SQL - Fusionar conjuntos de datos

en R y SAS

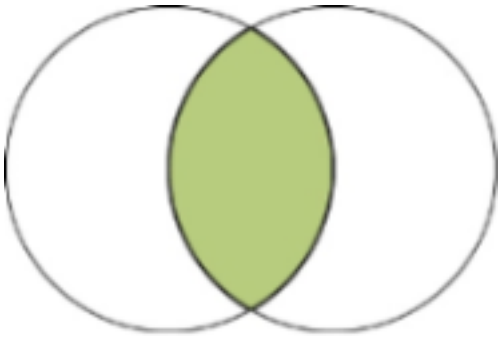
César Mignoni

Introducción

Las formas, que se ha vistos, de ralizar consultas con SQL, es utilizando una sola tabla. Sin embargo, a menudo es necesario obtener datos de tablas independientes. Cuando especifican varias *tablas* de consulta en la cláusula FROM, SQL las procesa para formar un conjunto de datos. El *tablas* resultante contiene datos de cada *tablas* fuente. Estas consultas se conocen como *combinaciones*.

Conceptualmente, cuando se especifican dos conjuntos de datos, SQL hace coincidir cada fila del primero con todas las filas del segundo para producir un conjunto de datos interno o intermedio conocido como el *producto cartesiano*. El producto cartesiano de grandes conjuntos de datos puede ser enorme, pero normalmente se obtienen subconjuntos de datos declarando el tipo de combinación. Hay dos tipos de combinaciones:

- **Combinación interna**: devuelve una tabla cuyas filas son todas las filas de la tabla de la isquierda que tienen una o más filas coincidentes en la tabla de la derecha. Busca coincidencias entre las dos tablas, en función a una columna que tienen en común. De tal modo que sólo la intersección se mostrará en los resultados.



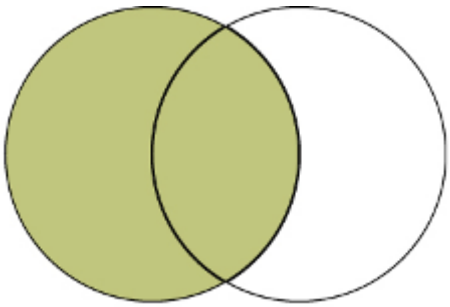
- **Combinaciones externas** son combinaciones internas que se aumentan con filas que no coinciden con ninguna fila de la otra tabla en la combinación. Hay tres tipos de combinaciones externas: izquierda, derecha y completa.

La tarea central es unir tablas para obtener un detalle sobre la consulta de tablas individuales.

Unión izquierda

Es un tipo de *combinación externa* en la que la tabla de resultados incluye todas las observaciones de la tabla izquierda independientemente de que se encuentre una coincidencia en la tabla especificada a la derecha.

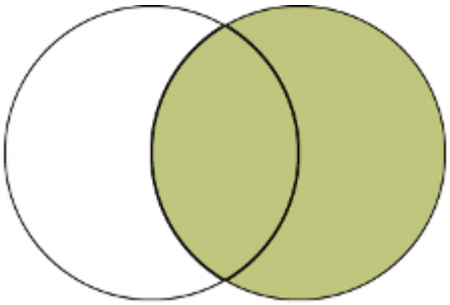
El siguiente diagrama representa Una combinación izquierda entre dos tablas.



Unión derecha

Es idéntica a la *unión izquierda*, excepto que la tabla de resultados incluye todas las observaciones de la tabla derecha, si se encuentra o no una coincidencia para ellas en la tabla de la izquierda.

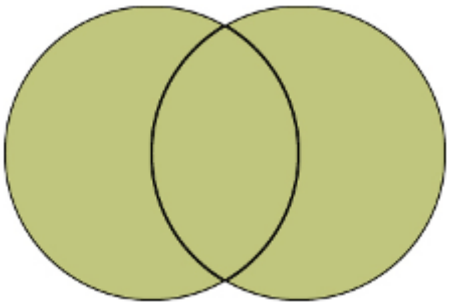
Una combinación derecha entre dos tablas se puede representar gráficamente como se muestra en el siguiente diagrama.



