## 

línea horizontal

Actividad UT 9 -Consultas Múltiples SQL

Lunes 22 de Enero

**─**

Hirahi Mejías Delgado

1 DAM-A

# 

# Actividad UT 9 - Consultas Múltiples SQL

# EJERCICIOS A REALIZAR

**Base De Datos Mundo**

**--1. Para todos los países mostrar el nombre y las ciudades más importantes.**

**select p.Name as pais,c.Name as Ciudad**

**from Country as p**

**inner join City as c**

**on p.Code=c.CountryCode**

**order by C.Population desc**

**go**

***haremos un inner join para unir las tablas country y city ya que necesitamos la información de estas tablas y ordenamos por population desc para que salgan primero los que tengan más población***

****

**--2. Para todos los países de Europa, contar cuántas ciudades y sumar la población de las ciudades.**

**select p.Name as pais,count (c.Name) as nºCiudades,Sum (c.Population) as SumaPoblacion**

**from Country as p**

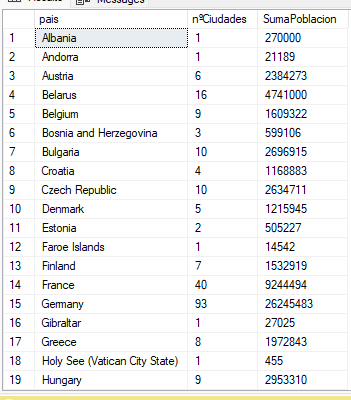
**inner join City as c**

**on p.Code=c.CountryCode**

**where p.Continent='Europe'**

**group by p.Name**

***en el select contaremos los nombres de ciudades con el count,sumamos población con sum y un inner join para unir tablas country y city con la condición de que el continente sea europa y agrupados por nombre del país***

****

**--3. Cuántos hablantes hay por lengua en América del Sur.**

**select sum(Percentage),p.name,l.Language as lenguaje**

**from CountryLanguage as l**

**inner join Country as p**

**on p.Code=l.CountryCode**

**where p.Continent like 'South America'**

**group by p.name,l.Language**

***en el select tenemos la suma del porcentaje de hablantes,nombre del país y lenguaje en el que haremos un inner join para unir las tablas country y country language,y que la condición sea que el continente sea américa del sur,agrupados por nombre del país y lenguaje***

****

**--4. Pares de países diferentes que ambos hablen Español.**

**SELECT DISTINCT p1.Name AS Pais1, p2.Name AS Pais2, lp.Language**

**FROM Country AS p1**

**CROSS JOIN Country AS p2**

**INNER JOIN CountryLanguage AS lp ON p1.Code = lp.CountryCode**

**INNER JOIN CountryLanguage AS lp2 ON p2.Code = lp2.CountryCode**

**WHERE p1.Code < p2.Code AND lp.Language = 'Spanish' AND lp2.Language = 'Spanish';**

*haremos un cross join con la misma tabla country seguido de dos inner join con country language ya que como tenemos que comparar países diferente,en el where pondremos que los códigos sean distintos y que el lenguaje de los dos países sea español*

****

**--5. 10 Ciudades con más habitantes indicando el nombre del país donde están.**

**select top 10 c.Name as ciudad, p.Name as pais, c.Population**

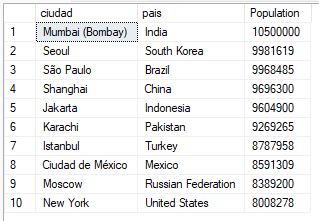
**from City as c**

**inner join Country as p**

**on c.CountryCode=p.Code**

**order by c.population desc**

*en el select pondremos select top 10 ciudades donde después del inner join de city con country lo ordenaremos por ciudad population*

****

**--6. Países que no tengan más de 2 idiomas.**

**select p.name as Paises, count (language) as lenguaje**

**from country as p**

**inner join countrylanguage as c**

**on c.CountryCode=p.Code**

**group by p.name**

**having count (language)<=2**

*hacemos un inner join de country con country language agrupamos por nombre del país y en la condición ponemos que cuente lenguaje y sea igual o mayor a 2*

