Advanced SQL

Profile

Mathematics

Data Scientist –
Stream Intelligence

Senior Data Analyst –
Tokopedia

Farhan Reza Gumay

DigitalSkola
Uncover The World Of Digital Skills With Us

Quote of The Day

You can have data without information, but you cannot have information without data.

- Daniel Keys Moran

Table of Content

What will We Learn Today?

1. Date Function
2. Join Tables
3. Sub Query

Date Functions

Date Functions

Handling Date in Database sometimes need special care, especially when we are multi-national company, have overseas market or working together with partners. Different timezone, time granularity, and updated time are the most common problem. That is why we must have a standardized date recording.

But when it comes to analyze the data, we also need capability to transform date to be something that we need. That is why date functions are important for database.

Getting Current Time

Get Datetime (2021-01-21 01:26:52)

now()

current_timestamp

Get Date (2021-01-21)

current_date

Get Time (01:26:52)

current_time

Changing Timezone

Date usually is recorded at UTC timezone (+00:00). By using standardized date timezone, we can match any date from any source. But when it comes to analytics need, sometimes change it again to our local time would help.

```
SELECT   created_at   at time zone   'Asia/Jakarta’
FROM   transaction
```

References for timezones:

https://popsql.com/learn-sql/postgresql/how-to-convert-utc-to-local-time-zone-in-

Making Date & Time

Make Date

make_date(year int, month int, day int)

make_date(2013, 7, 15)

Make Time

make_time(hour int, min int, sec double precision)

make_time(8, 15, 23.5)

Make Timestamp

make_timestamp(year int, month int, day int, hour int, min int, sec double precision)

Extract Part of Date

Extract Part of Date

```
date_part(text, timestamp)

        date_part('hour', timestamp  '2001-02-16 20:38:40')
```

Date Transformation

Transform Date to Character or String

```
to_char('string', 'format’)

        to_char('2017-03-31', 'DD/MM/YYYY’)
```

Transform String to Date

```
to_date('string', 'format’)

        to_date('20170103','YYYYMMDD’)
```

Transform String to Timestamp

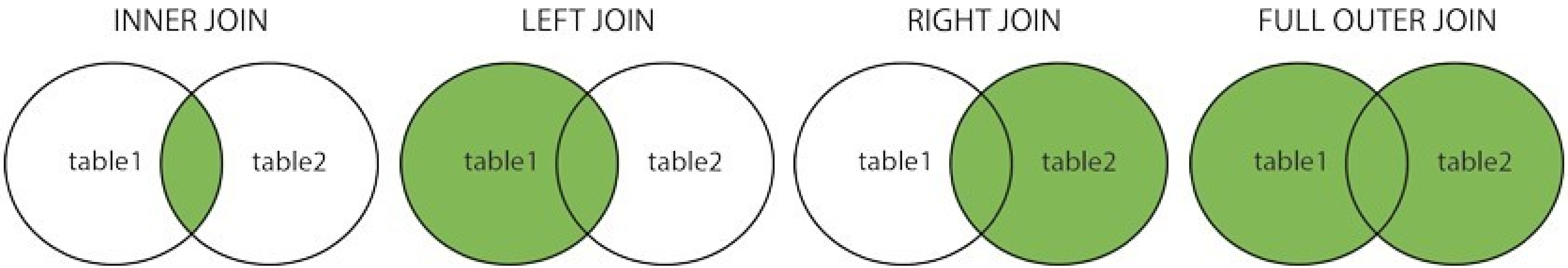
```
to_timestamp('string', 'format’)

        to_timestamp('2017-03-31 9:30:20', 'YYYY-MM-DD HH:MI:SS')
```

Join Tables

Join Tables

A JOIN clause is used to combine rows from two or more tables, based on related columns between them.



Inner Join

INNER JOIN selects records that have matching values in both tables.

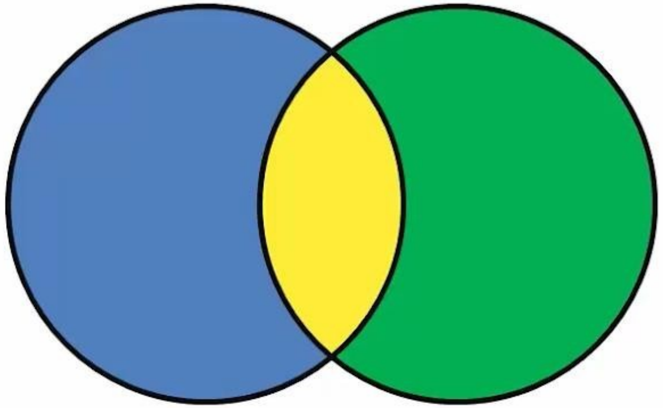
```
SELECT
   column_name(s)
FROM table1
INNER JOIN table2
   ON table1.column_name = table2.column_name;
```

Example:

```
SELECT
   Orders.OrderID,
Customers.CustomerName
FROM Orders
INNER JOIN Customers
   ON Orders.CustomerID =
Customers.CustomerID;
```