Unión completa

En este tipo de unión la tabla resultante incluye todas las observaciones de ambas tablas para las cuales haya coincidencia , más las filas de cada tabla que no coinciden con ninguna fila en la otra tabla.

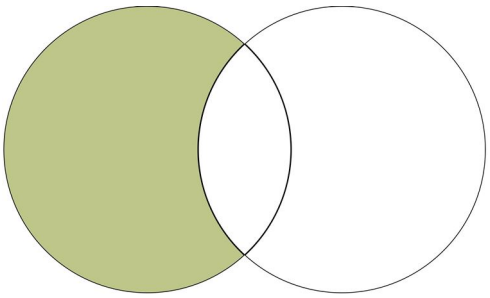
La representación visual de la unión externa completa se muestra en el siguiente diagrama.



Otras variantes

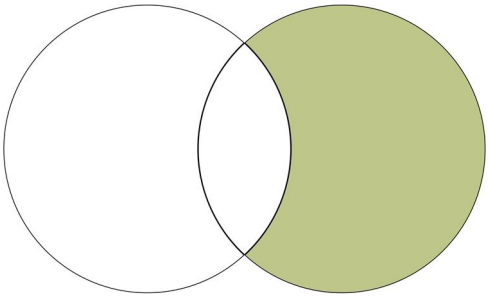
Exclución interna Izquierda

Esta consulta devolverá todos los registros de la tabla de la izquierda que no coincidancon ningún registro de la tabla de la derecha. La representación visual de esta unión semuestra en el siguiente diagrama:



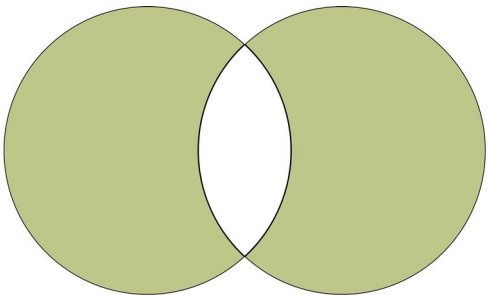
Exclución interna Derecha

Esta consulta devolverá todos los registros de la tabla de la derecha que no coincidan conningún registro de la tabla de la izquierda. La representación visual de esta unión se muestra en el siguiente diagrama:



Exclución interna Exterior

Esta consulta devolverá todos los registros de la tabla de la izquierda y todos los registros de la tabla de la derecha que no coincidan. La representación visual de esta unión se muestra en el siguiente diagrama:



Fusión de conjuntos de datos (interna y externa) en R

Se puede combinar dos conjuntos de datos en **R** usando la función *merge()*. Como requiseto, los conjuntos de datos deben tener los mismos nombres en las columna que se utilizan para realizar la unión. La función *merge()* en **R** es similar a la operación de unión de tablas en base de datos con SQL. Los diferentes argumentos para *merge()* permiten realizar combinaciones internas, así como combinaciones externas izquierda, derecha y completa.

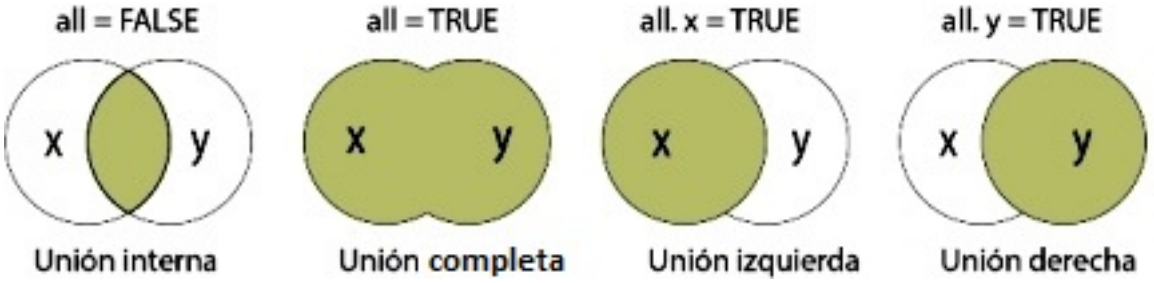
Los argumentos de la función *merge()* son:

- **"x"** : conjunto de datos 1.

