

Base de datos desde cero

Clase 6 - 19/10/2022

MDL - Lenguaje de Manipulación de Datos - Inner Join - Repaso

Junta entre dos tablas

32	RAMIREZ GERARDO RAMON	3	Posadas
33	GARCIA JUAN DOMINGO	3	Posadas
34	CESPEDES RICARDO OMAR	2	Oberá
35	OLIVERA JOSE LUIS	3	Posadas

Logramos una junta entre dos tablas, vinculándolas por medio de la primary key de una tabla con la foreign key de la otra tabla.

En el caso propuesto como ejemplo arriba, la primary key de la tabla localidad se debería vincular con la foreign key de la tabla Alumno formada por el campo idlocalidad.

dni	nombreyapellido	idlocalidad
19	TSCHIRSCH LILI MARGARITA	3
20	LOPEZ FABIANA NOEMI	2
21	MORAD DORA ARGENTINA	3
22	ROMERO JULIA GABRIELA	2

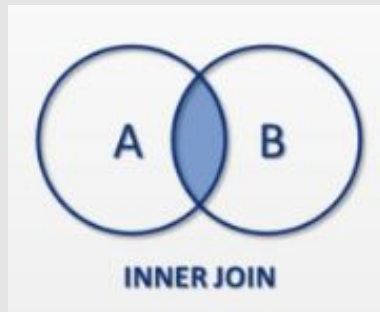
IDLOCALIDAD	LOCALIDAD
1	25 de Mayo
2	Oberá
3	Posadas

Columns
Indexes
Foreign Keys
FK_GenXAlu
FK_LocaXAlu

MDL - Lenguaje de Manipulación de Datos - Inner Join - Repaso

Si tenemos dos tablas A y B, el Inner Join trae los registros donde el/los atributos igualados en el on responden a dicha igualdad, o sea, traerá todos los registros de A donde el atributo indicado sea igual al atributo de B que se haya igualado en el on, y viceversa.

Si lo vemos gráficamente, podemos decir que se vería así:



Siempre teniendo en cuenta que se iguala uno o más atributos.

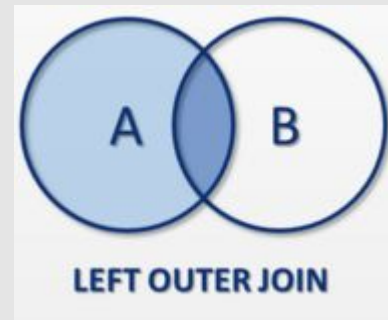
MDL - Lenguaje de Manipulación de Datos - Left Join

```
select count(*)  
  from alumno a left join localidad l  
    on a.IDLOCALIDAD = l.IDLOCALIDAD;
```

El Left Join trae todos los registros de la tabla que se encuentra a la izquierda y solo los registros que se igualan por el registro del join de la tabla de la derecha.

Por ejemplo, en la consulta de arriba traería todos los registros de la tabla alumno y solo los registros de la tabla localidad que igualen los idlocalidad de la tabla alumno.

Por eso esta consulta traerá el total de los alumnos, o sea 1056.



MDL - Lenguaje de Manipulación de Datos - Left Join

```
select count(*)  
  from alumno as a inner join localidad as l  
    on a.IDLOCALIDAD = l.idlocalidad;
```

```
select count(*)  
  from alumno as a left join localidad as l  
    on a.IDLOCALIDAD = l.idlocalidad;
```

```
select count(*)  
  from alumno as a right join localidad as l  
    on a.IDLOCALIDAD = l.idlocalidad;
```

Inner Join

Result Grid	
	count(*)
▶	1046

Left Join

Result Grid	
	count(*)
▶	1056

Right Join

Result Grid	
	count(*)
▶	1048

MDL - Lenguaje de Manipulación de Datos - Right join

Exactamente el mismo comportamiento que el Left join, pero en este caso, se traen todos los registros de la tabla de la derecha y solo los que igualan los campos unidos por el on.

Se puede intercalar Inner Join, Left Join y Right Join sin problemas.
Ejemplo:

```
select count(*)
  from alumno a inner join localidad l
    on a.IDLOCALIDAD = l.IDLOCALIDAD
 left join exaxmatxalu ex
    on a.dni = ex.dni
```

Result Grid	
	count(*)
▶	1195

MDL - Lenguaje de Manipulación de Datos - Full Join

MySQL no soporta Full Join

```
select count(*) from  
(  
  select dni  
    from alumno as a left outer join localidad as l  
      on a.IDLOCALIDAD = l.idlocalidad  
  union  
  select dni  
    from alumno as a right outer join localidad as l  
      on a.IDLOCALIDAD = l.idlocalidad  
) as Origen
```

Result Grid	
	count(*)
▶	1057

MDL - Lenguaje de Manipulación de Datos - Group by

```
select count(*)  
  from alumno as A
```

