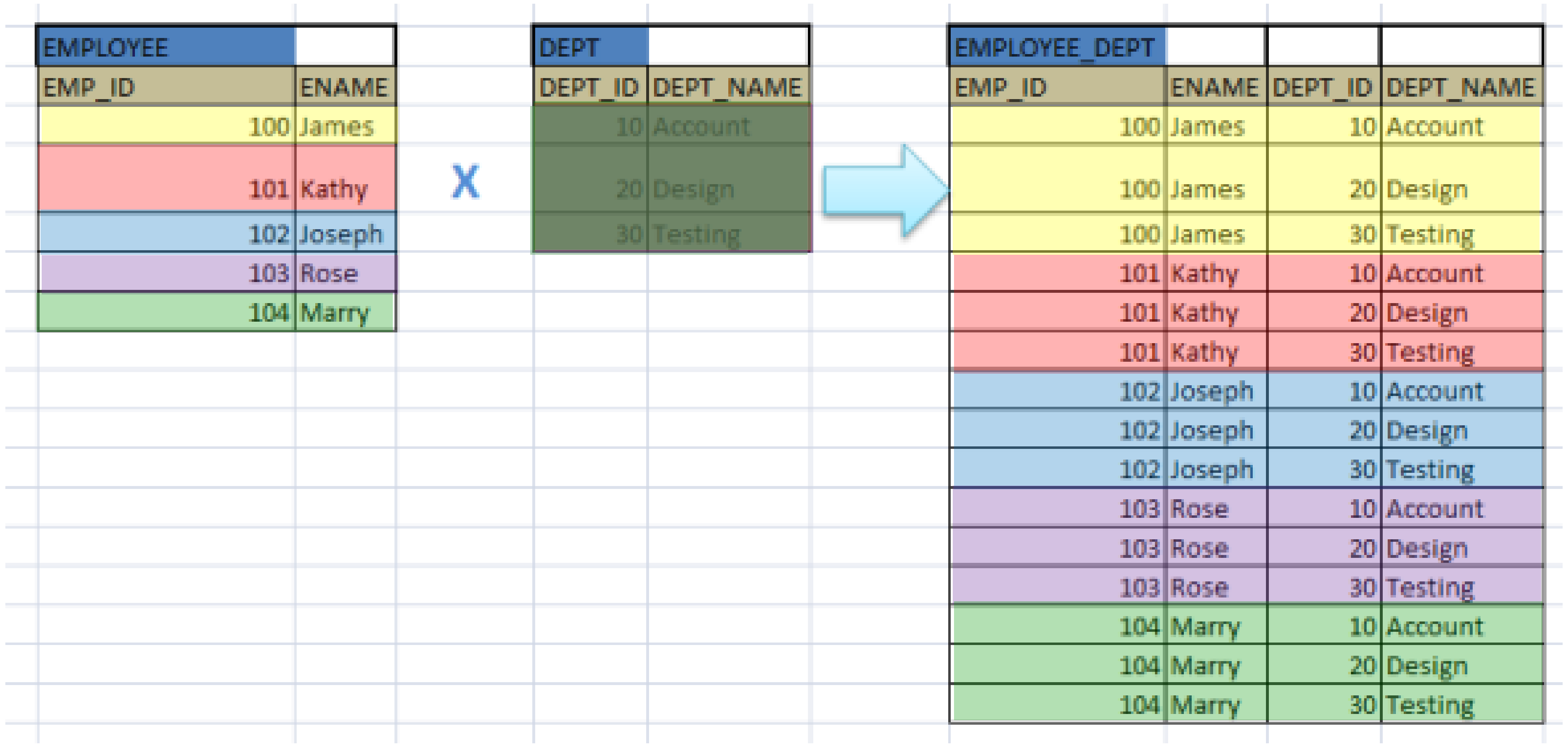
List the order number, the order date, the customer number, the customer name, the customer's state, and the shipping state in which the customer's state differs from the shipping's state.

Solution:



```
SELECT OrderNo, OrderDate, CustNo, `Order`.CustName,
    Customer.State AS 'Customer State',
    Order.State AS 'Shipping State'
FROM Customer, `Order`
```

Cartesian Product: EMPLOYEE X DEPT



EMPLOYEE	
EMP_ID	ENAME
100	James
101	Kathy
102	Joseph
103	Rose
104	Marry

X

DEPT	
DEPT_ID	DEPT_NAME
10	Account
20	Design
30	Testing

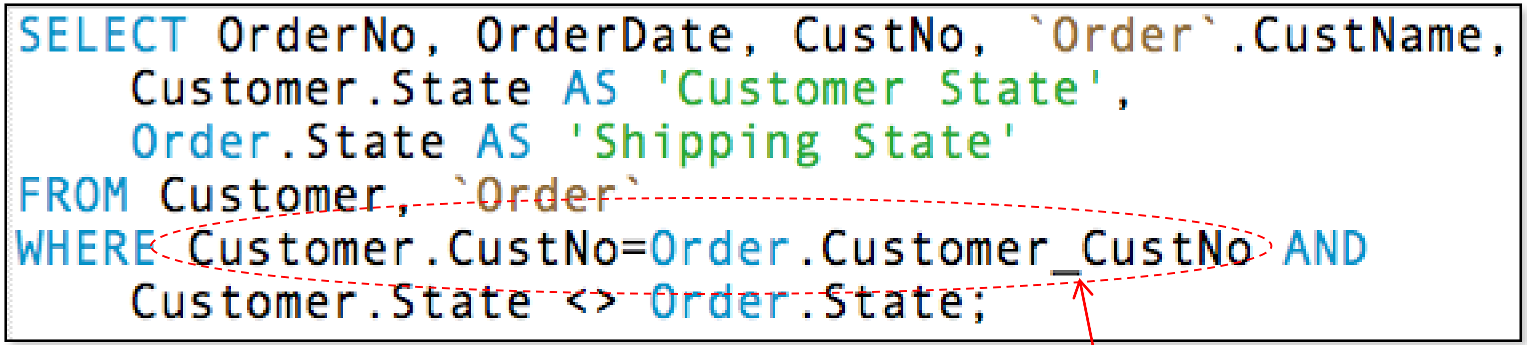
EMPLOYEE_DEPT			
EMP_ID	ENAME	DEPT_ID	DEPT_NAME
100	James	10	Account
100	James	20	Design
100	James	30	Testing
101	Kathy	10	Account
101	Kathy	20	Design
101	Kathy	30	Testing
102	Joseph	10	Account
102	Joseph	20	Design
102	Joseph	30	Testing
103	Rose	10	Account
103	Rose	20	Design
103	Rose	30	Testing
104	Marry	10	Account
104	Marry	20	Design
104	Marry	30	Testing

Joining Tables with Cross Products

Query:

List the order number, the order date, the customer number, the customer name, the customer's state, and the shipping state in which the customer's state differs from the shipping's state.

Solution:



```
SELECT OrderNo, OrderDate, CustNo, `Order`.CustName,
    Customer.State AS 'Customer State',
    Order.State AS 'Shipping State'
FROM Customer, `Order`
WHERE Customer.CustNo=Order.Customer_CustNo AND
    Customer.State <> Order.State;
```

Join condition in the WHERE clause

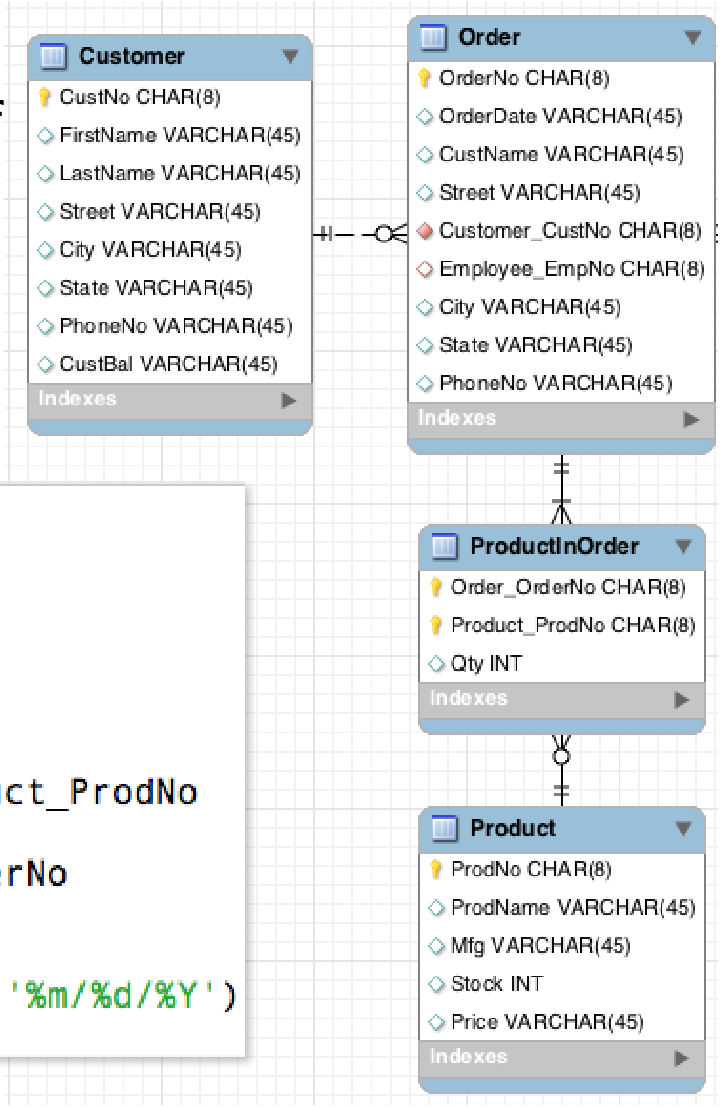
Joining Four Tables

Query:

List the product number, name and price of products ordered by customer C0954327 in January 2007.

Solution:

```
SELECT ProdNo, ProdName, Price, OrderDate
FROM
    Customer INNER JOIN (
        `Order` INNER JOIN (
            ProductInOrder INNER JOIN (
                Product
            )
            ON Product.ProdNo=ProductInOrder.Product_ProdNo
        )
        ON Order.OrderNo=ProductInOrder.Order_OrderNo
    )
    ON Customer.CustNo=Order.Customer_CustNo
WHERE CustNo='C0954327' AND STR_TO_DATE(OrderDate,'%m/%d/%Y')
        BETWEEN '2007-1-1' AND '2007-1-31';
```



STR_TO_DATE() in MySQL

https://dev.mysql.com/doc/refman/5.5/en/date-and-time-functions.html#function_str-to-date

- mysql> **SELECT STR_TO_DATE('01,5,2013','%d,%m,%Y');**
- -> '2013-05-01'
- mysql> **SELECT STR_TO_DATE('May 1, 13','%M %d,%y');**
- -> '2013-05-01’

https://dev.mysql.com/doc/refman/5.5/en/date-and-time-functions.html#function_date-format

%Y	Year, numeric, four digits
%y	Year, numeric (two digits)
%M	Month name (January..December)
%m	Month, numeric (00..12)

Joining Four Tables

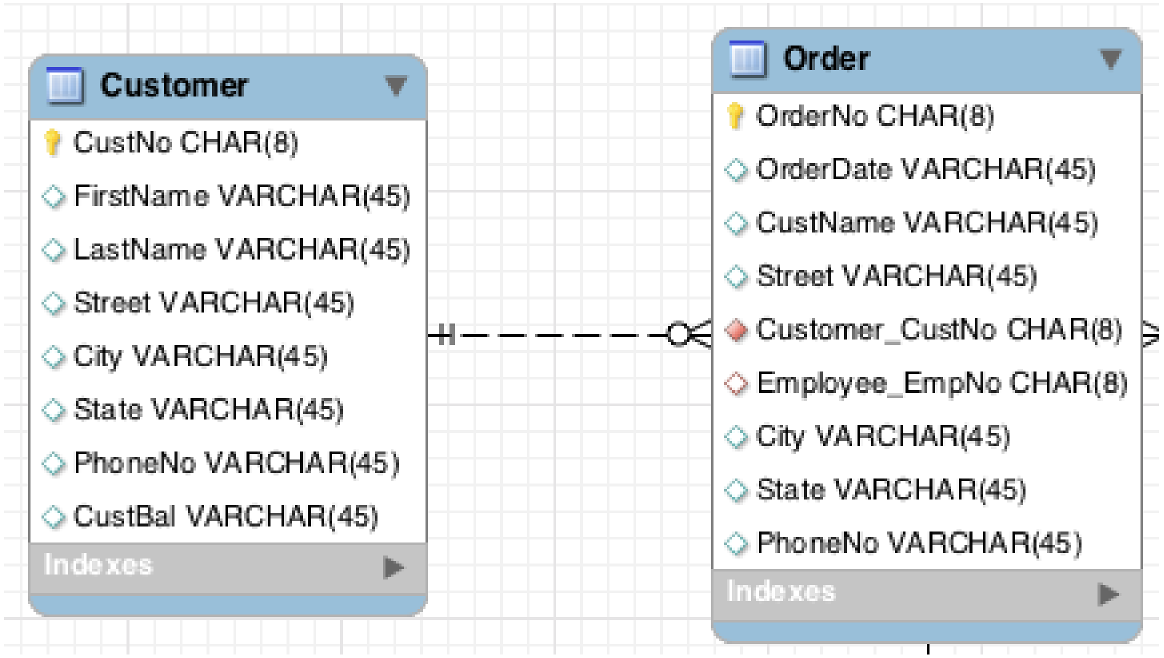
Exercise:

Rewrite the above query with join conditions putting in WHERE clause.

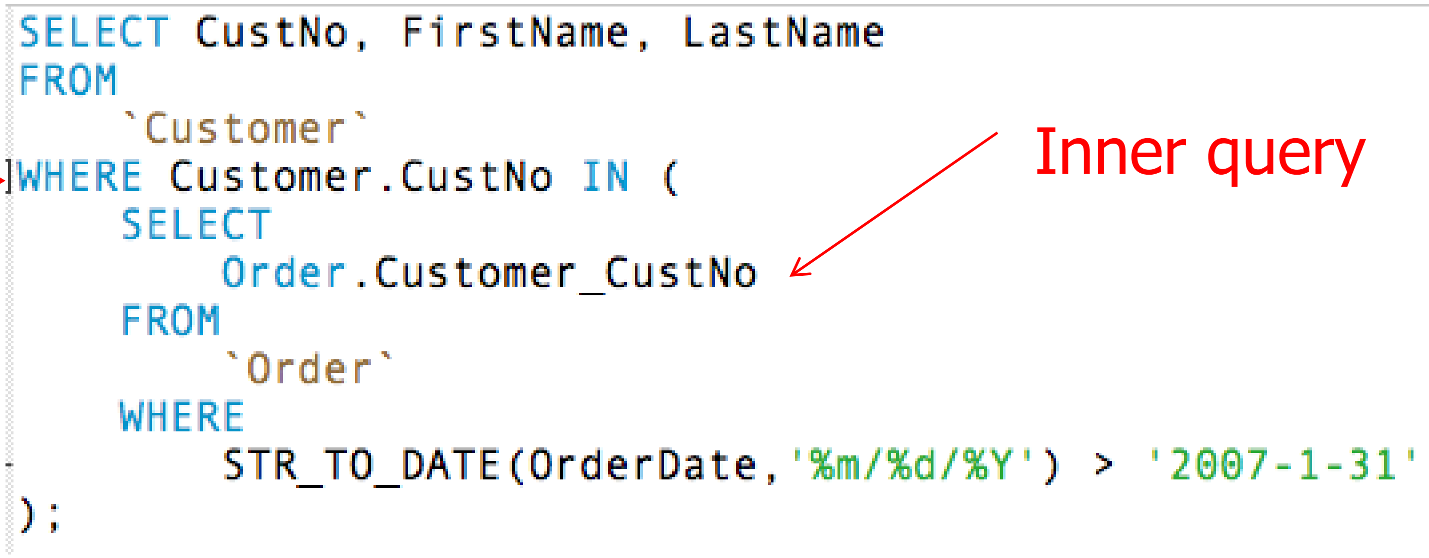
Solution:

Nested Queries

- Nested queries are queries inside queries (similar to nested for-loops in Java)
- Use the `IN` comparison operator
- E.g., list the ID and names of customers who have placed order after Jan 2007

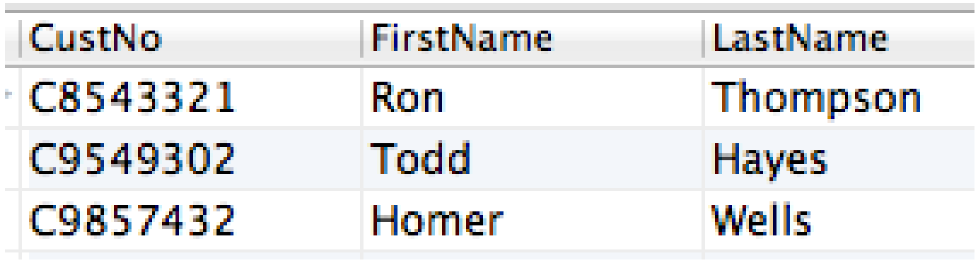


Nested Queries



```
SELECT CustNo, FirstName, LastName
FROM
    `Customer`
WHERE Customer.CustNo IN (
    SELECT
        Order.Customer_CustNo
    FROM
        `Order`
    WHERE
        STR_TO_DATE(OrderDate,'%m/%d/%Y') > '2007-1-31'
);
```

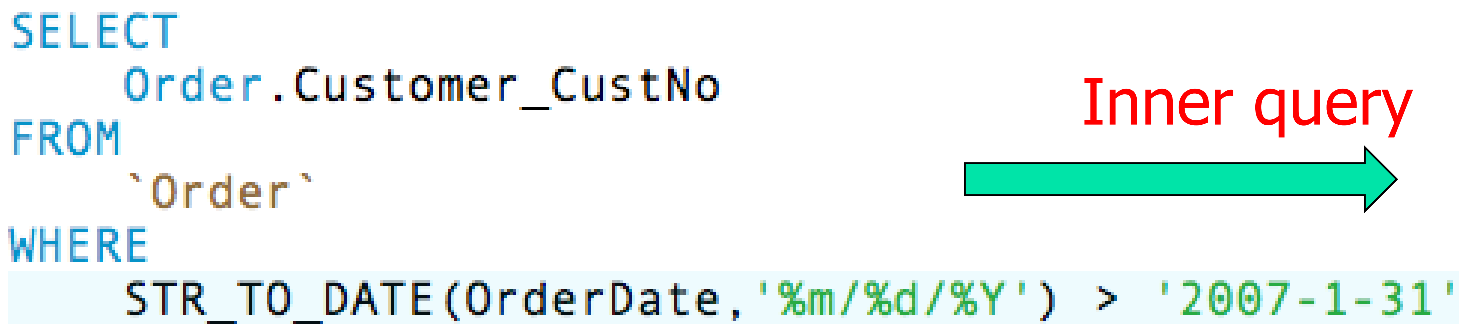
Inner query



CustNo	FirstName	LastName
C8543321	Ron	Thompson
C9549302	Todd	Hayes
C9857432	Homer	Wells

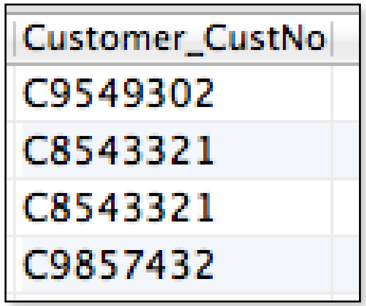
Only list customers whose CustNo matches the
CustNo found in the inner query

Nested Queries

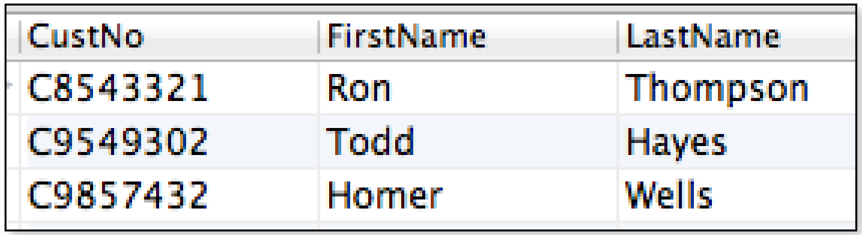


```
SELECT
    Order.Customer_CustNo
FROM
    `Order`
WHERE
    STR_TO_DATE(OrderDate,'%m/%d/%Y') > '2007-1-31'
```

Inner query

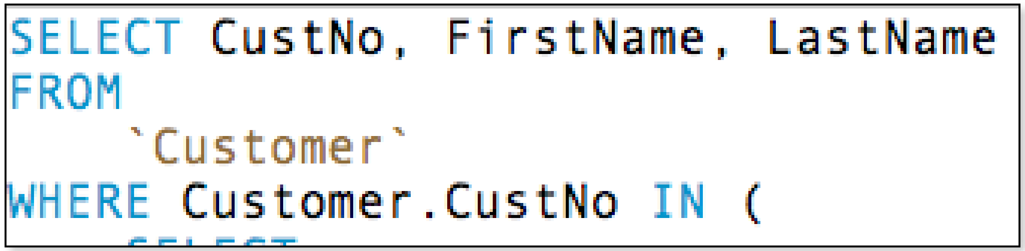


Customer_CustNo
C9549302
C8543321
C8543321
C9857432



CustNo	FirstName	LastName
C8543321	Ron	Thompson
C9549302	Todd	Hayes
C9857432	Homer	Wells

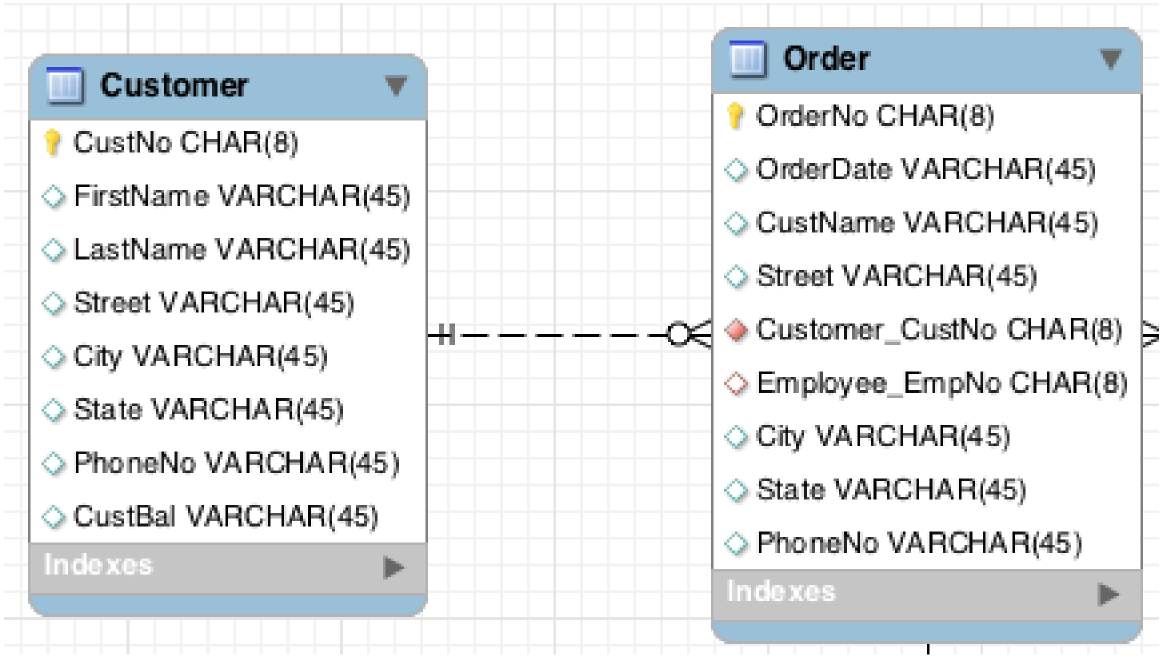
Outer query



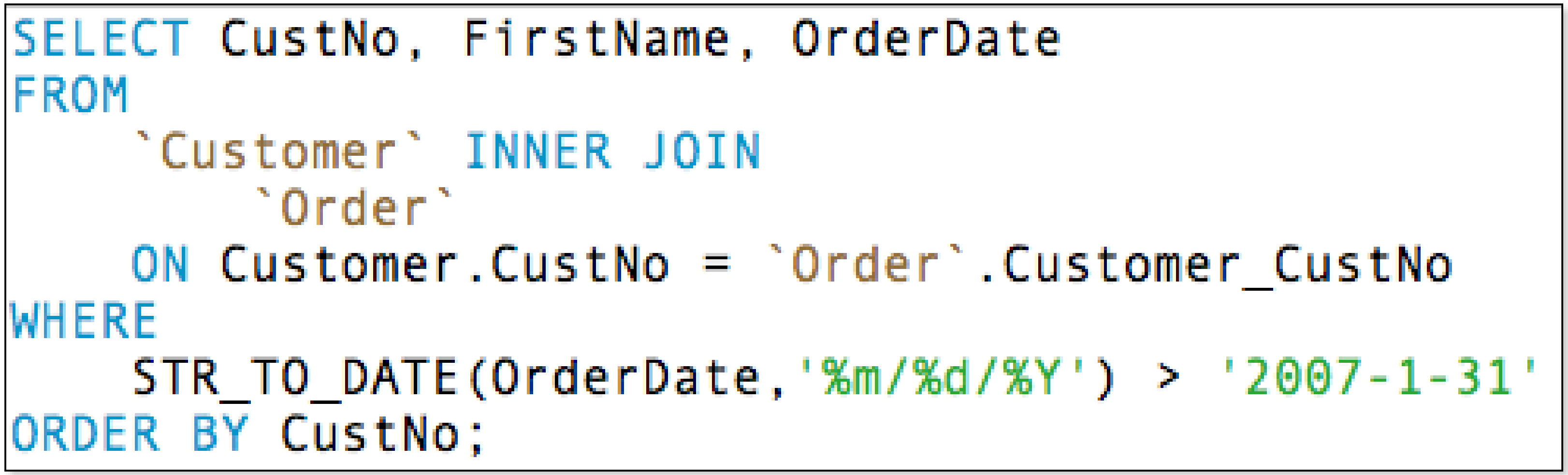
```
SELECT CustNo, FirstName, LastName
FROM
    `Customer`
WHERE Customer.CustNo IN (
```

When Nested Queries Fail

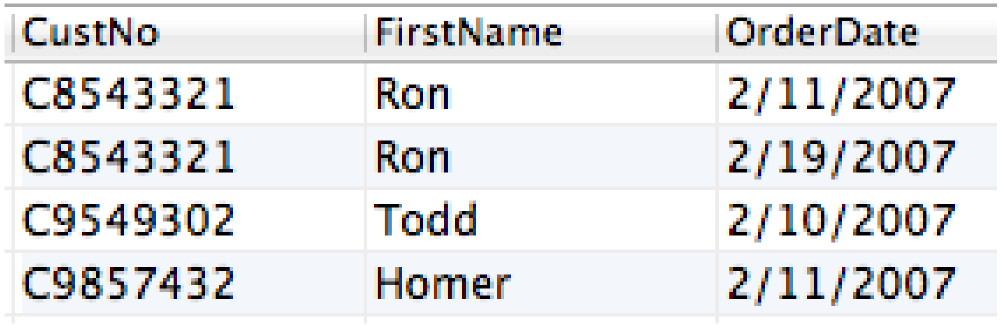
- The previous example can only display information of the table in the outer query (e.g., `Customer`).
- To display information of the table in the inner query, we need to use the join operator.
- E.g., Also display order date (`` `Order`.OrderDate ``)



When Nested Queries Fail



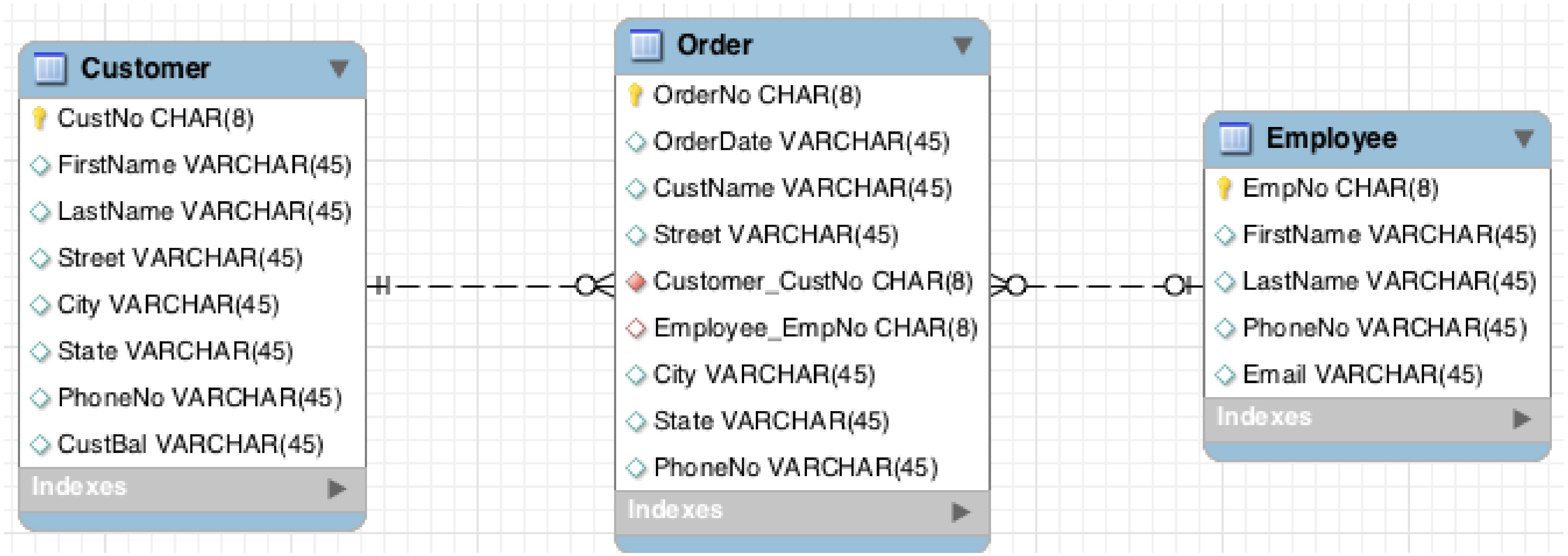
```
SELECT CustNo, FirstName, OrderDate
FROM
    `Customer` INNER JOIN
        `Order`
    ON Customer.CustNo = `Order`.Customer_CustNo
WHERE
    STR_TO_DATE(OrderDate,'%m/%d/%Y') > '2007-1-31'
ORDER BY CustNo;
```



