SE212 Database System and design

Advance SQL and Join query (PART 1)

Announcement

id, fname

team members

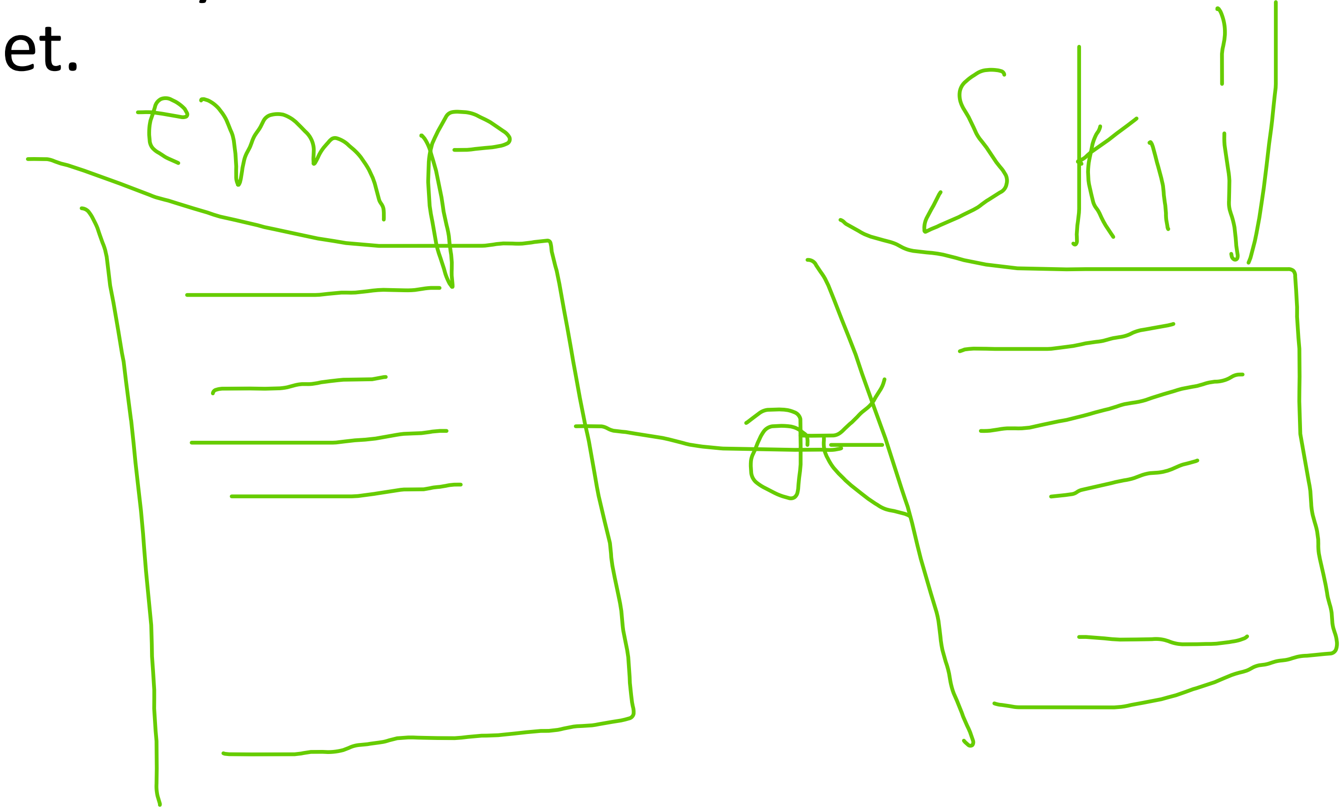
team name

- Term project.
  - Team of three people (max 3 people)
  - Introduction
  - Motivation (Why there is a need to DBMS in your business?) SME-MSME
  - Business rules
  - Derivable (a project report, ER, EER, Schemas, Mapping, FD, Normalized schemas, DDL, DML) , Gantchart
  - At least 10 samples for each relation(s)
- Final Exam 17th October 8am-11am

10

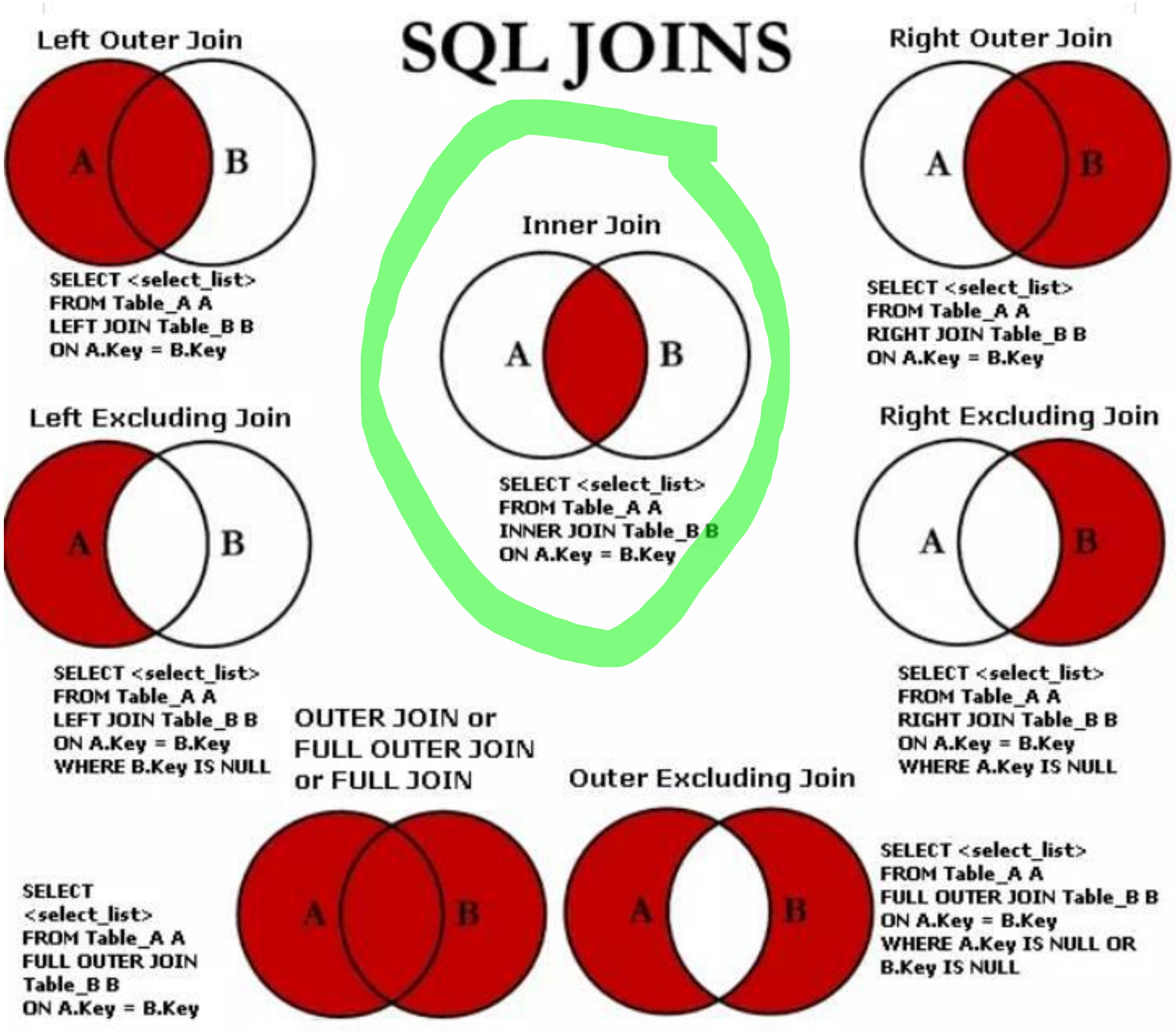
Join

- How to retrieve data from 2 or more table?
- A **Join** statement lets you combine data from two or more tables into a single result set.



Types of Join

1. **JOIN or INNER JOIN**
2. **OUTER JOIN**
   - ✓**LEFT OUTER JOIN or LEFT JOIN**
   - ✓**RIGHT OUTER JOIN or RIGHT JOIN**
   - ✓**FULL OUTER JOIN or FULL JOIN**
3. **NATURAL JOIN**



SQL JOINS

Left Outer Join

```
SELECT <select_list>
FROM Table_A A
LEFT JOIN Table_B B
ON A.Key = B.Key
```

Inner Join

```
SELECT <select_list>
FROM Table_A A
INNER JOIN Table_B B
ON A.Key = B.Key
```

Right Outer Join

```
SELECT <select_list>
FROM Table_A A
RIGHT JOIN Table_B B
ON A.Key = B.Key
```

Left Excluding Join

```
SELECT <select_list>
FROM Table_A A
LEFT JOIN Table_B B
ON A.Key = B.Key
WHERE B.Key IS NULL
```

Right Excluding Join

```
SELECT <select_list>
FROM Table_A A
RIGHT JOIN Table_B B
ON A.Key = B.Key
WHERE A.Key IS NULL
```

OUTER JOIN or FULL OUTER JOIN or FULL JOIN

```
SELECT
<select_list>
FROM Table_A A
FULL OUTER JOIN
Table_B B
ON A.Key = B.Key
```

Outer Excluding Join

```
SELECT <select_list>
FROM Table_A A
FULL OUTER JOIN Table_B B
ON A.Key = B.Key
WHERE A.Key IS NULL OR
B.Key IS NULL
```