Cuenta la cantidad de Alumnos

```
select L.LOCALIDAD, count(*)  
  from alumno as A  
    inner join localidad as L  
      on A.IDLOCALIDAD = L.IDLOCALIDAD  
   group by L.Localidad  
  order by L.LOCALIDAD
```

Cuenta la cantidad de Alumnos pero agrupando por la localidad a la que pertenece

MDL - Lenguaje de Manipulación de Datos - Group By

Localidad	Dni
25 de Mayo	316
25 de Mayo	318
25 de Mayo	320
Oberá	12
Oberá	14
Posadas	59
Posadas	61
Posadas	63
Posadas	65

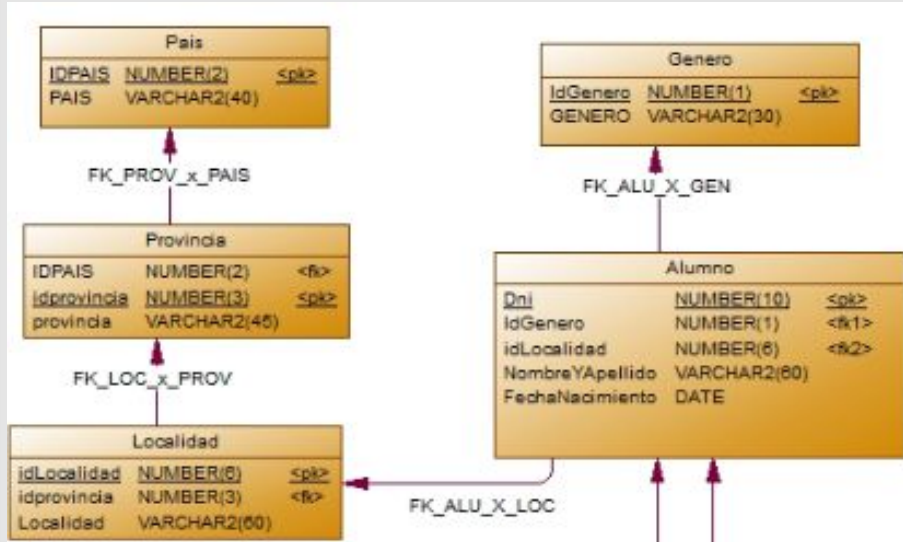
3

2

4

```
select L.LOCALIDAD, count(*)  
  from alumno as A  
    inner join localidad as L  
      on A.IDLOCALIDAD = L.IDLOCALIDAD  
 group by L.Localidad  
 order by L.LOCALIDAD
```

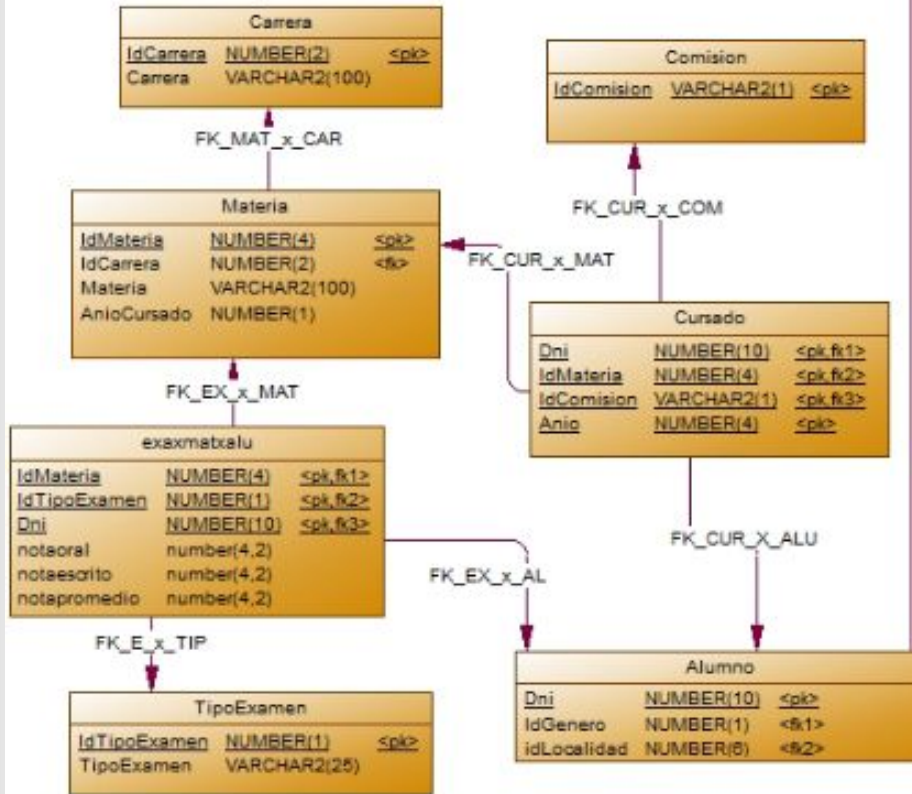
Escenario



Los Alumnos tienen un género asociado, la relación es un género, muchos alumnos.