Inner Join

On: A.Customer = B.Employee

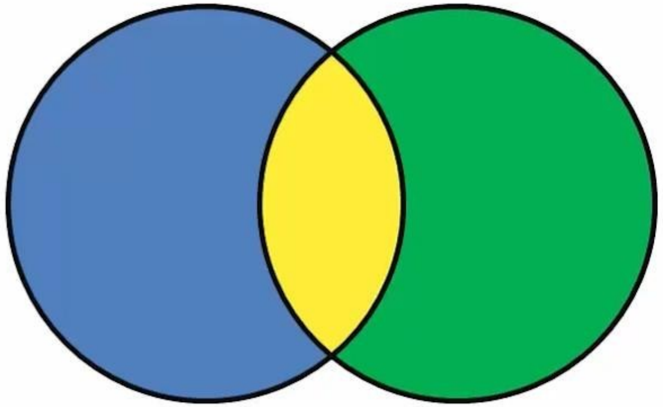


Customer	Gender	Age
Adam	male	24
Benjamin	male	32
Jack	male	29
Nick	male	37
Susan	female	31

Employee	Title	Wage
Jack	Clerk	17 $/hr
John	Clerk	19 $/hr
Mary	Mgr	21 $/hr
Susan	Mgr	19 $/hr

Inner Join

On: A.Customer = B.Employee

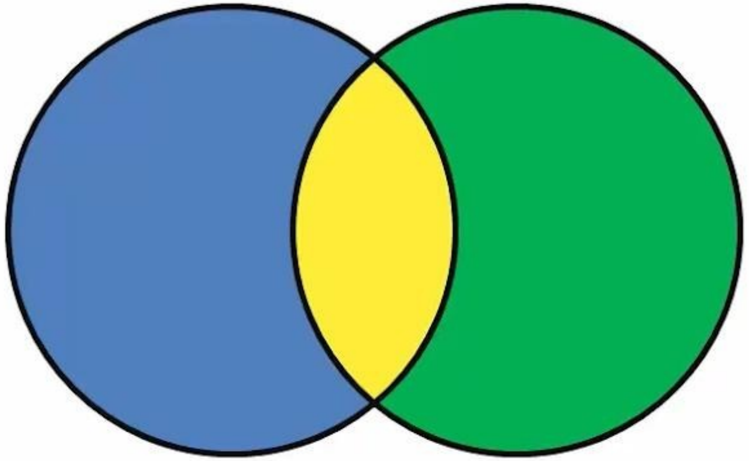


Customer	Gender	Age
~~Adam~~	~~male~~	~~24~~
~~Benjamin~~	~~male~~	~~32~~
Jack	male	29
~~Nick~~	~~male~~	~~37~~
Susan	female	31

Employee	Title	Wage
Jack	Clerk	17 $/hr
~~John~~	~~Clerk~~	~~19 $/hr~~
~~Mary~~	~~Mgr~~	~~21 $/hr~~
Susan	Mgr	19 $/hr

Inner Join

On: A.Customer = B.Employee



Customer	Gender	Age	Employee	Title	Wage
Jack	male	29	Jack	Clerk	17 $/hr
Susan	female	31	Susan	Mgr	19 $/hr