Types of Join

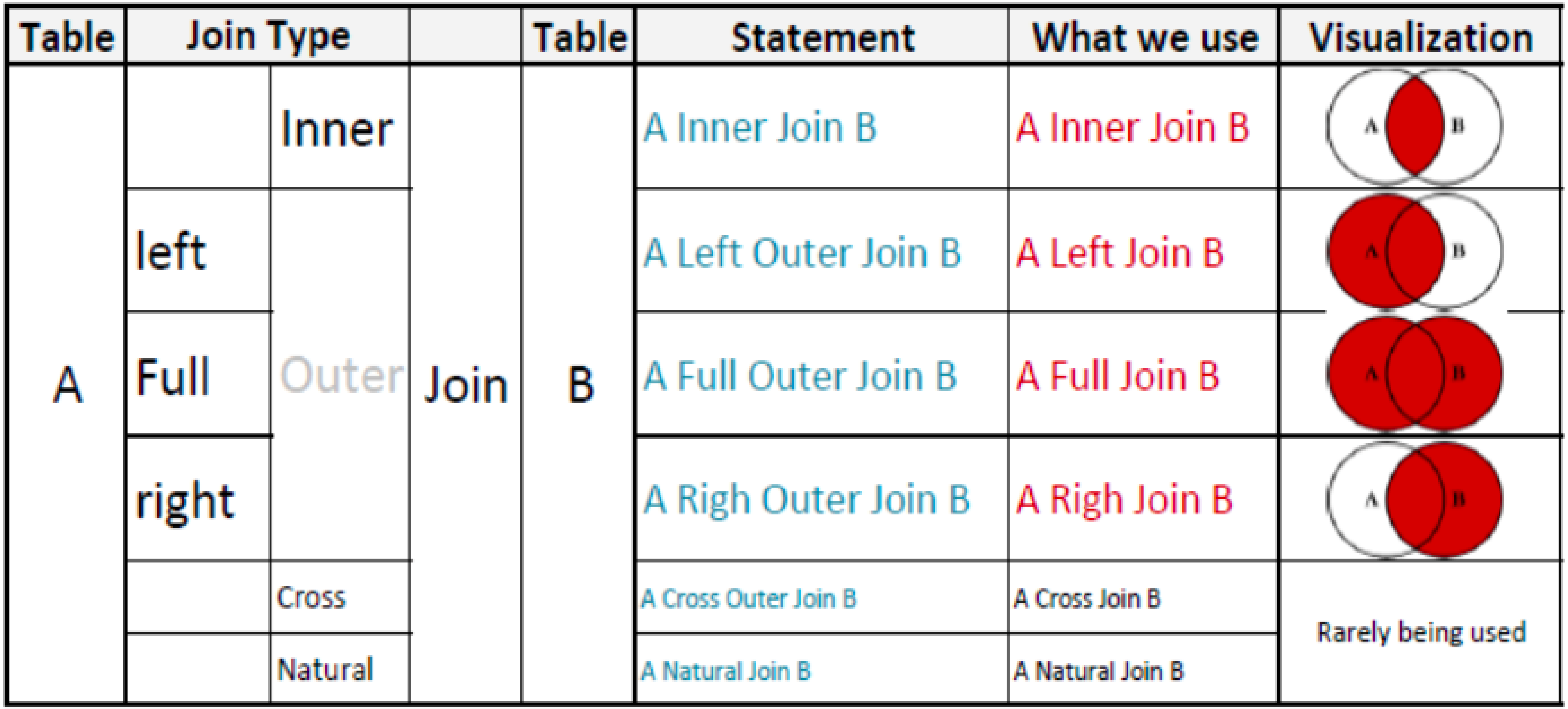
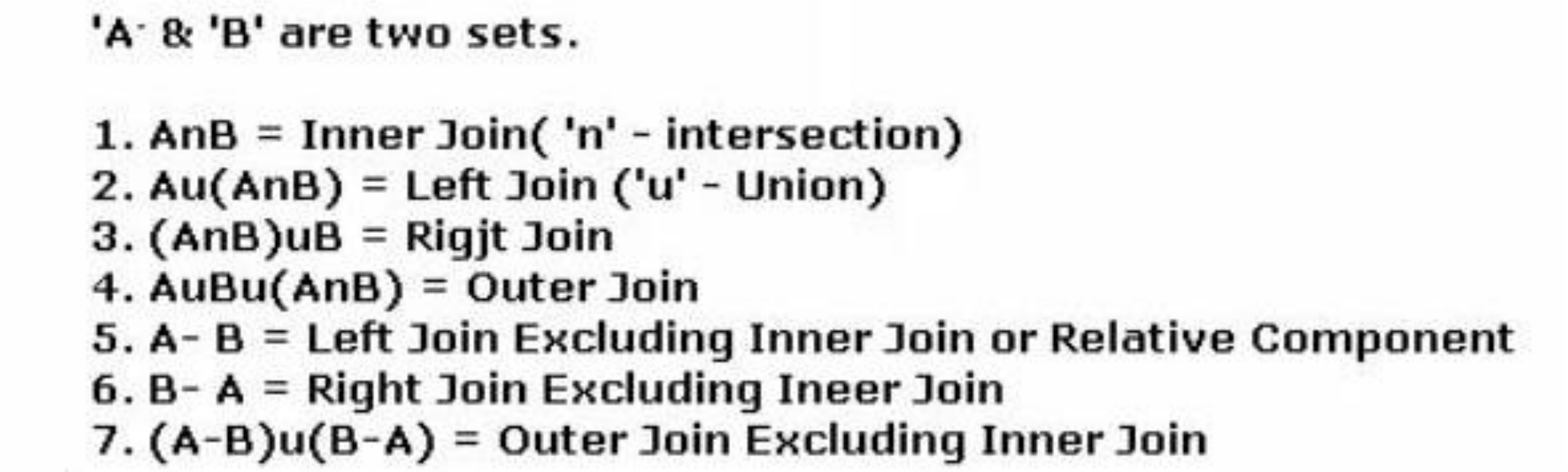


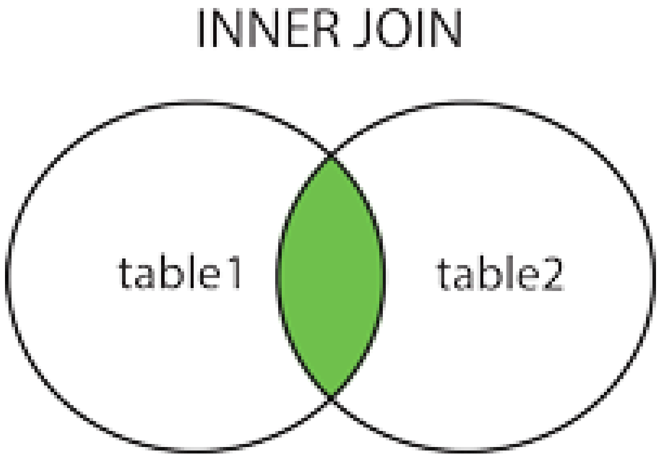
Table	Join Type			Table	Statement	What we use	Visualization
A		Inner	Join	B	A Inner Join B	A Inner Join B	
	left	Outer			A Left Outer Join B	A Left Join B	
	Full				A Full Outer Join B	A Full Join B	
	right				A Righ Outer Join B	A Righ Join B	
		Cross			A Cross Outer Join B	A Cross Join B	Rarely being used
		Natural			A Natural Join B	A Natural Join B	

JOIN



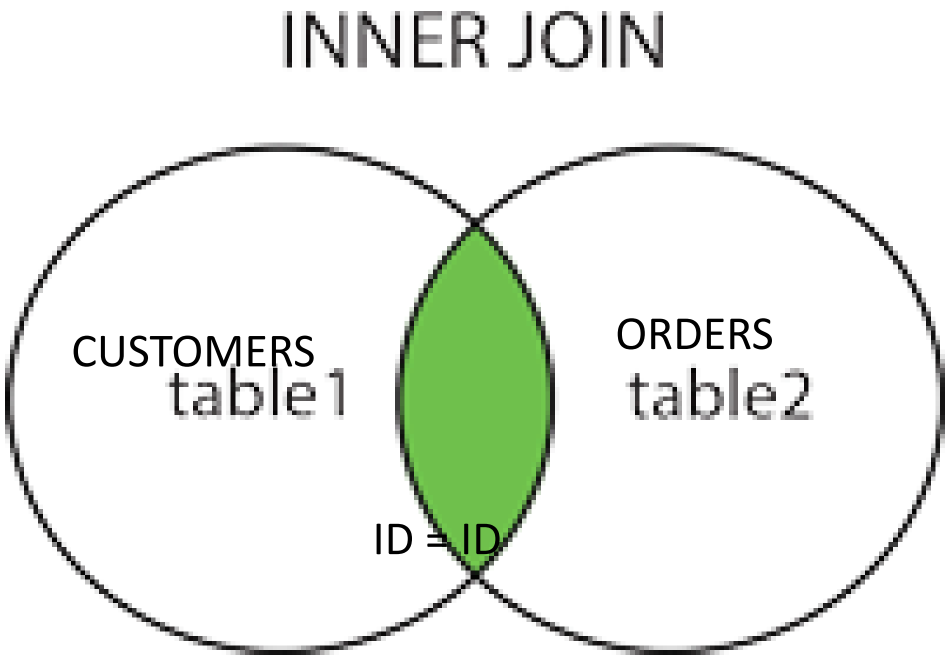
'A· & 'B' are two sets.

1. AnB = Inner Join('n' - intersection)
2. Au(AnB) = Left Join ('u' - Union)
3. (AnB)uB = Rigjt Join
4. AuBu(AnB) = Outer Join
5. A- B = Left Join Excluding Inner Join or Relative Component
6. B- A = Right Join Excluding Ineer Join
7. (A-B)u(B-A) = Outer Join Excluding Inner Join



1. JOIN or INNER JOIN

INNER JOIN clause



- The INNER JOIN clause matches rows in one table with rows in other tables and allows you to query rows that contain columns from both tables.