****

**Base de datos videojuegos**

**1. Para los juegos que contengan FIFA dar las plataformas en las que funciona y sus desarrolladores y distribuidores**

**select j.Juego,p.Plataforma,dis.Distribuidor,d.Desarrollador**

**from Juego as j**

**inner join JuegoPlataforma as jp**

**on j.IdJuego=jp.IdJuego**

**inner join Plataforma as p**

**on jp.IdPlataforma=p.IdPlataforma**

**inner join Desarrollador as d**

**on jp.IdDesarrollador=d.IdDesarrollador**

**inner join Distribuidor as dis**

**on jp.IdDistribuidor=dis.IdDistribuidor**

**where j.Juego like '%FIFA%'**

*hacemos inner join de las tablas donde necesitamos informacion y despues en la condicion buscamos los que contengan FIFA*

****

**2. Nombre de clientes que hayan votado juegos de la plataforma PC**

**select c.Nombre,p.Puntuacion,j.Juego,pla.Plataforma**

**from Cliente as c**

**inner join Puntuacion as p**

**on c.id=p.id**

**inner join JuegoPlataforma as jp**

**on p.id=jp.id**

**inner join Juego as j**

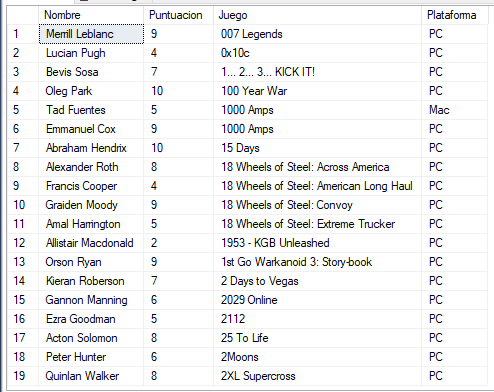
**on j.IdJuego=jp.IdJuego**

**inner join Plataforma as Pla**

**on jp.IdPlataforma=pla.IdPlataforma**

**where Plataforma like 'PC' or Plataforma like 'Mac';**

*hacemos los inner join de las tablas que necesitamos informacion y en la condición pondremos que sean de plataforma like pc or mac*

****

**3. Para cada juego de tipo Simulación dar en cuántas plataformas funciona.**

**select j.Juego,t.Tipo,count(Plataforma) as plataforma**

**from Tipo as t**

**inner join Juego as j**

**on j.IdTipo=t.IdTipo**

**inner join JuegoPlataforma as jp**

**on j.IdJuego=jp.IdJuego**

**inner join Plataforma as p**

**on jp.IdPlataforma=p.IdPlataforma**

**where Tipo='Simulación'**

**group by j.Juego,t.Tipo**

*haremos los inner join correspondientes y en la condición pondremos que el tipo sea simulación agrupados por el nombre del juego y el tipo*

****

**4. Clientes que no hayan puntuado**

**select c.Nombre,p.Puntuacion**

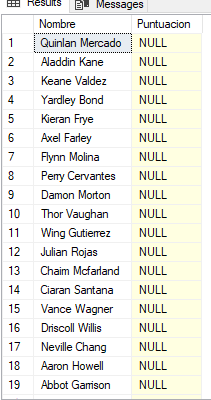
**from Cliente as c**

**left join Puntuacion as p**

**on c.id=p.id**

**where Idcliente is null;**

*en este caso tendremos que hacer un left join para que nos salgan los valores nulos y en la condición pondremos que id cliente sea nula*

****

**5. Videojuegos sin puntuación.**

**select j.Juego,p.Puntuacion**

**from Juego as j**

**left join JuegoPlataforma as jp**

**on j.IdJuego=jp.IdJuego**

**left join Puntuacion as p**

**on jp.Id=p.Id**

**where Puntuacion is null**

**group by j.juego,p.Puntuacion**

*como en el caso anterior pondremos left joins con las tablas que necesitemos y en la condición pondremos qué puntuación sea nula y agrupamos por nombre del juego y puntuación*

****

**6. Dando todos los videojuegos. Hallar la media de las puntuaciones recibidas y cuántas**

**select j.Juego,avg(p.Puntuacion) as media,count(Puntuacion) as NºPuntuaciones**

**from Juego as j**

**inner join JuegoPlataforma as jp**

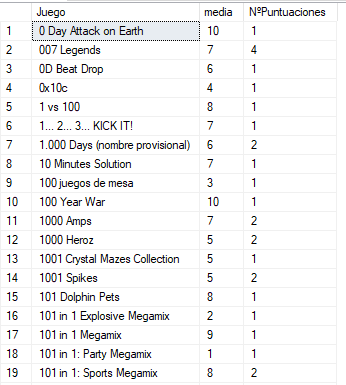
**on j.IdJuego=jp.IdJuego**

**inner join Puntuacion as p**

**on jp.Id=p.Id**

**group by j.Juego**

*en el select ponemos la media de la puntuación y contamos las puntuaciones que hay hacemos los inner join correspondientes y agrupamos por nombre del juego*

