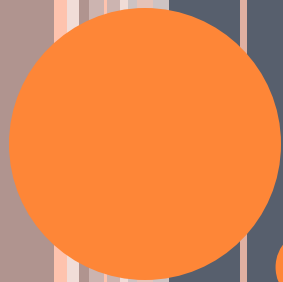


# **BASE DE DATOS 2**

**Ing. Jacqueline Guzman**



JOINS

# JOINS – CONSULTAS COMBINADAS

- Permite combinar varias tablas para obtener informacion que se encuentran en distintas tablas.
- TIPOS DE JOINS
  - INNER JOIN
  - NATURAL JOIN (no en SQL Server)
  - LEFT JOIN
  - RIGTH JOIN
  - FULL JOIN



# JOINS – CONSULTAS COMBINADAS

## INNER JOIN:


Permite combinar tuplas de distintas tablas filtrando el resultado para quedarse solo con las tuplas que estén en ambas tablas

```
SELECT campo1, campok...  
FROM TBLNOM1 INNER JOIN TBLNOM2  
ON TBLNOM1.CAMPO_R = TBLNOM2.CAMPO_L  
WHERE COND
```

La condicion del Join va separada de las demás condiciones (ON)

## NATURAL JOIN:

Simiar a Inner Join pero supone que campos con igual nombre deben tener igual valor. NO ES SOPORTADO POR SQLSERVER



# JOINS – CONSULTAS COMBINADAS

## Ejemplo INNER JOIN

```
SELECT P.nombreP, V.FechaV  
FROM Ventas V INNER JOIN Productos P  
ON V.codP = P.CodP
```

## Solución Equivalente:

```
SELECT P.nombreP, V.FechaV  
FROM Ventas V, Productos P  
WHERE V.codP = P.CodP
```



# JOINS – CONSULTAS COMBINADAS

## LEFT JOIN:

Combina las tuplas de ambas tablas a través de la condición indicada en el ON, pero además si hay tuplas en la tabla de la IZQUIERDA sin una tupla correspondiente en la tablas de la derecha igual se agrega al resultado completando los datos faltantes con NULL.

Es decir todas las tuplas de la tabla de la izquierda aparece en el resultado

```
SELECT *  
FROM Productos P LEFT JOIN Ventas V ON P.codP = V.codP
```

**Resultado:** datos de TODOS los productos y para aquellos productos vendidos se muestran los datos de cada venta.

**Observación:** aquellos productos no vendidos aparecerán en el resultado, completando con NULL los datos correspondientes a las ventas.



# JOINS – CONSULTAS COMBINADAS

## **RIGHT JOIN:**

Combina las tuplas de ambas tablas a través de la condición indicada en el ON, pero además si hay tuplas en la tabla de la DERECHA sin una tupla correspondiente en la tablas de la izquierda igual se agrega al resultado completando los datos faltantes con NULL.

Es decir todas las tuplas de la tabla de la derecha aparece en el resultado

```
SELECT *  
FROM Ventas V RIGHT JOIN Vendedores D ON V.codV=D.codV
```

**Resultado:** datos de TODOS los vendedores y para aquellos que han realizado alguna venta se muestran los datos de cada venta.

**Observación:** aquellos vendedores que no han realizado ninguna venta aparecerán en el resultado, completando con NULL los datos correspondientes a ventas.



# JOINS – CONSULTAS COMBINADAS

## **FULL JOIN:**

Combina las tuplas de ambas tablas a través de la condición indicada en el ON, pero además

- 1) si hay tuplas en la tabla de la DERECHA sin una tupla correspondiente en la tablas de la izquierda igual se agrega al resultado completando los datos faltantes con NULL.

Y ademas

- 2) si hay tuplas en la tabla de la IZQUIERDA sin una tupla correspondiente en la tablas de la derecha igual se agrega al resultado completando los datos faltantes con NULL.





## EJEMPLO: CASO DE ESTUDIO

Para cada producto, del proveedor 5, devolver la cantidad que se vendió en el 2016.

Los productos no vendidos en el 2016 deben aparecer en la solución.

Suponer el siguiente esquema:

`VENTAS(codVta, fechVta, codProd, cantidadProd, precioTot)`

`PRODUCTOS(codProd, nomProd, ..., codProv)`



# EJEMPLO: CASO DE ESTUDIO

VENTAS(codVta, fechVta, codProd, cantidadProd, precioTot)  
PRODUCTOS(codProd, nomProd, ... , codProv)

Comparar los resultados de las siguientes consultas:

SQL 1:

```
SELECT *  
FROM Productos P LEFT JOIN Ventas V  
ON P.codP = V.codP  
WHERE Year(V.fechaV) = 2016 and CodProv = 5
```

SQL 2:

```
SELECT *  
FROM Productos P LEFT JOIN Ventas V  
ON P.codP = V.codP and Year(V.fechaV) = 2016  
WHERE CodProv = 5
```