Inner join example

- CUSTOMERS Table
- +----+----------+-----+-----------+----------+
- | ID | NAME | AGE | ADDRESS | SALARY |
- +----+----------+-----+-----------+----------+
- | 1 | Ramesh | 32 | Ahmedabad | 2000.00 |
- | 2 | Khilan | 25 | Delhi | 1500.00 |
- | 3 | kaushik | 23 | Kota | 2000.00 |
- | 4 | Chaitali | 25 | Mumbai | 6500.00 |
- | 5 | Hardik | 27 | Bhopal | 8500.00 |
- | 6 | Komal | 22 | MP | 4500.00 |
- | 7 | Muffy | 24 | Indore | 10000.00 |
- +----+----------+-----+-----------+----------+

- ORDERS Table
- +-----+---------------------+-------------+--------+
- | OID | DATE | CUSTOMER_ID | AMOUNT |
- +-----+---------------------+-------------+--------+
- | 102 | 2009-10-08 00:00:00 | 3 | 3000 |
- | 100 | 2009-10-08 00:00:00 | 3 | 1500 |
- | 101 | 2009-11-20 00:00:00 | 2 | 1560 |
- | 103 | 2008-05-20 00:00:00 | 4 | 2060 |
- +-----+---------------------+-------------+--------+

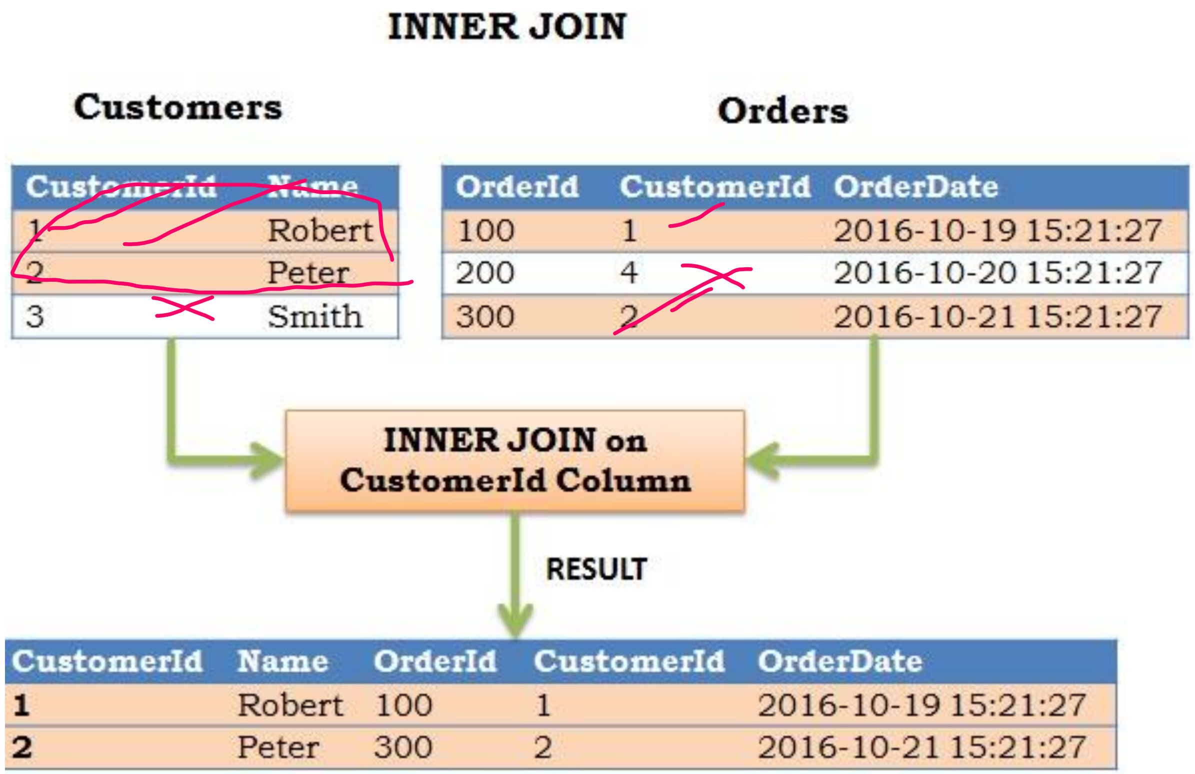
Inner join example

- *SQL>* SELECT ID, NAME, AMOUNT, DATE
- FROM CUSTOMERS
- JOIN ORDERS
- ON CUSTOMERS.ID = ORDERS.CUSTOMER_ID;

- Result

```
+----+----------+--------+---------------------+
| ID | NAME     | AMOUNT | DATE                |
+----+----------+--------+---------------------+
|  3 | kaushik  |   3000 | 2009-10-08 00:00:00 |
|  3 | kaushik  |   1500 | 2009-10-08 00:00:00 |
|  2 | Khilan   |   1560 | 2009-11-20 00:00:00 |
|  4 | Chaitali |   2060 | 2008-05-20 00:00:00 |
+----+----------+--------+---------------------+
```

Inner join example



INNER JOIN

Customers

CustomerId	Name
1	Robert
2	Peter
3	Smith

Orders

OrderId	CustomerId	OrderDate
100	1	2016-10-19 15:21:27
200	4	2016-10-20 15:21:27
300	2	2016-10-21 15:21:27

INNER JOIN on CustomerId Column

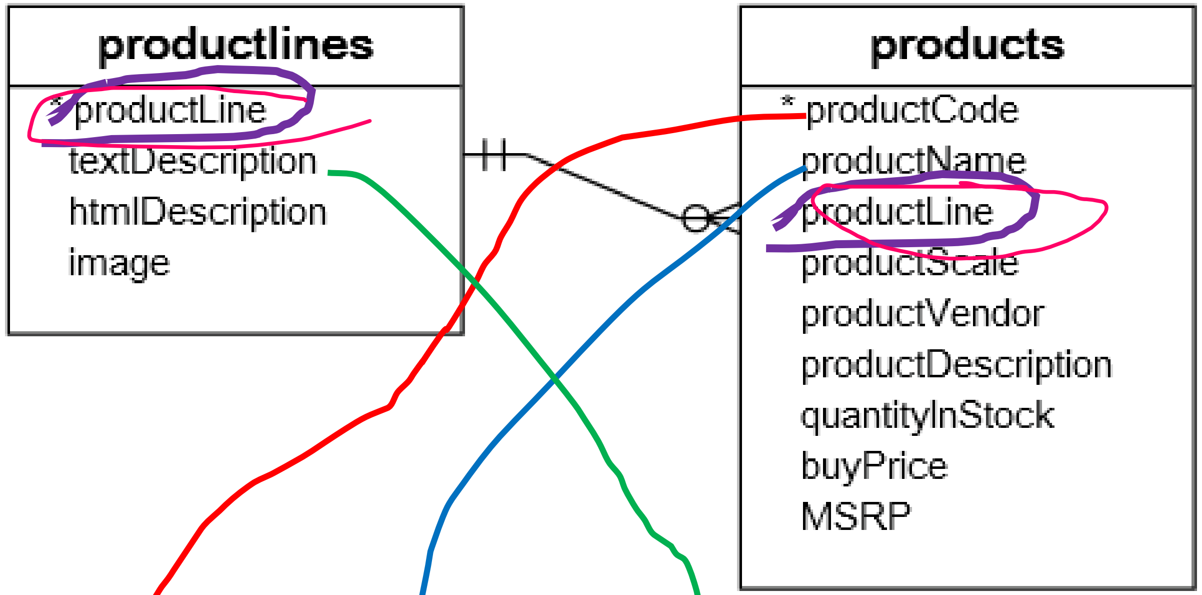
RESULT

CustomerId	Name	OrderId	CustomerId	OrderDate
1	Robert	100	1	2016-10-19 15:21:27
2	Peter	300	2	2016-10-21 15:21:27

INNER JOIN Clause

- First, specify the main table that appears in the FROM clause.
- Second, specify the table that you want to join with the main table, which appears in the INNER JOIN clause. ~~Theoretically, you can join a table with many tables. However, for better query performance, you should limit the number of tables to join.~~
- Third, **specify the join condition** or join predicate. The join condition appears after the keyword ON of ~~the INNER JOIN~~ clause. The join condition is the rule for matching rows between the main table and the other tables.

INNER JOIN



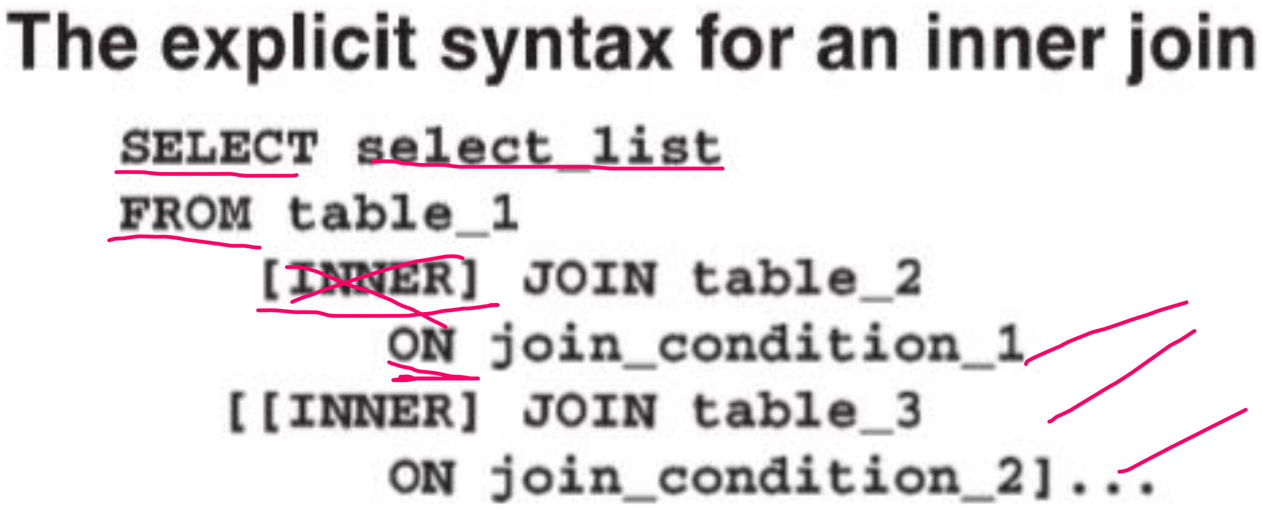
combine data from two tables

specify the join condition

```
SELECT productCode, productName, textDescription
    FROM products T1
    INNER JOIN productlines T2 ON T1.productline =
T2.productline;
```

How to work with inner joins

- Format:



The explicit syntax for an inner join

```
SELECT select_list
FROM table_1
    [INNER] JOIN table_2
        ON join_condition_1
   [[INNER] JOIN table_3
        ON join_condition_2]...
```

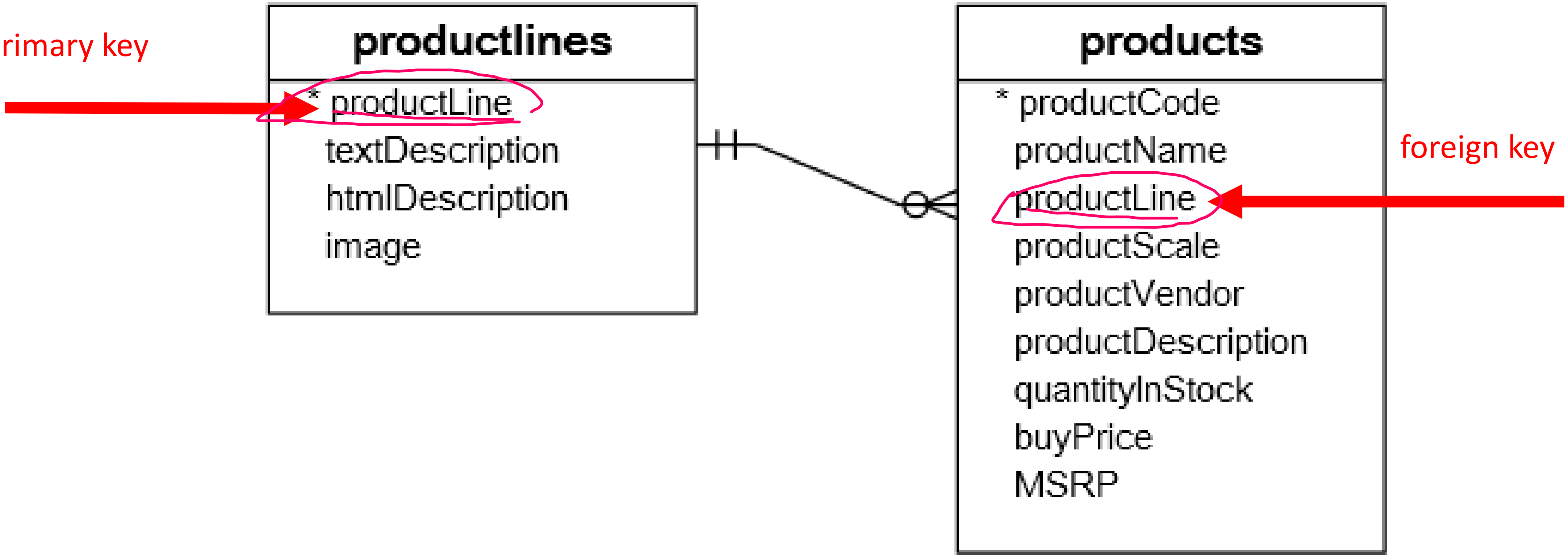
An inner join

- A join condition names a column in each of the two tables involved in the join and indicates how the two columns should be compared
- Tables are typically joined on the relationship between the primary key in one table and a foreign key in the other table
- If the two columns in a join condition have the same name, you must qualify them with the table name so MySQL can distinguish between them

An inner join

You have two tables to retrieve data: **productlines and products**

Which column can be used to join/compare these tables?



primary key

productlines

* productLine
textDescription
htmlDescription
image

products

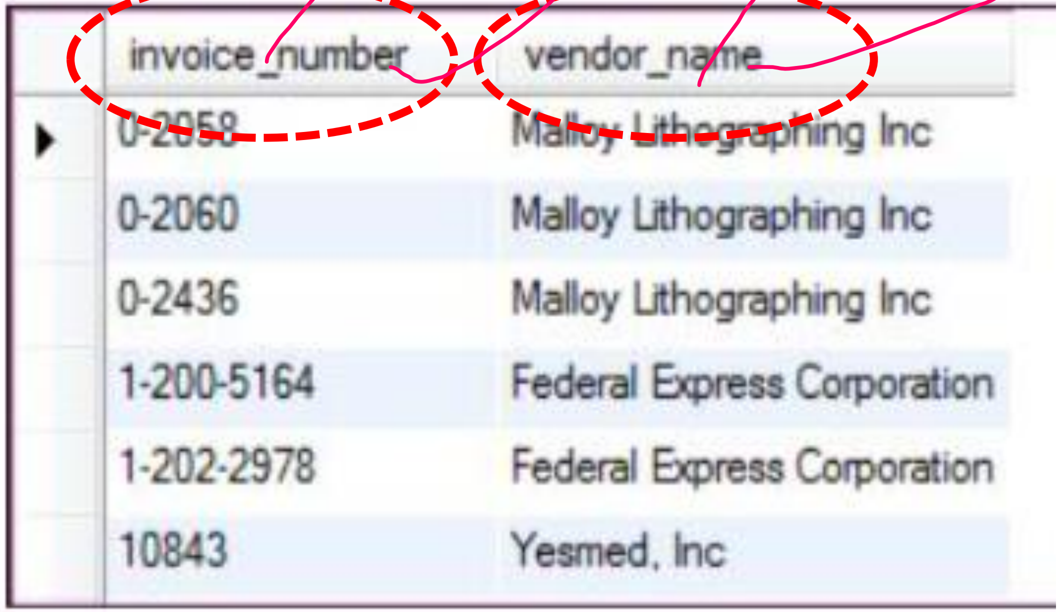
* productCode
productName
productLine
productScale
productVendor
productDescription
quantityInStock
buyPrice
MSRP

foreign key

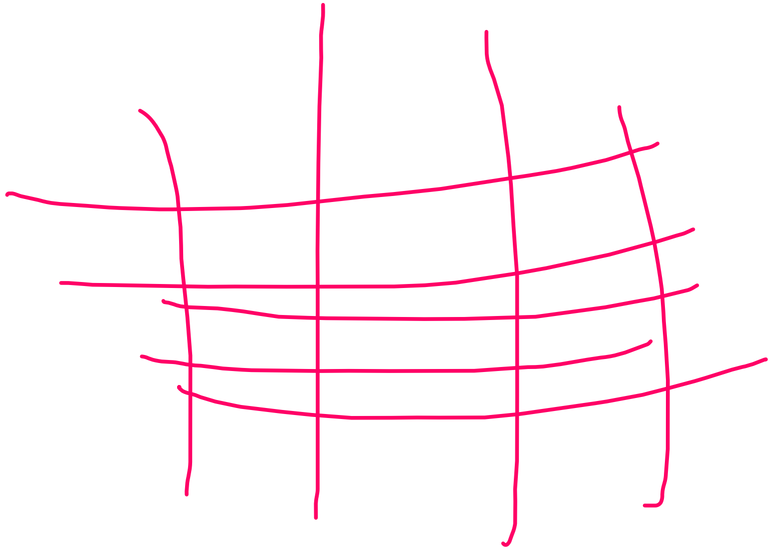
How many columns will be shown in the result?

An inner join of the Vendors and Invoices tables

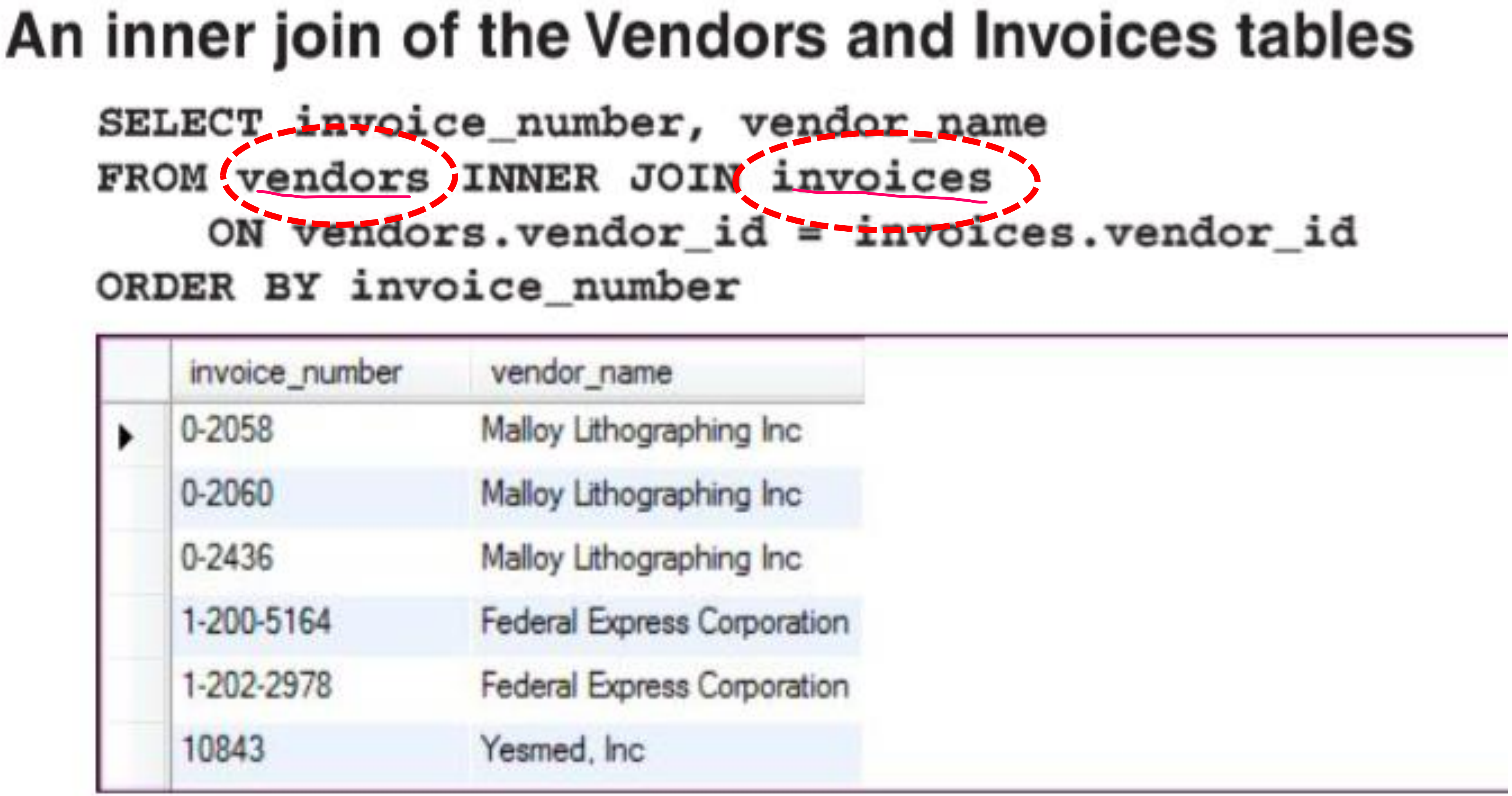
```
SELECT invoice_number, vendor_name
FROM vendors INNER JOIN invoices
    ON vendors.vendor_id = invoices.vendor_id
ORDER BY invoice_number
```



invoice_number	vendor_name
0-2058	Malloy Lithographing Inc
0-2060	Malloy Lithographing Inc
0-2436	Malloy Lithographing Inc
1-200-5164	Federal Express Corporation
1-202-2978	Federal Express Corporation
10843	Yesmed, Inc



How many tables were joined?