****

**Base de datos (EMPRESAS INFORMÁTICAS)**

**--1. Número de Facturas por tienda**

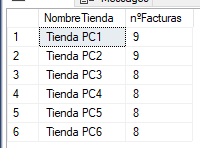
**select t.NombreTienda,count(f.NFactura) as nºFacturas**

**from Tienda as t**

**inner join Factura as f**

**on t.IdTienda=f.idTienda**

**group by t.NombreTienda**

****

*contamos los n Factura,hacemos los inner join correspondientes y agrupamos por nombre tienda*

**--2. Componentes para los que alguna vez el precioaplicado sea menor que el precio.**

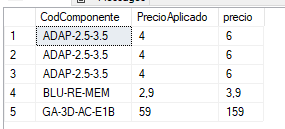
**select CodComponente,fc.PrecioAplicado,c.precio**

**from FacturaComponente as fc**

**inner join Componente as c**

**on fc.CodComponente=c.clave**

**where PrecioAplicado<precio**

****

*inner join de las tablas correspondientes y en la condición ponemos que el precio aplicado sea menos que el precio*

**--3. Modificar precio aplicado en facturas de cliente Ana disminuyéndolo en un 10%.**

**update FacturaComponente**

**set precioAplicado=PrecioAplicado\*0.9**

**from FacturaComponente as fc**

**inner join Factura as f**

**on fc.NFactura=f.NFactura**

**where Cliente='Ana'**

*haremos el update de la tabla Factura componente donde el set será precioAplicado=PrecioAplicado\*0.9 y la condición que el cliente sea ANA*

****

**--4. 6 componentes que más se han vendido en cantidad, sacando los empates.**

**select top 6 with ties CodComponente,Cantidad**

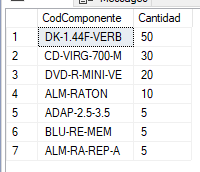
**from FacturaComponente as fc**

**inner join Componente as c**

**on fc.CodComponente=c.clave**

**order by Cantidad desc**

*haremos un select top 6 with ties para que aparezcan los empates,inner join correspondientes y ordenando el TOP por cantidad descendente*

****

**--5. Artículos sin ventas**

**select c.clave,NFactura**

**from Componente as c**

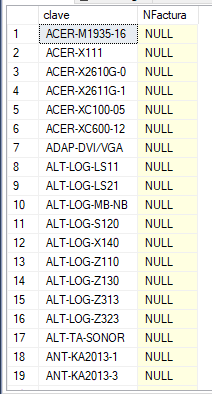
**left join FacturaComponente as fc**

**on c.clave=fc.CodComponente**

**where NFactura is null**

**group by c.clave,NFactura**

*haremos left join tabla componente con factura componente en la condición pondremos que Factura sea nula y agrupamos por clave y N Factura*

****

**--6. Borrar los artículos sin ventas del tipo BROTHER: FAX**

**Delete Componente**

**from Componente as c**

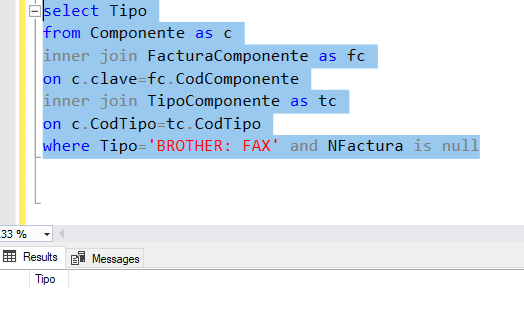
**inner join FacturaComponente as fc**

**on c.clave=fc.CodComponente**

**inner join TipoComponente as tc**

**on c.CodTipo=tc.CodTipo**

**where Tipo='BROTHER: FAX' and NFactura is null**

****

**Base de datos Discos**

**1. Número de discos por tipo sacando los que no tienen ninguno.**

**select count(distinct dt.IdDisco) as NºDiscos,t.tipo**

**from Tipo as t**

**left join DiscoTipo as dt**

**on dt.IdTipo=t.IdTipo**

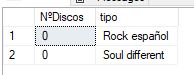
**left join Disco as d**

**on d.IdDisco=dt.IdDisco**

**where d.IdDisco is null**

**group by t.tipo**

*contamos los iddisco que sean distintos haremos left join correspondientes donde la condición sea que id disco sea null y agrupamos por tipo*