Los Alumnos son de una Localidad, la Localidades de una Provincia y la Provincia es de un País.

Escenario



Los Alumnos cursan Materias, en un año, en una comisión.
Las Materias pertenecen a una Carrera, los exámenes son de un tipo en particular: parcial, recuperatorio, final

MDL - Lenguaje de Manipulación de Datos - Group by

	localidad	Cantidad
►	25 de Mayo	13
	Oberá	37
	Posadas	596
	Leandro N Alem	47
	Candelaria	84
	Garupá	62
	Santa Ana	5
	San Pedro	29
	San Ignacio	20
	Roca	18
	Resistencia	17
	San Juan	8

```
select loc.localidad, count(*) Cantidad
  from alumno as alu
    inner join localidad as loc
      on alu.idlocalidad = loc.idlocalidad
 group by loc.localidad;
```

Cantidad de alumnos por localidad

MDL - Lenguaje de Manipulación de Datos - Group by

PROVINCIA	localidad	Cantidad
Chaco	Resistencia	17
Chaco	Saenz Peña	8
Misiones	25 de Mayo	13
Misiones	Oberá	37
Misiones	Posadas	596
Misiones	Leandro N Alem	47
Misiones	Candelaria	84
Misiones	Garupá	62

Cantidad de alumnos por provincia y por localidad.

```
select pro.PROVINCIA, loc.localidad, count(*) Cantidad
  from alumno as alu
    inner join localidad as loc
      on alu.idlocalidad = loc.idlocalidad
    inner join provincia pro on pro.IDPROVINCIA = loc.IDPROVINCIA
 group by pro.PROVINCIA, loc.localidad;
```


MDL - Lenguaje de Manipulación de Datos - Group by con Order by

PROVINCIA	localidad	Cantidad
Altos del Paraná	Ciudad del Este	28
Chaco	Resistencia	17
Chaco	Saenz Peña	8
Guairá	Villarrica	12
Misiones	25 de Mayo	13
Misiones	Candelaria	84
Misiones	Garupá	62
Misiones	Leandro N Alem	47

Ordenando por los campos que queremos

```
select pro.PROVINCIA, loc.localidad, count(*) Cantidad
  from alumno as alu
    inner join localidad as loc
      on alu.idlocalidad = loc.idlocalidad
    inner join provincia pro on pro.IDPROVINCIA = loc.IDPROVINCIA
group by pro.PROVINCIA, loc.localidad
order by pro.provincia, loc.localidad;
```


MDL - Lenguaje de Manipulación de Datos - Group by con Having

PROVINCIA	localidad	Cantidad
Altos del Paraná	Ciudad del Este	28
Misiones	Candelaria	84
Misiones	Garupá	62
Misiones	Leandro N Alem	47
Misiones	Los Helechos	21
Misiones	Oberá	37
Misiones	Posadas	596
Misiones	San Pedro	70

Filtrar los resultados, posterior a la ejecución de cualquiera de las funciones agregadas.

```
select pro.PROVINCIA, loc.localidad, count(*) Cantidad
  from alumno as alu
    inner join localidad as loc
      on alu.idlocalidad = loc.idlocalidad
    inner join provincia pro on pro.IDPROVINCIA = loc.IDPROVINCIA
group by pro.PROVINCIA, loc.localidad
having count(*) >20
order by pro.provincia, loc.localidad;
```

MDL - Lenguaje de Manipulación de Datos - Having

Having tiene el mismo comportamiento para el Group By que el Where para el From.

Esto significa, que ante el resultado enviado por el Group by el Having filtra según las expresiones lógicas indicadas.

El Where actúa sobre el conjunto de datos devuelto por el From.

IMPORTANTE:

- Para filtrar un origen de datos usar Where
 - Para filtrar sobre un agrupamiento realizado utilizar Having.
- Se puede utilizar los dos en una misma consulta, el where hará primero su trabajo, o sea filtrará el origen de datos y el having filtrará sobre el resultado devuelto a posterior por el group by.

MDL - Lenguaje de Manipulación de Datos - Having

```
select L.LOCALIDAD, count(*)
  from alumno as A
    inner join localidad as L
      on A.IDLOCALIDAD = L.IDLOCALIDAD
 where a.NOMBREYAPELLIDO like 'A%'
 group by L.Localidad
 order by L.LOCALIDAD
```

LOCALIDAD	count(*)
Camboriú	4
Candelaria	7
Ciudad del Este	3
Garupá	6
Leandro N Alem	4
Oberá	2
Posadas	52
Resistencia	1

¡Muchas gracias!

    /poloticmisiones