



Universidad Tecnológica de Puebla

Carrera: TICS

Producto:

Materia: Base de Datos II

Profesor: José Francisco Espinosa Garita

Grupo: 3"H"

Alumnos:

Gallegos Cuellar Ellioth

Michel Ernesto Duran Gonzales

Contenido

Contenido

Consultas Avanzadas.....	3
Descripción General SGBD SQL Server.....	4
Tablas con registros para hacer las consultas.....	5
INNER JOIN.....	6
Creacion de tablas por medio de cliente	6
LEFT JOIN.....	7
RIGTH JOIN.....	8
Ejemplos.....	8
Explicación de Conceptos	12
Diferencia entre producto cartesiano y el uso de Joins en consultas multitabla.	12
Bibliografía	12

Consultas Avanzadas

Las consultas son una de las partes fundamentales de las bases de datos ya que permiten obtener información para las empresas u organizaciones, que les sirve para tomar decisiones.

Una consulta SQL es tipo de consulta a una base de datos empleando lenguaje SQL.

Una consulta SQL básica puede constar con un máximo de seis cláusulas, de las cuales sólo dos son obligatorias (SELECT y FROM). Las cláusulas se especifican en el siguiente orden:

- SELECT < lista de atributos >
- FROM < lista de tablas >
- WHERE < condición >
- GROUP BY < atributo(s) de agrupación >
- HAVING < condición de agrupación >
- ORDER BY < lista de atributos >

SELECT: Indica qué atributos o funciones se van a recuperar.

FROM: Especifica todas las relaciones (tablas) que se necesitan en la consulta.

WHERE: Especifica las condiciones, si es que hacen falta, para seleccionar tuplas de esas relaciones, incluyendo las condiciones de reunión.

GROUP BY: Especifica atributos de agrupación.

HAVING: Especifica una condición que deben cumplir los grupos seleccionados, no las tuplas individuales. Las funciones agregadas integradas COUNT, SUM, MIN, MAX y AVG se usan junto con la agrupación.

ORDER BY: Especifica un orden para presentar el resultado de una consulta.

Descripción General SGBD SQL Server

SQL Server es un sistema de gestión de bases de datos relacionales (RDBMS) de Microsoft que está diseñado para el entorno empresarial. SQL Server se ejecuta en T-SQL (Transact -SQL), un conjunto de extensiones de programación de Sybase y Microsoft que añaden varias características a SQL estándar, incluyendo control de transacciones, excepción y manejo de errores, procesamiento fila, así como variables declaradas.

Tablas con registros para hacer las consultas

Results Messages				
	Idcomida	Nombre	Descripcion	Precio
1	1	Entrada	Flor de calabaza	85
2	2	Entrada	Tostas de hongos	27
3	3	Crema	Champiñones	26
4	4	Sopa	Puntitos	13
5	5	Ensalada	Cesar	65
6	6	Ensalada	Frutos rojos	95
7	7	Fuerte	Pato a la naranja	150
8	8	Fuerte	Pescado con adereso	120
9	9	Postre	Pay con helado	50
10	10	Postre	Pastel	35

Results Messages			
	Idbebida	Descripcion	Precio
1	1	Cerveza Modelo	18
2	2	Cerveza Corona	19
3	3	Cerveza XX	22
4	4	Cerveza Carta Blanca	15.5
5	5	Vino Tinto	250
6	6	Cerveza Tecate	25
7	7	Torres V	180
8	8	Tequila Corralejo	260
9	9	Vodka Absolut	650
10	10	Coca-cola	32

Results Messages			
	Idmesas	HoraEntrada	HoraSalida
1	1	09:00:00.0000000	11:00:00.0000000
2	2	12:00:00.0000000	02:59:00.0000000
3	3	02:00:00.0000000	04:40:00.0000000
4	4	06:00:00.0000000	07:10:00.0000000
5	5	05:30:00.0000000	06:20:00.0000000

Results Messages				
	Idmesero	Nombre	Apellido_Patemo	Apellido_Matemo
1	1	jose	garcia	juarez
2	2	angel	ruiz	aguilera
3	3	alfredo	guevara	martinez
4	4	margarita	contreras	cortez
5	5	andrea	lozano	guterrez

	Idreservaciones	FechaEntrada	FechaSalida
1	1	2017-06-02	2017-06-02
2	2	2017-07-13	2017-0-13
3	3	2017-06-01	2017-06-01
4	4	2017-05-29	2017-05-30

