

Chapter 3: Combining Tables Horizontally Using Proc SQL

Dennis Eggett

Stat 224 Fall 2015

Cartesian Products

When joining tables together, SAS creates a Cartesian Product, a set containing each row of the first table combined with every row of the second table.

SAS Code

```
data table1;  
input A B $;  
datalines;  
1 a  
2 b  
3 c  
4 a  
5 b  
6 a  
;
```

SAS Code

```
data table2;  
input A C $;  
datalines;  
1 d  
2 e  
3 g  
4 f  
5 d  
6 g  
;
```

SAS Code

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

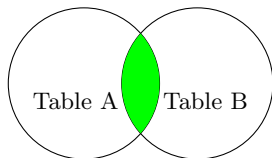
SAS Output

1	a	1	d
1	a	2	e
1	a	3	g
1	a	4	f
1	a	5	d
1	a	6	g
:	:	:	:
:	:	:	:

Note that the full query result contains 36 observations (the number of observations from each dataset multiplied together). A WHERE clause is essential to narrow the Cartesian Product down to the desired query.

Inner Products

An inner join combines two datasets and selects only those rows from the first table that match those of the second.



SAS Code

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

The syntax includes utilizing a WHERE clause to limit the Cartesian Plane to observations where variable values from each dataset match.

Customizations of Inner Joins

Inner joins are customizable in many ways:

- Eliminate duplicate columns
 - `select one.*, b`
- Rename a column using an alias
 - `select one.x as ID`
- Create an alias for the table so a shorter name can be referenced
 - `from sasuser.staffmaster as s, sasuser.payrollmaster as p`
`where s.empid=p.empid;`
- Combine variables into a single variable
 - `select substr(lname,1,1)|| ' ' || fname as name`
- Use summary functions
 - `avg(Salary) as LaborCost`

Multiple Values

Note that if you have duplicate values in the matching variable, inner joins will display all combinations of merges between the two datasets

SAS Code

```
data table1;  
input A B $;  
datalines;  
1 a  
2 b  
2 c  
4 a  
;
```

SAS Code

```
data table2;  
input A C $;  
datalines;  
1 g  
1 f  
2 d  
2 s  
3 a  
;
```

SAS Code

```
proc sql;  
    select table1.*, C  
    from table1,table2  
    where table1.A=table2.A;  
quit;
```

SAS Output

	A	B	C
1	a		g
1	a		f
2	b		d
2	c		d
2	b		s
2	c		s

Outer Joins

Outer joins contain the data from inner joins, plus mismatched rows from one or more of the tables.

- Left outer joins combine all rows from the left table and matching rows from the right table.
- Right outer joins combine all rows from the right table and matching rows from the first table.
- Full outer joins combine matching and non-matching rows from both tables.

Outer Join Examples

SAS Code

```
#Left Join
proc sql;
    select *
    from one
    left join
    two
    on one.x=two.x;
```

```
#Right Join
proc sql;
    select *
    from one
    right join
    two
    on one.x=two.x;
```

SAS Code

```
#Full Join
proc sql;
    select *
    from one
    full join
    two
    on one.x=two.x;
```

```
#Alternate Inner Join Syntax
proc sql;
    select *
    from one
    inner join
    two
    on one.x=two.x;
```

Match-Merge vs. Joins

- If all values match, SQL inner joins produce the same results as match-merging.
- When only some of the values match, left, right, or full joins may be used.
- The COALESCE function allows you to overlay the columns as does match-merging.

In-line View

In general, a view is a saved query. It can be referred to in SAS and SQL as if it were a table, but whenever it's referenced it is recreated from its underlying tables.

An in-line view is a virtual table only available during the execution of an SQL statement. It is written just like a subquery, but is in the FROM clause instead of WHERE or SELECT.

SAS Code

```
... from ( select flights.number, flights.date,  
            boarded/capacity as full, delay from flights, delays  
            where flights.number = delays.number and  
            flights.date = delays.date );
```