Chapter 5: SQL Joins

5.1: Introduction to SQL Joins

5.2: Complex SQL Joins

Chapter 5: SQL Joins

5.1: Introduction to SQL Joins

5.2: Complex SQL Joins

Objectives

- Horizontally combine data from multiple tables.
- Distinguish between inner and outer SQL joins.
- Compare SQL joins to DATA step merges.

Combining Data from Multiple Tables

SQL uses set operators to combine tables vertically.

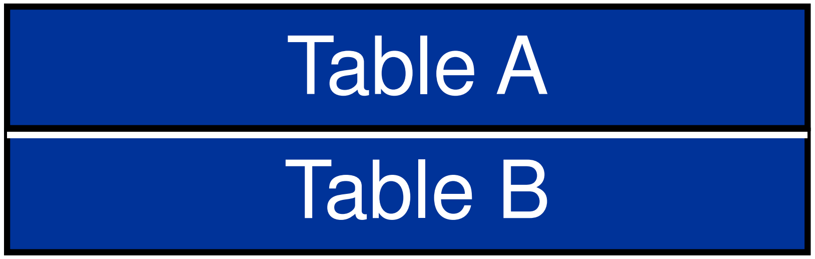


Table A

Table B

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

Combining Data from Multiple Tables

SQL uses joins to combine tables horizontally.

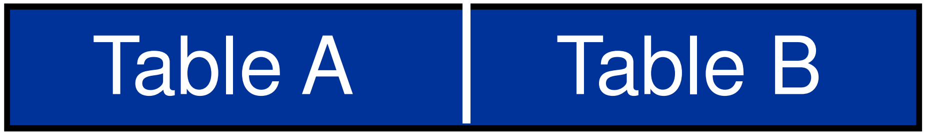


Table A

Table B

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

Poll

Quiz

5.01 Multiple Choice Poll

Which of these DATA step statements is used to combine tables horizontally?

a. SET

b. APPEND

c. MERGE

d. INPUT

e. INFILE

5.01 Multiple Choice Poll – Correct Answer

Which of these DATA step statements is used to combine tables horizontally?

a. SET

b. APPEND

[✓ c.] MERGE

d. INPUT

e. INFILE

Types of Joins

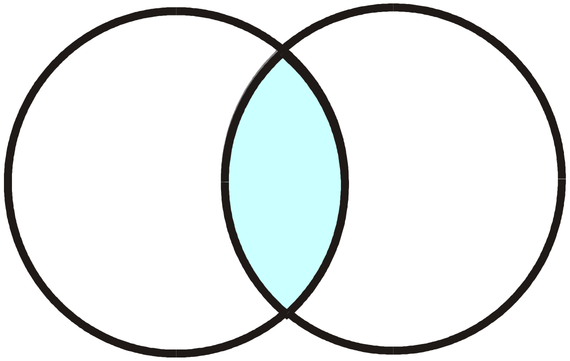
PROC SQL supports two types of joins:

- inner joins
- outer joins

Types of Joins

Inner joins

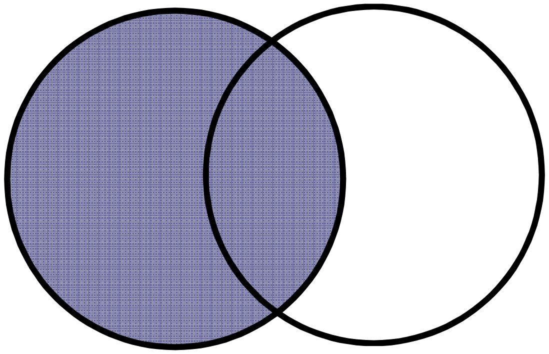
- return only matching rows
- enable a maximum of 256 tables to be joined at the same time.



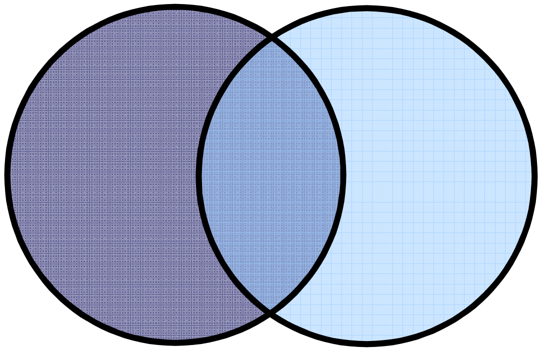
Types of Joins

Outer joins

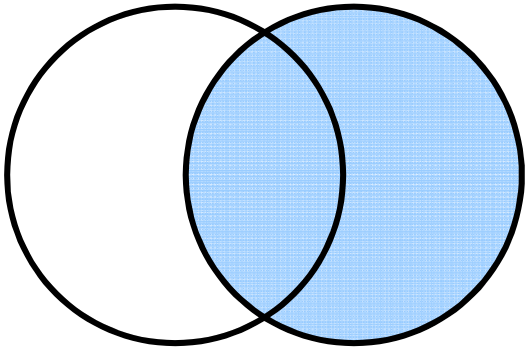
- return all matching rows, plus nonmatching rows from one or both tables
- can be performed on only two tables or views at a time.