- **"y"** : conjunto de datos 2.
- **"by"** , **"x"** , **"by.y"** : los nombres de las columnas que son comunes a **"x"** e **"y"** . El valor predeterminado es usar las columnas con nombres comunes entre los dos conjuntos de datos.
- **"all,all.x,all.y"** : valores lógicos que especifican el tipo de fusión. El valor predeterminado es "all=FALSE" (lo que significa que solo se devuelven las filas coincidentes).

Definición de argmentos para los diferentes tipos de fusión:

- ***Unión interna*** : para mantener solo las filas que coinciden con los conjuntos de datos, especifique el argumento *"all=FALSE"*.
- ***Unión completa***: para mantener todas las filas de ambos conjuntos de datos, se especifica *"all=TRUE"*.
- ***Unión izquierda***: para incluir todas las filas del conjunto de datos de la izquierda **"x"** y solo aquellas filas del de la derecha **"y"** que coincidan, se especifica *"all.x=TRUE"*.
- ***Unión derecha***: para incluir todas las filas del conjunto de datos de la derecha **"y"** y solo las filas del de la izquierda **"x"** que coinciden, se especifica *"all.y=TRUE"* .



Ejemplos prácticos.

Primero se definien dos conjuntos de datos que serán la base para los ejemplos posteriores. Por un lado, la tabla *Empleados* que almacena una lista de empleados y el id del departamento al quepertenecen:

Empleados

```
##   departamentoID      Nombre
## 1             31       Sales
## 2             33 Engineering
## 3             34    Clerical
## 4             35   Marketing
```

Y por otro lado, la tabla *Departamentos* con la lista de departamentos que existen en la empresa.

Departamentos

```
##       Nombre departamentoID
## 1   Rafferty             31
## 2      Jones             33
## 3 Heisenberg             33
## 4   Robinson             34
## 5      Smith             34
## 6   Williams             NA
```

Unión interna

Por ejemplo, si se quiere listar los empleados e indicar el nombre del departamento al que pertenecen, se reliza lo siguiente:

```
df <- merge(x = df1, y = df2, by = "departamentoID")
```

Con este comando el resultado es:

```
##   departamentoID    Nombre.x   Nombre.y
## 1             31       Sales   Rafferty
## 2             33 Engineering      Jones
## 3             33 Engineering Heisenberg
## 4             34    Clerical   Robinson
## 5             34    Clerical      Smith
```

Y apartir de aquí podemos notar lo siguiente:

El empleado "Williams" no aparece en los resultados, ya que no pertenece a ningún departamento existente.

También hay que tener en cuenta que, en los resultados se ven 3 columnas. Las 2 primeras corresponden a la tabla Empleados y la última a Departamentos.

Unión completa:

```
df <- merge(x = df1, y = df2, by = "departamentoID", all = TRUE)
```

El conjunto de datos" df" resultante será:

```
##   departamentoID    Nombre.x   Nombre.y
## 1             31       Sales   Rafferty
## 2             33 Engineering      Jones
## 3             33 Engineering Heisenberg
## 4             34    Clerical   Robinson
## 5             34    Clerical      Smith
## 6             35   Marketing       <NA>
## 7             NA        <NA>   Williams
```

Unión izquierda: devuelve todas las filas de la tabla izquierda y cualquier fila con teclas coincidentes de la tabla derecha.

```
df <- merge(x = df1, y = df2, by = "departamentoID", all.x = TRUE)
```

El conjunto de datos" df" resultante será:

```
##   departamentoID    Nombre.x   Nombre.y
## 1             31       Sales   Rafferty
## 2             33 Engineering      Jones
## 3             33 Engineering Heisenberg
## 4             34    Clerical   Robinson
## 5             34    Clerical      Smith
## 6             35   Marketing       <NA>
```

Unión derecha: devuelve todas las filas de la tabla derecha y cualquier fila con teclas coincidentes de la tabla izquierda.

```r
df <- merge(x = df1, y = df2, by = "departamentoID", all.y = TRUE)
```