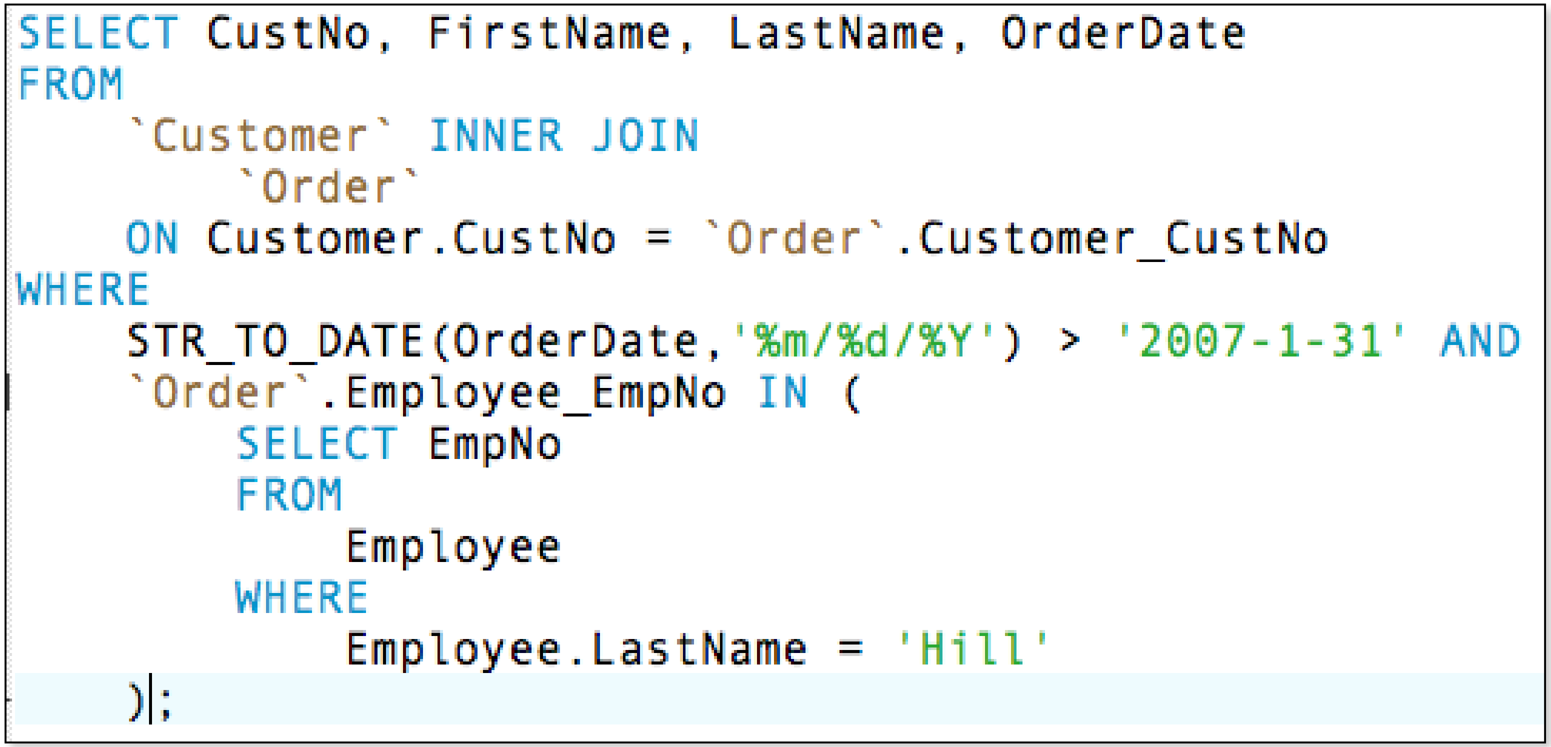
CustNo	FirstName	OrderDate
C8543321	Ron	2/11/2007
C8543321	Ron	2/19/2007
C9549302	Todd	2/10/2007
C9857432	Homer	2/11/2007

Nested Queries + Join

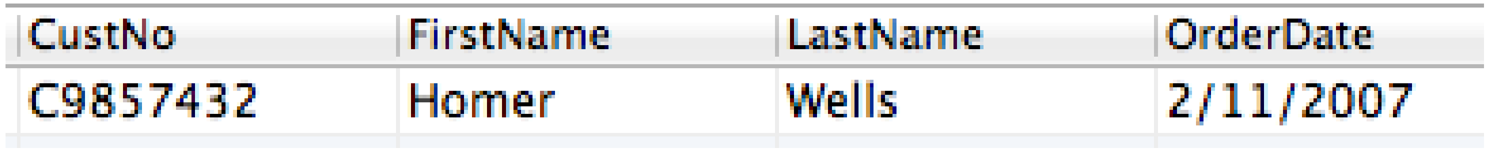
- What if we want the joined table to match with another table?
- We may combine join operation and nested queries.
- E.g., display customer info and order date for those order done by the employee Mr. Hill.



Nested Queries + Join



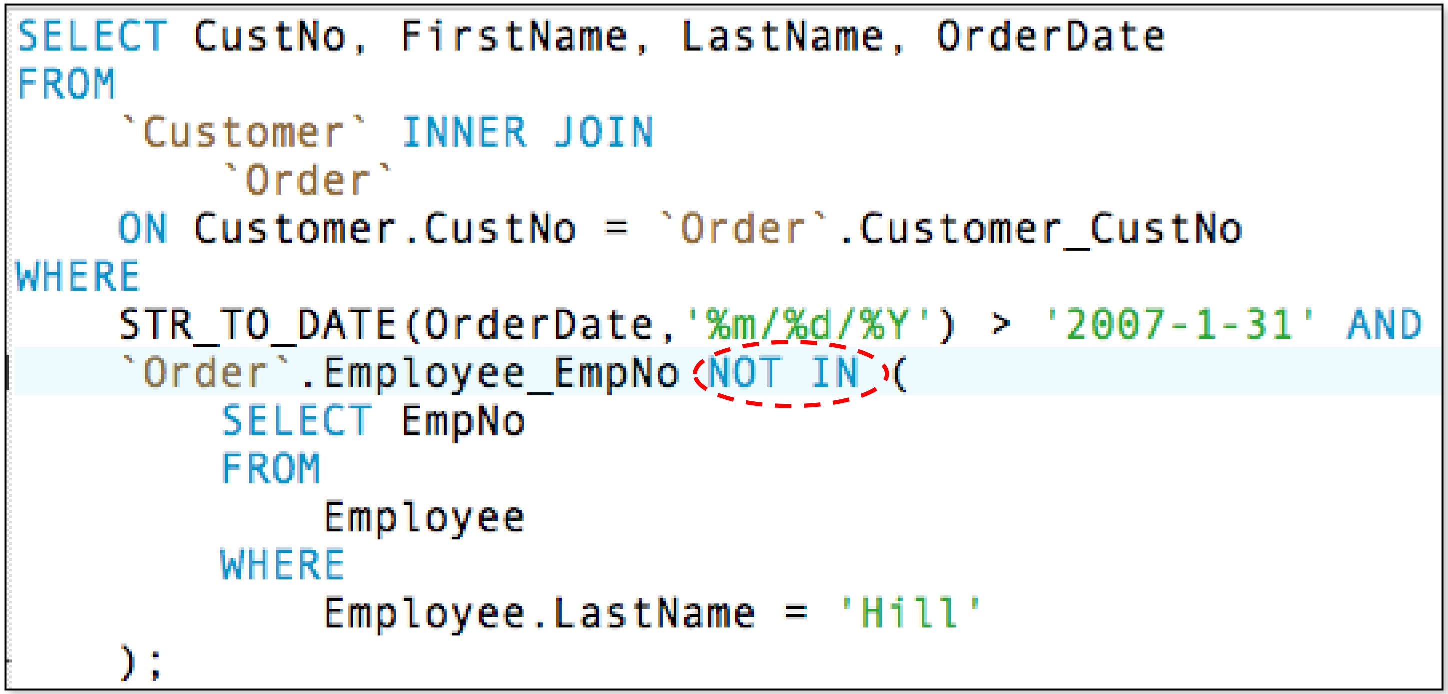
```
SELECT CustNo, FirstName, LastName, OrderDate
FROM
    `Customer` INNER JOIN
        `Order`
    ON Customer.CustNo = `Order`.Customer_CustNo
WHERE
    STR_TO_DATE(OrderDate,'%m/%d/%Y') > '2007-1-31' AND
    `Order`.Employee_EmpNo IN (
        SELECT EmpNo
        FROM
            Employee
        WHERE
            Employee.LastName = 'Hill'
    );
```



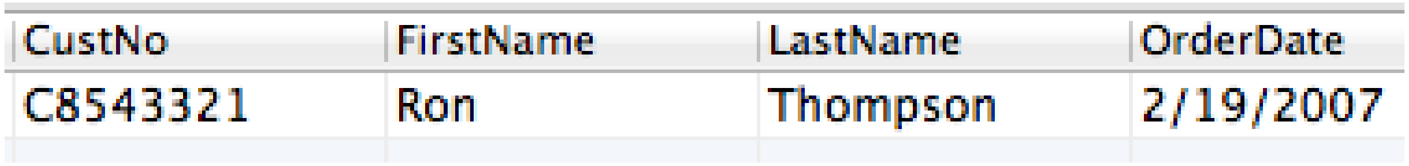
CustNo	FirstName	LastName	OrderDate
C9857432	Homer	Wells	2/11/2007

Nested Queries + Join

- E.g., display customer info and order date for those order NOT done by the employee Mr. Hill.



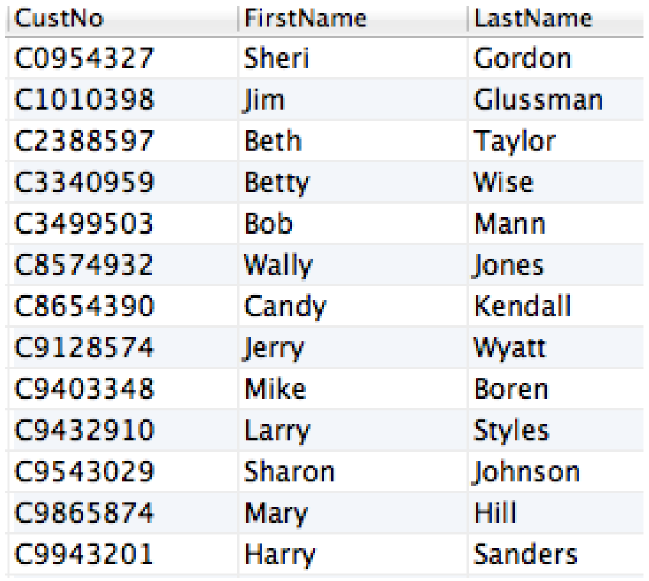
```
SELECT CustNo, FirstName, LastName, OrderDate
FROM
    `Customer` INNER JOIN
        `Order`
    ON Customer.CustNo = `Order`.Customer_CustNo
WHERE
    STR_TO_DATE(OrderDate,'%m/%d/%Y') > '2007-1-31' AND
    `Order`.Employee_EmpNo NOT IN (
        SELECT EmpNo
        FROM
            Employee
        WHERE
            Employee.LastName = 'Hill'
    );
```



CustNo	FirstName	LastName	OrderDate
C8543321	Ron	Thompson	2/19/2007

Nested Query Exercise

- List the customer number and name of those customers who have ***not*** placed any order since Feb 2007.

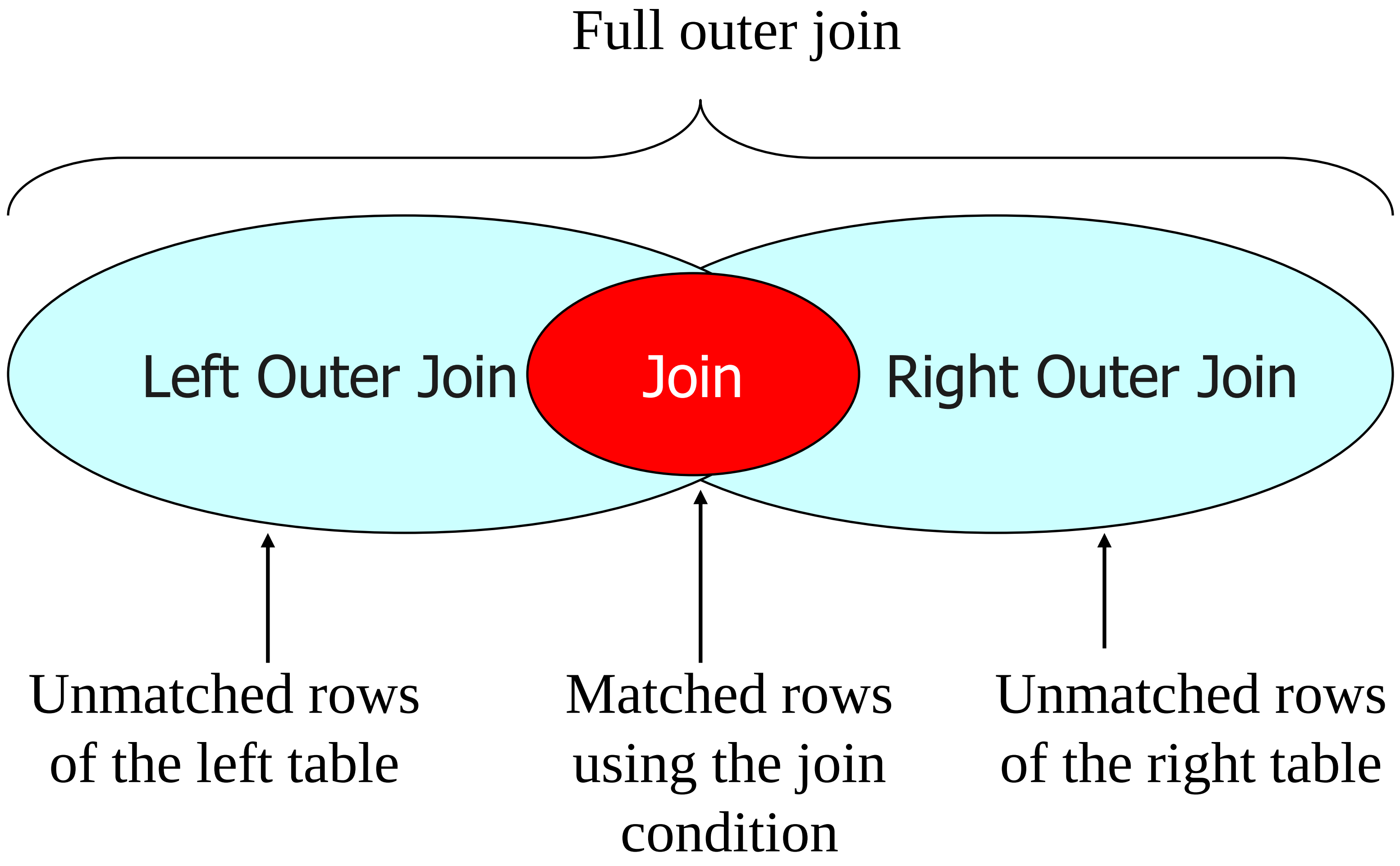


CustNo	FirstName	LastName
C0954327	Sheri	Gordon
C1010398	Jim	Glussman
C2388597	Beth	Taylor
C3340959	Betty	Wise
C3499503	Bob	Mann
C8574932	Wally	Jones
C8654390	Candy	Kendall
C9128574	Jerry	Wyatt
C9403348	Mike	Boren
C9432910	Larry	Styles
C9543029	Sharon	Johnson
C9865874	Mary	Hill
C9943201	Harry	Sanders