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;
```

s105d01

Cartesian Product

Table One

X	A
1	a
4	d
2	b

Table Two

X	B
2	x
3	y
5	v

s105d01

...

Cartesian Product

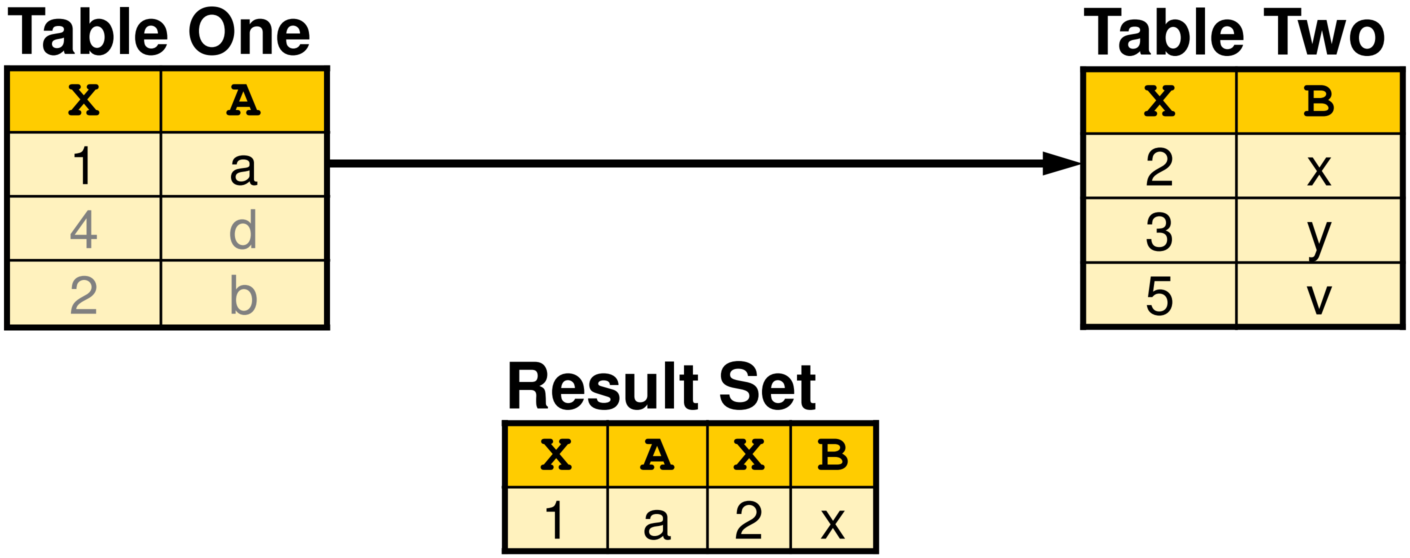


Table One

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

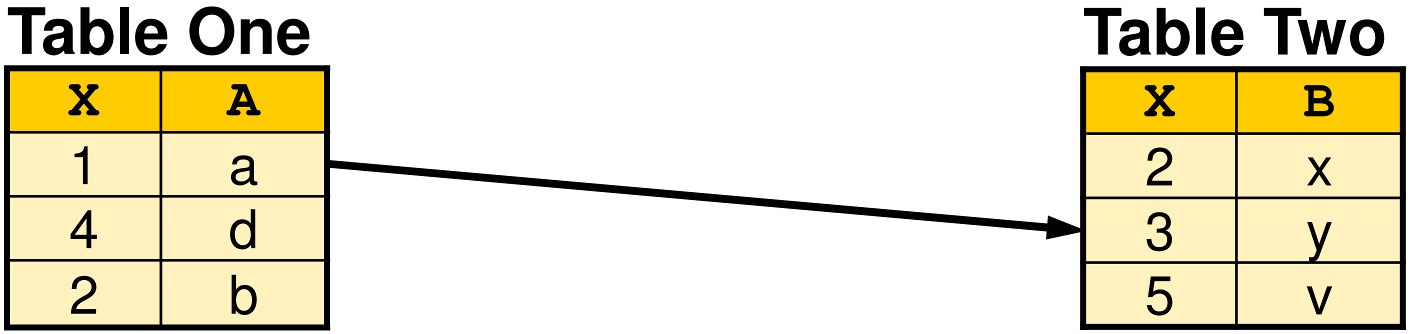


Table One

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

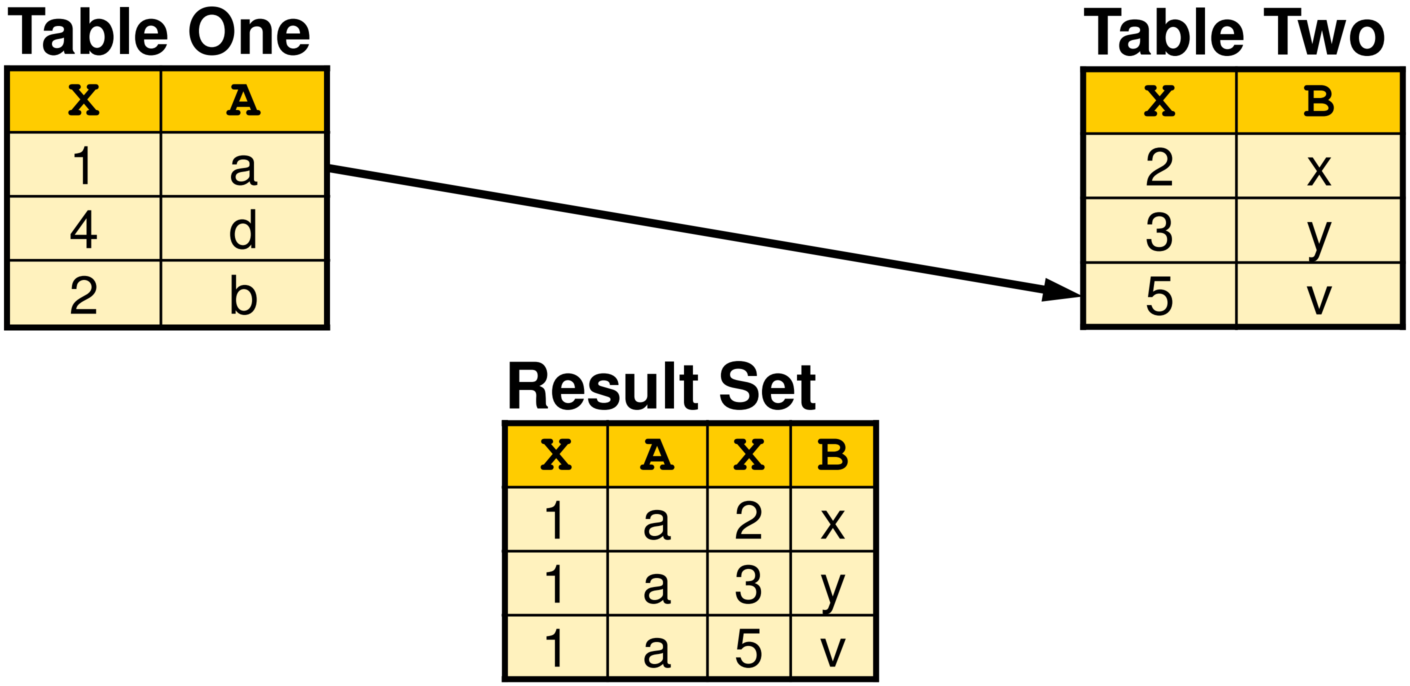


Table One

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

s105d01

...

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
2	b	5	v

s105d01

...

Cartesian Product

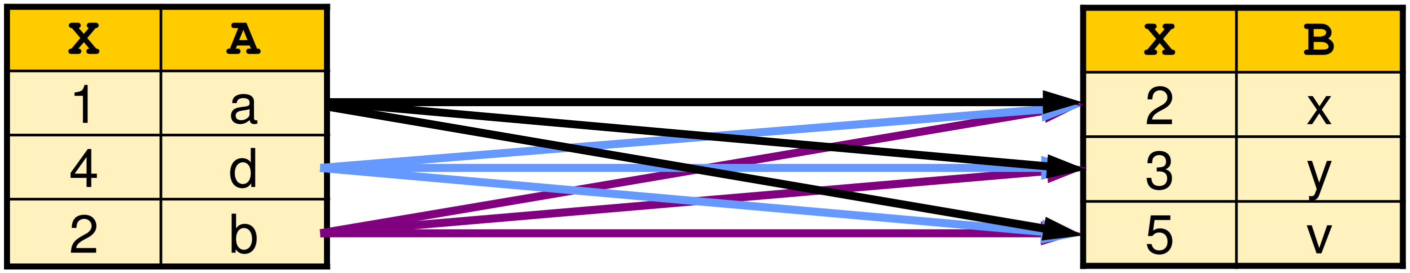


Table One

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
2	b	5	v

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
2	b	5	v

...

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
2	b	5	v

...

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
2	b	5	v

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 x 3 = 9

1,000 x 1,000 = 1,000,000

100,000 x 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;
```