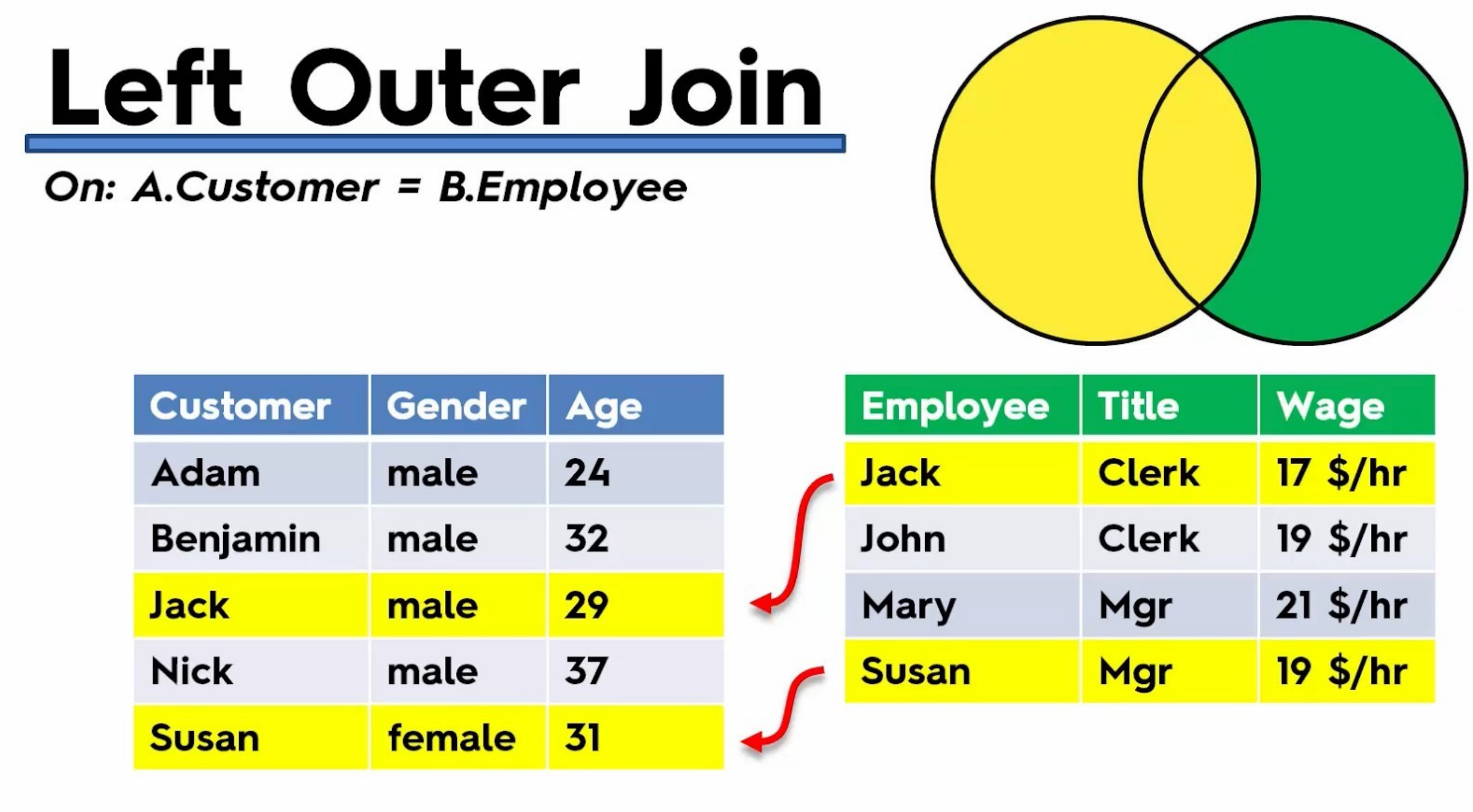
Left Join /Left Outer Join

LEFT JOIN returns all records from the left table (table1) and the matched records from the right table (table 2).

```
SELECT
  column_name(s)
FROM table1
LEFT JOIN table2
  ON table1.column_name =table2.column_name;
```

Example:

```
SELECT
  Orders.OrderID,
Customers.CustomerName
FROM Orders
LEFT JOIN Customers
  ON Orders.CustomerID =
Customers.CustomerID;
```

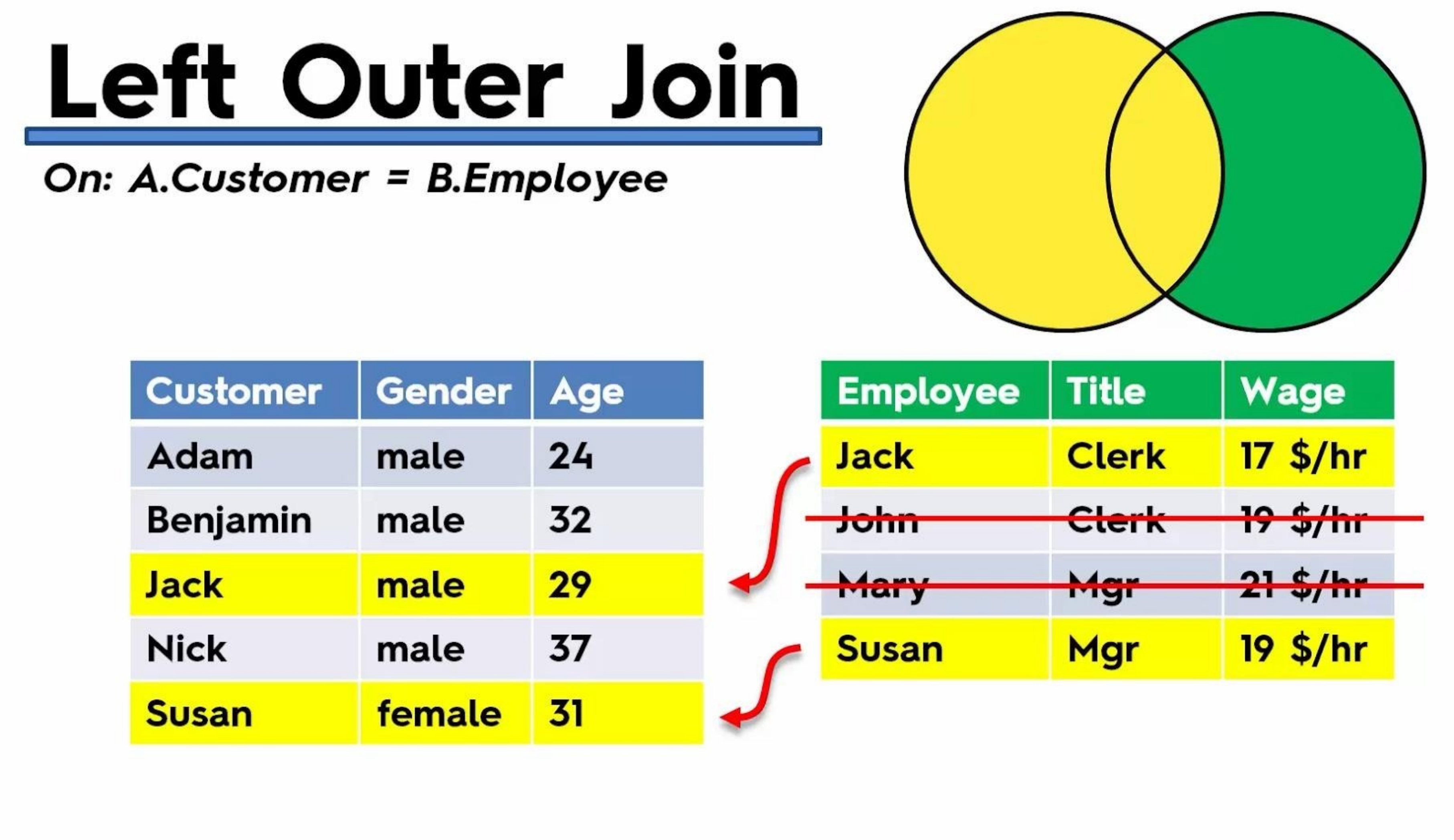


Left Outer Join

On: A.Customer = B.Employee

Customer	Gender	Age
Adam	male	24
Benjamin	male	32
Jack	male	29
Nick	male	37
Susan	female	31

Employee	Title	Wage
Jack	Clerk	17 $/hr
John	Clerk	19 $/hr
Mary	Mgr	21 $/hr
Susan	Mgr	19 $/hr