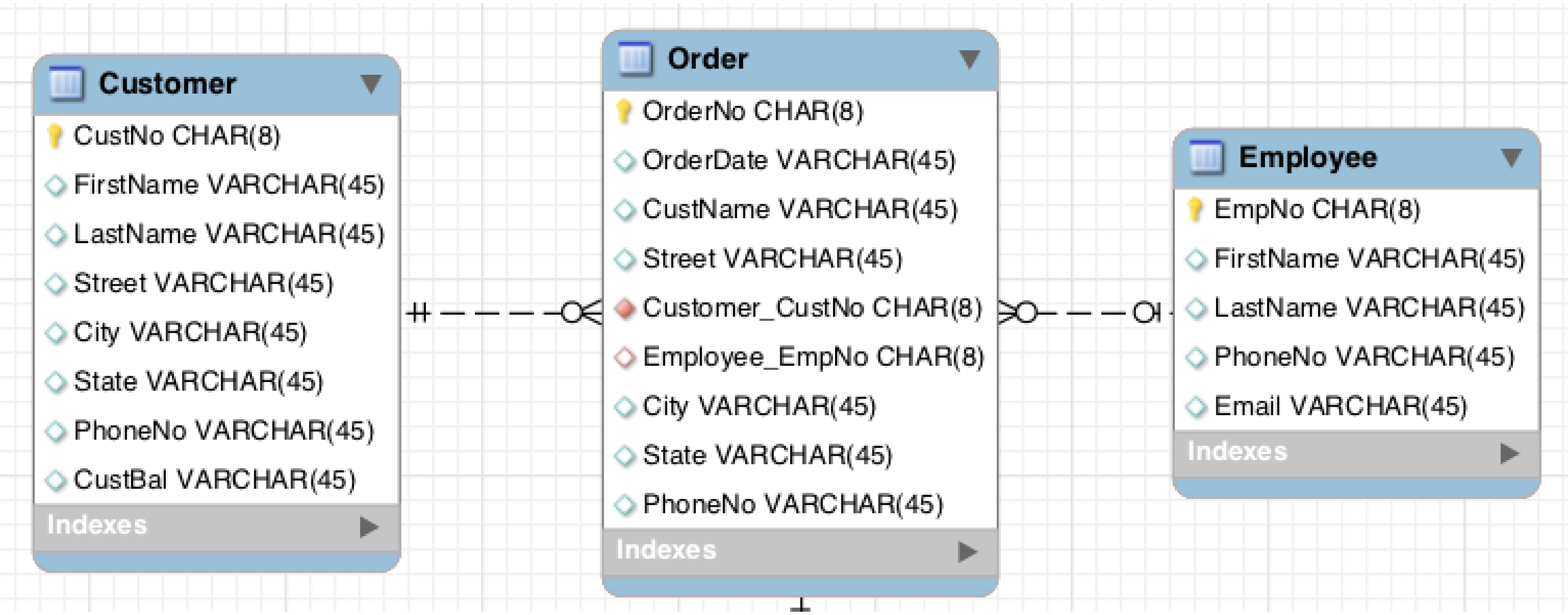
Outer Joins

- Join: excludes non matching rows
- One-sided outer join: generate a new table with the matching rows + non-matching rows from one of the tables
  - LEFT JOIN: include non-matching rows of the **left** table
  - RIGHT JOIN: include non-matching rows of the **right** table

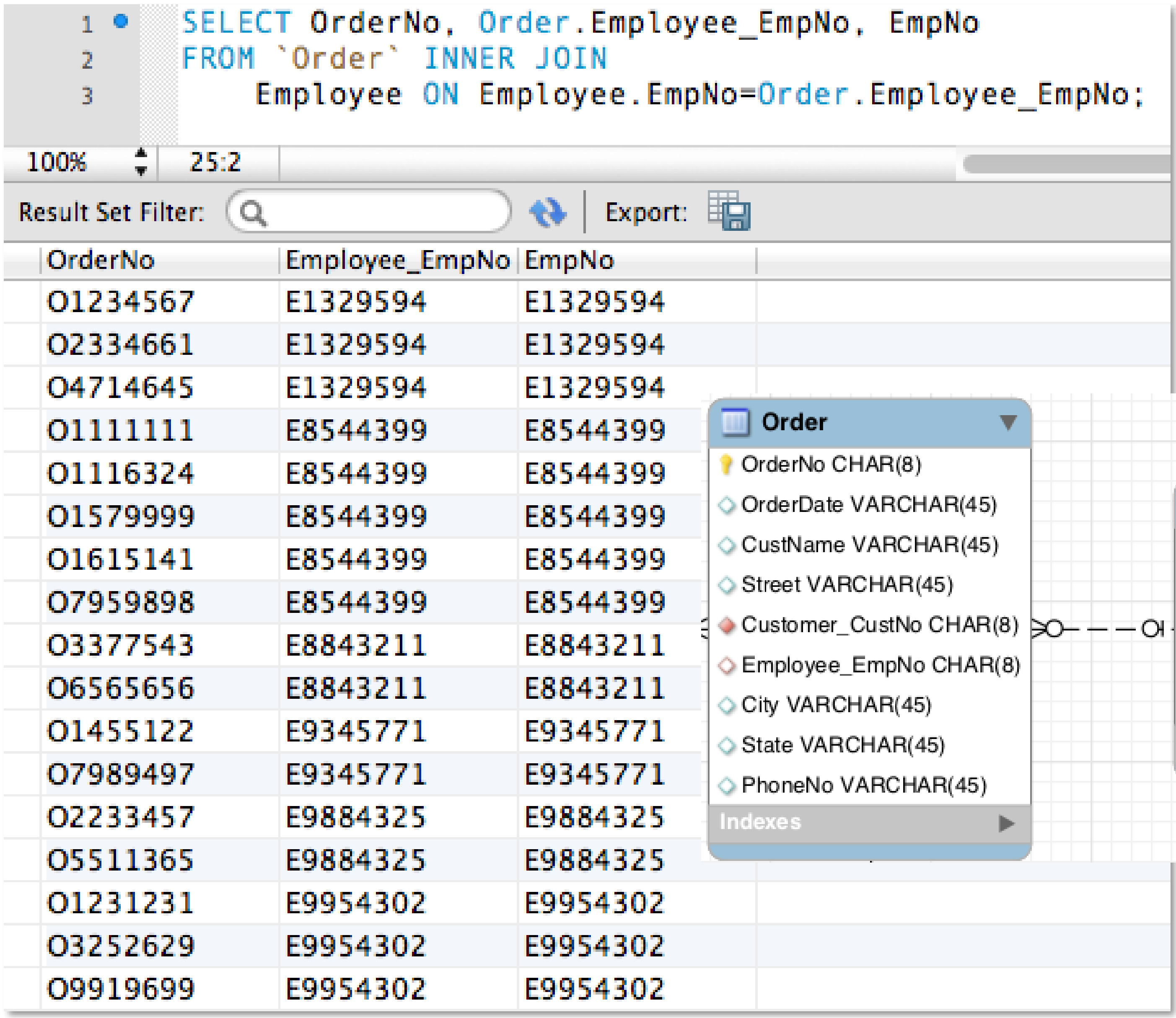
Outer Joins


Full outer join
Left Outer Join
Join
Right Outer Join
Unmatched rows of the left table
Matched rows using the join condition
Unmatched rows of the right table

Example of Outer Joins



Example: Inner Join

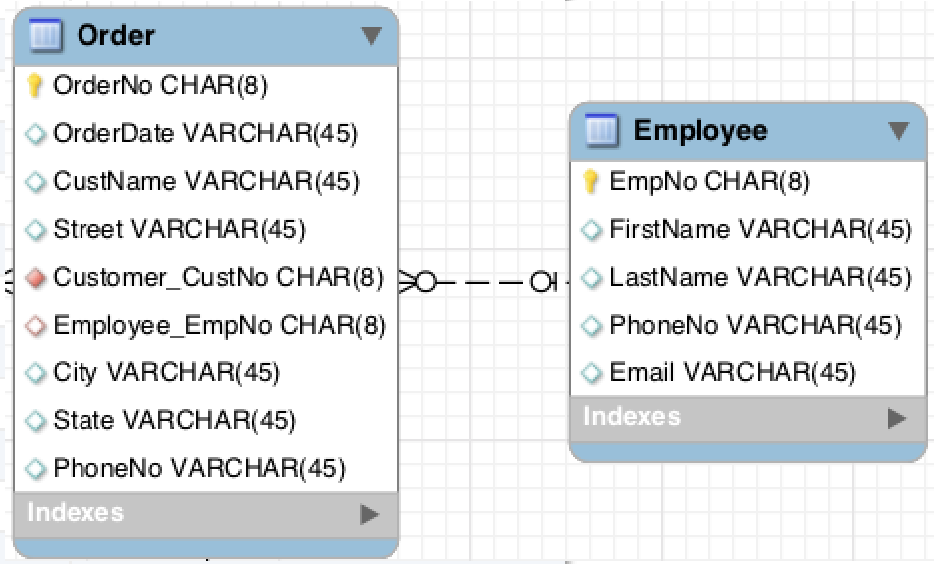


```
SELECT OrderNo, Order.Employee_EmpNo, EmpNo
FROM `Order` INNER JOIN
    Employee ON Employee.EmpNo=Order.Employee_EmpNo;
```

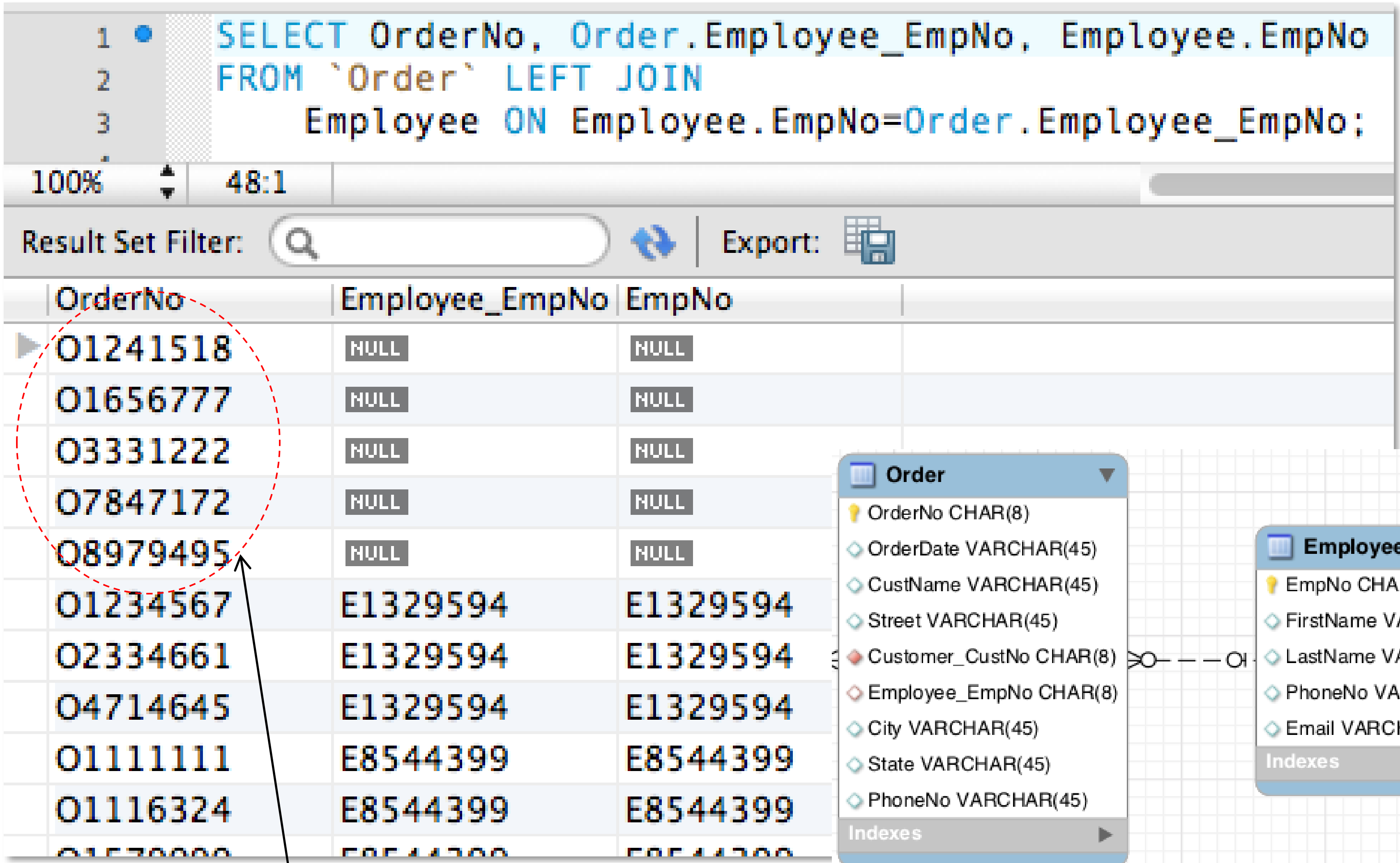
100% 25:2

Result Set Filter: Export:

OrderNo	Employee_EmpNo	EmpNo
O1234567	E1329594	E1329594
O2334661	E1329594	E1329594
O4714645	E1329594	E1329594
O1111111	E8544399	E8544399
O1116324	E8544399	E8544399
O1579999	E8544399	E8544399
O1615141	E8544399	E8544399
O7959898	E8544399	E8544399
O3377543	E8843211	E8843211
O6565656	E8843211	E8843211
O1455122	E9345771	E9345771
O7989497	E9345771	E9345771
O2233457	E9884325	E9884325
O5511365	E9884325	E9884325
O1231231	E9954302	E9954302
O3252629	E9954302	E9954302
O9919699	E9954302	E9954302

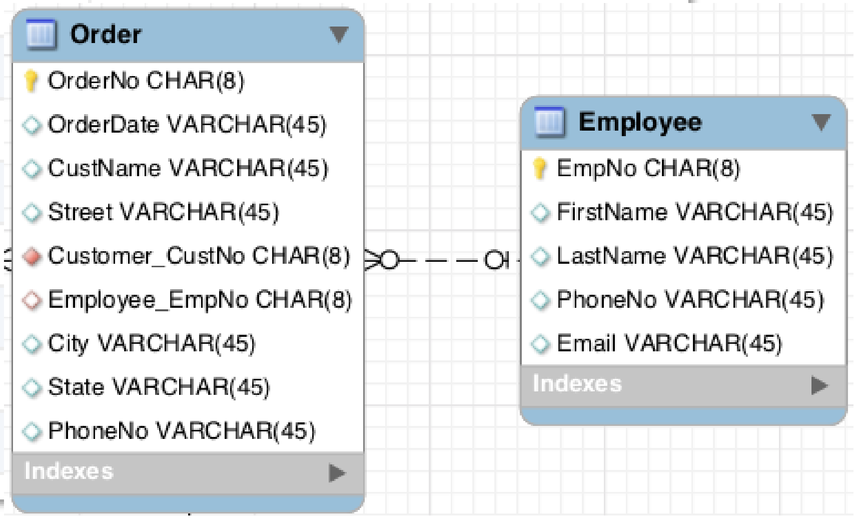


Example: Left Join



```
SELECT OrderNo, Order.Employee_EmpNo, Employee.EmpNo
FROM `Order` LEFT JOIN
    Employee ON Employee.EmpNo=Order.Employee_EmpNo;
```

OrderNo	Employee_EmpNo	EmpNo
O1241518	NULL	NULL
O1656777	NULL	NULL
O3331222	NULL	NULL
O7847172	NULL	NULL
O8979495	NULL	NULL
O1234567	E1329594	E1329594
O2334661	E1329594	E1329594
O4714645	E1329594	E1329594
O1111111	E8544399	E8544399
O1116324	E8544399	E8544399