An inner join of the Vendors and Invoices tables

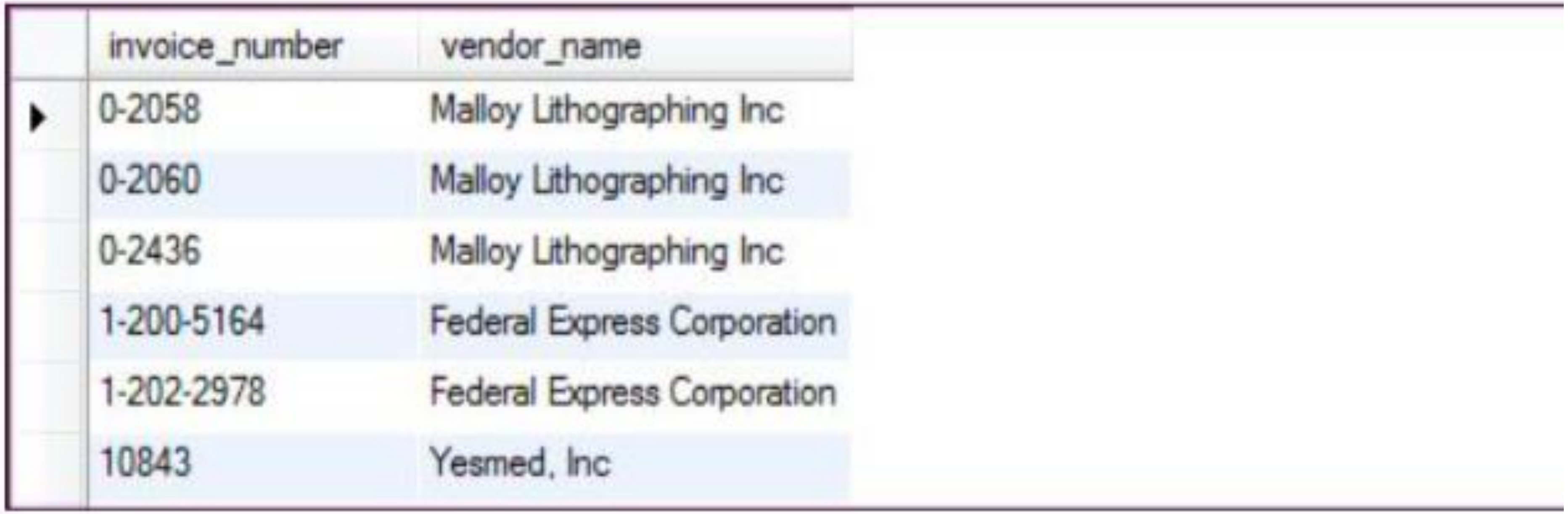
```
SELECT invoice_number, vendor_name
FROM vendors INNER JOIN invoices
    ON vendors.vendor_id = invoices.vendor_id
ORDER BY invoice_number
```

invoice_number	vendor_name
0-2058	Malloy Lithographing Inc
0-2060	Malloy Lithographing Inc
0-2436	Malloy Lithographing Inc
1-200-5164	Federal Express Corporation
1-202-2978	Federal Express Corporation
10843	Yesmed, Inc

Which column was joined?

An inner join of the Vendors and Invoices tables

```
SELECT invoice_number, vendor_name
FROM vendors INNER JOIN invoices
    ON vendors.vendor_id = invoices.vendor_id
ORDER BY invoice_number
```



invoice_number	vendor_name
0-2058	Malloy Lithographing Inc
0-2060	Malloy Lithographing Inc
0-2436	Malloy Lithographing Inc
1-200-5164	Federal Express Corporation
1-202-2978	Federal Express Corporation
10843	Yesmed, Inc

Problem during implementation?

- We type a long SQL statement to retrieve data from many tables.
- What mistakes are from typing?

- What should we do to solve problems?

Aliases

- A table alias is an alternative/temporary name.
- Usually, use a letter or two letter
- Alias makes the code and reading easier, especially if the table names are long

Aliases (cont.)

- SELECT *column_name* AS *alias_name*
  FROM *table_name;* ✓

- SELECT *column_name(s)*
  FROM *table_name* AS *alias_name;* 2) ✓

Aliases example

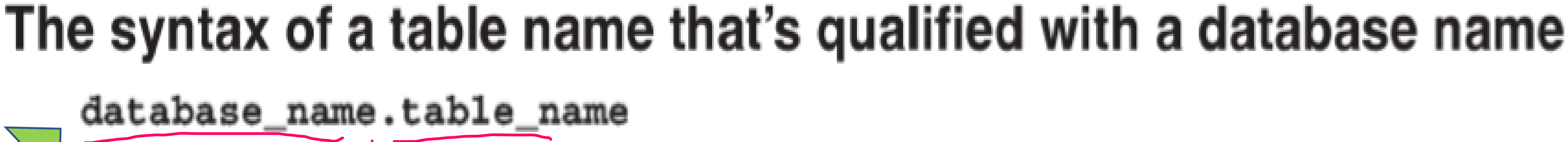
```
SELECT o.OrderID, o.OrderDate, c.CustomerName
FROM Customers AS c, Orders AS o
WHERE c.CustomerName="Around the Horn" AND c.CustomerID =
o.CustomerID;
```

(Note, you may ignore AS)

How to join tables from different databases?

- How does computer know which database you want to connect to?
- How does computer know which tables you want to retrieve data from?

You should reference the database name



The syntax of a table name that's qualified with a database name

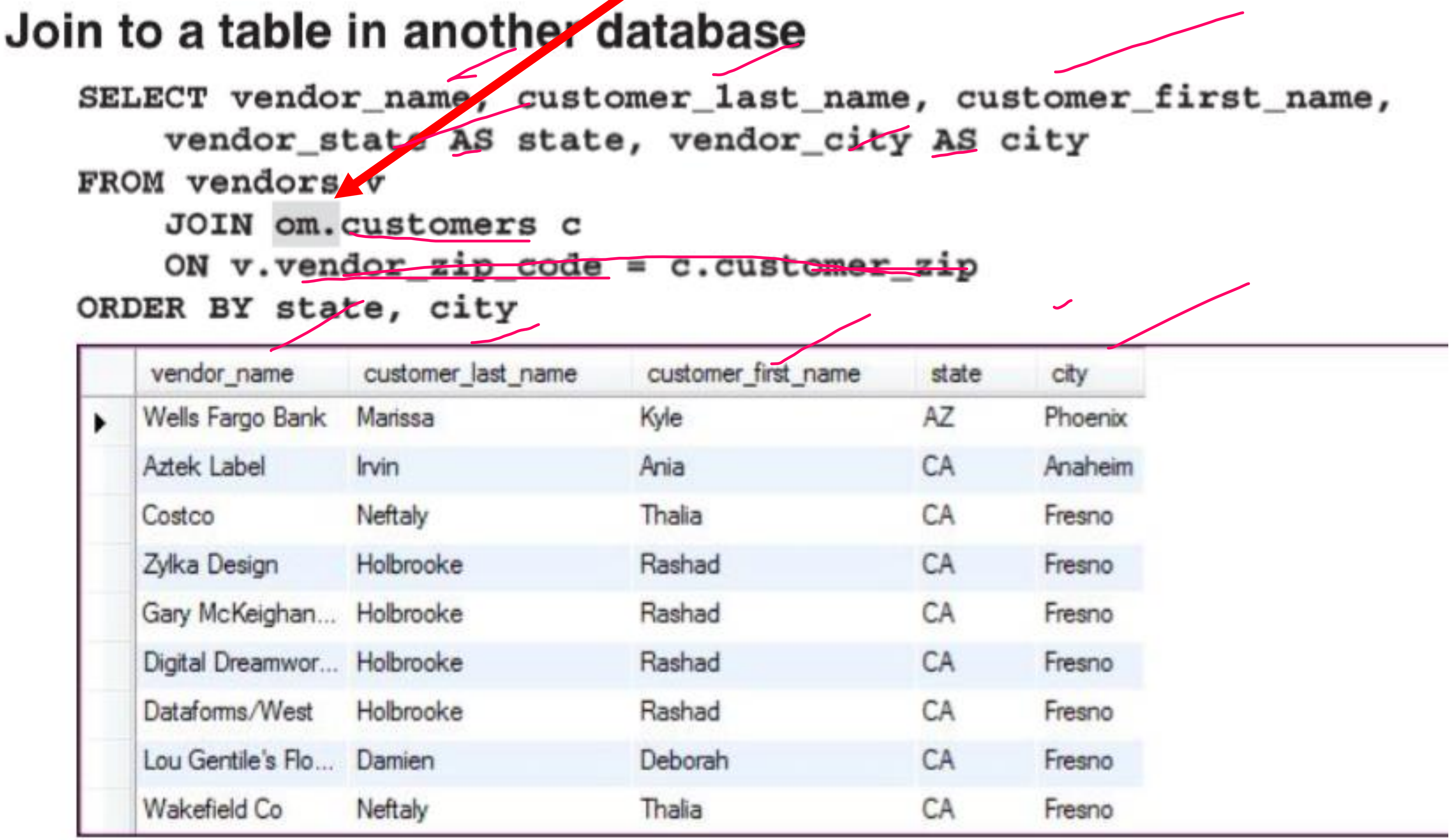
```
database_name.table_name
```

Database

- A database server can store tables in multiple databases
- Run a SELECT statement against one database, you can join a table in another ~~database if you have approp~~riate permissions
- Use prefix the table name in the other database with the name of that database

