UNIVERSIDAD PRIVADA DE TACNA

FACULTAD DE INGENIERIA

Escuela Profesional de Ingeniería de Sistemas

INFORME DE SEMANA N°01

"Laboratorio N° 02 - Utilizando Operadores de Conjuntos de Datos"

Curso: BASE DE DATOS II

Docente: Mag. Ing. Patrick Cuadros Quiroga

Robles Flores, Anthony Richard (2016056192)

Tacna – Perú
2019

INDICE

LABORATORIO N° 02

UTILIZANDO OPERADORES DE CONJUNTOS DE DATOS

I. INFORMACIÓN GENERAL

- **Objetivos:**

  Utilizando Aure Data Studio para poder realizar las consultas que se verán en este informe:

  - Utilización de la base de datos TSQL.
  - Ejecución de consultas.
  - Analizar y realizar consultas bajas medias y avanzadas.

- **Equipos, materiales, programas y recursos utilizados:**

  Para poder realizar las consultas que se observaran se utilizara lo siguiente:

  - Computadora con sistema operativo Windows que soporte SQL SERVER 2016.
  - Software “Azure Data Studio”.
  - Explicación de docente del curso

II. MARCO TEORICO

OPERACIONES DE CONJUNTOS

SQL proporciona varias operaciones de conjuntos muy eficaces. Por ejemplo, incluye operadores de conjuntos del estilo de los de SQL como UNION, INTERSECT, EXCEPT y EXISTS. Entity SQL también es compatible con operadores para la eliminación de duplicados (SET), la prueba de pertenencia a un grupo (IN) y las combinaciones (JOIN). En los temas siguientes se describen los operadores de conjuntos de SQL.

- UNION, disponible en todas las versiones de SQL Server.
- EXCEPT, nuevo en SQL Server 2005.
- INTERSECT, nuevo en SQL Server 2005.

Para utilizar operaciones de conjuntos debemos cumplir una serie de normas.

·Las consultas a unir deben tener el mismo número campos, y además los campos deben ser del mismo tipo.

Sólo puede haber una única clausula ORDER BY al final de la sentencia SELECT.

UNIÓN

Devuelve la suma de dos o más conjuntos de resultados. El conjunto obtenido como resultado de UNION tiene la misma estructura que los conjuntos originales.

El siguiente ejemplo muestra el uso de UNION

```
SELECT Nombre, Apellido1 , Apellido2, NifCif, FxNacimiento
FROM EMPLEADOS
UNION
SELECT Nombre, Apellido1 , Apellido2, NifCif, FxNacimiento
FROM CLIENTES
```

Cuando realizamos una consulta con UNION internamente se realiza una operacion DISTINCTsobre el conjunto de resultados final. Si queremos obtener todos los valores debemos utiliza UNION ALL.

EXCEPT

Devuelve la diferencia (resta) de dos o más conjuntos de resultados. El conjunto obtenido como resultado de EXCEPT tiene la misma estructura que los conjuntos originales.

El siguiente ejemplo muestra el uso de EXCEPT

```
SELECT Nombre, Apellido1 , Apellido2, NifCif, FxNacimiento
FROM EMPLEADOS
UNION ALL
SELECT Nombre, Apellido1 , Apellido2, NifCif, FxNacimiento
FROM CLIENTES
```

El uso de EXCEPT, como norma general, es mucho más rápido que utilizar condiciones NOT IN oEXISTS en la clausula WHERE.

INTERSECT

Devuelve la intersección entre dos o más conjuntos de resultados en uno. El conjunto obtenido como resultado de INTERSECT tiene la misma estructura que los conjuntos originales.

El siguiente ejemplo muestra el uso de INTERSECT

```
SELECT Nombre, Apellido1 , Apellido2, NifCif, FxNacimiento
FROM EMPLEADOS
EXCEPTSELECT Nombre, Apellido1 , Apellido2, NifCif, FxNacimiento
FROM CLIENTES
```

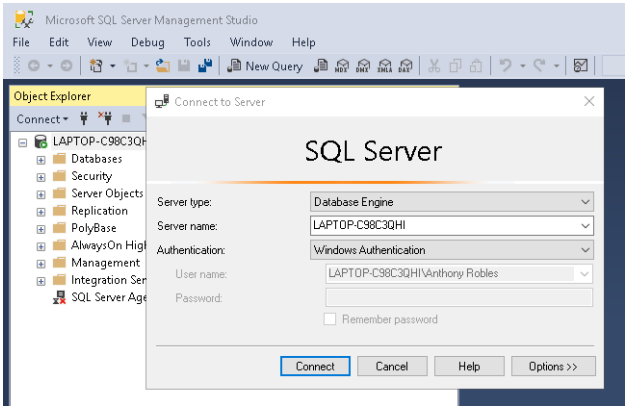
```
< class="texto">


SELECT Nombre, Apellido1 , Apellido2, NifCif, FxNacimiento
FROM EMPLEADOS
INTERSECTSELECT Nombre, Apellido1 , Apellido2, NifCif, FxNacimiento
FROM CLIENTES
```

III. PROCEDIMIENTO

Comencemos asegurando que nuestro SQL pueda ingresar correctamente.

