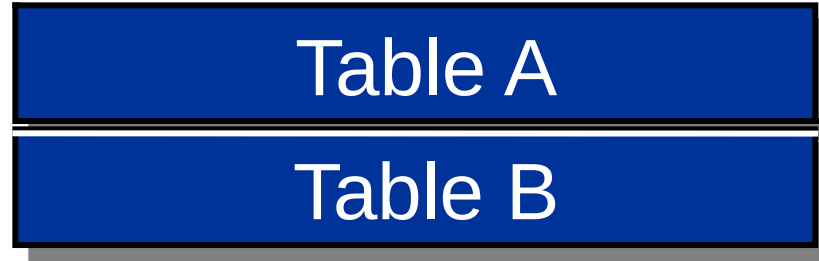


SQL Joins



Combining Data from Multiple Tables

SQL uses set operators to combine tables vertically.



This produces results that can be compared to a DATA step concatenation.

Types of Joins

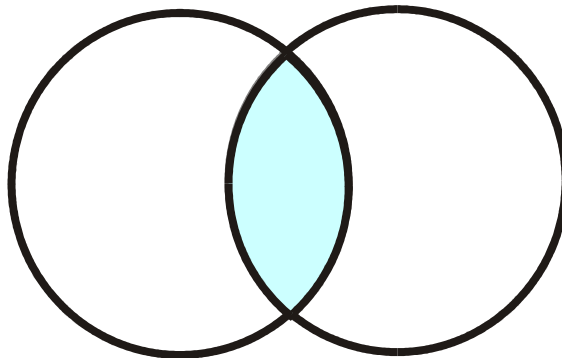
SQL supports two types of joins:

- inner joins
- outer joins

Types of Joins

Inner joins

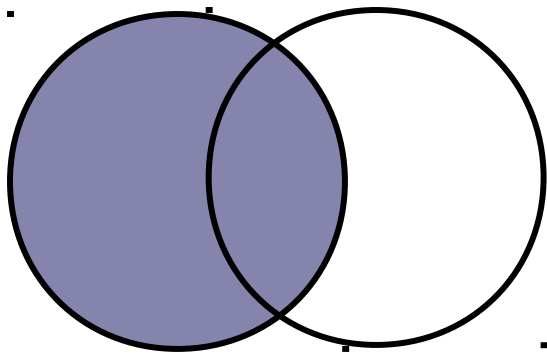
- return only matching rows
- enable a maximum of 256 tables to be joined at the same time.



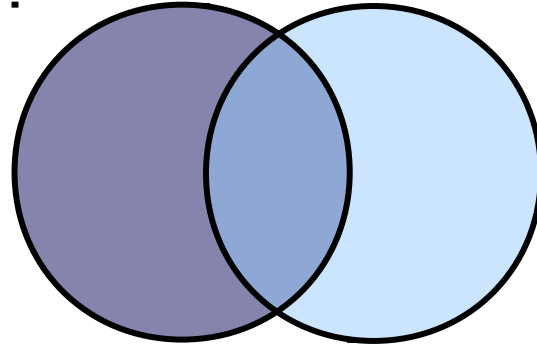
Types of Joins

Outer joins

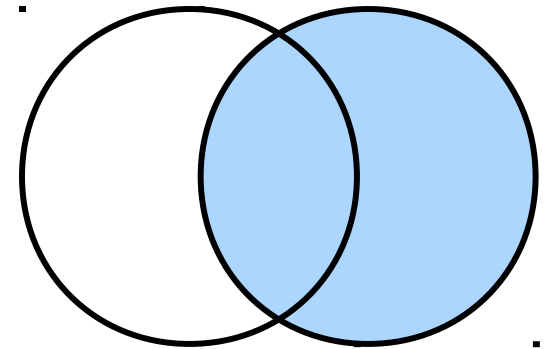
- return all matching rows, plus nonmatching rows from one or both tables
- can be performed on only two tables or views at a time.



Left



Full



Right

Cartesian Product

To understand how SQL processes a join, it is important to understand the concept of the Cartesian product.