Example:

om is the database name



Join to a table in another database

```
SELECT vendor_name, customer_last_name, customer_first_name,
    vendor_state AS state, vendor_city AS city
FROM vendors v
    JOIN om.customers c
    ON v.vendor_zip_code = c.customer_zip
ORDER BY state, city
```

vendor_name	customer_last_name	customer_first_name	state	city
Wells Fargo Bank	Marissa	Kyle	AZ	Phoenix
Aztek Label	Irvin	Ania	CA	Anaheim
Costco	Neftaly	Thalia	CA	Fresno
Zylka Design	Holbrooke	Rashad	CA	Fresno
Gary McKeighan...	Holbrooke	Rashad	CA	Fresno
Digital Dreamwor...	Holbrooke	Rashad	CA	Fresno
Dataforms/West	Holbrooke	Rashad	CA	Fresno
Lou Gentile's Flo...	Damien	Deborah	CA	Fresno
Wakefield Co	Neftaly	Thalia	CA	Fresno

If you have more conditions to compare, what should you do?

Forget about it!

or

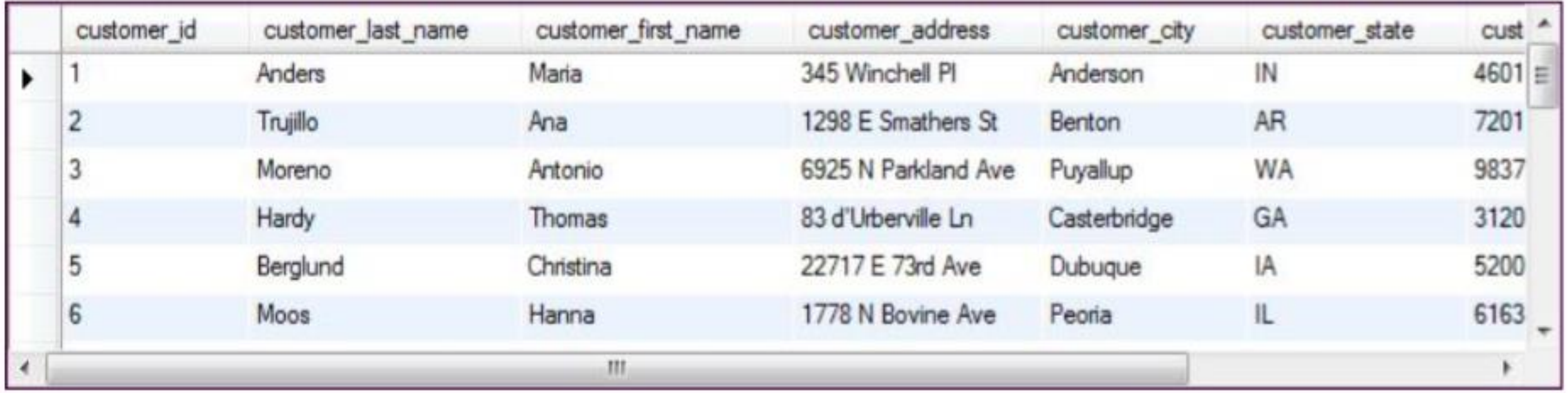
Find a statement to retrieve the data!

How to use compound join conditions

- Join conditions usually consists of a single comparison
- You can use ‘AND’ and ‘OR’ to include two or more comparisons in a join condition

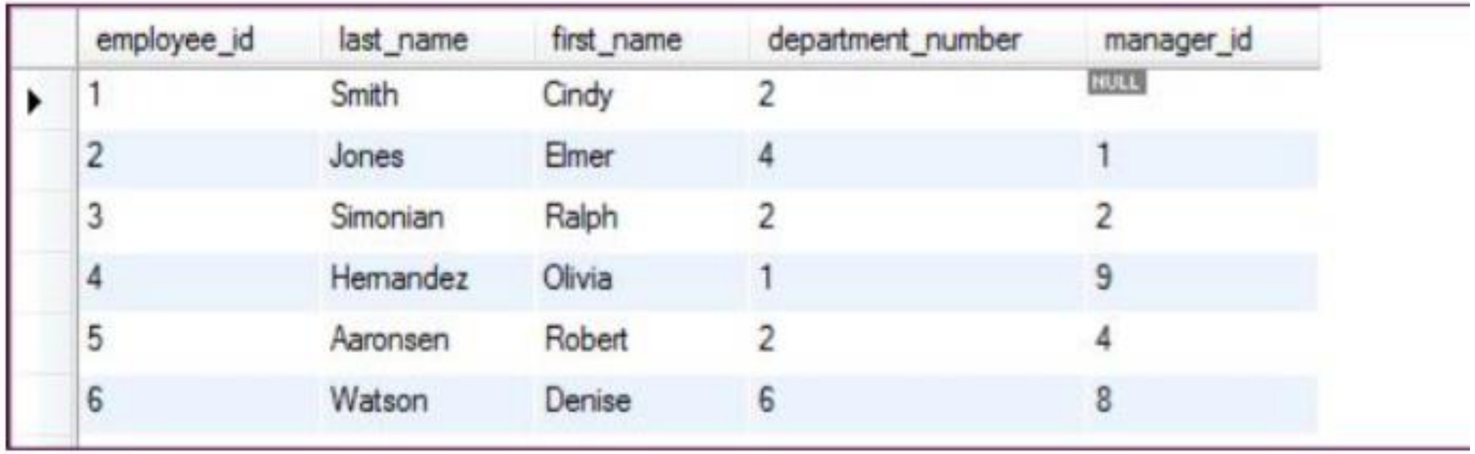
How to use compound join conditions

The Customers table

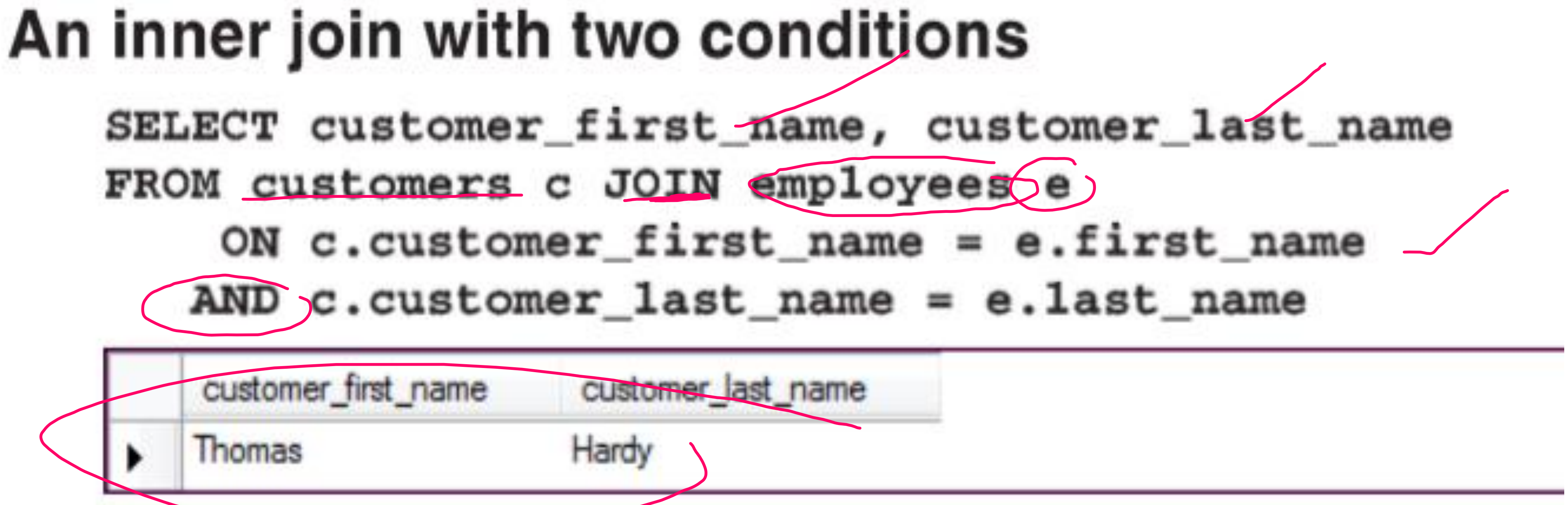


customer_id	customer_last_name	customer_first_name	customer_address	customer_city	customer_state	cust
1	Anders	Maria	345 Winchell Pl	Anderson	IN	4601
2	Trujillo	Ana	1298 E Smathers St	Benton	AR	7201
3	Moreno	Antonio	6925 N Parkland Ave	Puyallup	WA	9837
4	Hardy	Thomas	83 d'Urberville Ln	Casterbridge	GA	3120
5	Berglund	Christina	22717 E 73rd Ave	Dubuque	IA	5200
6	Moos	Hanna	1778 N Bovine Ave	Peoria	IL	6163

The Employees table



employee_id	last_name	first_name	department_number	manager_id
1	Smith	Cindy	2	NULL
2	Jones	Elmer	4	1
3	Simonian	Ralph	2	2
4	Hernandez	Olivia	1	9
5	Aaronsen	Robert	2	4
6	Watson	Denise	6	8