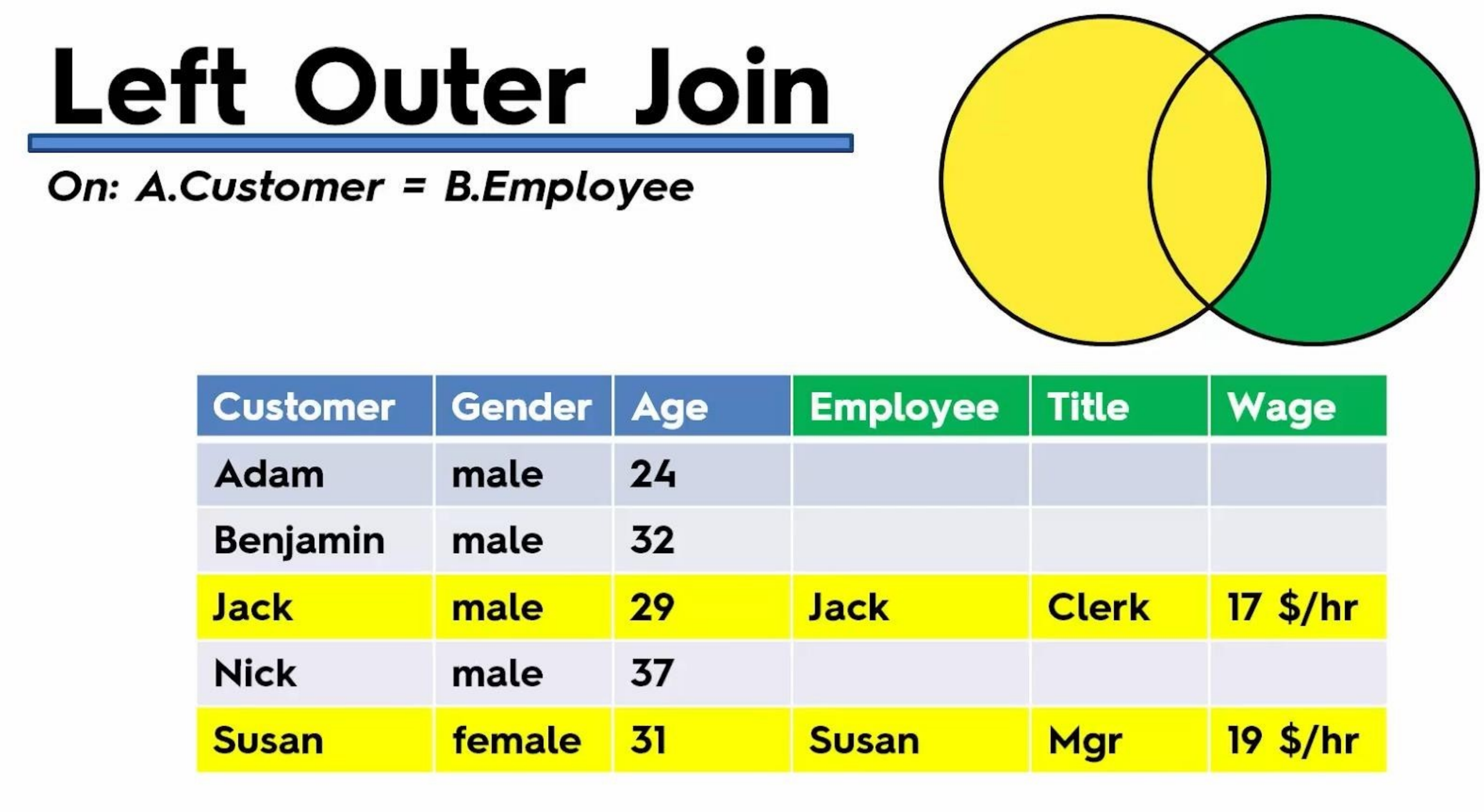


Left Outer Join

On: A.Customer = B.Employee

Customer	Gender	Age
Adam	male	24
Benjamin	male	32
Jack	male	29
Nick	male	37
Susan	female	31

Employee	Title	Wage
Jack	Clerk	17 $/hr
~~John~~	~~Clerk~~	~~19 $/hr~~
~~Mary~~	~~Mgr~~	~~21 $/hr~~
Susan	Mgr	19 $/hr



Left Outer Join

On: A.Customer = B.Employee

Customer	Gender	Age	Employee	Title	Wage
Adam	male	24			
Benjamin	male	32			
Jack	male	29	Jack	Clerk	17 $/hr
Nick	male	37			
Susan	female	31	Susan	Mgr	19 $/hr