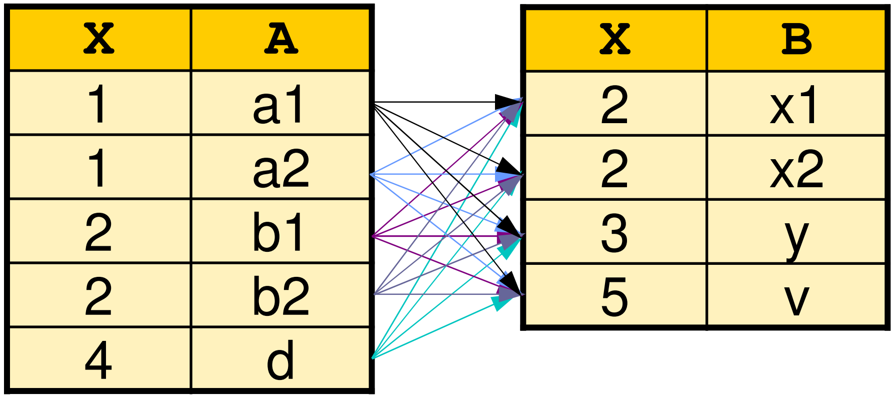


Table Three

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

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.

Inner Joins

Conceptually, when processing an inner join, PROC SQL does the following:

1. builds the Cartesian product of all the tables listed
2. applies the WHERE clause to limit the rows returned

Inner Joins: Cartesian Product Built

Table One

X	A
1	a
4	d
2	b

Table Two

X	B
2	x
3	y
5	v

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

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

s105d02

...

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

s105d02

...

Inner Joins: Results Are Returned

Table One

X	A
1	a
4	d
2	b

Table Two

X	B
2	x
3	y
5	v

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

X	A	X	B
2	b	2	x

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

s105d02

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

s105d03

Inner Joins

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

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

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

s105d04

...

Inner Joins

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

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

Results Set

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

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

s105d04

Poll

Quiz

Setup for the Poll

Run program **s105a02** and review the results to determine how many rows (observations) the DATA step MERGE statement produces in the output table.

Three

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

Four

X	B
2	x1
2	x2
3	y
5	v

```
data new;
   merge three (in=InThree)
         four (in=InFour);
   by x;
   if InThree and InFour;
run;

proc print data=new;
run;
```

s105a02

5.03 Multiple Choice Poll

How many rows (observations) result from the DATA step MERGE statement in program **s105a02**?

a. 4

b. 2

c. 6

d. 20

e. None of the above

5.03 Multiple Choice Poll – Correct Answer

How many rows (observations) result from the DATA step MERGE statement in program **s105a02**?

a. 4

[✓ b.] 2

c. 6

d. 20

e. None of the above

Three

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

Four

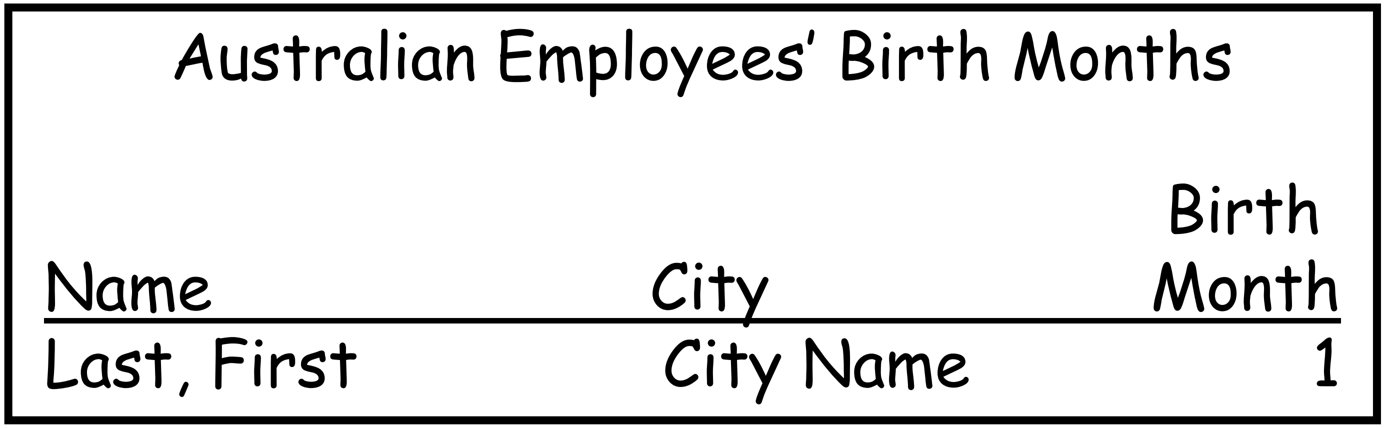
X	B
2	x1
2	x2
3	y
5	v

New

X	A	B
2	b1	x1
2	b2	x2

Business Scenario

Display the name, city, and birth month of all Australian employees. Here is a sketch of the desired report:



Australian Employees' Birth Months

Name	City	Birth Month
Last, First	City Name	1

Business Scenario

Considerations:

- **orion.Employee_Addresses** contains employee name, country, and city data.
- **orion.Payroll** contains employee birth dates.
- Both **orion.Employee_Addresses** and **orion.Payroll** contain **Employee_ID**.
- Names are stored in the **Employee_Name** column as Last, First.