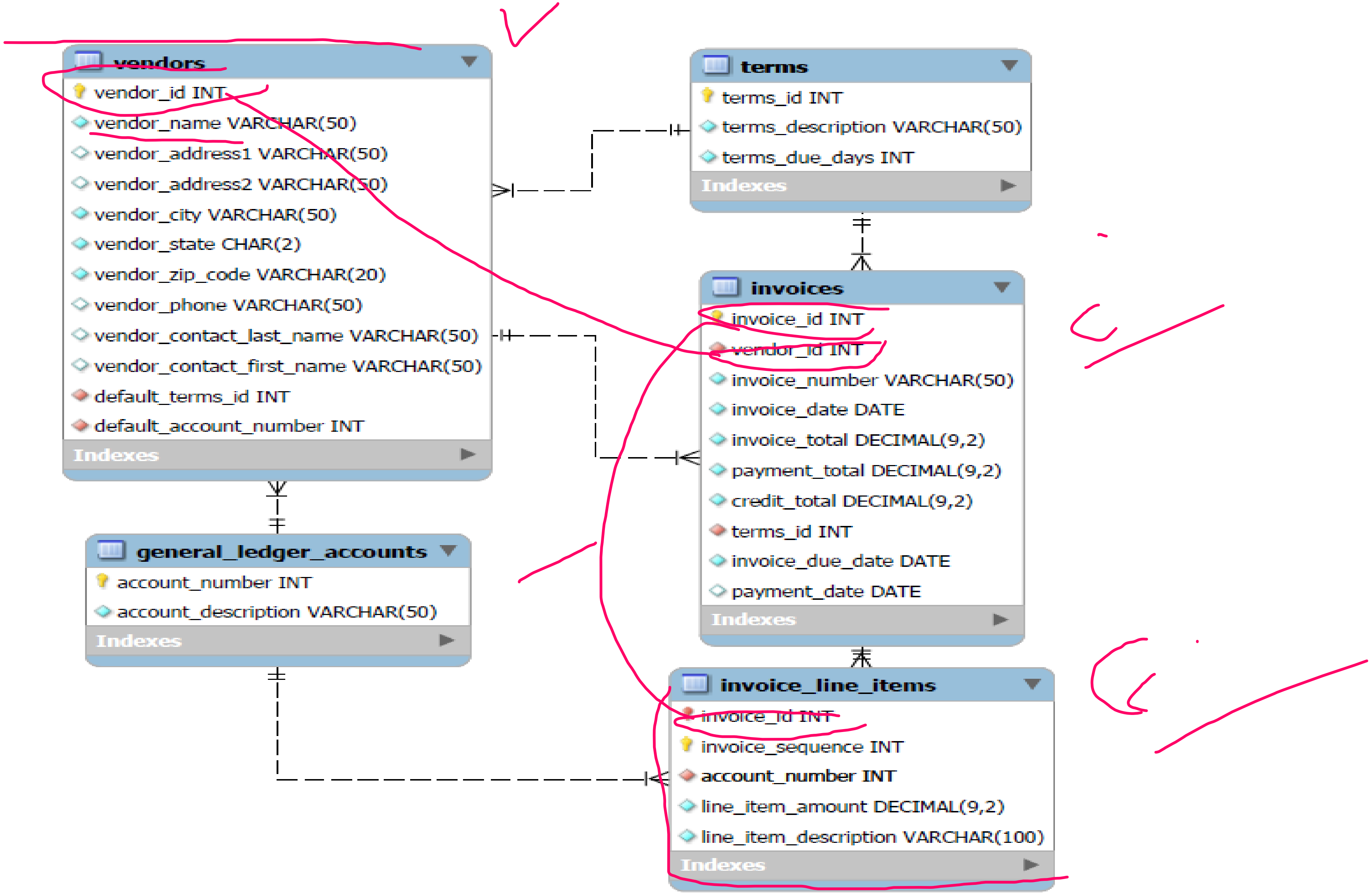


An inner join with two conditions

```
SELECT customer_first_name, customer_last_name
FROM customers c JOIN employees e
    ON c.customer_first_name = e.first_name
   AND c.customer_last_name = e.last_name
```

customer_first_name	customer_last_name
Thomas	Hardy

- A join condition can include two or more conditions by AND or OR operators
- Thomas Hardy is employee and customer at the same time.



vendors
vendor_id INT
vendor_name VARCHAR(50)
vendor_address1 VARCHAR(50)
vendor_address2 VARCHAR(50)
vendor_city VARCHAR(50)
vendor_state CHAR(2)
vendor_zip_code VARCHAR(20)
vendor_phone VARCHAR(50)
vendor_contact_last_name VARCHAR(50)
vendor_contact_first_name VARCHAR(50)
default_terms_id INT
default_account_number INT
Indexes

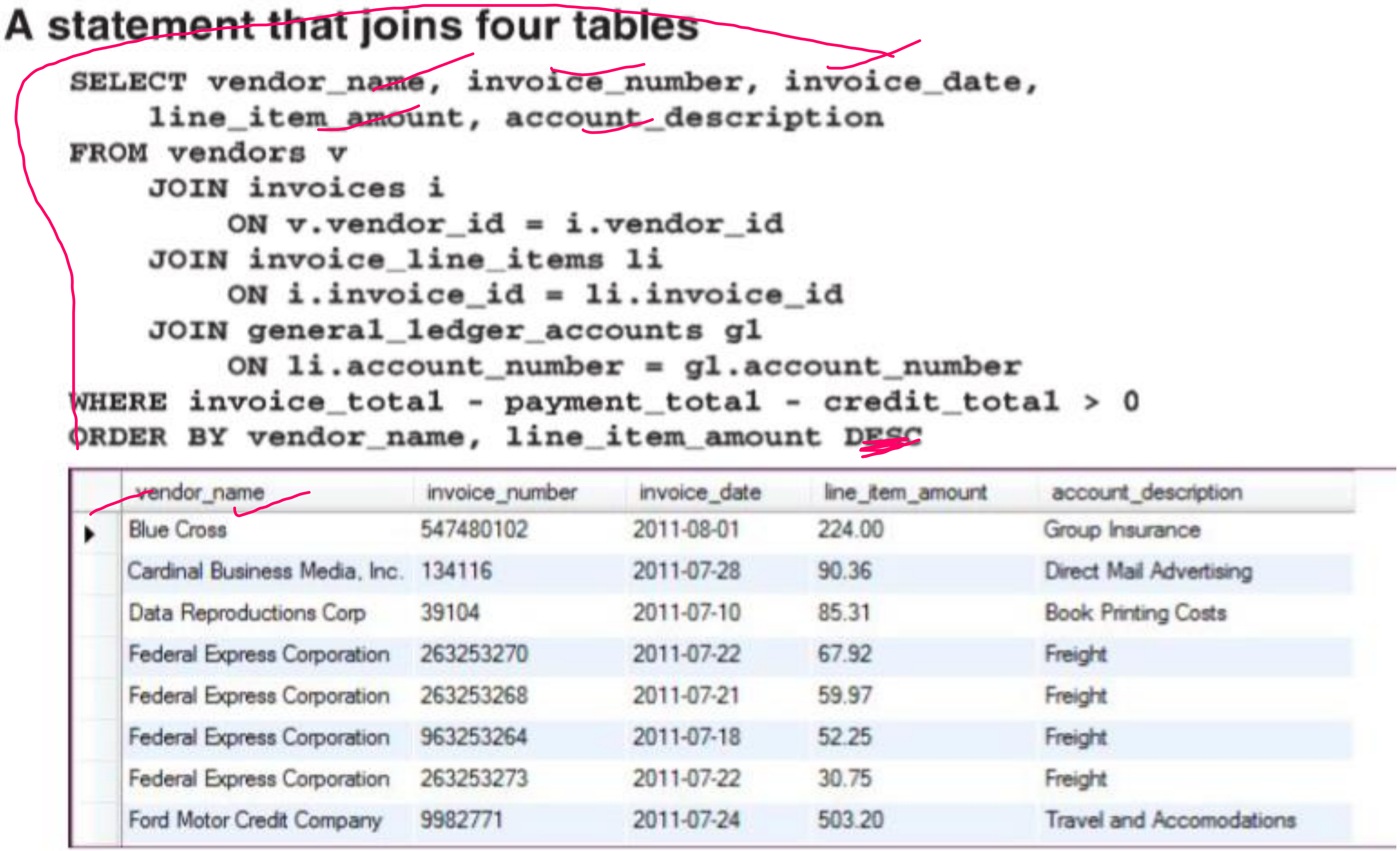
terms
terms_id INT
terms_description VARCHAR(50)
terms_due_days INT
Indexes

invoices
invoice_id INT
vendor_id INT
invoice_number VARCHAR(50)
invoice_date DATE
invoice_total DECIMAL(9,2)
payment_total DECIMAL(9,2)
credit_total DECIMAL(9,2)
terms_id INT
invoice_due_date DATE
payment_date DATE
Indexes

general_ledger_accounts
account_number INT
account_description VARCHAR(50)
Indexes

invoice_line_items
invoice_id INT
invoice_sequence INT
account_number INT
line_item_amount DECIMAL(9,2)
line_item_description VARCHAR(100)
Indexes

