



UNIVERSIDAD NACIONAL DE LA MATANZA

DEPARTAMENTO DE INGENIERÍA
E INVESTIGACIONES TECNOLÓGICAS

INGENIERIA EN INFORMATICA

BASE DE DATOS

Clase practica #9
Procesamiento de consultas

Grupo(id , nombre, fechCreación)	1 Bloque
Teatro(id , nombre, disponible, calle, nro, localidad, fechCreación)	5 Bloques
Recital(idTeatro , idGrupo , fecha , sala)	43 Bloques
Butaca(nro , teatroID)	295 Bloques
Venta(butacaNro , teatroID , monto, fecha)	600 Bloques

Para las siguiente consulta arme el árbol canónico y luego optimice teniendo en cuenta el MR planteado.

```

SELECT t.nombre, t.localidad
FROM butaca b, teatro t, venta v, grupo g, recital r
  WHERE g.id = r.idGrupo
    AND r.idTeatro = t.id
    AND b.TeatroID = t.id
    AND t.fechCreación > '2014-01-01' {
    AND (v.monto = 500 OR v.fecha > t.fechaCreación)

```

```
SELECT t.nombre, t.localidad
FROM butaca b, teatro t, venta v, grupo g, recital r
WHERE g.id = r.idGrupo
AND r.idTeatro = t.id
AND b.TeatroID = t.id
AND t.fechCreación > '2014-01-01' {
AND (v.monto = 500 OR v.fecha > t.fechaCreación)
```

Primero resolvemos el árbol canónico:

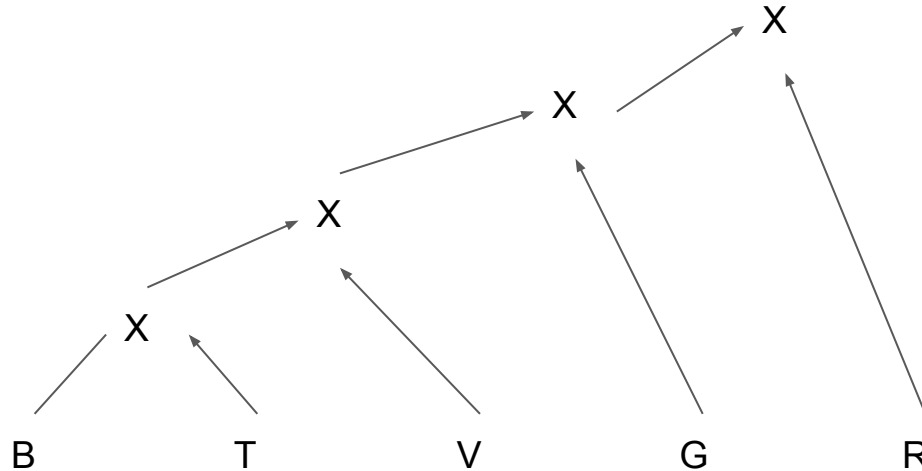
Ordenamos las tablas tal cual aparecen en la consulta.

B T V G R

```
SELECT t.nombre, t.localidad
FROM butaca b, teatro t, venta v, grupo g, recital r
WHERE g.id = r.idGrupo AND r.idTeatro = t.id
AND b.TeatroID = t.id AND t.fechCreación > '2014-01-01'
AND (v.monto = 500 OR v.fecha > t.fechaCreación)
```

PROYECCION (T.NOMBRE, T.LOCALIDAD)

SELECCION (g.id = r.idGrupo AND r.idTeatro = t.id
AND b.TeatroID = t.id AND t.fechCreación >
'2014-01-01' AND (v.monto = 500 OR v.fecha >
t.fechaCreación))



```
SELECT t.nombre, t.localidad
FROM butaca b, teatro t, venta v, grupo g, recital r
WHERE g.id = r.idGrupo AND r.idTeatro = t.id
AND b.TeatroID = t.id AND t.fechCreación > '2014-01-01'
AND (v.monto = 500 OR v.fecha > t.fechaCreación)
```

2do-Ordeno las tablas por tamaño en bloques

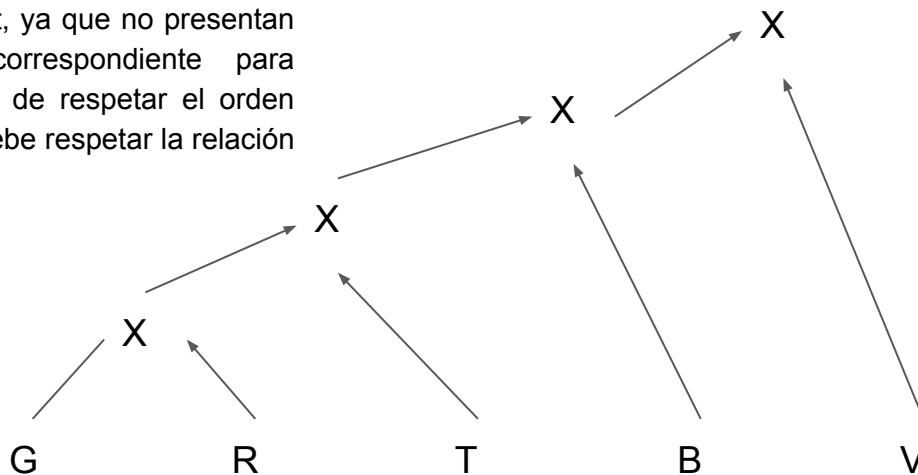
PROYECCION (T.NOMBRE, T.LOCALIDAD)

Ordenamos las relaciones de menor a mayor de acuerdo a la cantidad de bloques.