Inner Joins

```
proc sql;
title "Australian Employees' Birth Months";
select Employee_Name as Name format=$25.,
       City format=$25.,
       month(Birth_Date) 'Birth Month' format=3.
   from orion.Employee_Payroll,
        orion.Employee_Addresses
   where Employee_Payroll.Employee_ID=
         Employee_Addresses.Employee_ID
         and Country='AU'
   order by 3,City, Employee_Name;
quit;
```

s105d05

Inner Joins

Partial PROC SQL Output

```
              Australian Employees Birthday Months
_______________________________________________________________
                                                          Birth
Name                                   City               Month

Aisbitt, Sandy                         Melbourne              1
Graham-Rowe, Jannene                   Melbourne              1
Hieds, Merle                           Melbourne              1
Sheedy, Sherie                         Melbourne              1
Simms, Doungkamol                      Melbourne              1
Tannous, Cos                           Melbourne              1
Body, Meera                            Sydney                 1
Clarkson, Sharryn                      Sydney                 1
Dawes, Wilson                          Sydney                 1
Rusli, Skev                            Sydney                 1
Glattback, Ellis                       Melbourne              2
Gromek, Gladys                         Melbourne              2
```

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.

Inner Join Alternate Syntax

```
proc sql;
title "Australian Employees' Birth Months";
select Employee_Name as Name format=$25.,
       City format=$25.,
       month(Birth_Date) 'Birth Month' format=3.
   from orion.Employee_Payroll
        inner join
        orion.Employee_Addresses
        on Employee_Payroll.Employee_ID=
           Employee_Addresses.Employee_ID
   where Country='AU'
   order by 3,City, Employee_Name;
quit;
```

s105d06

Poll

Quiz

5.04 Multiple Choice Poll

How many tables can be combined using a single inner join?

a. 2

b. 32

c. 256

d. 512

e. Limited only by my computer's resources

f. No limit

5.04 Multiple Choice Poll – Correct Answer

How many tables can be combined using a single inner join?

a. 2

b. 32

[✓ c.] 256

d. 512

e. Limited only by my computer’s resources

f. No limit

Question & Answer

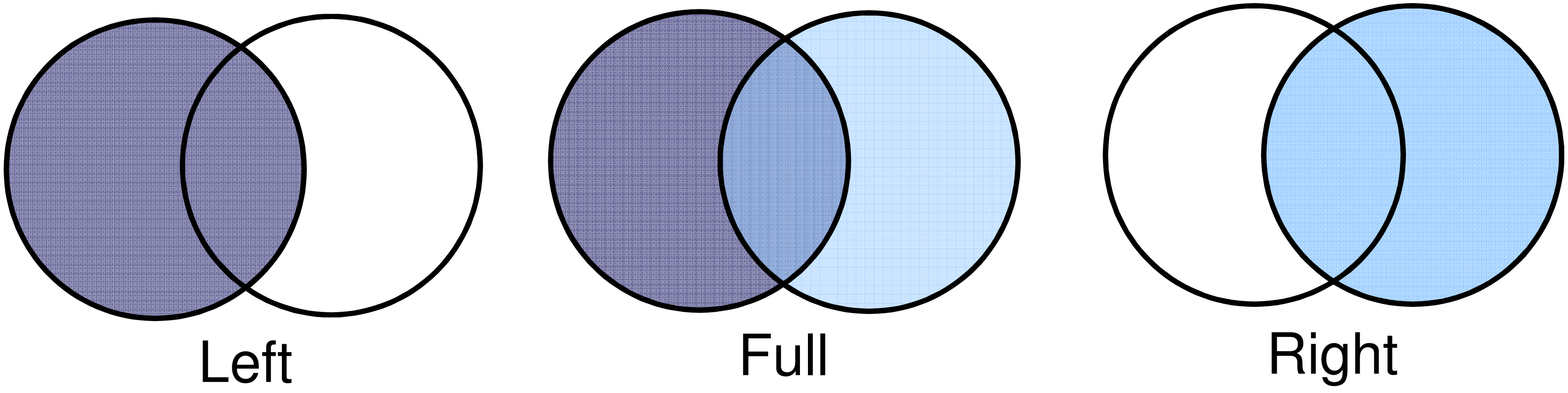
Outer Joins

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.



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	■ 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:

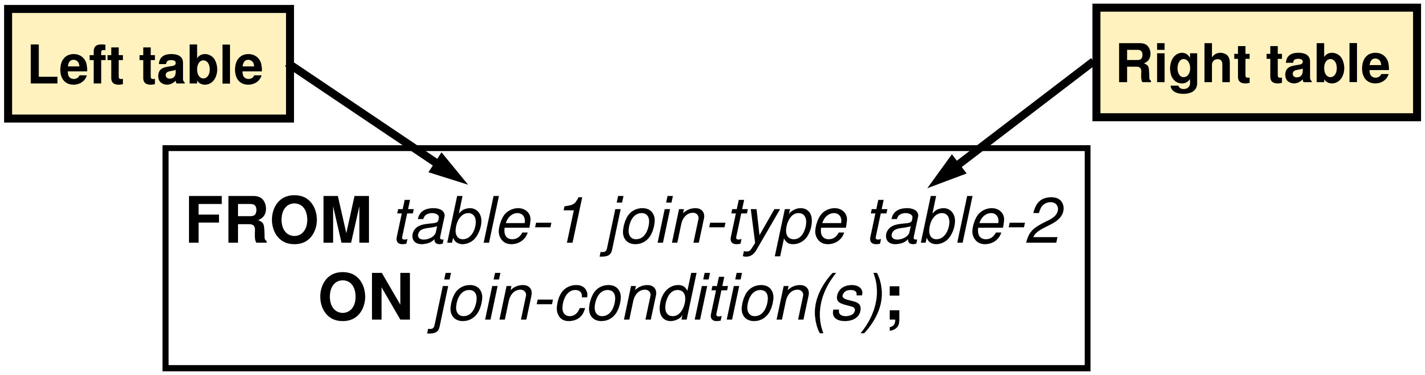
```
SELECT column-1 <, …column-n>
      FROM table-1
      LEFT|RIGHT|FULL JOIN
               table-2
      ON join-condition(s)
           <other clauses>;
```