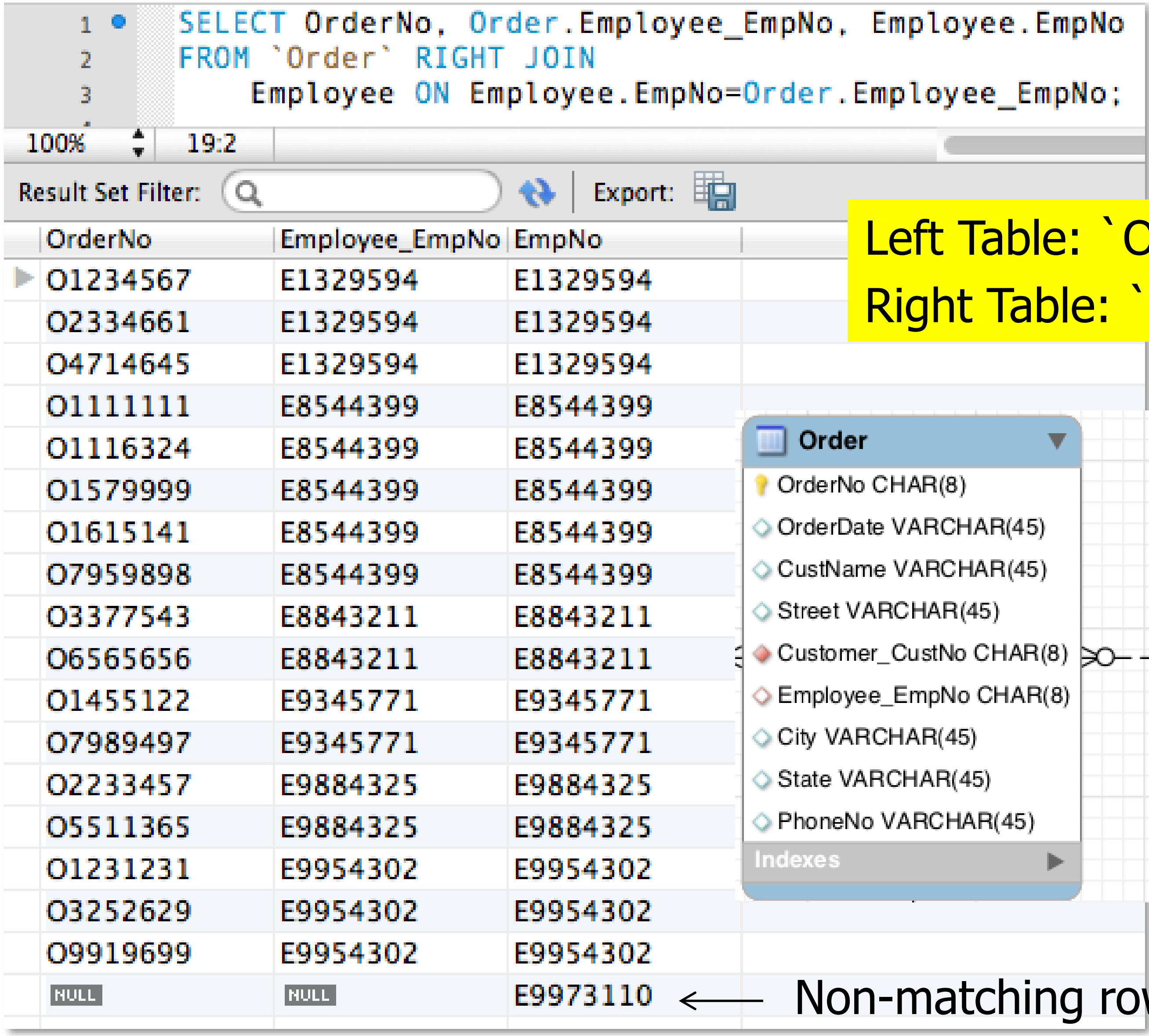


Non-matching rows of left table

Left Table: `Order`

Right Table: `Employee`

Example: Right Join

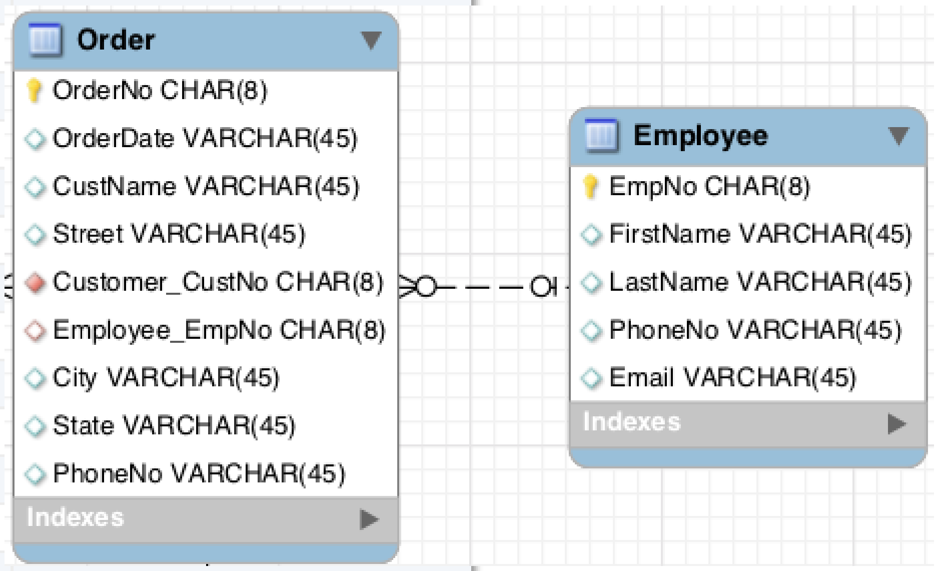


```
SELECT OrderNo, Order.Employee_EmpNo, Employee.EmpNo
FROM `Order` RIGHT JOIN
    Employee ON Employee.EmpNo=Order.Employee_EmpNo;
```

OrderNo	Employee_EmpNo	EmpNo
O1234567	E1329594	E1329594
O2334661	E1329594	E1329594
O4714645	E1329594	E1329594
O1111111	E8544399	E8544399
O1116324	E8544399	E8544399
O1579999	E8544399	E8544399
O1615141	E8544399	E8544399
O7959898	E8544399	E8544399
O3377543	E8843211	E8843211
O6565656	E8843211	E8843211
O1455122	E9345771	E9345771
O7989497	E9345771	E9345771
O2233457	E9884325	E9884325
O5511365	E9884325	E9884325
O1231231	E9954302	E9954302
O3252629	E9954302	E9954302
O9919699	E9954302	E9954302
NULL	NULL	E9973110

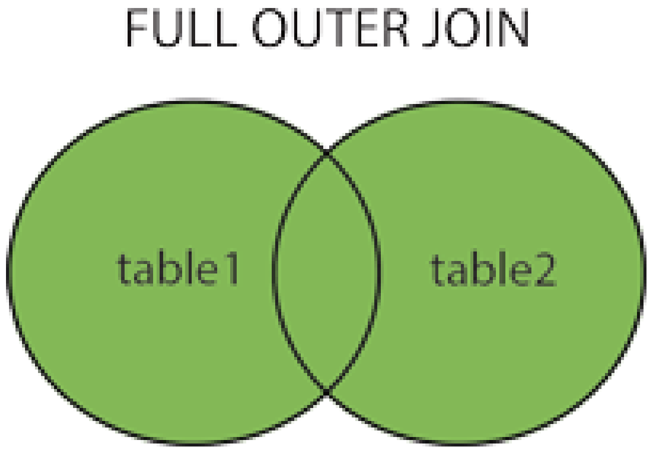
Non-matching rows of right table

Left Table: `Order`
Right Table: `Employee`

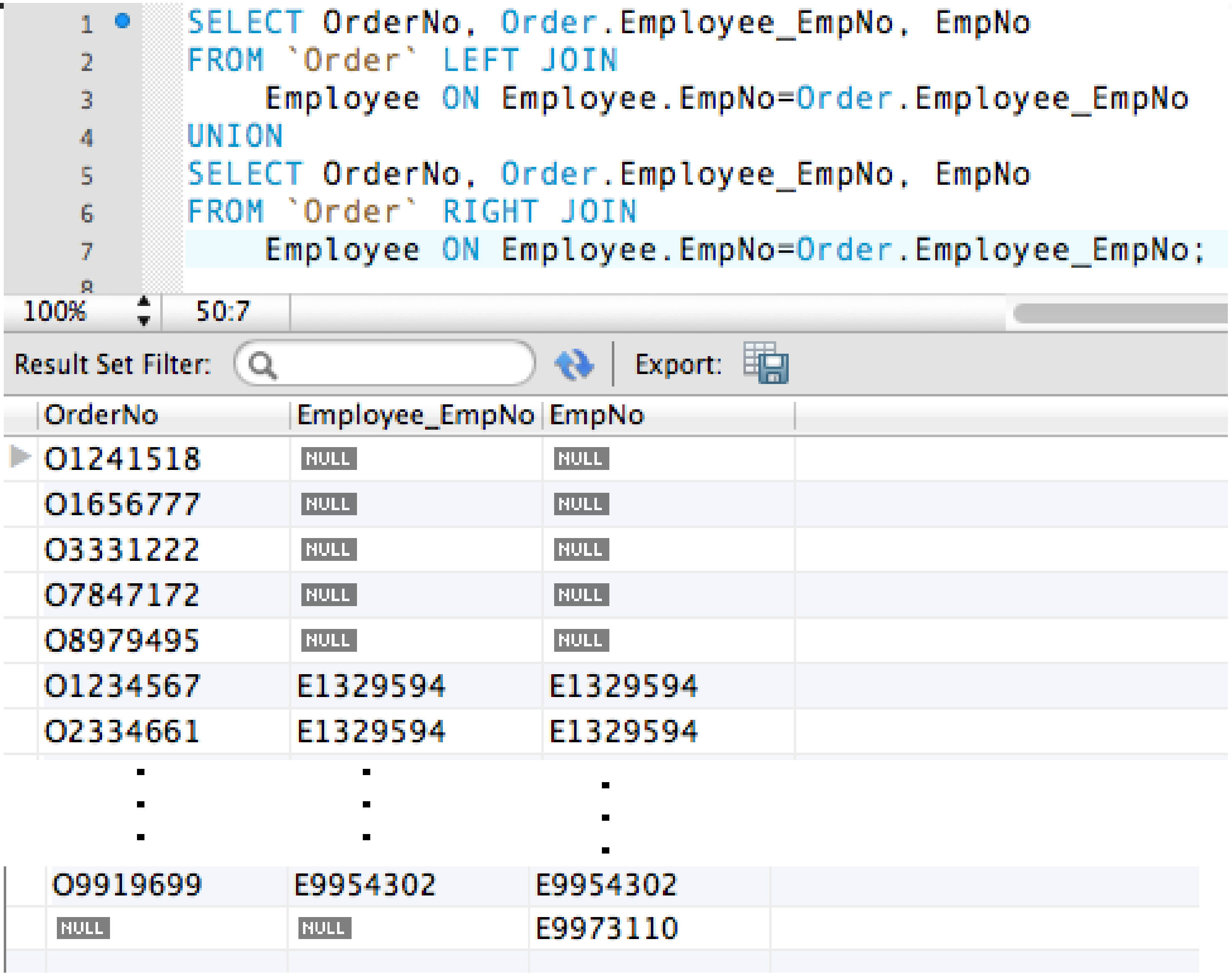


Example: Full Outer Join

- The FULL OUTER JOIN keyword returns all rows from the left table (table1) and from the right table (table2).
- The FULL OUTER JOIN keyword combines the result of both LEFT and RIGHT joins.
- Not supported by MySQL. But can be worked around (see next page)



Example: Emulating Full Join



```sql
SELECT OrderNo, Order.Employee_EmpNo, EmpNo
FROM `Order` LEFT JOIN
    Employee ON Employee.EmpNo=Order.Employee_EmpNo
UNION
SELECT OrderNo, Order.Employee_EmpNo, EmpNo
FROM `Order` RIGHT JOIN
    Employee ON Employee.EmpNo=Order.Employee_EmpNo;
```

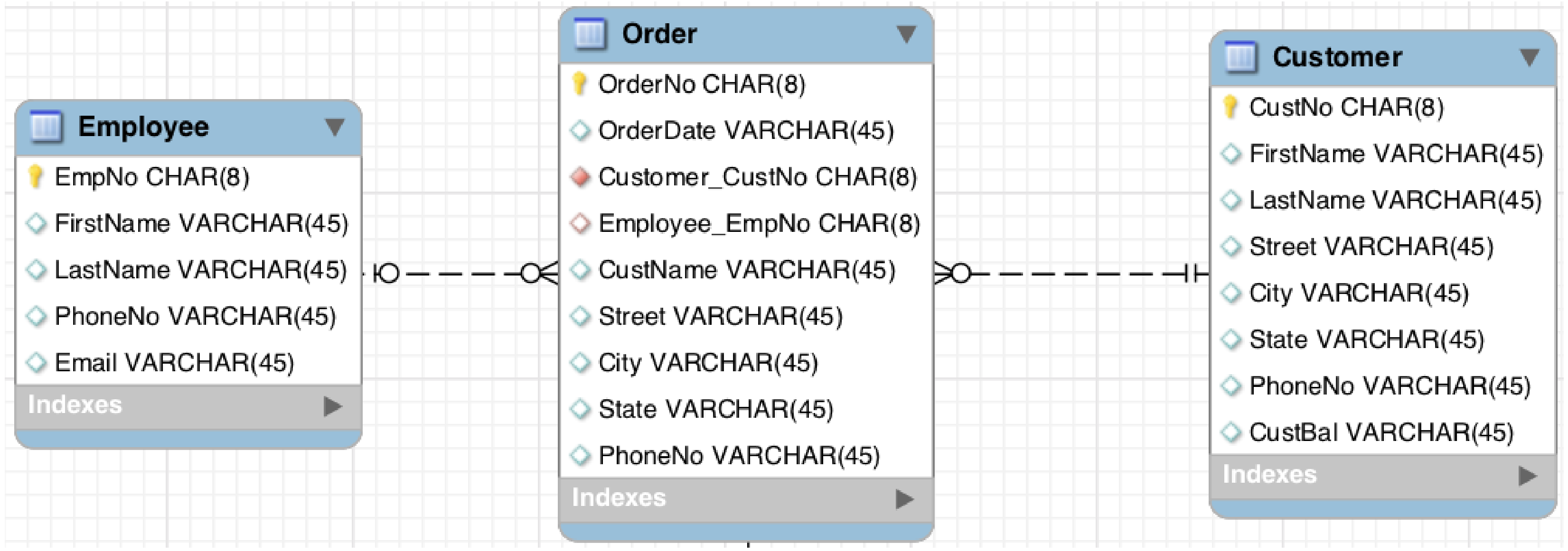
100% 50:7

Result Set Filter: Export:

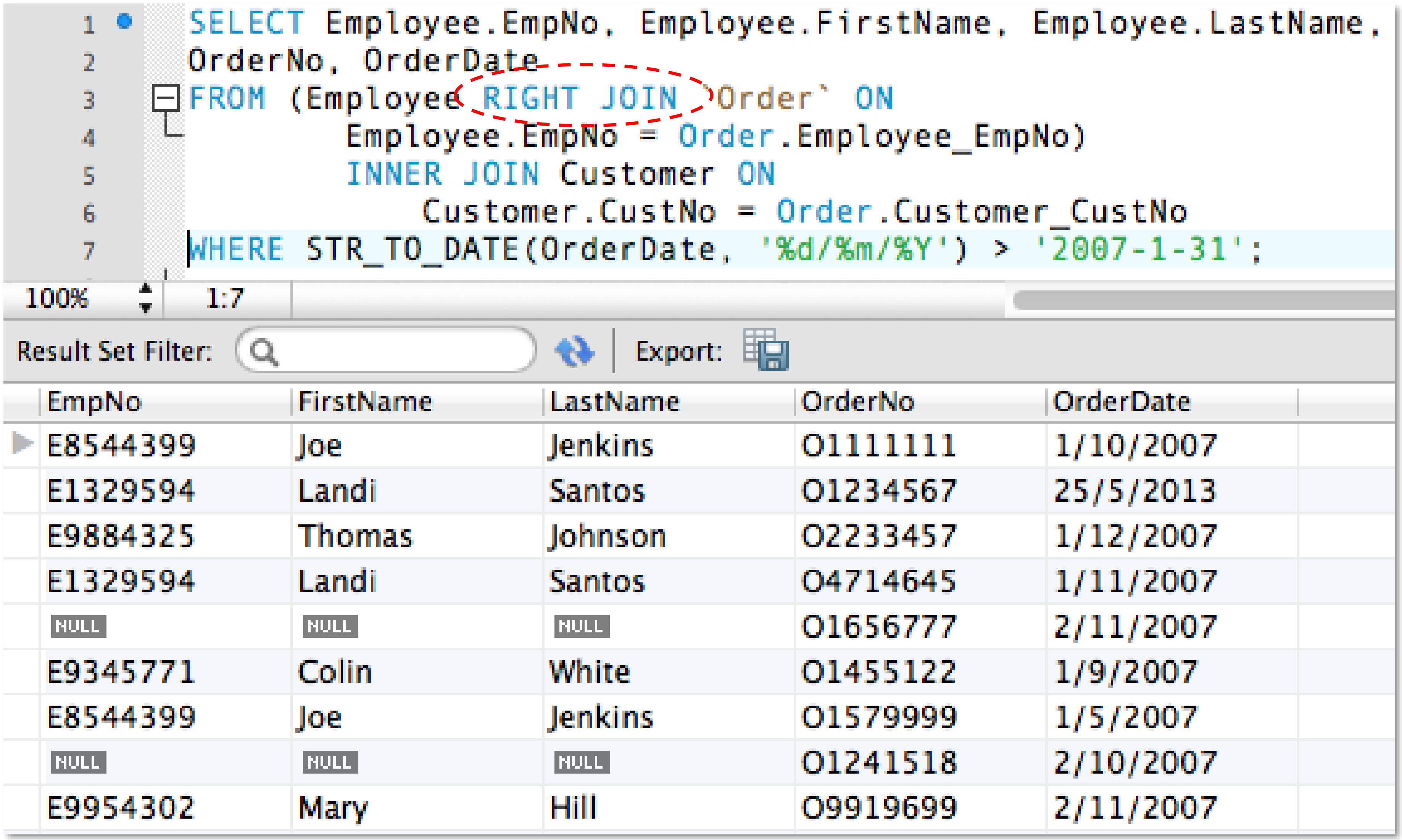
OrderNo	Employee_EmpNo	EmpNo
O1241518	NULL	NULL
O1656777	NULL	NULL
O3331222	NULL	NULL
O7847172	NULL	NULL
O8979495	NULL	NULL
O1234567	E1329594	E1329594
O2334661	E1329594	E1329594
⋮	⋮	⋮
O9919699	E9954302	E9954302
NULL	NULL	E9973110

Mixing Outer and Inner Join

- Combine Customer, Order and Employee
- List employee no., employee name, order number, and order date, where date of order is after Jan 2007
- Include rows that do not have employees.