****

**2. Cliente, Título e Intérprete de todos los discos votados.**

**select c.Nombre,d.Titulo,i.Interprete,p.Puntuacion**

**from Cliente as c**

**inner join Puntuacion as p**

**on c.id=p.id**

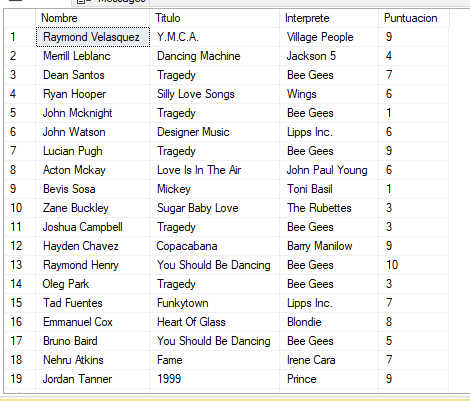
**inner join Disco as d**

**on d.idDisco=p.iddisco**

**inner join Interprete as i**

**on i.IdInterprete=d.IdInterprete**

*seleccionamos lo que queremos mostrar y hacemos los inner join donde este la información que necesitamos*

****

**3. Intérpretes que hayan interpretado discos de tipo que contenga Rock**

**select i.interprete,d.Titulo,t.tipo**

**from Interprete as i**

**inner join disco as d**

**on i.IdInterprete=d.IdInterprete**

**inner join DiscoTipo as dt**

**on d.IdDisco=dt.IdDisco**

**inner join Tipo as t**

**on dt.IdTipo=t.IdTipo**

**where t.tipo like '%rock%'**

*hacemos los inner join que necesitamos y en el where ponemos que el tipo contenga ‘Rock’*

****

**4. Pares de clientes que hayan nacido en décadas diferentes.**

**select**

**C1.Nombre AS NombreCliente1,**

**C2.Nombre AS NombreCliente2,**

**C1.FechaNacimiento AS NacimientoCliente1,**

**C2.FechaNacimiento AS NacimientoCliente2**

**from Cliente C1**

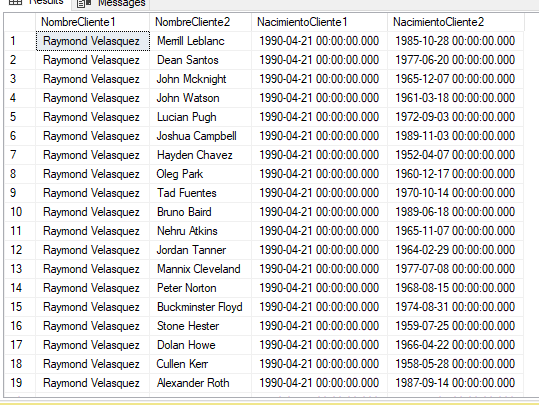
**INNER JOIN**

**Cliente C2 ON C1.id < C2.id AND**

**YEAR(C1.FechaNacimiento) / 10 <> YEAR(C2.FechaNacimiento) / 10**

**order by**

**C1.id, C2.id;**

****

**5. Título, Tipo e intérprete de discos sin puntuaciones.**

**select d.Titulo,t.tipo,i.Interprete,p.Puntuacion**

**from Interprete as i**

**left join Disco as d**

**on i.IdInterprete=d.IdInterprete**

**left join Puntuacion as p**

**on p.iddisco=d.IdDisco**

**left join DiscoTipo as dt**

**on d.IdDisco=dt.IdDisco**

**left join tipo as t**

**on dt.IdTipo=t.IdTipo**

**where p.Puntuacion is null**

*hacemos left join para que aparezcan los nulos de las tablas que necesitamos y en la condicion pondremos que la puntuación sea nula*

****

**6. 5 Intérpretes con mayor número de puntuaciones, incluyendo empates.**

**select top 5 with ties i.Interprete,p.Puntuacion**

**from interprete as i**

**inner join Disco as d**

**on i.IdInterprete=d.IdInterprete**

**inner join Puntuacion as p**

**on d.IdDisco=p.iddisco**

**order by p.Puntuacion desc**

*en el select pondremos top 5 with ties intérpretes ordenados en el order by por puntuación descendentes después de hacer los inner join correspondientes*

****