Right Join /Right Outer Join

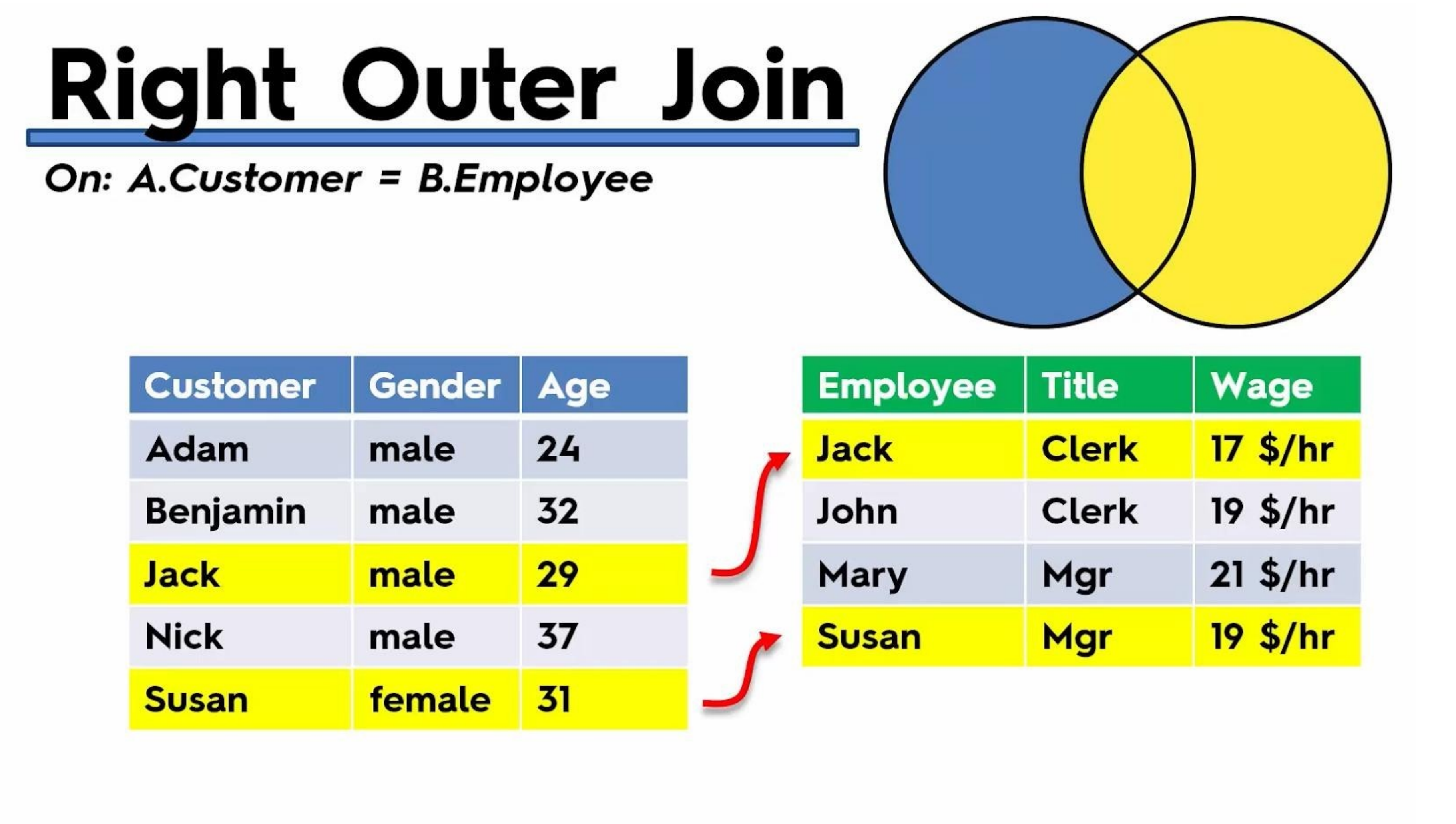
RIGHT JOIN returns all records from the right table (table 2) and the matched records

from the left table (table1).

```
SELECT
  column_name(s)
FROM table1
RIGHT JOIN table2
  ON table1.column_name =table2.column_name;
```

Example:

```
SELECT
  Orders.OrderID,
Customers.CustomerName
FROM Orders
RIGHT JOIN Customers
  ON Orders.CustomerID =
Customers.CustomerID;
```