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 table

Right table

FROM *table-1 join-type table-2*
ON *join-condition(s)***;**

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

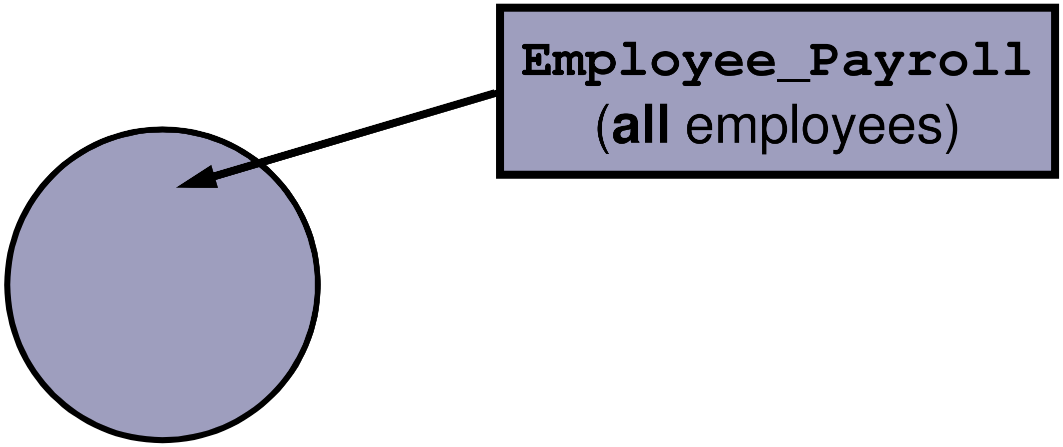
Business Scenario

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 **orion.Employee_Payroll** contains gender and marital status information.

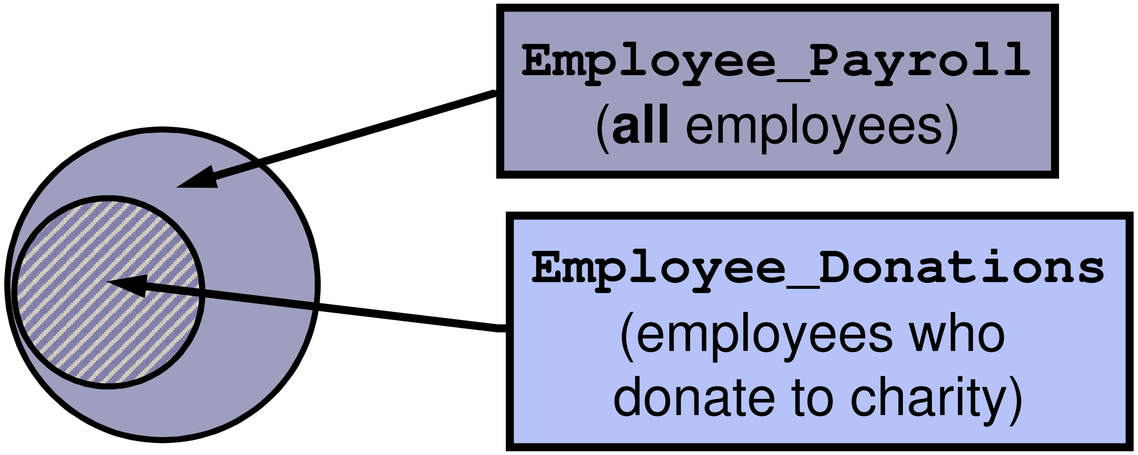


...

Business Scenario

Considerations:

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



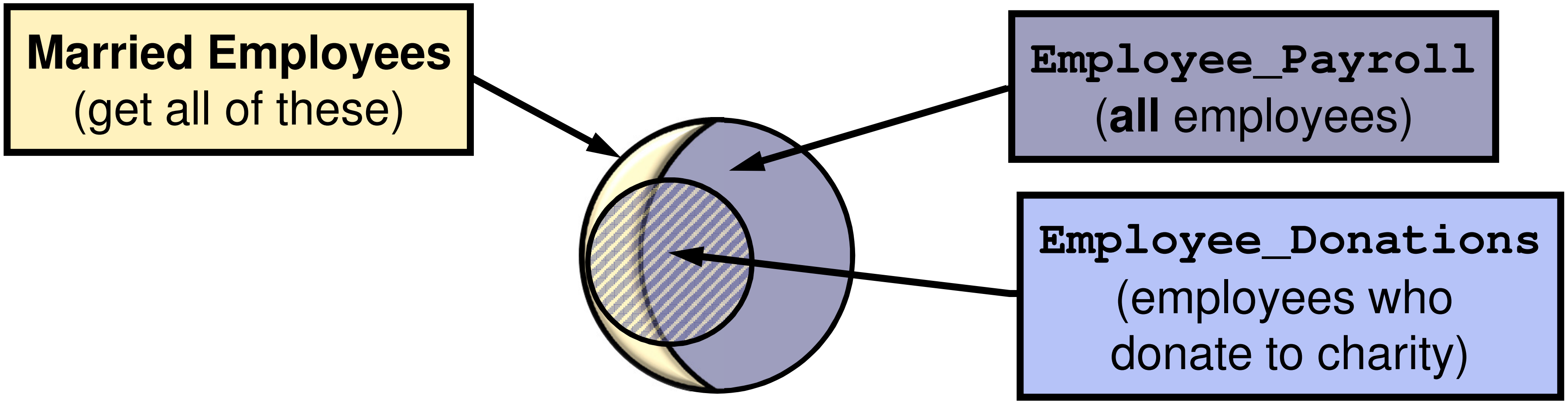
Employee_Payroll
(**all** employees)

Employee_Donations
(employees who donate to charity)

Business Scenario

Considerations:

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



Married Employees
(get all of these)

Employee_Payroll
(**all** employees)