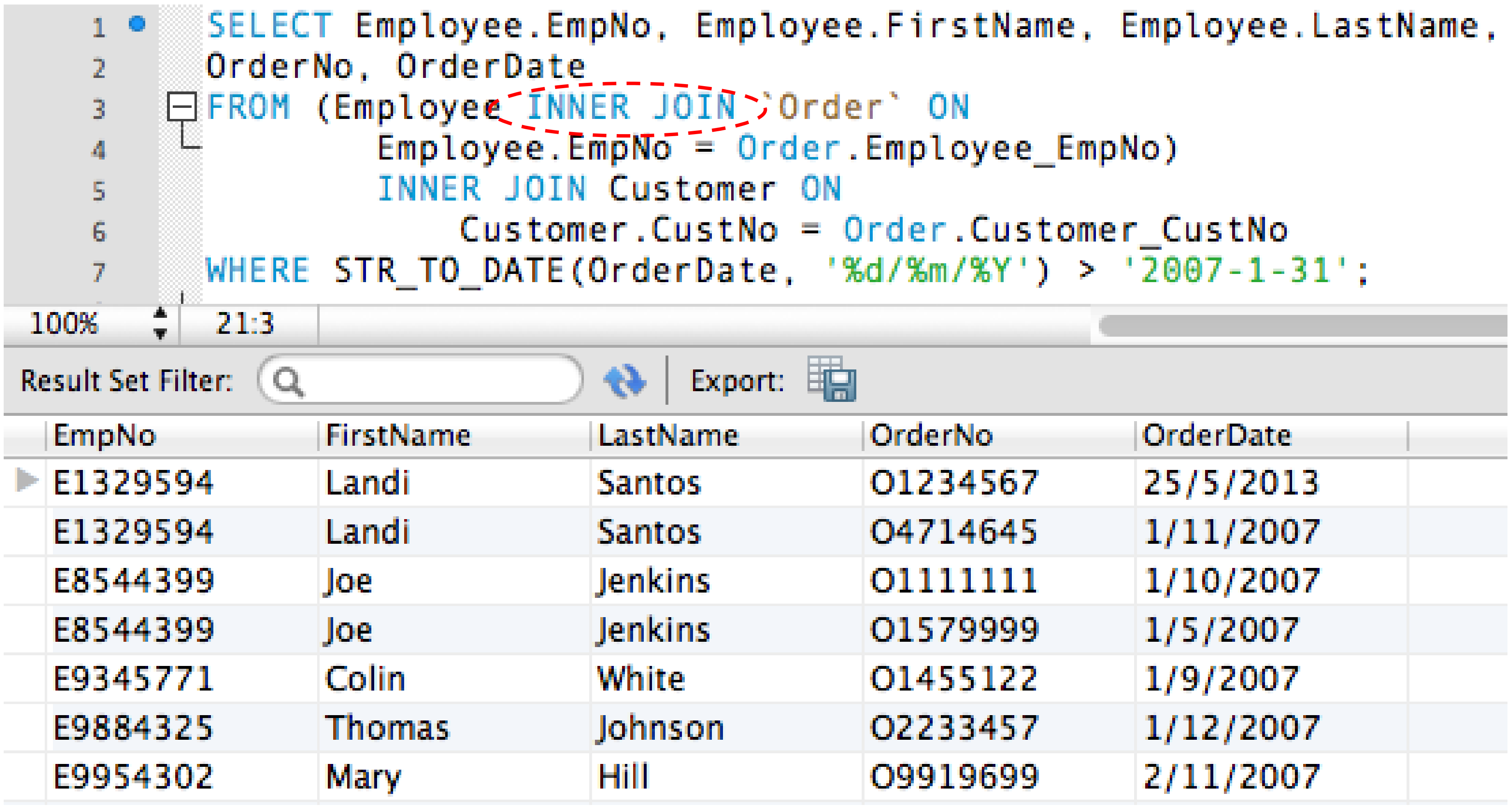
Mixing Outer and Inner Join



```
SELECT Employee.EmpNo, Employee.FirstName, Employee.LastName,
OrderNo, OrderDate
FROM (Employee RIGHT JOIN `Order` ON
        Employee.EmpNo = Order.Employee_EmpNo)
        INNER JOIN Customer ON
            Customer.CustNo = Order.Customer_CustNo
WHERE STR_TO_DATE(OrderDate, '%d/%m/%Y') > '2007-1-31';
```

EmpNo	FirstName	LastName	OrderNo	OrderDate
E8544399	Joe	Jenkins	O1111111	1/10/2007
E1329594	Landi	Santos	O1234567	25/5/2013
E9884325	Thomas	Johnson	O2233457	1/12/2007
E1329594	Landi	Santos	O4714645	1/11/2007
NULL	NULL	NULL	O1656777	2/11/2007
E9345771	Colin	White	O1455122	1/9/2007
E8544399	Joe	Jenkins	O1579999	1/5/2007
NULL	NULL	NULL	O1241518	2/10/2007
E9954302	Mary	Hill	O9919699	2/11/2007

Compared with Inner Join



```
SELECT Employee.EmpNo, Employee.FirstName, Employee.LastName,
OrderNo, OrderDate
FROM (Employee INNER JOIN `Order` ON
        Employee.EmpNo = Order.Employee_EmpNo)
        INNER JOIN Customer ON
            Customer.CustNo = Order.Customer_CustNo
WHERE STR_TO_DATE(OrderDate, '%d/%m/%Y') > '2007-1-31';
```

100% 21:3

Result Set Filter: Export:

EmpNo	FirstName	LastName	OrderNo	OrderDate
E1329594	Landi	Santos	O1234567	25/5/2013
E1329594	Landi	Santos	O4714645	1/11/2007
E8544399	Joe	Jenkins	O1111111	1/10/2007
E8544399	Joe	Jenkins	O1579999	1/5/2007
E9345771	Colin	White	O1455122	1/9/2007
E9884325	Thomas	Johnson	O2233457	1/12/2007
E9954302	Mary	Hill	O9919699	2/11/2007

Summarizing Tables

- Row summary: compress multiple rows into a single row
- Row summaries are important for decision-making tasks
  - Contains statistical (aggregate) functions, e.g.,
    - counts the no. of customers
    - compute the average balance of customers
  - Conditions involve aggregate functions, e.g.,
    - Average balance < 10

Summarizing Tables

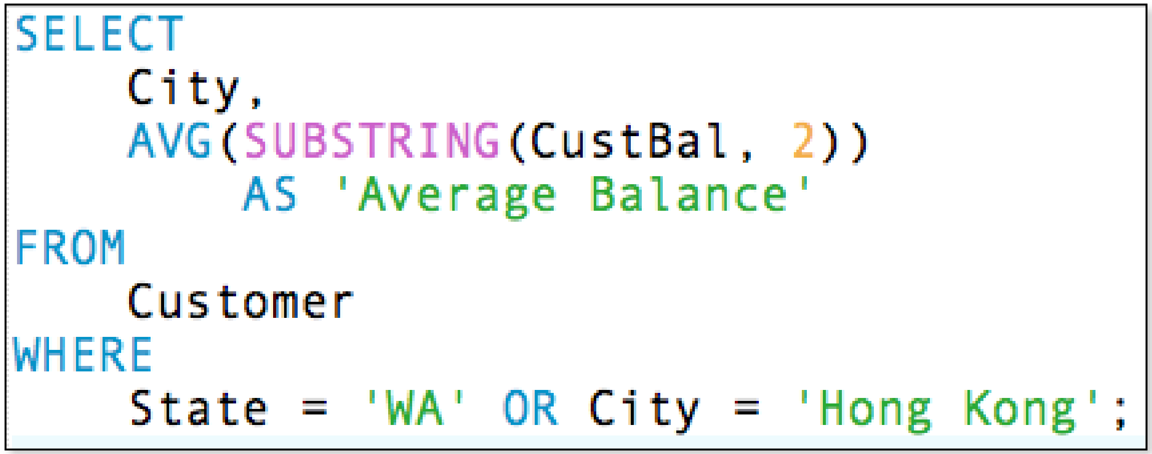
- SQL keywords:
  - Standard aggregate functions
    - `COUNT, MIN, MAX, SUM, AVG`
  - `GROUP BY` columns:
    - indicate columns to summarize on
  - `Having` (optional)
    - Indicate group conditions

Example of GROUP BY

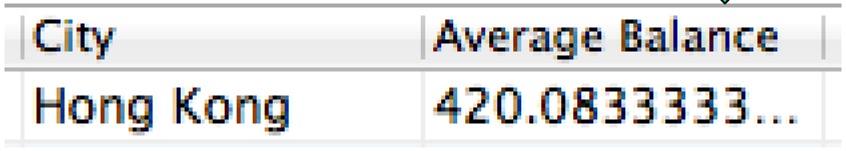
Query:

List the *average balance* of customers by *city*. Include only customers residing in Washington State (WA) or residing in Hong Kong.

Without using GROUP BY:



```
SELECT
    City,
    AVG(SUBSTRING(CustBal, 2))
        AS 'Average Balance'
FROM
    Customer
WHERE
    State = 'WA' OR City = 'Hong Kong';
```



City	Average Balance
Hong Kong	420.0833333...

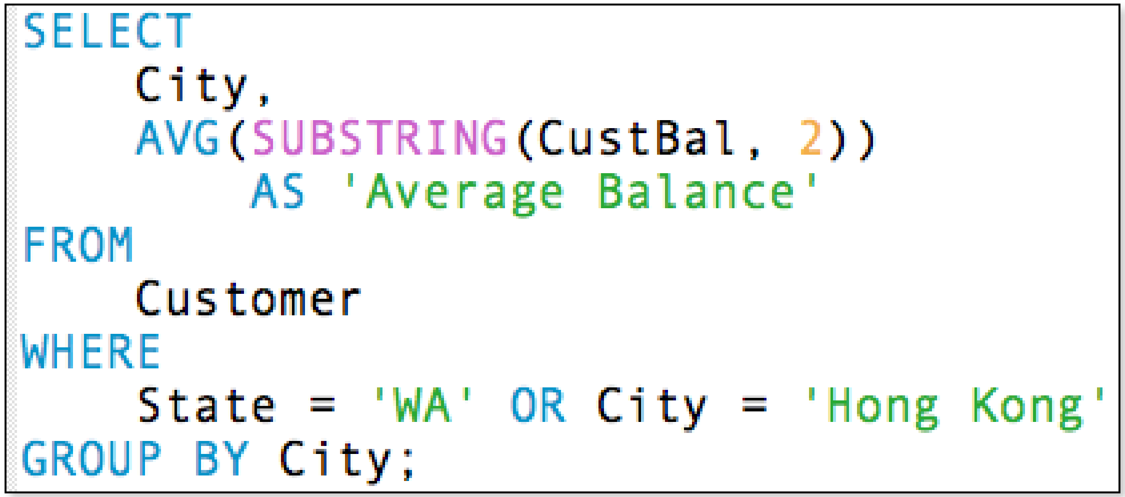
Without "GROUP BY", can only display the overall average

Example of GROUP BY

Query:

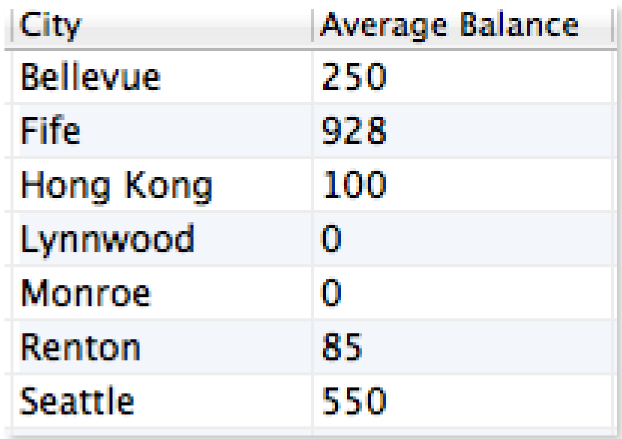
List the *average balance* of customers by *city*. Include only customers residing in Washington State (WA) or residing in Hong Kong.