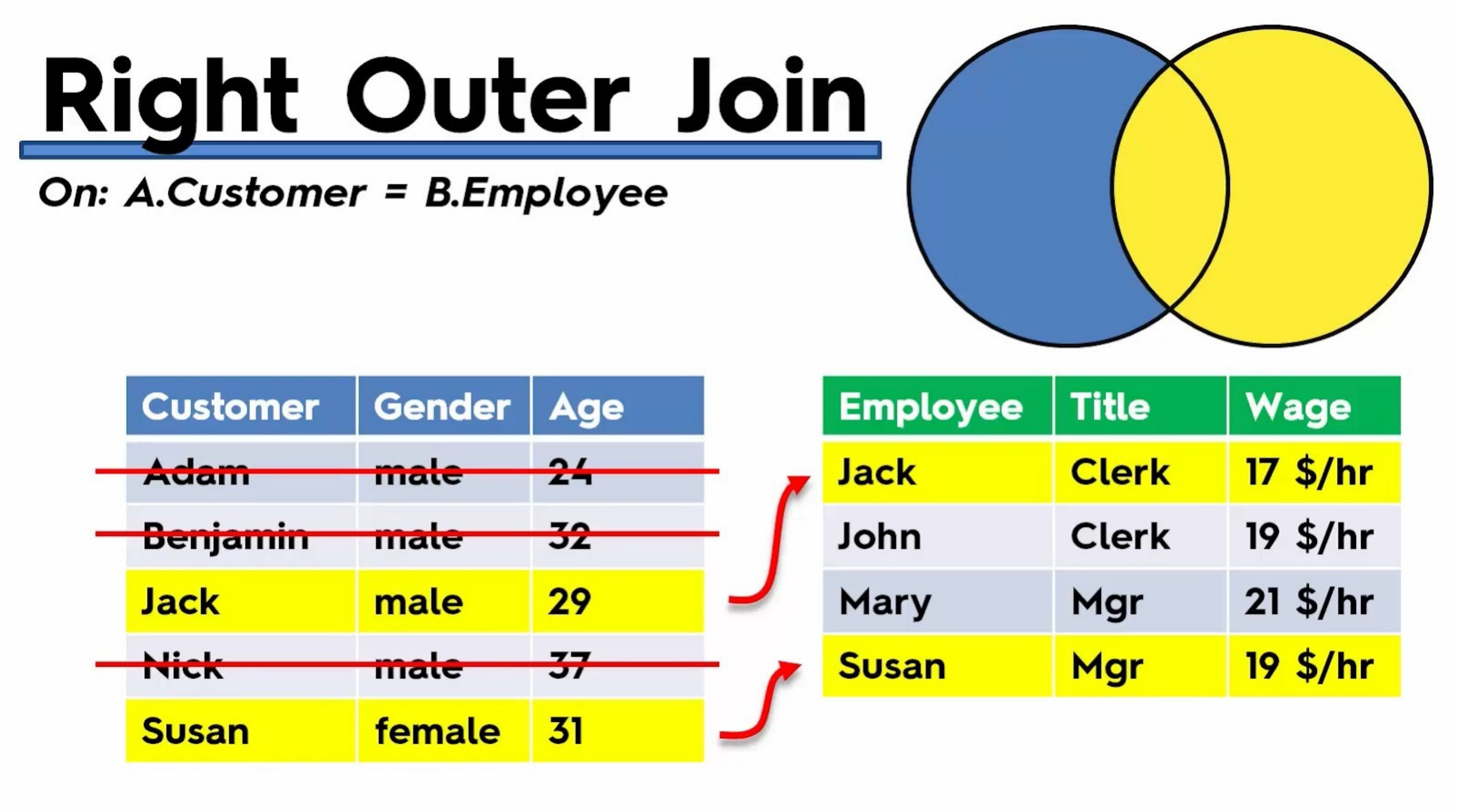


Right Outer Join

On: A.Customer = B.Employee

Customer	Gender	Age
Adam	male	24
Benjamin	male	32
Jack	male	29
Nick	male	37
Susan	female	31

Employee	Title	Wage
Jack	Clerk	17 $/hr
John	Clerk	19 $/hr
Mary	Mgr	21 $/hr
Susan	Mgr	19 $/hr

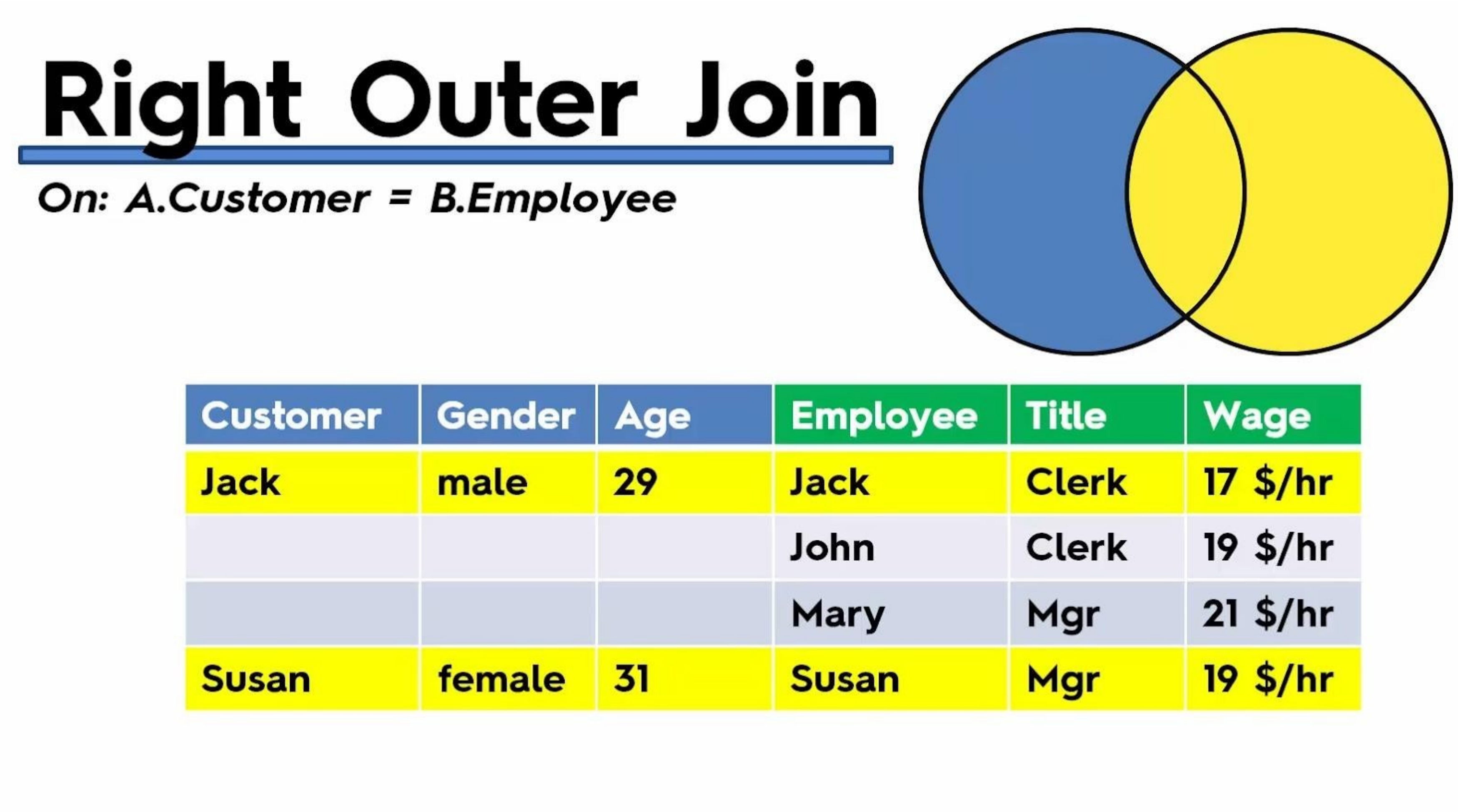


Right Outer Join

On: A.Customer = B.Employee

Customer	Gender	Age
~~Adam~~	~~male~~	~~24~~
~~Benjamin~~	~~male~~	~~32~~
Jack	male	29
~~Nick~~	~~male~~	~~37~~
Susan	female	31