A query that lists multiple tables in the FROM clause without a WHERE clause produces all possible combinations of rows from all tables. This result is called the *Cartesian product*.

```
select *  
  from one, two;
```

Cartesian Product

Table One

X	A
1	a
4	d
2	b

Table Two

X	B
2	x
3	y
5	v

Cartesian Product

Table One

X	A
1	a
4	d
2	b

Table Two

X	B
2	x
3	y
5	v

Result Set

X	A	X	B
1	a	2	x

Cartesian Product

Table One

X	A
1	a
4	d
2	b

Table Two

X	B
2	x
3	y
5	v

Result Set

X	A	X	B
1	a	2	x
1	a	3	y

Cartesian Product

Table One

X	A
1	a
4	d
2	b

Table Two

X	B
2	x
3	y
5	v

Result Set

X	A	X	B
1	a	2	x
1	a	3	y
1	a	5	v

Cartesian Product

Table One

X	A
1	a
4	d
2	b

Table Two

X	B
2	x
3	y
5	v

Result Set

X	A	X	B
1	a	2	x
1	a	3	y
1	a	5	v
4	d	2	x

Cartesian Product

Table One

X	A
1	a
4	d
2	b

Table Two

X	B
2	x
3	y
5	v

Result Set

X	A	X	B
1	a	2	x
1	a	3	y
1	a	5	v
4	d	2	x
4	d	3	y

Cartesian Product

Table One

X	A
1	a
4	d
2	b

Table Two

X	B
2	x
3	y
5	v

Result Set

X	A	X	B
1	a	2	x
1	a	3	y
1	a	5	v
4	d	2	x
4	d	3	y
4	d	5	v

Cartesian Product

Table One

X	A
1	a
4	d
2	b

Table Two

X	B
2	x
3	y
5	v

Result Set

X	A	X	B
1	a	2	x
1	a	3	y
1	a	5	v
4	d	2	x
4	d	3	y
4	d	5	v
2	b	2	x

Cartesian Product

Table One

X	A
1	a
4	d
2	b

Table Two

X	B
2	x
3	y
5	v

Result Set

X	A	X	B
1	a	2	x
1	a	3	y
1	a	5	v
4	d	2	x
4	d	3	y
4	d	5	v
2	b	2	x
2	b	3	y

Cartesian Product

Table One

X	A
1	a
4	d
2	b

Table Two

X	B
2	x
3	y
5	v

Result Set

X	A	X	B
1	a	2	x
1	a	3	y
1	a	5	v
4	d	2	x
4	d	3	y
4	d	5	v
2	b	2	x
2	b	3	y

Cartesian Product

Table One

X	A
1	a
4	d
2	b

} 3 rows

Table Two

X	B
2	x
3	y
5	v

Result Set

X	A	X	B
1	a	2	x
1	a	3	y
1	a	5	v
4	d	2	x
4	d	3	y
4	d	5	v
2	b	2	x
2	b	3	y

Cartesian Product

Table One

X	A
1	a
4	d
2	b

3 rows

Table Two

X	B
2	x
3	y
5	v

3 rows

Result Set

X	A	X	B
1	a	2	x
1	a	3	y
1	a	5	v
4	d	2	x
4	d	3	y
4	d	5	v
2	b	2	x
2	b	3	y

Cartesian Product

Table One

X	A
1	a
4	d
2	b

3 rows

X

3 rows

Table Two

X	B
2	x
3	y
5	v

Result Set

X	A	X	B
1	a	2	x
1	a	3	y
1	a	5	v
4	d	2	x
4	d	3	y
4	d	5	v
2	b	2	x
2	b	3	y

9 rows

Cartesian Product

The number of rows in a Cartesian product is the product of the number of rows in the contributing tables.

$$3 \times 3 = 9$$

$$1,000 \times 1,000 = 1,000,000$$

$$100,000 \times 100,000 = 10,000,000,000$$

A Cartesian product is rarely the **desired** result of a query.

Poll

Quiz



5.02 Quiz

How many rows are returned from this query?

```
select *  
  from three, four;
```

Table Three

X	A
1	a1
1	a2
2	b1
2	b2
4	d

Table Four

X	B
2	x1
2	x2
3	y
5	v

5.02 Quiz – Correct Answer

How many rows are returned from this query?