El conjunto de datos" df" resultante será:

```
##   departamentoID    Nombre.x   Nombre.y
## 1             31       Sales   Rafferty
## 2             33 Engineering      Jones
## 3             33 Engineering Heisenberg
## 4             34    Clerical   Robinson
## 5             34    Clerical      Smith
## 6             NA        <NA>   Williams
```

Combinación cruzada: una combinación cruzada (también conocida como combinación cartesiana) da como resultado que cada fila de una tabla se una a cada fila de otra tabla.

```r
df<-merge(x = df1, y = df2, by = NULL)
```

El conjunto de datos" df" resultante será:

```
##    departamentoID.x    Nombre.x   Nombre.y departamentoID.y
## 1                31       Sales   Rafferty               31
## 2                33 Engineering   Rafferty               31
## 3                34    Clerical   Rafferty               31
## 4                35   Marketing   Rafferty               31
## 5                31       Sales      Jones               33
## 6                33 Engineering      Jones               33
## 7                34    Clerical      Jones               33
## 8                35   Marketing      Jones               33
## 9                31       Sales Heisenberg               33
## 10               33 Engineering Heisenberg               33
## 11               34    Clerical Heisenberg               33
## 12               35   Marketing Heisenberg               33
## 13               31       Sales   Robinson               34
## 14               33 Engineering   Robinson               34
## 15               34    Clerical   Robinson               34
## 16               35   Marketing   Robinson               34
## 17               31       Sales      Smith               34
## 18               33 Engineering      Smith               34
## 19               34    Clerical      Smith               34
## 20               35   Marketing      Smith               34
## 21               31       Sales   Williams               NA
## 22               33 Engineering   Williams               NA
## 23               34    Clerical   Williams               NA
## 24               35   Marketing   Williams               NA
```

Fusión de conjuntos de datos (interna y externa) en SAS

PROC SQL implementa el lenguaje de consulta estándar y permite al usuario la unión de *dataset* mediante consultas de combinación.

Como se ha descripto en el material "*Lenguaje de Consulta Estructurado en SAS y R*", en **PROC SQL** la cláusula *FROM* se utiliza en una expresión de consulta para especificar el/los

conjunto(s) de datos fuente, y que se combinan para producir el resultado de la unión.

Además de los diversos tipos de combinaciones (internas y externas) que se describen, los ejemplos que se incluyen muestran la igualdad entre los valores de columna provenientes de las tablas que se están uniendo; comparación entre valores calculados; etc. La cláusula *WHERE* o la cláusula *ON* contienen las condiciones bajo las cuales algunas filas son guardadas o eliminadas en la tabla de resultados. *WHERE* se usa para seleccionar filas de uniones internas. *ON* se utiliza para seleccionar filas de uniones internas o externas.

Ejemplos

Unión interna

```
proc sql;
   title 'Oil Production/Reserves of Countries';
   select p.country, barrelsperday 'Production',
          barrels 'Reserves'
      from oilprod p, oilrsrvs r
      where p.country = r.country
      order by barrelsperday desc;
quit;
```

Unión externa izquierda

```
proc sql;
   title 'Coordinates of Capital Cities';
   select Capital format=$20.,
          Name 'Country' format=$20.,
          Latitude, Longitude
      from countries a left join
           worldcitycoords b
           on a.Capital = b.City and
              a.Name = b.Country
      order by Capital;
quit;
```

Unión externa derecha

```
proc sql;
   title 'Populations of Capitals Only';
   select City format=$20.,
          Country 'Country' format=$20.,
          Population
      from countries right join
           worldcitycoords
           on Capital = City and
              Name = Country
      order by City;
quit;
```

Unión externa completa

```
proc sql;
   title 'Populations/Coordinates of World Cities';
   select City '#City#(WORLDCITYCOORDS)' format=$20.,
```

```
          Capital '#Capital#(COUNTRIES)' format=$20.,
          Population, Latitude, Longitude
      from countries full join
           worldcitycoords
           on Capital = City and
              Name = Country;
quit;
```

Borrar los títulos

```
title;
```