INNER JOIN

```
select C.idCliente, NombreC, Descripcion, Precio
from Clientes as C inner join bebida as b on C.idCliente=b.Idbebida
```

	idCliente	NombreC	Descripcion	Precio
1	1	Luis	Cerveza Modelo	18
2	2	juan	Cerveza Corona	19
3	3	Selina	Cerveza XX	22
4	4	Hilce	Cerveza Carta Blanca	15.5

```
select r.Idreservaciones, FechaEntrada, FechaSalida, Nombre, Apellido_Paterno, Apellido_Materno
from reservaciones as r inner join mesero as m on r.Idreservaciones=m.Idmesero
```

	Idreservaciones	FechaEntrada	FechaSalida	Nombre	Apellido_Paterno	Apellido_Materno
1	1	2017-06-02	2017-06-02	jose	garcia	juarez
2	2	2017-07-13	2017-0-13	angel	ruiz	aguilera
3	3	2017-06-01	2017-06-01	alfredo	guevara	martinez
4	4	2017-05-29	2017-05-30	margarita	contreras	cortez

Creacion de tablas por medio de cliente

```
select idCliente, Direccion, Telefono
into cliente
from Clientes
where idCliente between 1 and 4;
select * from cliente
```

	idCliente	Direccion	Telefono
1	1	Av SanAntonio	6-36-18-30
2	2	Av Rusia int 1	7-68-82-79
3	3	Calle Gregorio de gante	8-36-06-36
4	4	Av Robles #7894	2-15-27-42

```

select idCliente, NombreC
into cliente2
from Clientes
where idCliente between 1 and 4;
select * from cliente2

```

100 %

Results Messages

	idCliente	NombreC
1	1	Luis
2	2	juan
3	3	Selina
4	4	Hilce

LEFT JOIN

```

select * from cliente c left join cliente2 c2 on c.idCliente=c2.idCliente;

```

100 %

Results Messages

	idCliente	Direccion	Telefono	idCliente	NombreC
1	1	Av San Antonio	6-36-18-30	1	Luis
2	2	Av Rusia int 1	7-68-82-79	2	juan
3	3	Calle Gregorio de gante	8-36-06-36	3	Selina
4	4	Av Robles #7894	2-15-27-42	4	Hilce

Cambiando el orden

```

select * from cliente2 c2 left join cliente c on c.idCliente=c2.idCliente;

```

100 %

Results Messages

	idCliente	NombreC	idCliente	Direccion	Telefono
1	1	Luis	1	Av San Antonio	6-36-18-30
2	2	juan	2	Av Rusia int 1	7-68-82-79
3	3	Selina	3	Calle Gregorio de gante	8-36-06-36
4	4	Hilce	4	Av Robles #7894	2-15-27-42

RIGHT JOIN

```
select * from cliente c right join cliente2 c2 on c.idCliente=c2.idCliente;
```

100 %

Results Messages

	idCliente	Direccion	Telefono	idCliente	NombreC
1	1	Av SanAntonio	6-36-18-30	1	Luis
2	2	Av Rusia int 1	7-68-82-79	2	juan
3	3	Calle Gregorio de gante	8-36-06-36	3	Selina
4	4	Av Robles #7894	2-15-27-42	4	Hilce

```
select * from cliente2 c2 right join cliente c on c.idCliente=c2.idCliente;
```

100 %

Results Messages

	idCliente	NombreC	idCliente	Direccion	Telefono
1	1	Luis	1	Av SanAntonio	6-36-18-30
2	2	juan	2	Av Rusia int 1	7-68-82-79
3	3	Selina	3	Calle Gregorio de gante	8-36-06-36
4	4	Hilce	4	Av Robles #7894	2-15-27-42

Ejemplos

```
select count (Descripcion)[idcomida], Descripcion
from Cliente e inner join comida m on e.idCliente=m.idComida
group by Descripcion
having Descripcion='Queso panela a la parrilla con chiles'
```

100 %

Results Messages

	idcomida	Descripcion
1	2	Queso panela a la parrilla con chiles


```

select count (Descripcion)[idcomida], Descripcion,(NombreC)
from Cliente e left join comida m on e.idCliente=m.idComida
group by Descripcion,NombreC
having Descripcion='Queso panela a la parrilla con chiles' and NombreC='Michel Ernesto Duran Gonzalez'

```

100 %

Results Messages

	idcomida	Descripcion	NombreC
1	1	Queso panela a la parrilla con chiles	Michel Ernesto Duran Gonzalez

Consultas unión, intersección, complemento

```

select NombreC
from Cliente
union
select Descripcion
from comida