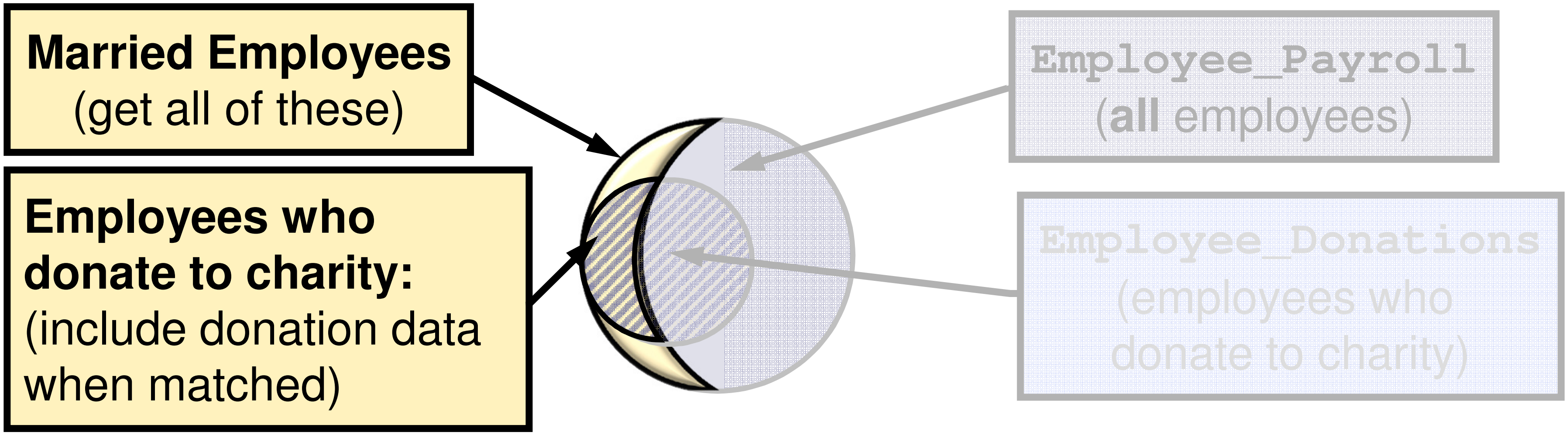
Employee_Donations
(employees who donate to charity)

...

Business Scenario

Considerations:

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



Married Employees
(get all of these)

Employees who donate to charity:
(include donation data when matched)

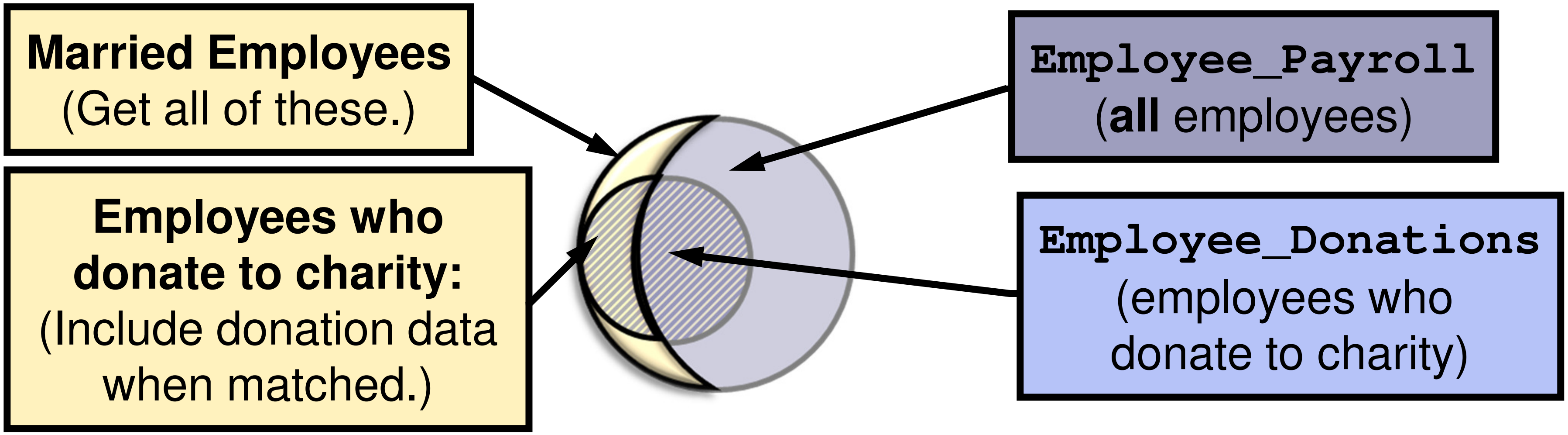
Employee_Payroll
(**all** employees)

Employee_Donations
(employees who donate to charity)

Business Scenario

Considerations:

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



Married Employees
(Get all of these.)

Employees who donate to charity:
(Include donation data when matched.)

`Employee_Payroll`
(**all** employees)

`Employee_Donations`
(employees who donate to charity)

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

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

s105d10

Outer Joins

Partial PROC SQL Output (Rows 203-215)

```
             Employee_
Employee_ID  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

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

s105d11

DATA Step Merge (Review)

A DATA step with MERGE and BY statements automatically overlays same-name columns.

Table One

X	A
1	a
4	d
2	b

Table Two

X	B
2	x
3	y
5	v

Table One must be sorted or indexed on column X before a merge can be performed.

```
data merged;
   merge one two;
   by x;
run;
proc print data=merged;
run;
```

Output

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

s105d12

...

SQL Join versus DATA Step Merge

SQL joins do not automatically overlay same-named columns.

Table One

X	A
1	a
4	d
2	b

Table Two

X	B
2	x
3	y
5	v

```
proc sql;
   select one.x, a, b
      from one full join two
      on one.x=two.x
;
quit;
```

Output

```
X   A   B
1   a
2   b   x
○       y
4   d
○       v
```

The COALESCE Function

The COALESCE function returns the value of the first non-missing argument.

General form of the COALESCE function:

COALESCE(*argument-1*,*argument-2*<, ...*argument-n*)

argument can be a constant, expression, or variable name. When all arguments are missing, COALESCE returns a missing value.

✎ All arguments must be of the same type (character or numeric).

SQL Join versus DATA Step Merge

You can use the COALESCE function to overlay columns.

Table One

X	A
1	a
4	d
2	b

Table Two

X	B
2	x
3	y
5	v

```
proc sql;
   select coalesce(one.x,two.x)
          as x,a,b
      from one full join two
      on one.x=two.x;
quit;
```

Output

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

s105d12

SQL Join versus DATA Step Merge

Key Points	SQL Join	DATA Step Merge
Explicit sorting of data before join/merge	Not required	Required
Same-named columns in join/merge expressions	Not required	Required
Equality in join or merge expressions	Not required	Required

Question & Answer

Exercise

This exercise reinforces the concepts discussed previously.

Chapter 5: SQL Joins

5.1: Introduction to SQL Joins

5.2: Complex SQL Joins

Objectives

- Create and use in-line views.
- Use in-line views and subqueries to simplify coding a complex query.

In-Line Views

In-line views are often useful when you build complex SQL queries.

An *inline view* is

- a temporary “virtual table” that exists only during query execution
- created by placing a query expression in a FROM clause where a table name would normally be used.

In-Line Views

An in-line view is a query expression (SELECT statement) that resides in a FROM clause. It acts as a virtual table, used in place of a physical table in a query.