Employee	Title	Wage
Jack	Clerk	17 $/hr
John	Clerk	19 $/hr
Mary	Mgr	21 $/hr
Susan	Mgr	19 $/hr



Right Outer Join

On: A.Customer = B.Employee

Customer	Gender	Age	Employee	Title	Wage
Jack	male	29	Jack	Clerk	17 $/hr
			John	Clerk	19 $/hr
			Mary	Mgr	21 $/hr
Susan	female	31	Susan	Mgr	19 $/hr

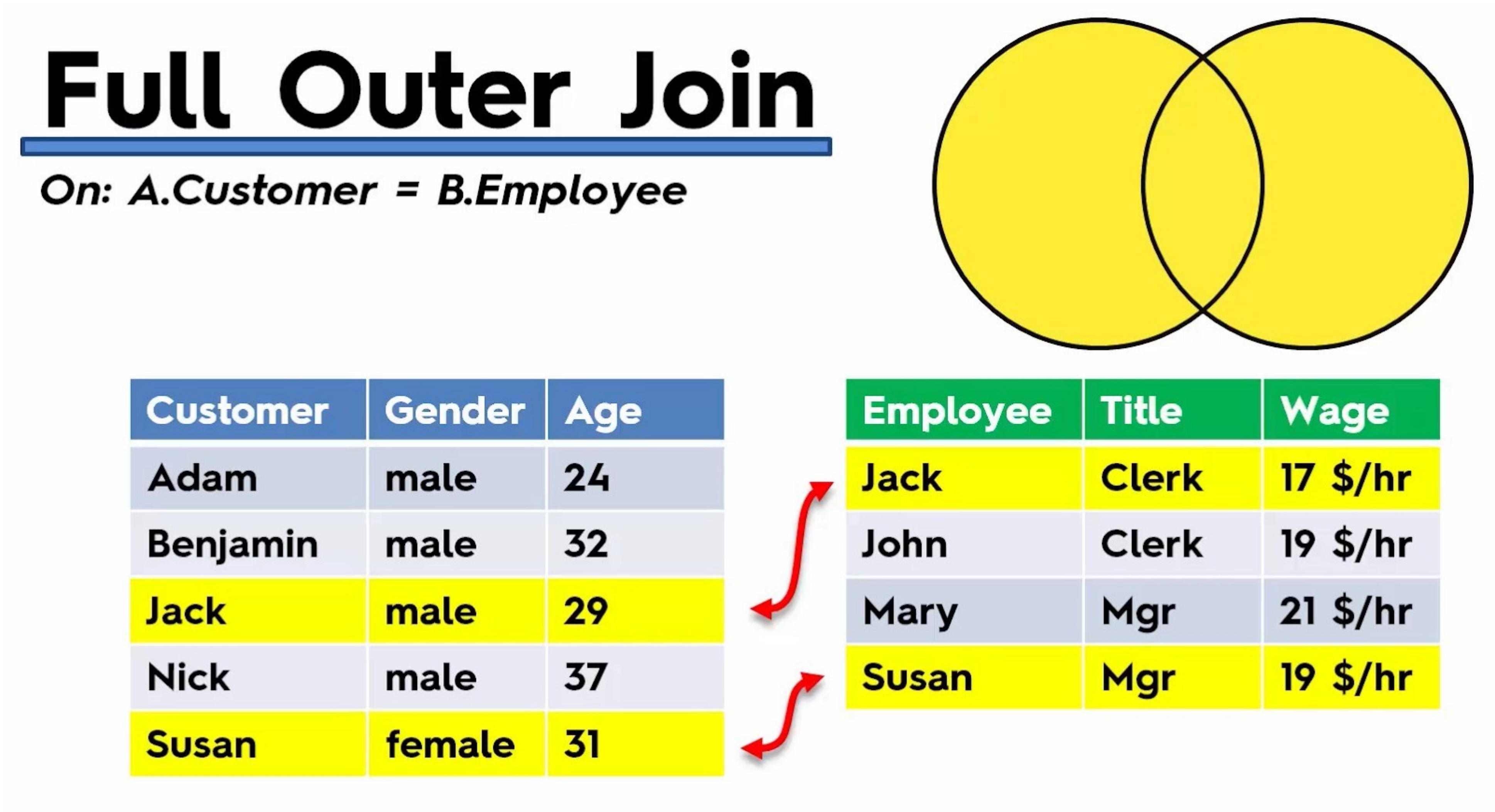
Full Outer Join

FULL OUTER JOIN returns all records when there is match in either left table (table1) or right table (table2).

```
SELECT
   column_name(s)
FROM table1
FULL OUTER JOIN table2
   ON table1.column_name =table2.column_name;
```

Example:

```
SELECT
   Orders.OrderID,
Customers.CustomerName
FROM Orders
FULL OUTER JOIN Customers
   ON Orders.CustomerID =
Customers.CustomerID;
```