The query produces 20 rows.

```
select *  
  from three, four;
```

Table Three

X	A
1	a1
1	a2
2	b1
2	b2
4	d

Table Four

X	B
2	x1
2	x2
3	y
5	v

5*4=20

Partial Results Set

X	A	X	B
1	a1	2	x1
1	a1	2	x2
1	a1	3	y
1	a1	5	v
1	a2	2	x1
1	a2	2	x2
1	a2	3	y
1	a2	5	v
2	b1	2	x1
2	b1	2	x2
2	b1	3	y
2	b1	5	v
2	b2	2	x1
2	b2	2	x2
2	b2	3	y
2	b2	5	v
4	d	2	x1
4	d	2	x2
4	d	3	y
4	d	5	v

Inner Joins

Inner join syntax resembles Cartesian product syntax, but a WHERE clause restricts which rows are returned.

General form of an inner join:

```
SELECT column-1<, ...column-n>  
  FROM table-1|view-1<, ... table-n|view-n>  
  WHERE join-condition(s)  
        <AND other subsetting conditions>  
        <other clauses>;
```


Inner Joins

Inner join syntax resembles Cartesian product syntax, but a WHERE clause restricts which rows are returned.

General form of an inner join:

```
SELECT column-1<, ...column-n>
FROM table-1|view-1<, ... table-n|view-n>
WHERE join-condition(s)
        <AND other subsetting conditions>
        <other clauses>;
```

Significant syntax changes from earlier queries:

- The FROM clause references multiple tables.
- The WHERE clause includes join conditions in addition to other subsetting specifications.

SYNTAX:

SELECT column_name(s)

FROM table1

INNER JOIN table2 ON table1.column_name =
table2.column_name;

Inner Joins: Cartesian Product Built

Table One

X	A
1	a
4	d
2	b

```
select *  
  from one, two
```

Table Two

X	B
2	x
3	y
5	v

X	A	X	B
1	a	2	x
1	a	3	y
1	a	5	v
4	d	2	x
4	d	3	y
4	d	5	v
2	b	2	x
2	b	3	y
2	b	5	v

Inner Joins: WHERE Clause Restricts Rows

Table One

X	A
1	a
4	d
2	b

```
select *  
  from one, two  
 where one.x=two.x;
```

Table Two

X	B
2	x
3	y
5	v

X	A	X	B
1	a	2	x
1	a	3	y
1	a	5	v
4	d	2	x
4	d	3	y
4	d	5	v
2	b	2	x
2	b	3	y
2	b	5	v



Inner Joins: Results Are Returned

Table One

X	A
1	a
4	d
2	b

```
select *  
  from one, two  
 where one.x=two.x;
```

Table Two

X	B
2	x
3	y
5	v

X	A	X	B
2	b	2	x

- ✎ Tables do not have to be sorted before they are joined.

Inner Joins

One method of displaying the X column only once is to use a table qualifier in the SELECT list.

Table One

X	A
1	a
4	d
2	b

Table Two

X	B
2	x
3	y
5	v

```
select one.x, a, b  
  from one, two  
 where one.x=two.x;
```

X	A	B
2	b	x

Inner Joins

Display all combinations of rows with matching keys, including duplicates.

Table Three

X	A
1	a1
1	a2
2	b1
2	b2

Table Four

X	B
2	x1
2	x2
3	y
5	v

```
select *  
  from three, four  
 where three.x=four.x;
```

Inner Joins

Display all combinations of rows with matching keys, including duplicates.

Table Three

X	A
1	a1
1	a2
2	b1
2	b2

Table Four

X	B
2	x1
2	x2
3	y
5	v

Results Set

X	A	X	B
2	b1	2	x1
2	b1	2	x2
2	b2	2	x1
2	b2	2	x2

```
select *  
  from three, four  
 where three.x=four.x;
```


Inner Join Alternate Syntax

An inner join can also be accomplished using an alternate syntax, which limits the join to a maximum of two tables.

General form of an inner join:

```
SELECT column-1 <, ...column-n>  
FROM table-1  
INNER JOIN  
      table-2  
ON join-condition(s)  
<other clauses>;
```

- ✎ This syntax is common in SQL code produced by code generators such as SAS Enterprise Guide. The ON clause specifies the JOIN criteria; a WHERE clause **can** be added to subset the results.