Solution:



```
SELECT
    City,
    AVG(SUBSTRING(CustBal, 2))
        AS 'Average Balance'
FROM
    Customer
WHERE
    State = 'WA' OR City = 'Hong Kong'
GROUP BY City;
```

⇨



City	Average Balance
Bellevue	250
Fife	928
Hong Kong	100
Lynnwood	0
Monroe	0
Renton	85
Seattle	550

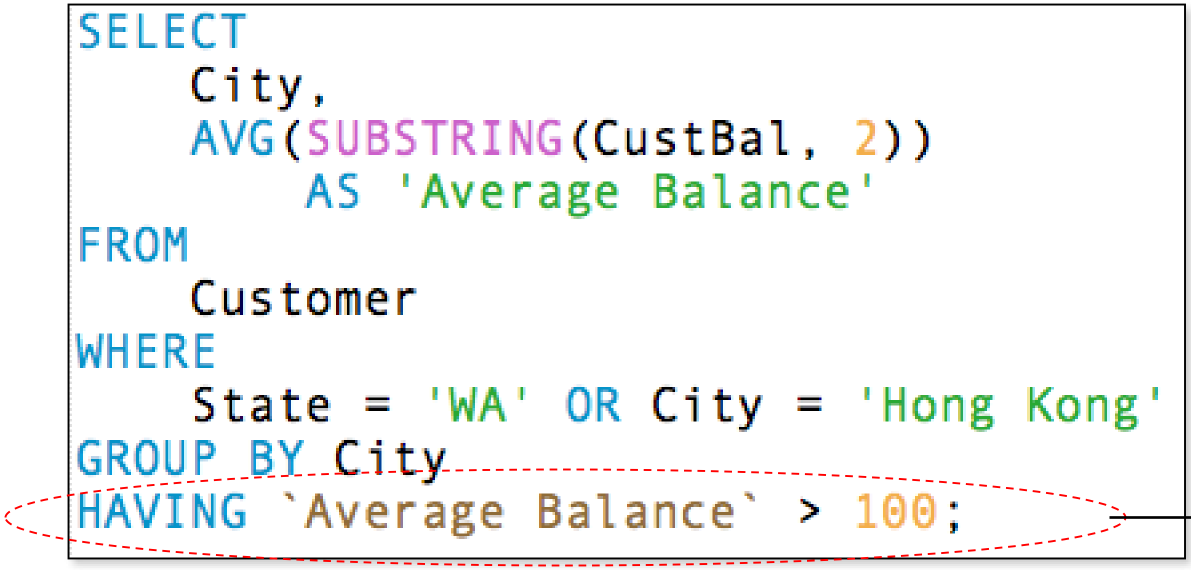
With "GROUP BY", we can display the average balance of each individual cities

Example of HAVING

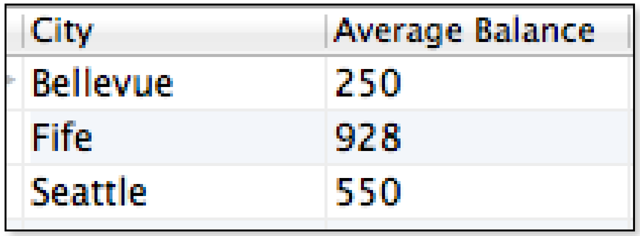
Query:

List the *average balance* of customers by *city*. Include only customers residing in Washington State (WA) or residing in Hong Kong. *Only list the city whose balance is greater than 100.*

Solution:



```
SELECT
    City,
    AVG(SUBSTRING(CustBal, 2))
        AS 'Average Balance'
FROM
    Customer
WHERE
    State = 'WA' OR City = 'Hong Kong'
GROUP BY City
HAVING `Average Balance` > 100;
```



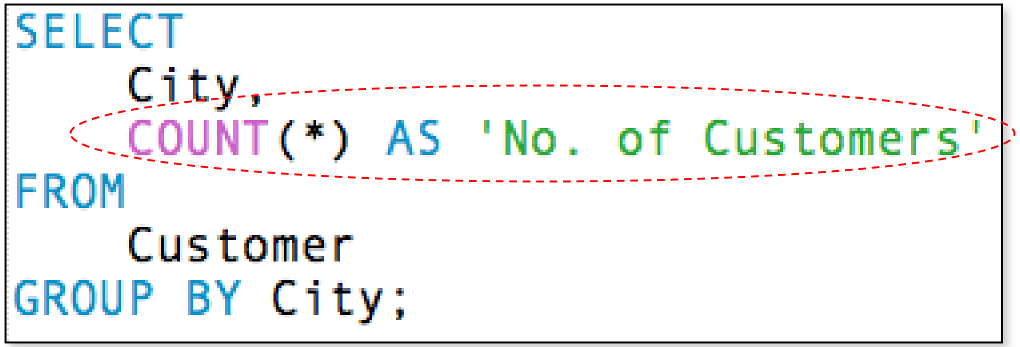
City	Average Balance
Bellevue	250
Fife	928
Seattle	550

Example of COUNT

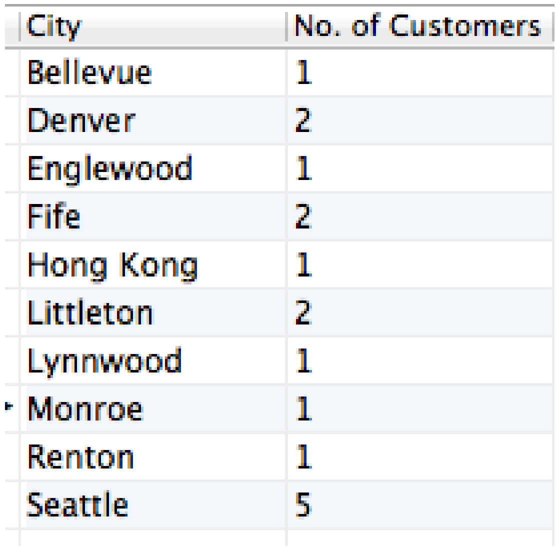
Query:

List the *number of customers* by city.

Solution:



```
SELECT
    City,
    COUNT(*) AS 'No. of Customers'
FROM
    Customer
GROUP BY City;
```

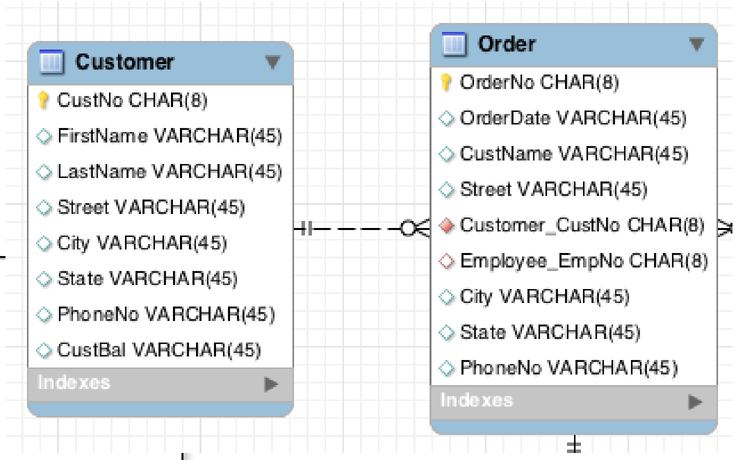


City	No. of Customers
Bellevue	1
Denver	2
Englewood	1
Fife	2
Hong Kong	1
Littleton	2
Lynnwood	1
Monroe	1
Renton	1
Seattle	5

Example of COUNT + JOIN

Query:

List the *number of customers* whose *shipping orders* are to the state 'WA'



Solution:

```
SELECT
    Customer.City,
    `Order`.State,
    COUNT(*) AS 'No. of Customers'
FROM
    Customer, `Order`
WHERE `Order`.Customer_CustNo = Customer.CustNo
    AND `Order`.State = 'WA'
GROUP BY City;
```

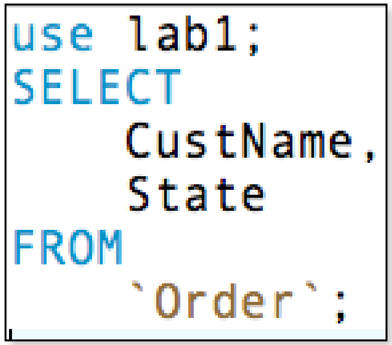
City	State	No. of Customers
Bellevue	WA	1
Fife	WA	2
Littleton	WA	3
Lynnwood	WA	1
Monroe	WA	1
Renton	WA	2
Seattle	WA	6

Rules for GROUP BY

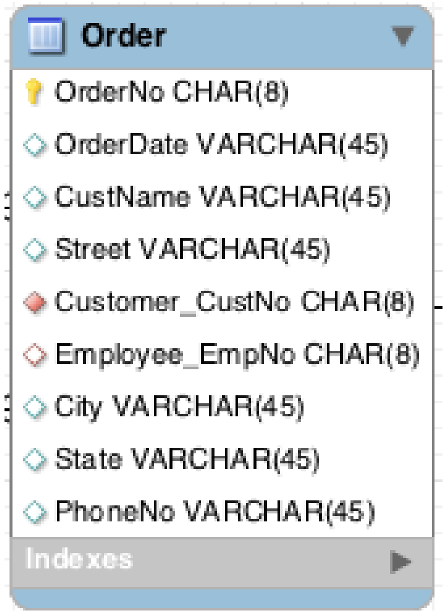
Query:

List all customer (recipient) names and their residing state.

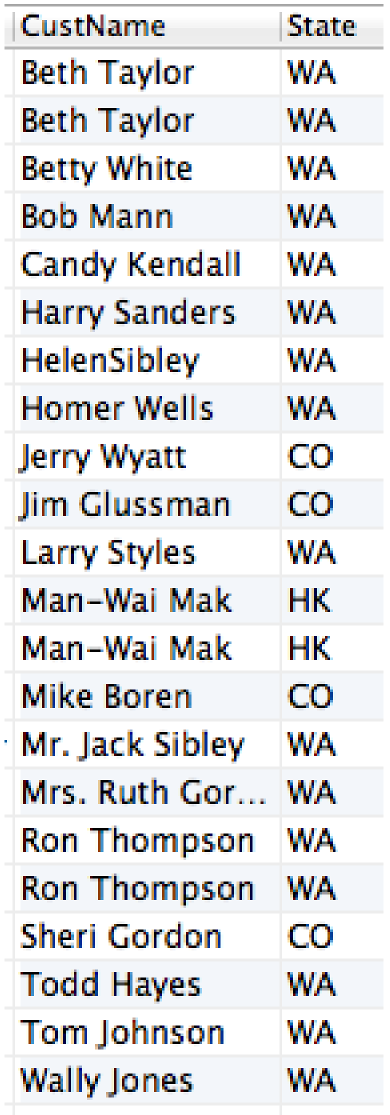
Solution:



```
use lab1;
SELECT
    CustName,
    State
FROM
    `Order`;
```



```
Order
OrderNo CHAR(8)
OrderDate VARCHAR(45)
CustName VARCHAR(45)
Street VARCHAR(45)
Customer_CustNo CHAR(8)
Employee_EmpNo CHAR(8)
City VARCHAR(45)
State VARCHAR(45)
PhoneNo VARCHAR(45)
Indexes
```



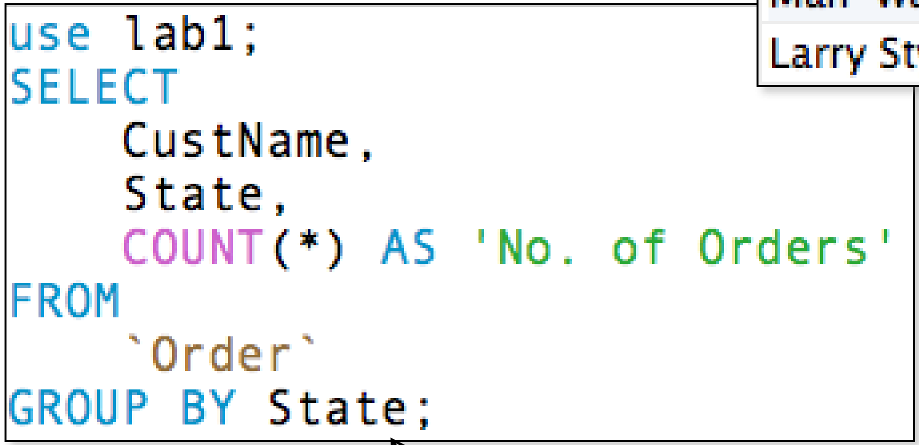
CustName	State
Beth Taylor	WA
Beth Taylor	WA
Betty White	WA
Bob Mann	WA
Candy Kendall	WA
Harry Sanders	WA
HelenSibley	WA
Homer Wells	WA
Jerry Wyatt	CO
Jim Glussman	CO
Larry Styles	WA
Man-Wai Mak	HK
Man-Wai Mak	HK
Mike Boren	CO
Mr. Jack Sibley	WA
Mrs. Ruth Gor...	WA
Ron Thompson	WA
Ron Thompson	WA
Sheri Gordon	CO
Todd Hayes	WA
Tom Johnson	WA
Wally Jones	WA

Rules for GROUP BY

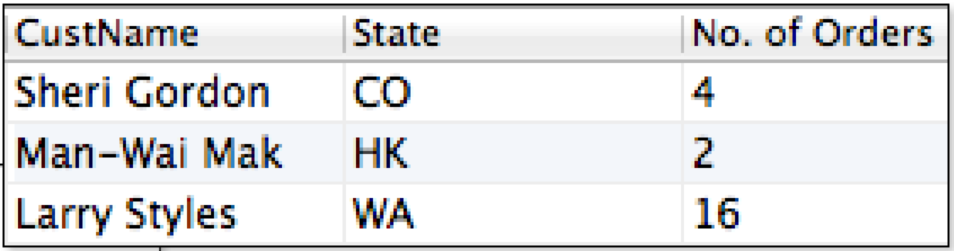
Query:

List all customer (recipient) names, their residing state, and the number of orders that they have placed.

***Incorrect* Solution:**



```
use lab1;
SELECT
    CustName,
    State,
    COUNT(*) AS 'No. of Orders'
FROM
    `Order`
GROUP BY State;
```



CustName	State	No. of Orders
Sheri Gordon	CO	4
Man-Wai Mak	HK	2
Larry Styles	WA	16

Missing many customers, e.g. Wally Jones

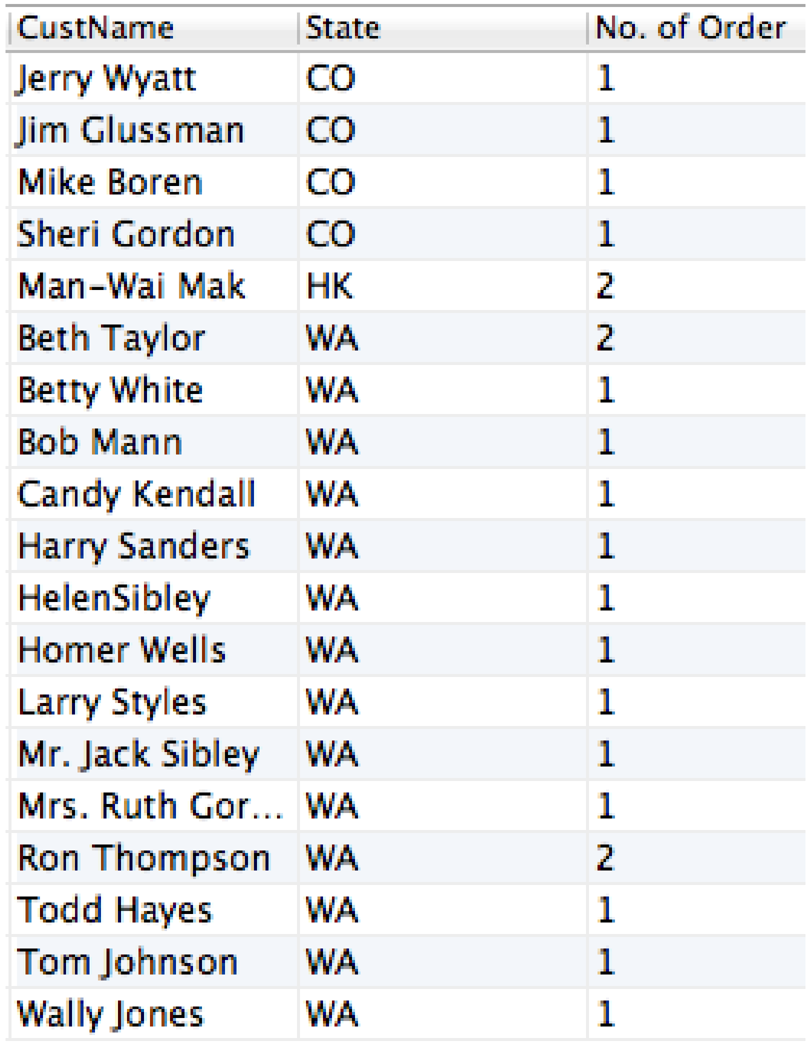
Missing CustName

Rules for GROUP BY

***Correct* Solution:**

```
SELECT
    CustName,
    State,
    COUNT(*) AS 'No. of Orders'
FROM
    `Order`
GROUP BY State, CustName;
```

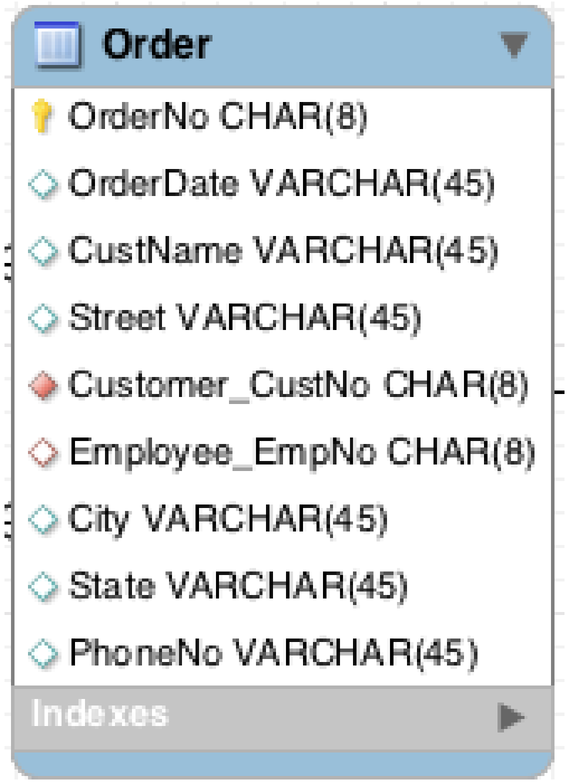
All columns not part of aggregate function must appear here