Paso 1:

- Abrir SQL Server tomando en cuenta que en los servicios de Windows se encuentre iniciado.



Paso 2:

- Abriremos Azure DataStudio tomando en cuenta que el usuario de ingreso es el mismo con el que se ingresa a SQL SERVER.
- Conectar con la BD TSQL

```
[2]     1 USE TSQL;
        2 GO
```

Paso 3:

- Escriba una instrucción SELECT para devolver las columnas productid y productname de la tabla Production.Products. Filtre los resultados para incluir solo productos que tengan un valor de categoría 4.

```
1 select productid , productname from Production.Products
2 where categoryid = 4
```

	productid	productname
1	11	Product QMVUN
2	12	Product OSFNS
3	31	Product XWOXC
4	32	Product NUNAW
5	33	Product ASTMN
6	59	Product UKXRI
7	60	Product WHBYK
8	69	Product COAXA
9	71	Product MYMOI
10	72	Product GEEOO

Paso 4:

- Escriba una instrucción SELECT para devolver las columnas productid y productname de la tabla Production.Products. Filtre los resultados para incluir solo productos que tengan un monto total de ventas de más de $ 50,000. Para el monto total de ventas, deberá consultar la tabla Sales.OrderDetails y agregar todos los valores de línea de pedido (cantidad * precio unitario) para cada producto

```
1 SELECT
2 d.productid, p.productname
3 FROM Sales.OrderDetails d
4 INNER JOIN Production.Products p ON p.productid = d.productid
5 GROUP BY d.productid, p.productname
6 HAVING SUM(d.qty * d.unitprice) > 50000;
```

	productid	productname
1	29	Product VJXYN
2	38	Product QDOMO
3	59	Product UKXRI
4	60	Product WHBYK

Paso 5:

- Escriba una instrucción SELECT para recuperar las columnas custid y contactname de la tabla Sales.Customers. Muestre los 10 principales clientes por importe de ventas para enero de 2008 y los 10 principales clientes por importe de ventas para febrero de 2008. (Sugerencia: escriba dos declaraciones SELECT, cada una uniendo Sales.Customers y Sales.OrderValues, y use el operador de conjunto apropiado).

```sql
SELECT
c1.custid, c1.contactname
FROM
(
SELECT TOP(10)
o.custid, c.contactname
FROM Sales.OrderValues AS o
INNER JOIN Sales.Customers AS c ON c.custid = o.custid
WHERE o.orderdate >='20080101' AND o.orderdate < '20080201'
GROUP BY o.custid, c.contactname
ORDER BY SUM(o.val) DESC
) AS c1

union

select c2.custid , c2.contactname
FROM
(
    select top (10)
    o.custid, c.contactname
    from Sales.OrderValues as o
    inner join Sales.Customers as c on c.custid = o.custid
    where o.orderdate >= '20080201' and o.orderdate < '20080301'
    group by o.custid , c.contactname
    order by SUM(o.val) desc
) as c2
```

	custid	contactname
1	9	Raghav, Amritansh
2	20	Kane, John
3	30	Shabalin, Rostislav
4	32	Krishnan, Venky
5	34	Cohen, Shy
6	37	Crăciun, Ovidiu V.
7	39	Song, Lolan
8	55	Egelund-Muller, Anja
9	62	Misiec, Anna
10	63	Veronesi, Giorgio
11	65	Moore, Michael
12	71	Navarro, Tomás
13	76	Gulbis, Katrin
14	81	Nagel, Jean-Philippe
15	89	Smith Jr., Ronaldo

Paso 6: Operador EXCEPT y INTERSECT

Escriba una instrucción SELECT para recuperar la columna custid de la tabla Sales.Orders. Filtre los resultados para incluir solo a los clientes que compraron más de 20 productos diferentes (según la columna productid de la tabla Sales.OrderDetails)

```sql
SELECT o.custid
from Sales.Orders as o
inner join Sales.OrderDetails as d on d.orderid = o .orderid
group by o.custid
having count (distinct d.productid) > 20
```

	custid
1	4
2	5
3	7
4	9
5	10

	custid
5	10
6	20
7	24
8	25
9	30

Paso 7:

- Escriba una instrucción SELECT para recuperar la columna custid de la tabla Sales.Orders. Filtre los resultados para incluir solo clientes del país EE. UU. Y excluir a todos los clientes del resultado anterior (tarea 1). (Sugerencia: use el operador EXCEPT y la consulta anterior).

```
1  select custid
2  from Sales.Customers
3  where country = 'USA'
4
5  EXCEPT
6
7  select
8      o.custid
9      from Sales.Orders as o
10     INNER join Sales.OrderDetails as d on d.orderid = o.orderid
11     group by o.custid
12     having count (distinct d.productid) > 20
```

	custid
1	32
2	36
3	43
4	45
5	48
6	55
7	75
8	77
9	78
10	82

Paso 8

- Escriba una instrucción SELECT para recuperar la columna custid de la tabla Sales.Orders. Filtre solo clientes que tengan un valor total de ventas superior a $ 10,000. Calcule el valor de ventas utilizando las columnas qty y unitprice de la tabla Sales.OrderDetails

```
1  select o.custid
2  from Sales.Orders as o
3  INNER JOIN Sales.OrderDetails as d on d.orderid = o.orderid
4  group by o.custid
5  having SUM(d.qty * d.unitprice) >10000
```

	custid
1	23
2	46
3	75
4	9
5	89
6	72

Paso 10: Utilizando el Operador APPLY

- Escriba una instrucción SELECT para recuperar las columnas productid y productname de la tabla Production.Products. Además, para cada producto, recupere las dos últimas filas de la tabla Sales.OrderDetails según el número de pedido.

```
1  select top (2)
2      d.orderid
3      from SAles.OrderDetails as d
4      where d.productid = 1
5      order by d.orderid desc
```

	orderid
1	11070
2	11047

Paso 11:

- Utilice el operador APLICACIÓN CRUZADA y una subconsulta correlacionada. Ordene el resultado por la columna productid

```
1  SELECT
2   p.productid , p.productname , o.orderid
3   from Production.Products as p
4   cross apply
5   (
6  select top (2)
7      d.orderid
8      from SAles.OrderDetails as d
9      where d.productid = p.productid
10     order by d.orderid desc
11  ) o
12
13 order by p.productid
```

	productid	productname	orderid
1	1	Product HHYDP	11070
2	1	Product HHYDP	11047
3	2	Product RECZE	11077
4	2	Product RECZE	11075
5	3	Product IMEHJ	11077
6	3	Product IMEHJ	11017
7	4	Product KSBRM	11077

Paso 12:

- Ejecutar la sentencia siguiente para crear la función fnGetTop3ProductsForCustomer.

```
DROP FUNCTION IF EXISTS dbo.fnGetTop3ProductsForCustomer;
GO
CREATE FUNCTION dbo.fnGetTop3ProductsForCustomer (@custid AS INT)
RETURNS TABLE AS
RETURN
SELECT TOP(3) d.productid, p.productname, SUM(d.qty * d.unitprice)
AS totalsalesamount
FROM Sales.Orders AS o
INNER JOIN Sales.OrderDetails AS d ON d.orderid = o.orderid
INNER JOIN Production.Products AS p ON p.productid = d.productid
WHERE custid = @custid
GROUP BY d.productid, p.productname
ORDER BY totalsalesamount DESC;
```

Paso 13:

- Escriba una instrucción SELECT para recuperar las columnas custid y contactname de la tabla Sales.Customers. Utilice el operador CROSS APPLY con la función dbo.fnGetTop3ProductsForCustomer para recuperar las columnas productid, productname y totalsalesamount para cada cliente

```
select c.custid , c.contactname , p.productid , p.productname , p.totalsalesamount
from Sales.Customers as c
cross apply dbo.fnGetTop3ProductsForCustomer (c.custid) as p
order by c.custid
```

	custid	contactname	productid	productname	totalsalesamount
1	1	Allen, Michael	63	Product ICKNK	878,0000
2	1	Allen, Michael	59	Product UKXRI	825,0000
3	1	Allen, Michael	28	Product OFBNT	775,2000
4	2	Hassall, Mark	72	Product GEEOO	348,0000
5	2	Hassall, Mark	60	Product WHBYK	340,0000
6	2	Hassall, Mark	32	Product NUNAW	320,0000
7	3	Peoples, John	11	Product QMVUN	1453,2000

IV. CUESTIONARIO

CONCLUSIONES

- ✓ Las consultas son uno de los análisis fundamentales dentro de un SIG. Básicamente, una consulta efectúa una pregunta acerca de la información contenida en una capa, y obtiene como resultado los elementos de la capa que dan respuesta a dicha pregunta. Las consultas son en general un elemento aplicado sobre capas vectoriales, y el resultado de la consulta se expresa mediante una selección de entidades dentro de aquellas que componen dicha capa.

WEBGRAFIA

- ✓ Operaciones de conjuntos
  Recuperado de: http://katyygaby.blogspot.com/p/operaciones-de-conjuntos.html

- ✓ Consultas SQL
  Recuperado de: https://volaya.github.io/libro-sig/chapters/Consultas.html