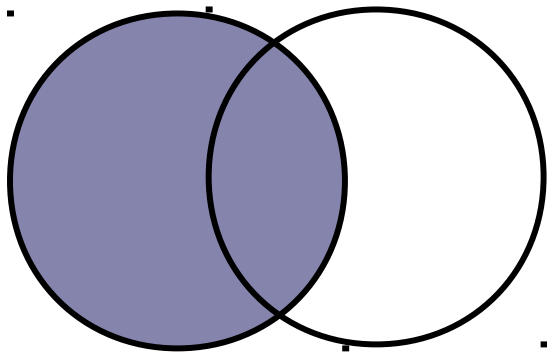
Outer Joins

Inner joins returned only matching rows. When you join tables, you might want to include nonmatching rows as well as matching rows.

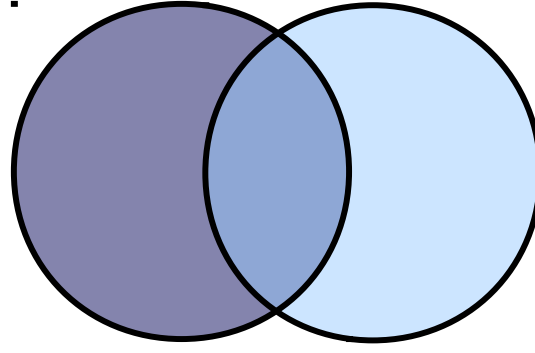
Outer Joins

You can retrieve both nonmatching and matching rows using an outer join.

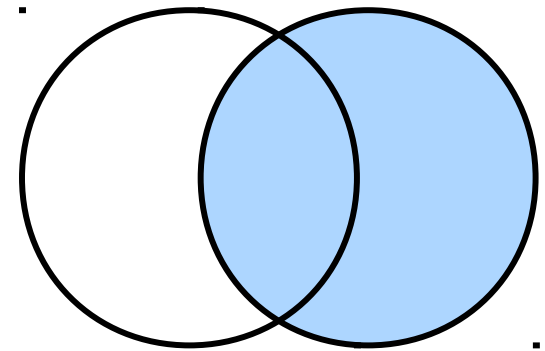
Outer joins include left, full, and right outer joins. Outer joins can process only two tables at a time.



Left



Full



Right

Compare Inner Joins And Outer Joins

The following table is a comparison of inner and outer join syntax and limitations:

Key Point	Inner Join	Outer Join
Table Limit	256	2
Join Behavior	Returns matching rows only	Returns matching and nonmatching rows
Join Options	Matching rows only	LEFT, FULL, RIGHT
Syntax changes	<ul style="list-style-type: none">■ Multiple tables in the FROM clause■ WHERE clause that specifies join criteria	ON clause that specifies join criteria

Outer Joins

Outer join syntax is similar to the inner join alternate syntax.

General form of an outer join:

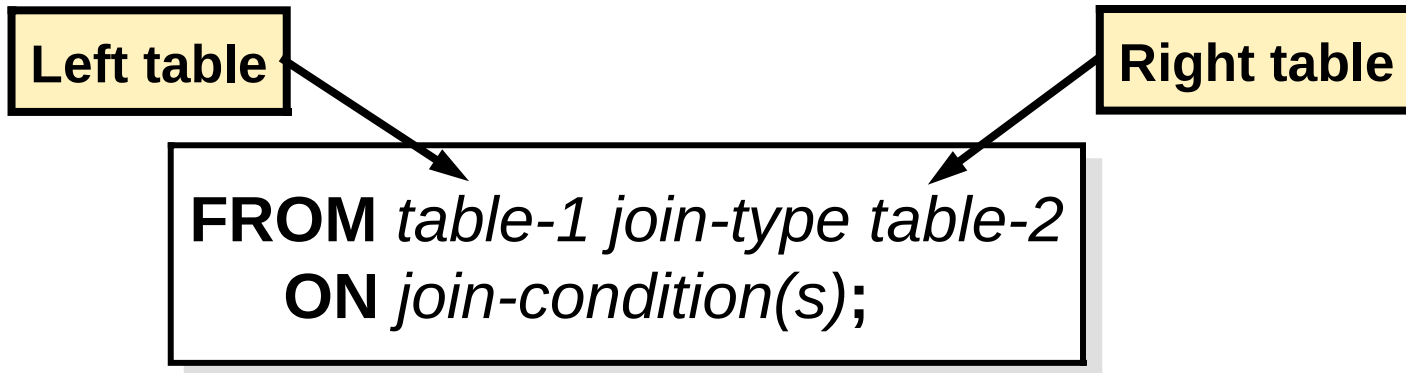
```
SELECT column-1 <, ...column-n>  
FROM table-1  
LEFT|RIGHT|FULL JOIN  
      table-2  
ON join-condition(s)  
    <other clauses>;
```

The ON clause specifies the join criteria in outer joins.