How to join more than two tables



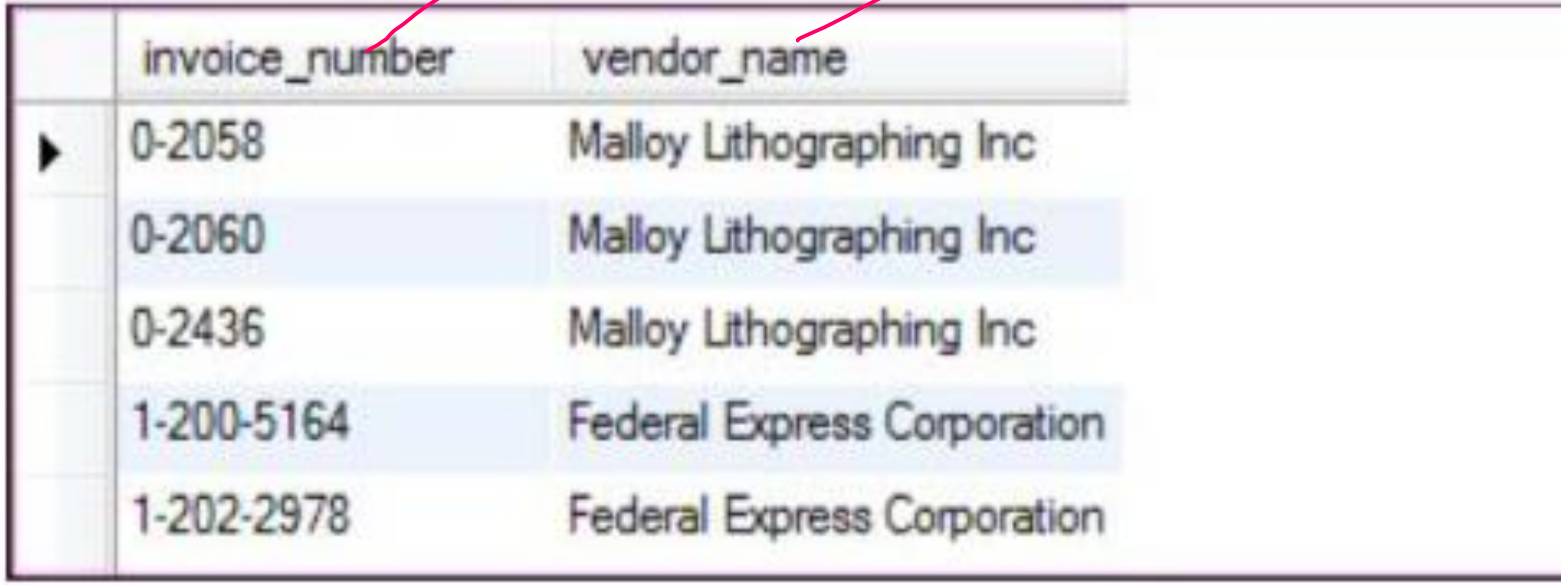
A statement that joins four tables

```
SELECT vendor_name, invoice_number, invoice_date,
    line_item_amount, account_description
FROM vendors v
    JOIN invoices i
        ON v.vendor_id = i.vendor_id
    JOIN invoice_line_items li
        ON i.invoice_id = li.invoice_id
    JOIN general_ledger_accounts gl
        ON li.account_number = gl.account_number
WHERE invoice_total - payment_total - credit_total > 0
ORDER BY vendor_name, line_item_amount DESC
```

vendor_name	invoice_number	invoice_date	line_item_amount	account_description
Blue Cross	547480102	2011-08-01	224.00	Group Insurance
Cardinal Business Media, Inc.	134116	2011-07-28	90.36	Direct Mail Advertising
Data Reproductions Corp	39104	2011-07-10	85.31	Book Printing Costs
Federal Express Corporation	263253270	2011-07-22	67.92	Freight
Federal Express Corporation	263253268	2011-07-21	59.97	Freight
Federal Express Corporation	963253264	2011-07-18	52.25	Freight
Federal Express Corporation	263253273	2011-07-22	30.75	Freight
Ford Motor Credit Company	9982771	2011-07-24	503.20	Travel and Accomodations

Join the Vendors and Invoices tables

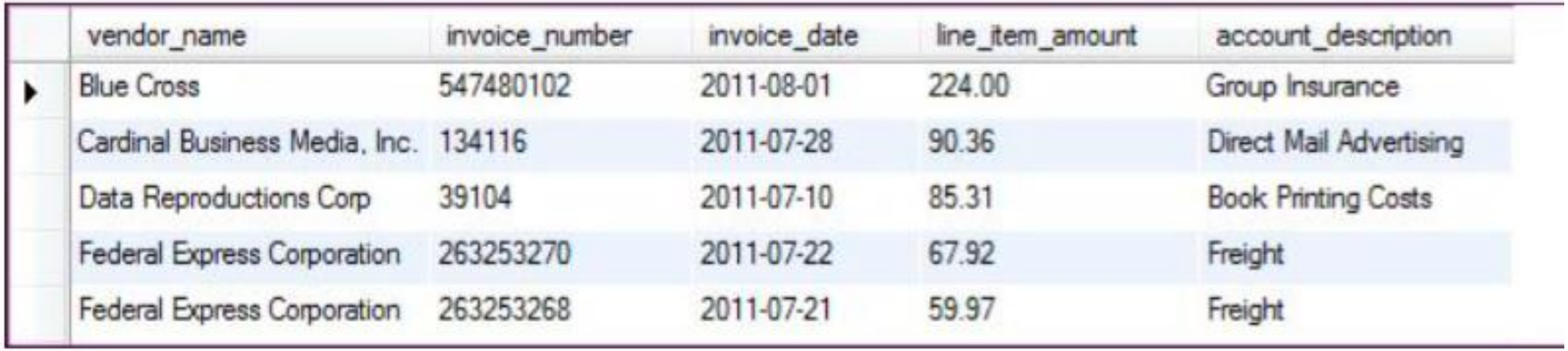
```
SELECT invoice_number, vendor_name
FROM vendors v, invoices i
WHERE v.vendor_id = i.vendor_id
ORDER BY invoice_number
```



invoice_number	vendor_name
0-2058	Malloy Lithographing Inc
0-2060	Malloy Lithographing Inc
0-2436	Malloy Lithographing Inc
1-200-5164	Federal Express Corporation
1-202-2978	Federal Express Corporation

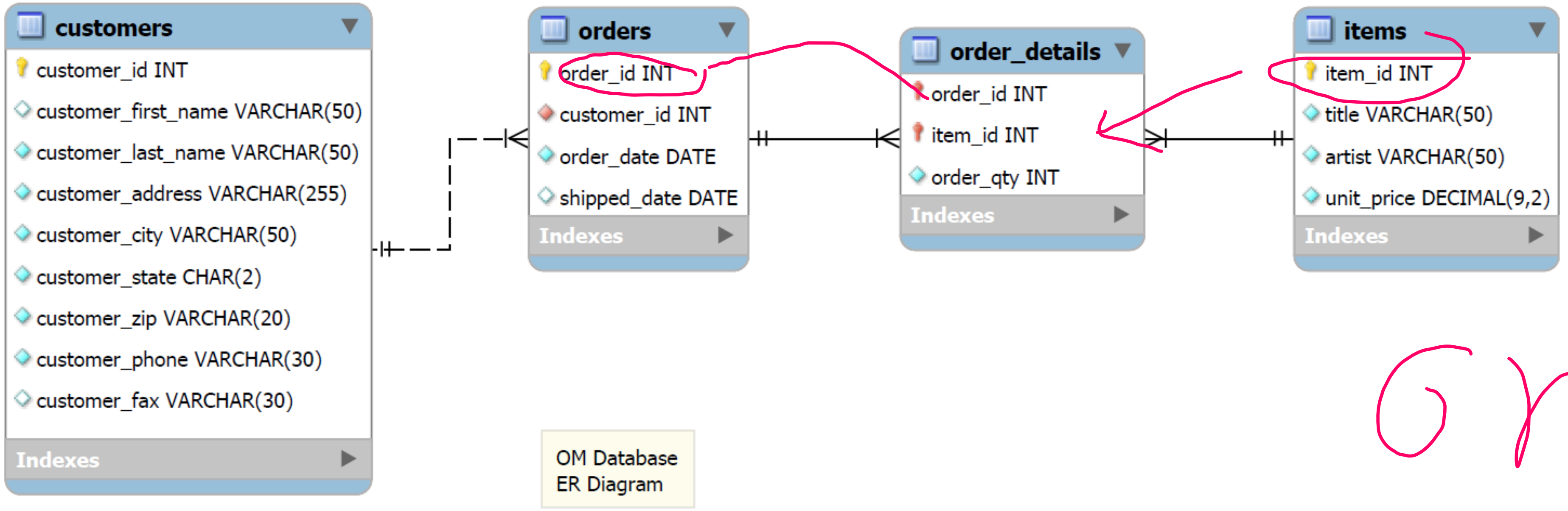
Join four tables

```
SELECT vendor_name, invoice_number, invoice_date,
    line_item_amount, account_description
FROM  vendors v, invoices i, invoice_line_items li,
    general_ledger_accounts gl
WHERE v.vendor_id = i.vendor_id
  AND i.invoice_id = li.invoice_id
  AND li.account_number = gl.account_number
  AND invoice_total - payment_total - credit_total > 0
ORDER BY vendor_name, line_item_amount DESC
```



vendor_name	invoice_number	invoice_date	line_item_amount	account_description
Blue Cross	547480102	2011-08-01	224.00	Group Insurance
Cardinal Business Media, Inc.	134116	2011-07-28	90.36	Direct Mail Advertising
Data Reproductions Corp	39104	2011-07-10	85.31	Book Printing Costs
Federal Express Corporation	263253270	2011-07-22	67.92	Freight
Federal Express Corporation	263253268	2011-07-21	59.97	Freight

ERD



customers
- customer_id INT
- customer_first_name VARCHAR(50)
- customer_last_name VARCHAR(50)
- customer_address VARCHAR(255)
- customer_city VARCHAR(50)
- customer_state CHAR(2)
- customer_zip VARCHAR(20)
- customer_phone VARCHAR(30)
- customer_fax VARCHAR(30)

Indexes

orders
- order_id INT
- customer_id INT
- order_date DATE
- shipped_date DATE

Indexes

order_details
- order_id INT
- item_id INT
- order_qty INT

Indexes

items
- item_id INT
- title VARCHAR(50)
- artist VARCHAR(50)
- unit_price DECIMAL(9,2)

Indexes

OM Database
ER Diagram

Om

https://cmu.to/u-y1J

Workshop 1(10 mins)

- Open MySQL Workbench
- Create schema and executes the previous given SQL scripts.
- Use **inner join / join** to list first name, last name of customer and order date in *descending* order.

Workshop 2

(15-20 min)

1. Use ‘**ap**’ database perform an inner join of invoice number and vender name order by invoice number.
2. Now, learn compound join and join more than 2 tables from the previous script. Your goal is to produce the same results as previous slides(31-34)