Full Outer Join

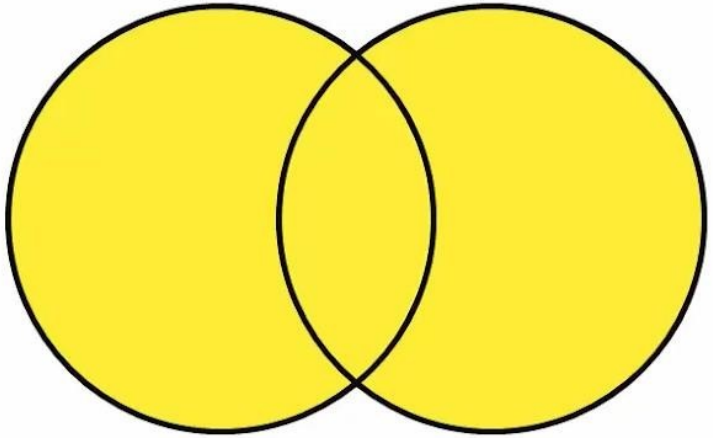
On: A.Customer = B.Employee

Customer	Gender	Age
Adam	male	24
Benjamin	male	32
Jack	male	29
Nick	male	37
Susan	female	31

Employee	Title	Wage
Jack	Clerk	17 $/hr
John	Clerk	19 $/hr
Mary	Mgr	21 $/hr
Susan	Mgr	19 $/hr

Full Outer Join

On: A.Customer = B.Employee

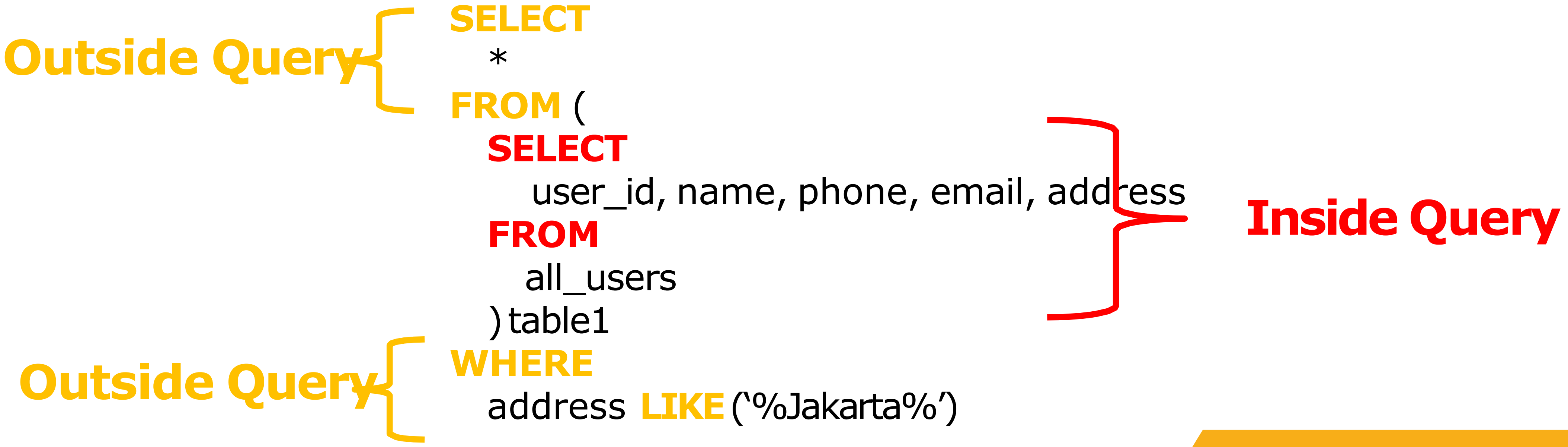


Customer	Gender	Age	Employee	Title	Wage
Adam	male	24			
Benjamin	male	32			
Jack	male	29	Jack	Clerk	17 $/hr
Nick	male	37			
Susan	female	31	Susan	Mgr	19 $/hr
			John	Clerk	19 $/hr
			Mary	Mgr	21 $/hr

Sub Query

Sub Query

A subquery is a SQL query nested inside a larger query. Outside query re-process inside query’s table. Note that subquery statements are enclosed between parenthesis.



```
SELECT
    *
FROM (
    SELECT
        user_id, name, phone, email, address
    FROM
        all_users
    ) table1
WHERE
    address LIKE (‘%Jakarta%’)
```

“Behind The Scene” of Sub Query

```
SELECT
      user_id
     ,name
     ,phone
     ,email
     ,address
  FROM
     all_users
```

table1

user_id	name	phone	email	address

```
SELECT
   column_name(s)
FROM
   table1
WHERE
   address LIKE ('%Jakarta%
```

Common Table Expression (CTE)

What is CTE?

```
with
cte_1 as(
    select
         column_1
    from table_1
)

select *
from cte_1
```

Why do you need CTE?

1. **Readability** - CTE’s promote readability. Rather than lump all you query logic into one large query, create several CTE’s, which are the combined later in the statement.
2. **Making query more effective**

Let’s Practice

1. What is the most sold category in Germany?

2. What is the most ship mode for Technology Category order that received in September 2011?

3. Which country and its city has the highest order in July 2011?

4. What is the average of total profit for segment "Home Office"?

Summary

These query functions enable us to transform date data and combine different tables.

Please master it carefully since most of the queries use these functions.

1. Date Functions
2. Join Tables
3. Inner Join
4. Left Join
5. Right Join
6. Full Outer Join
7. Sub Query

DigitalSkola

Uncover The World Of Digital Skills With Us

Thank YOU