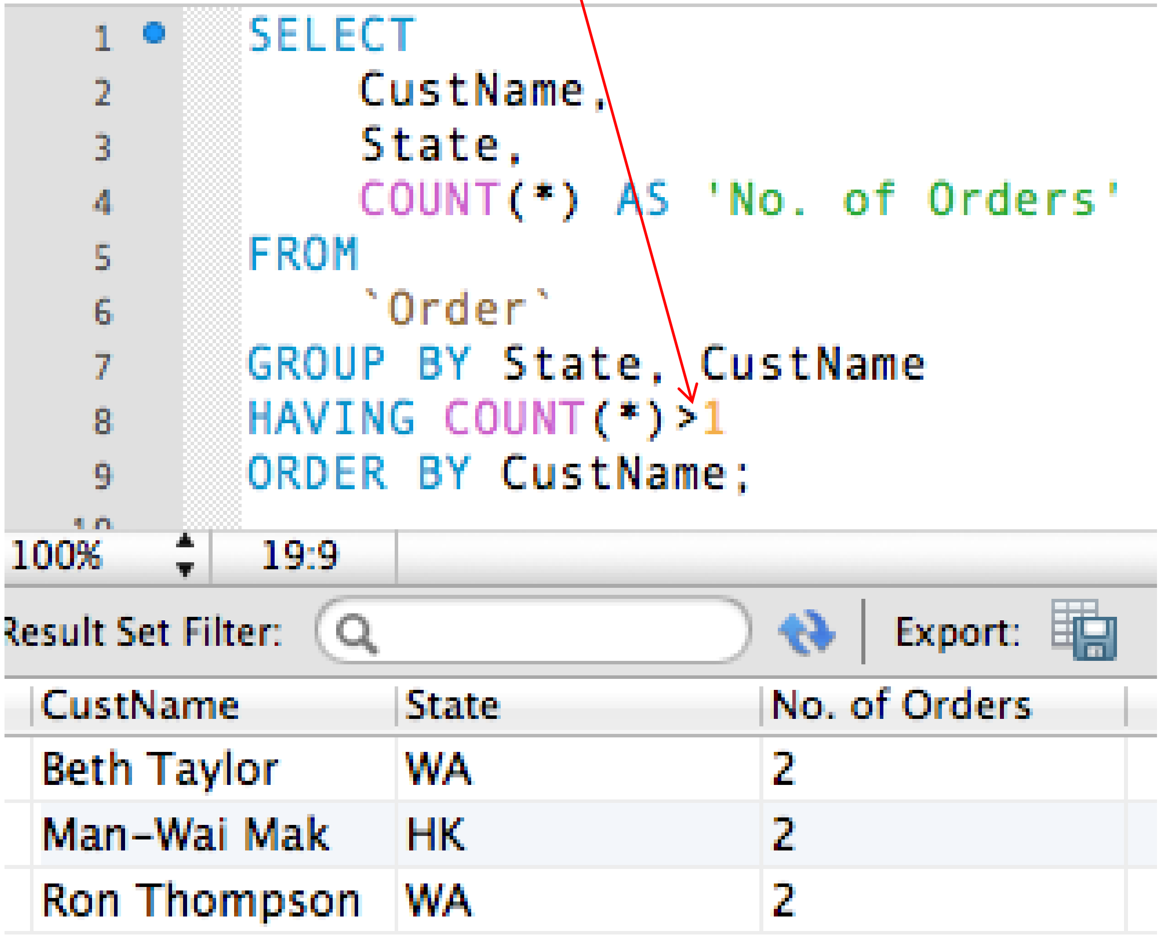


CustName	State	No. of Order
Jerry Wyatt	CO	1
Jim Glussman	CO	1
Mike Boren	CO	1
Sheri Gordon	CO	1
Man-Wai Mak	HK	2
Beth Taylor	WA	2
Betty White	WA	1
Bob Mann	WA	1
Candy Kendall	WA	1
Harry Sanders	WA	1
HelenSibley	WA	1
Homer Wells	WA	1
Larry Styles	WA	1
Mr. Jack Sibley	WA	1
Mrs. Ruth Gor...	WA	1
Ron Thompson	WA	2
Todd Hayes	WA	1
Tom Johnson	WA	1
Wally Jones	WA	1

Rules for HAVING

Only use conditions that involve an aggregate function



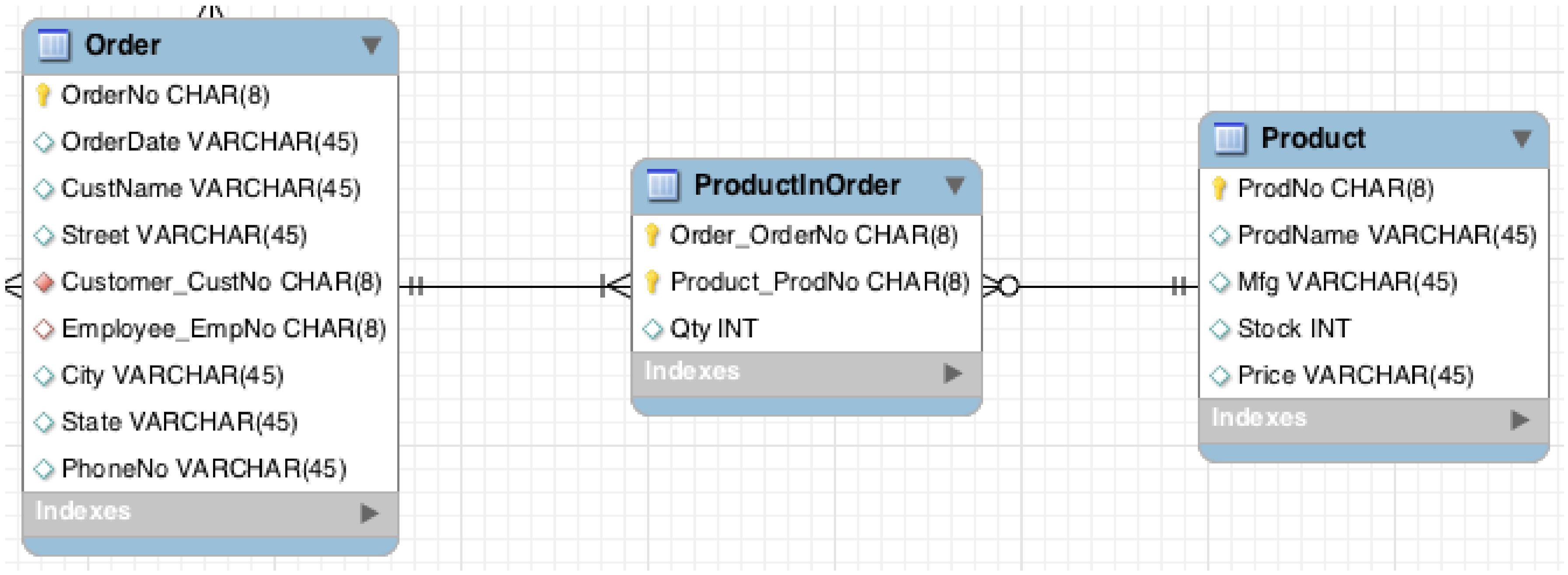


```
SELECT
    CustName,
    State,
    COUNT(*) AS 'No. of Orders'
FROM
    `Order`
GROUP BY State, CustName
HAVING COUNT(*)>1
ORDER BY CustName;
```

CustName	State	No. of Orders
Beth Taylor	WA	2
Man-Wai Mak	HK	2
Ron Thompson	WA	2

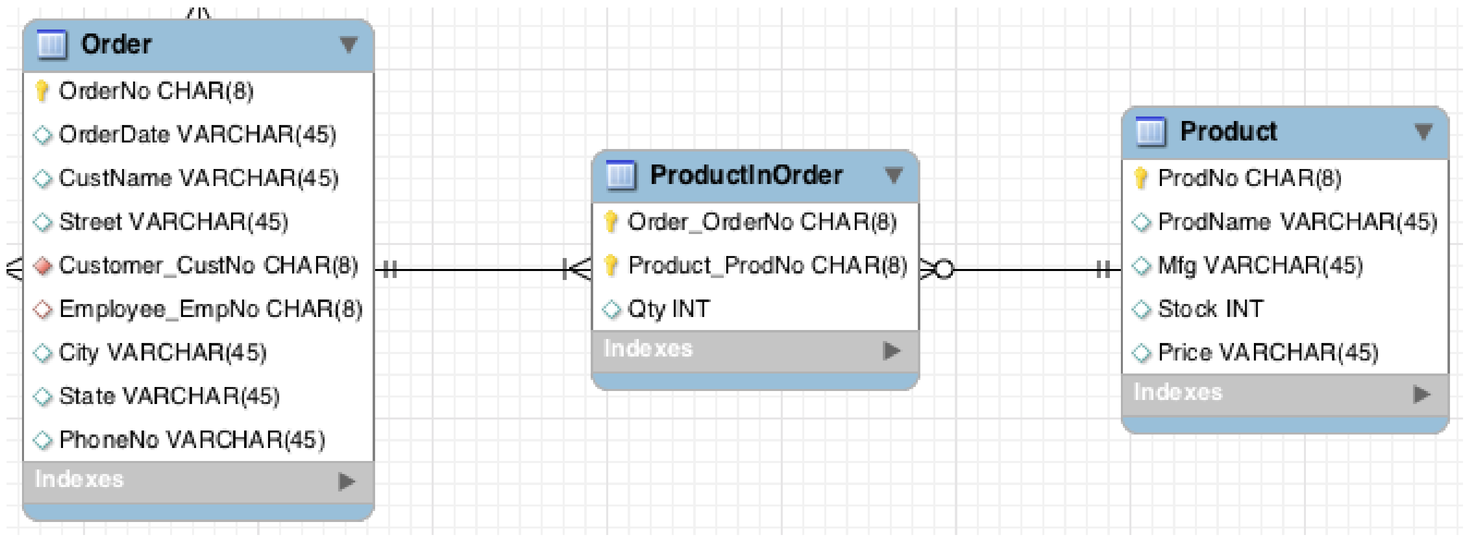
View Definition

- Define a view containing the details of orders placed in January 2007. Include all columns from Order, Qty from ProductInOrder and the ProdName in the view.



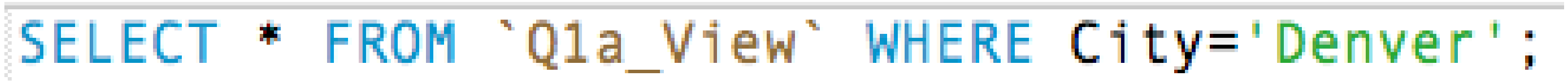
View Definition

```
CREATE VIEW `Q1a_View` AS
    SELECT
        `Order` . *, Qty, ProdName
    FROM
        `Order`,
        ProductInOrder,
        Product
    WHERE
        `Order`.OrderNo = ProductInOrder.Order_OrderNo
            AND ProductInOrder.Product_ProdNo = Product.ProdNo
            AND STR_TO_DATE(OrderDate, '%m/%d/%Y') BETWEEN '2007-1-1' AND '2007-1-31'
```

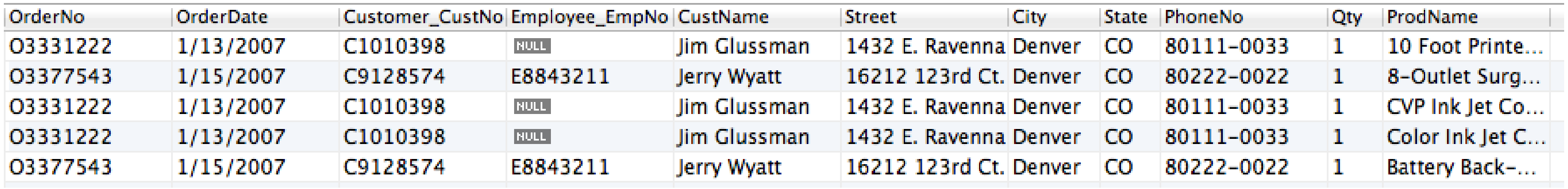


Retrieve from View

- List all rows in the view `Q1a_View` where the orders in Jan 2007 are from Denver.

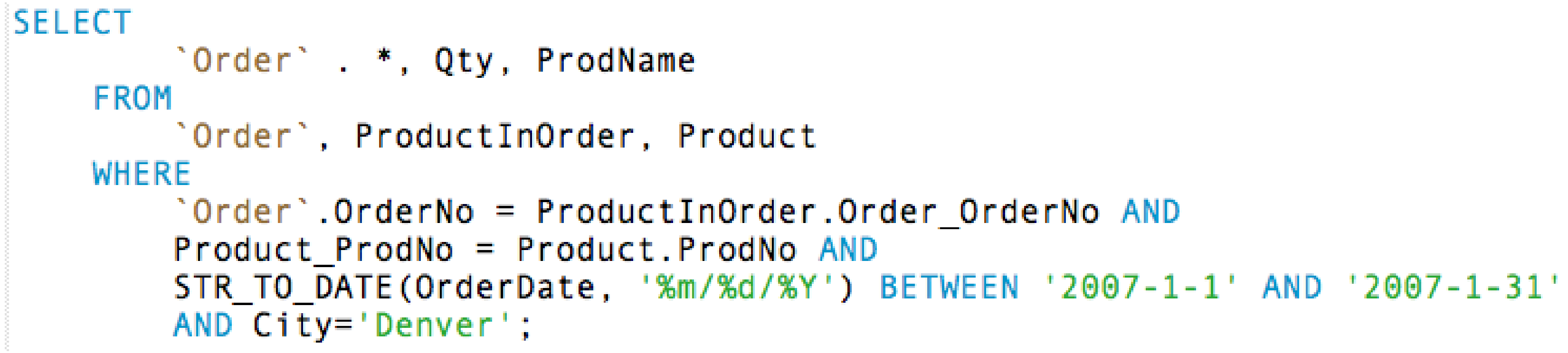


```
SELECT * FROM `Q1a_View` WHERE City='Denver';
```



OrderNo	OrderDate	Customer_CustNo	Employee_EmpNo	CustName	Street	City	State	PhoneNo	Qty	ProdName
O3331222	1/13/2007	C1010398	NULL	Jim Glussman	1432 E. Ravenna	Denver	CO	80111-0033	1	10 Foot Printe...
O3377543	1/15/2007	C9128574	E8843211	Jerry Wyatt	16212 123rd Ct.	Denver	CO	80222-0022	1	8-Outlet Surg...
O3331222	1/13/2007	C1010398	NULL	Jim Glussman	1432 E. Ravenna	Denver	CO	80111-0033	1	CVP Ink Jet Co...
O3331222	1/13/2007	C1010398	NULL	Jim Glussman	1432 E. Ravenna	Denver	CO	80111-0033	1	Color Ink Jet C...
O3377543	1/15/2007	C9128574	E8843211	Jerry Wyatt	16212 123rd Ct.	Denver	CO	80222-0022	1	Battery Back-...

- Same query on base tables.



```
SELECT
        `Order` . *, Qty, ProdName
    FROM
        `Order`, ProductInOrder, Product
    WHERE
        `Order`.OrderNo = ProductInOrder.Order_OrderNo AND
        Product_ProdNo = Product.ProdNo AND
        STR_TO_DATE(OrderDate, '%m/%d/%Y') BETWEEN '2007-1-1' AND '2007-1-31'
        AND City='Denver';
```