```

100 %

Results Messages

	NombreC
1	Angel Arturo Ruiz Aguilera
2	Carne asada a la tampiqueña
3	Chile ancho relleno de frijol
4	Consomé de pollo con tallarines y pollo
5	Daniel Cantor Hernandez
6	Eder Romero Tamariz
7	Elioth Gallegos Cuellar
8	Emanuel Barceñas García
9	Ensalada César
10	Ensalada de Higos
11	Ensalada Grieta

Query executed successfully. | PC35\MSSQLSERVER2016 (12.0 ... | PC35\PC119 (52) | master | 00:00:00 | 33 rows

```

select Descripción
from Bebida
union
select Descripción
from comida

```

100 %

Results Messages

	Descripción
1	Brandy Conde De Osbome 700 ml
2	Brandy Torres 10 700 Ml
3	Came asada a la tampiqueña
4	Cerveza Carta Blanca
5	Cerveza Corona
6	Cerveza Modelo
7	Cerveza Modelo Especial
8	Cerveza xx equis
9	Chile ancho relleno de frijol
10	Consomé de pollo con tallarines y pollo
11	Ensalada César
12	Ensalada de Higos
13	Ensalada Griega

Query executed successfully. PC35\MSSQLSERVER2016 (12.0 ... PC35\PC119 (52) master 00:00:00 43 rows

```

select Descripción
from comida
union
select Descripción
from Bebida

```

100 %

Results Messages

	Descripción
10	Consomé de pollo con tallarines y pollo
11	Ensalada César
12	Ensalada de Higos
13	Ensalada Griega
14	Ensalada Mixta
15	Ensalada Violeta
16	Filete de róbalo a los tres chiles
17	Flor de calabaza rellena de queso de...
18	Hot cakes con miel y mantequilla 3 P...
19	Parqué de pasas o nuez
20	Pastel con chocolate Pza
21	Pechuga de pollo rellena de camaró...
22	Pechuga de pollo rellena de queso d...
23	Pierna de ternera al horno
24	Plátanos con crema

Query executed successfully. PC35\MSSQLSERVER2016 (12.0 ... PC35\PC119 (52) master 00:00:00 43 rows

Subconsultas

SQLQuery1.sql - PC...te (PC67/PC02 (53))

```
select [idCliente],[idBebida]
from Bebida,Cliente
where idCliente=0 and idCliente<10 and NombreC in (select NombreC from Cliente where NombreC='Angel Arturo Ruiz Aguilera')
```

100 %

	idCliente	idBebida
4	2	4
5	2	5
6	2	6
7	2	7
8	2	8
9	2	9
10	2	10
11	2	11
12	2	12
13	2	13
14	2	14

Query executed successfully. PC67 (12.0 RTM) PC67/PC02 (53) Restaurante 00:00:00 20 rows

En este ejemplo solo mostro el idCliente correspondiente a Angel Arturo Ruiz Aguilera

Explicación de Conceptos

Diferencia entre producto cartesiano y el uso de Joins en consultas multitabla.

El producto cartesiano es un tipo de composición de tablas, aplicando el producto cartesiano a dos tablas se obtiene una tabla con las columnas de la primera tabla unidas a las columnas de la segunda tabla, y las filas de la tabla resultante son todas las posibles concatenaciones de filas de la primera tabla con filas de la segunda tabla.

La diferencia con el join es que no crea tablas y la relación puede ser mejor por eso es recomendable utilizar inner join, left join, right join.

Al final el resultado no es el mismo y no es necesario el producto cartesiano

Producto cartesiano e inner join

Producto cartesiano

El producto cartesiano es un tipo de composición de tablas, aplicando el producto cartesiano a dos tablas se obtiene una tabla con las columnas de la primera tabla unidas a las columnas de la segunda tabla, y las filas de la tabla resultante son todas las posibles concatenaciones de filas de la primera tabla con filas de la segunda tabla.

Inner join

La sentencia **INNER JOIN** es la sentencia JOIN por defecto, y consiste en combinar cada fila de una tabla con cada fila de la otra tabla, seleccionando aquellas filas que cumplan una determinada condición.

Diferencias entre producto cartesiano e inner join

Una diferencia entre ellos es que el producto cartesiano no es muy necesario en consultas ya que podrías obtener el mismo resultado en inner join de manera más detallada e implementando left join o right.

Tienen mucha similitud pero se puede diferenciar por sintaxis o en una consulta en sql server.

Bibliografía

<https://app.schoology.com/home>

<https://app.schoology.com/course/1032318648/materials/gp/1032324100>