Determining Left and Right

Consider the position of the tables in the FROM clause.

- Left joins include all rows from the first (left) table, even if there are no matching rows in the second (right) table.
- Right joins include all rows from the second (right) table, even if there are no matching rows in the first (left) table.
- Full joins include all rows from both tables, even if there are no matching rows in either table.



Left Join

Table One

X	A
1	a
4	d
2	b

Table Two

X	B
2	x
3	y
5	v

```
select *  
  from one left join two  
    on one.x = two.x;
```

X	A	X	B
1	a	.	
2	b	2	x
4	d	.	

Right Join

Table Two

X	B
2	x
3	y
5	v

Table One

X	A
1	a
4	d
2	b

```
select *  
  from two right join one  
    on one.x = two.x;
```

X	B	X	A
.		1	a
2	x	2	b
.		4	d

Full Join

Table One

X	A
1	a
4	d
2	b

Table Two

X	B
2	x
3	y
5	v

```
select *  
  from one full join two  
    on one.x = two.x;
```

X	A	X	B
1	a	.	
2	b	2	x
.		3	y
4	d	.	
		5	v

Business Scenario

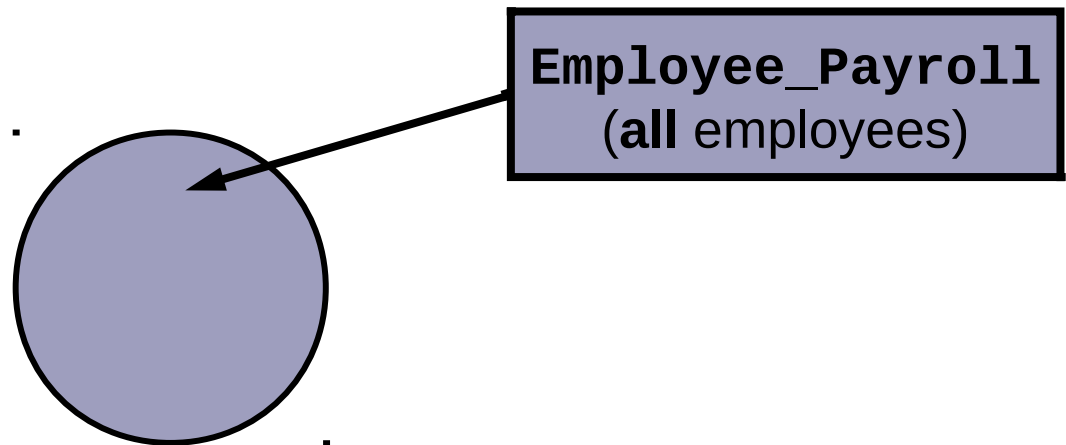
List the employee ID and gender for all married employees. Include the names of any charities to which the employee donates via the company program.



Business Scenario

Considerations:

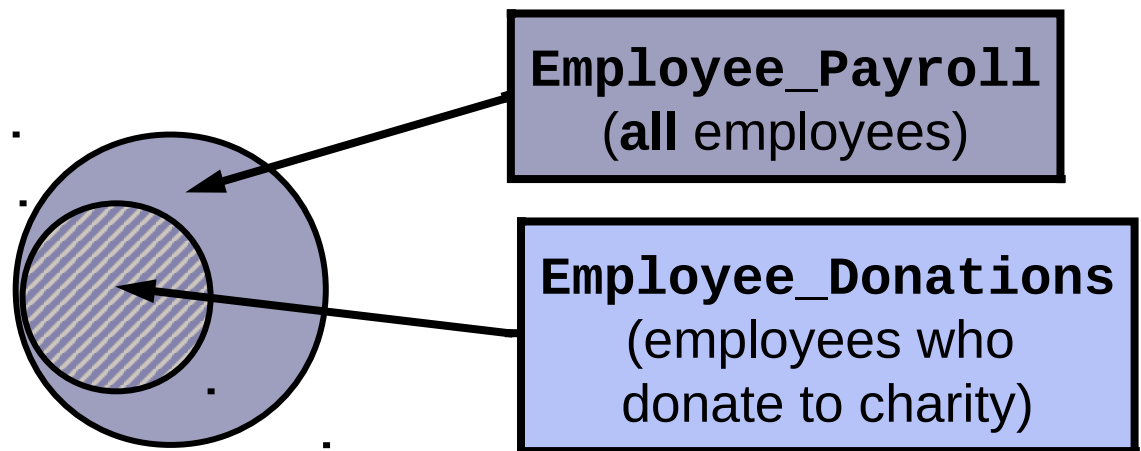
- The table **Employee_Payroll** contains gender and marital status information.



Business Scenario

Considerations:

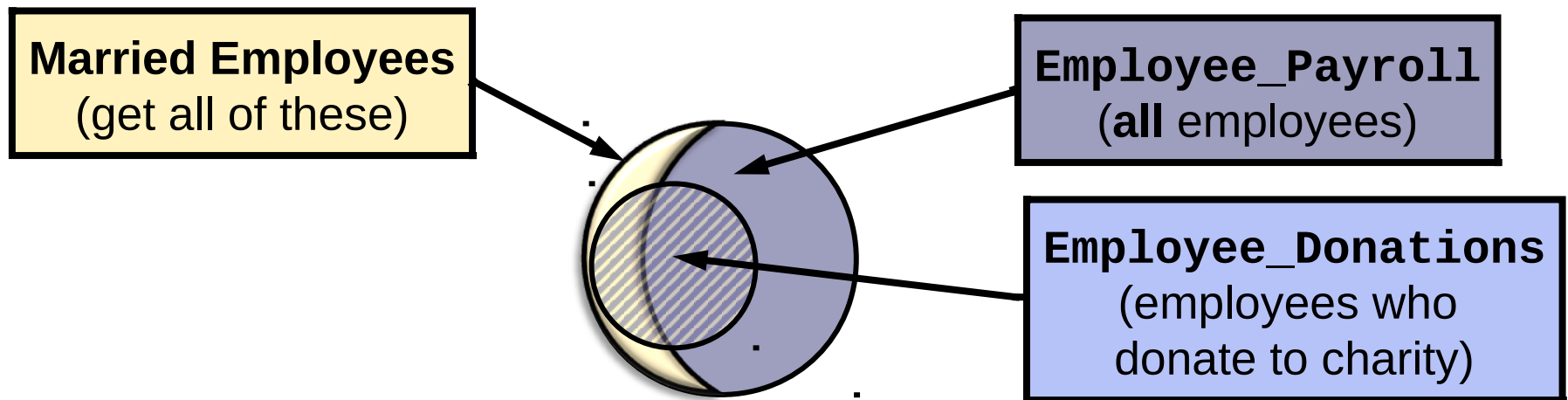
- The table **Employee_Payroll** contains gender and marital status information.
- The table **Employee_Donations** contains records only for those employees who donate to a charity via the company program.



Business Scenario

Considerations:

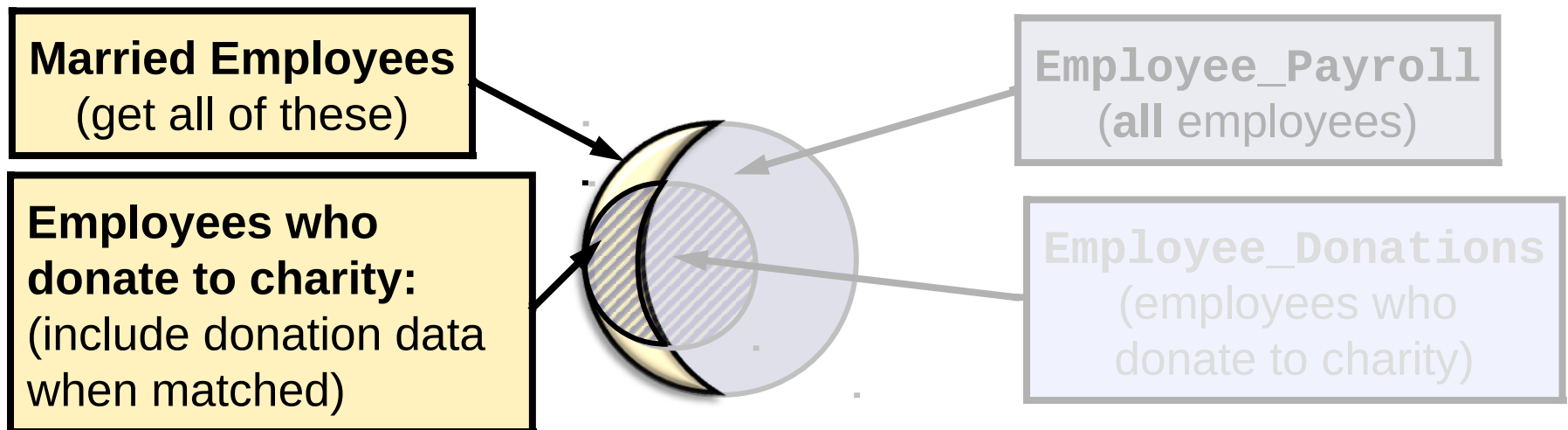
- The table **Employee_Payroll** contains gender and marital status information.
- The table **Employee_Donations** contains records only for those employees who donate to a charity via the company program.
- Less than half of all employees are married.



Business Scenario

Considerations:

- The table **Employee_Payroll** contains gender and marital status information.
- The table **Employee_Donations** contains records only for those employees who donate to a charity via the company program.
- Less than half of all employees are married.



Poll 

Quiz

5.05 Multiple Choice Poll

For the report, you need the data for all married employees from **orion.Employee_Payroll**. You also want to include the charity names from the **orion.Employee_Donations** table if **Employee_ID** matches. What type of join should you use to combine the information from these two tables?

- a. Inner Join
- b. Left Join
- c. Full Join
- d. None of the above

5.05 Multiple Choice Poll – Correct Answer

For the report, you need the data for all married employees from **orion.Employee_Payroll**. You also want to include the charity names from the **orion.Employee_Donations** table if **Employee_ID** matches. What type of join should you use to combine the information from these two tables?

- a. Inner Join
- ☒ b. Left Join
- c. Full Join
- d. None of the above

Outer Joins

```
select Employee_payroll.Employee_ID,  
       Employee_Gender, Recipients  
from Employee_payroll  
     left join  
       Employee_donations  
on Employee_payroll.Employee_ID=  
   Employee_donations.Employee_ID  
where Marital_Status="M"  
;
```

Outer Joins

SQL Output

Employee_ID	Employee_ Gender	Recipients
121128	F	Cancer Cures, Inc.
121131	M	Vox Victimas 40%, Conserve Nature, Inc. 60%
121132	M	EarthSalvors 50%, Vox Victimas 50%
121133	M	Disaster Assist, Inc.
121138	M	Cuidadores Ltd.
121139	F	
121142	M	AquaMissions International 10%, Child Survivors 90%
121143	M	Mitleid International 60%, Save the Baby Animals
40%		
121144	F	
121145	M	Save the Baby Animals
121146	F	
121147	F	Cuidadores Ltd. 50%, Mitleid International 50%
121148	M	

✍ Remember that output order is not guaranteed unless you use an ORDER BY clause.

Using a Table Alias

An *alias* is a table nickname. You can assign an alias to a table by following the table name in the FROM clause with the AS keyword and a nickname for the table. Then use the alias in other clauses of the QUERY statement.

General form of the FROM clause:

```
SELECT alias-1.column-1<, ...alias-2.column-n>  
FROM table-1 AS alias-1  
      join-type  
      table-2 AS alias-2  
ON join-condition(s)  
    <other clauses>;
```

Using a Table Alias

```
select p.Employee_ID, P.Employee_Gender,  
       P.Recipients  
from Employee_payroll as p  
left join  
       Employee_donations as d  
on p.Employee_ID=d.Employee_ID  
where Marital_Status="M"  
;
```