```
proc sql;
   select *
      from
        (in-line view query expression)
quit;
```

Business Scenario

List all active Sales employees having annual salaries significantly (more than 5%) lower than the average salary for everyone with the same job title.

Considerations

First, you must calculate the average salaries for active employees in the Sales department, grouped by job title.

Next, you must match each employee to a GROUP-BY job title.

Finally, you must compare the employee's salary to the group's average to determine if it is more than 5% below the group average.

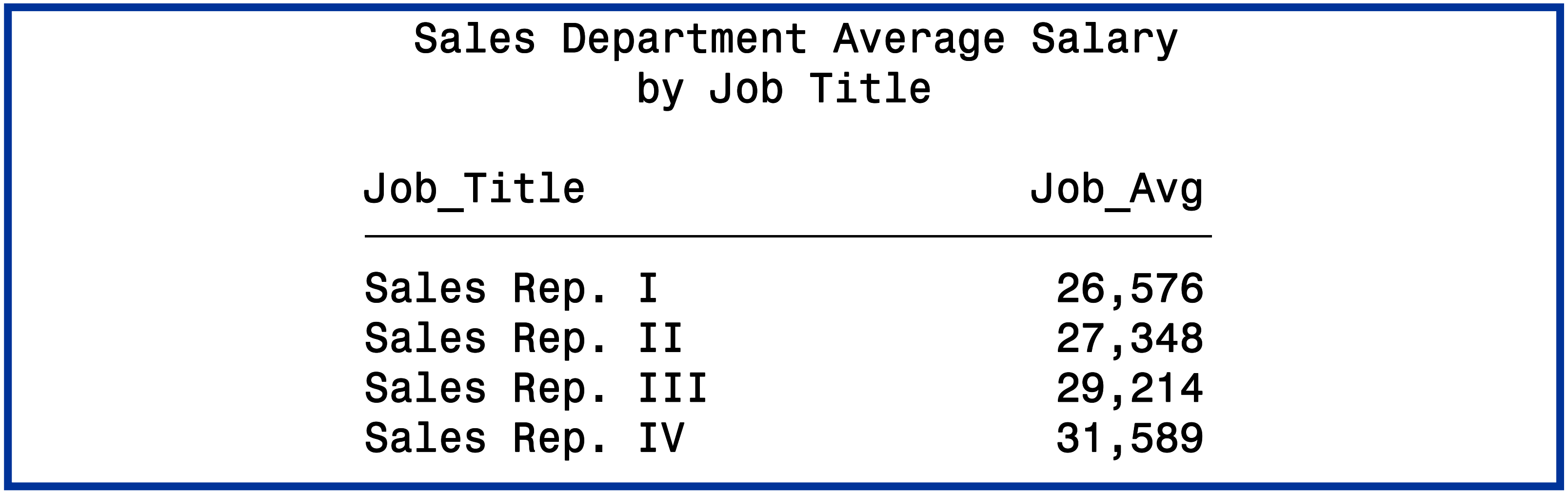
In-Line Views

Build a query to produce the aggregate averages.

```
proc sql;
title  'Sales Department Average Salary';
title2 'By Job Title';
   select Job_Title,
          avg(Salary) as Job_Avg
          format=comma7.
      from orion.Employee_payroll as p,
           orion.Employee_organization as o
      where p.Employee_ID=o.Employee_ID
            and not Employee_Term_Date
            and o.Department="Sales"
   group by Job_Title;
quit;
```

In-Line Views

PROC SQL Output



```
              Sales Department Average Salary
                       by Job Title

          Job_Title                           Job_Avg
          ____________________________________________

          Sales Rep. I                         26,576
          Sales Rep. II                        27,348
          Sales Rep. III                       29,214
          Sales Rep. IV                        31,589
```

In-Line Views

If you create a table from the results of the query, you can join this table and the **`orion.Employee_payroll`** table and subset the appropriate rows to get the answer. This adds unnecessary I/O.

Would it be useful to use only the query itself in place of a table?

In SQL, you can with an in-line view!

In-Line Views

Using a query in the FROM clause in place of a table causes the query output to be used as an in-line view.

```
proc sql;
title  'Employees with salaries less than';
title2 '95% of the average for their job';
   select Employee_Name, emp.Job_Title,
          Salary format=comma7., Job_Avg format=comma7.
      from (select Job_Title,
                   avg(Salary) as Job_Avg format=comma7.
               from orion.Employee_payroll as p,
                    orion.Employee_organization as o
               where p.Employee_ID=o.Employee_ID
                    and not Employee_Term_Date
                    and o.Department="Sales"
               group by Job_Title) as job,
           orion.Salesstaff as emp
       where emp.Job_Title=job.Job_Title
             and Salary < Job_Avg*.95
       order by Job_Title, Employee_Name;
```

s105d14

In-Line Views

PROC SQL Output

```
                  Employees with salaries less than
                  95% of the average for their job

                                                     Employee
                                                       Annual
Employee_Name          Employee Job Title              Salary   Job_Avg
-----------------------------------------------------------------------

Ould, Tulsidas         Sales Rep. I                    22,710    26,576
Polky, Asishana        Sales Rep. I                    25,110    26,576
Tilley, Kimiko         Sales Rep. I                    25,185    26,576
Voron, Tachaun         Sales Rep. I                    25,125    26,576
```

Question & Answer

Business Scenario

In 2003, Top Sports launched a premium line of sleeping bags called Expedition Zero, which was sold through Orion Star.

The CEO of Top Sports wants to send a letter of thanks to the manager of each employee who sold Expedition Zero sleeping bags in 2003, with a $50 reward certificate (in U.S. dollars) to be presented by the manager to the employee.

The Task:
Prepare a list of the managers' names and the cities in which they are located.

Planning the Complex Query

Step 1 Identify the employees who sold Expedition Zero merchandise in 2003.

Step 2 Find the employee identifier for the managers of these employees

Step 3 Obtain the managers' names and city information.

Complex Query: Step 1 Considerations

Step 1 Get employee IDs for employees who sold Expedition Zero merchandise in 2003.

Select the employee's identifier (**Employee_ID)** from the results of joining the **Order_Fact** and **Product_Dim** tables on **Product_ID**, where **Product_Name** contains Expedition Zero. Exclude Internet orders (**Employee_ID NE 99999999**).

Coding the Complex Query

Step 1

Write a query to obtain the employee ID of all employees who sold Expedition Zero merchandise in 2003.

```
select distinct Employee_ID
   from orion.Order_Fact as o,
        orion.Product_Dim as p
   where o.Product_ID=p.Product_ID
         and year(Order_Date)=2003
         and Product_Name contains
         'Expedition Zero'
         and Employee_ID ne 99999999;
```

s105d15

Coding the Complex Query

Step 1 PROC SQL Output

```
Employee ID
-----------

     120145
     120732
```

Complex Query: Step 2 Considerations

Step 2

Find the employee identifier for the managers of these employees.

Select the manager's identifier (**`Manager_ID`**) from the results of joining the **`Employee_Organization`** table with the first query's results on **`Employee_ID`**.

Poll ○ ✓ ○

Quiz

5.06 Multiple Choice Poll

To join the **`Employee_Organization`** table with the Step 1 query results, you use the query from Step 1 as which of the following?

a. an in-line view
b. a subquery

5.06 Multiple Choice Poll – Correct Answer

To join the **Employee_Organization** table with the Step 1 query results, you use the query from Step 1 as which of the following?

[✓ a.] an in-line view
b. a subquery

A query used in place of a physical table in a SELECT statement FROM clause is called an in-line view.

Coding the Complex Query

Step 2

Write a query to obtain the manager ID of the employee’s manager.

```
select Manager_ID
   from orion.Employee_Organization as o,
       (<Step 1 query results>) as ID
   where o.Employee_ID=ID.Employee_ID;
```

Employee_ID
120145
120732

Coding the Complex Query

Step 2

Write a query to obtain the manager ID of the employee’s manager.

```
select Manager_ID
   from orion.Employee_Organization as o,
      (select distinct Employee_ID
          from orion.Order_Fact as o,
                  orion.Product_Dim as p
          where o.Product_ID=p.Product_ID
               and year(Order_Date)=2003
               and Product_Name
               contains 'Expedition Zero'
               and Employee_ID ne 99999999)as ID
   where o.Employee_ID=ID.Employee_ID;
```

s105d16

Coding the Complex Query

Step 2 PROC SQL Output

```
Manager_ID
──────────
    120103
    120736
```

Complex Query: Step 3 Considerations

Step 3

Find the managers’ names and cities.

Select the employee’s name (**`Employee_Name`**) and **`City`** from the **`Employee_Addresses`** table, where **`Employee_ID`** matches **`Manager_ID`**
in the results of the previous query.

Poll
Quiz

5.07 Poll

Is it possible to use the entire query in Step 2 as a subquery?

○ Yes

○ No

5.07 Poll – Correct Answer

Is it possible to use the entire query in Step 2 as a subquery?

[✓ ○ Yes]

○ No

A subquery can return values for multiple rows, but must return values for only one column. When submitted on its own, the query in Step 2 returns two rows and only one column, so it can be used as a non-correlated subquery.

Coding the Complex Query

Step 3 Write a query to obtain the managers’ names and city information.

```
proc sql;
select Employee_Name format=$25. as Name, City
   from orion.Employee_Addresses
   where Employee_ID in
       (<Step 2 query results>);
```

Manager_ID
120145
120732

Coding the Complex Query

```
proc sql;
select Employee_Name format=$25. as Name
     , City
   from orion.Employee_Addresses
   where Employee_ID in
       (select Manager_ID
           from orion.Employee_Organization as o,
           (select distinct Employee_ID
               from orion.Order_Fact as o,
                     orion.Product_Dim as p
               where o.Product_ID=p.Product_ID
               and year(Order_Date)=2003
               and Product_Name contains
                    'Expedition Zero'
               and Employee_ID ne 99999999) as ID
            where o.Employee_ID=ID.Employee_ID);
```

Step 3

Coding the Complex Query

Step 3 PROC SQL Output

```
Name                      City

______________________________________

Dawes, Wilson             Sydney
Kiemle, Parie             Miami-Dade
```

Coding the Complex Query

You can also solve this problem using a multiway join.

```
proc sql;
   select distinct Employee_Name format=$25. as Name, City
      from orion.Order_Fact as of,
           orion.Product_Dim as pd,
           orion.Employee_Organization as eo,
           orion.Employee_Addresses as ea
      where of.Product_ID=pd.Product_ID
            and of.Employee_ID=eo.Employee_ID
            and ea.Employee_ID=eo.Manager_ID
            and Product_Name contains 'Expedition Zero'
            and year(Order_Date)=2003
            and eo.Employee_ID ne 99999999
;
quit;
```

s105d18

Chapter Review

1. How many rows are returned by the following query?

```
proc sql;
   select *
      from
      table1,table2;
quit;
```

Table1

X	A
1	a
3	d
2	b

Table2

X	B
2	x
1	y
3	v

Chapter Review Answers

1. How many rows are returned by the following query?

```
proc sql;
   select *
      from
      table1,table2;
quit;
```

Table1

X	A
1	a
3	d
2	b

Table2

X	B
2	x
1	y
3	v

This query produces a Cartesian product.

Nine rows will be returned.

Chapter Review

2. Which of the following statements describes an advantage of using a PROC SQL view?

   a. Views often save space, because a view is usually quite small compared with the data that it accesses.

   b. Views can provide users a simpler alternative to frequently retrieving and submitting query code to produce identical results.

   c. Views hide complex query details from users.

   d. All of the above

Chapter Review Answers

2. Which of the following statements describes an advantage of using a PROC SQL view?

   a. Views often save space, because a view is usually quite small compared with the data that it accesses.

   b. Views can provide users a simpler alternative to frequently retrieving and submitting query code to produce identical results.

   c. Views hide complex query details from users.

   [✓ d.] All of the above

Chapter Review

3. Outer and Inner Joins:
   a. An **outer join** can operate on a maximum of ___ tables simultaneously.
   b. An **inner join** can operate on a maximum of ___ tables simultaneously.

Chapter Review Answers

3. Outer and Inner Joins:

   a. An **outer join** can operate on a maximum of __2__ tables simultaneously.

   b. An **inner join** can operate on a maximum of __256 tables simultaneously.

Chapter Review

4. True or False:
   An in-line view can be used on a WHERE or HAVING clause and can return many rows of data, but must return only one column.

Chapter Review Answers

4. True or False:
   An in-line view can be used on a WHERE or HAVING clause and can return many rows of data, but must return only one column.

   False

   An in-line view is a query used in the FROM clause in place of a table. An in-line view can return any number of rows or columns.