G T R B V

SELECCION (g.id = r.idGrupo AND r.idTeatro = t.id
AND b.TeatroID = t.id AND t.fechCreación >
'2014-01-01' AND (v.monto = 500 OR v.fecha >
t.fechaCreación))

Este ordenamiento no queda del todo completo. No se puede realizar una junta entre g y t, ya que no presentan en su consulta, el código correspondiente para relacionarse. Por lo tanto además de respetar el orden según la cantidad de bloques, se debe respetar la relación existente entre sus relaciones.



```

SELECT t.nombre, t.localidad
FROM butaca b, teatro t, venta v, grupo g, recital r
WHERE g.id = r.idGrupo AND r.idTeatro = t.id
AND b.TeatroID = t.id AND t.fechaCreación > '2014-01-01'
AND (v.monto = 500 OR v.fecha > t.fechaCreación)

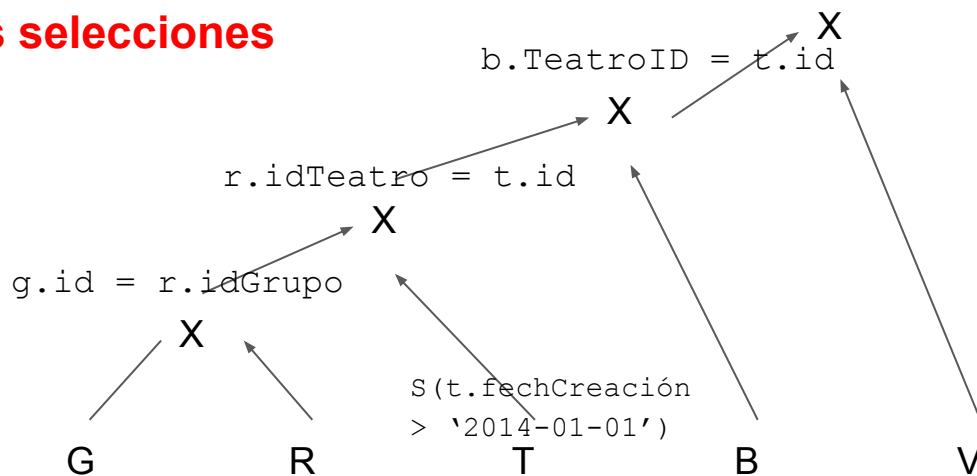
```

PROYECCION (T.NOMBRE, T.LOCALIDAD)

SELECCION

S(v.monto = 500 OR v.fecha > t.fechaCreación)

3ro-Bajo las selecciones



```

SELECT t.nombre, t.localidad
FROM butaca b, teatro t, venta v, grupo g, recital r
WHERE g.id = r.idGrupo AND r.idTeatro = t.id
AND b.TeatroID = t.id AND t.fechCreación > '2014-01-01'
AND (v.monto = 500 OR v.fecha > t.fechaCreación)

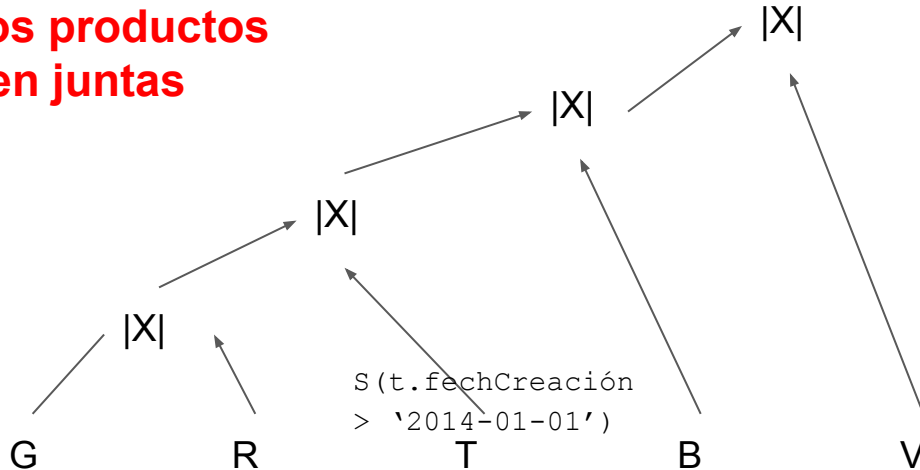
```

PROYECCION (T.NOMBRE, T.LOCALIDAD

SELECCION

g.id = r.idGrupo AND r.idTeatro = t.id
AND b.TeatroID = t.id AND t.fechCreación
> '2014-01-01'

**4to - Convierto los productos
cartesianos en juntas**



```

SELECT t.nombre, t.localidad
FROM butaca b, teatro t, venta v, grupo g, recital r
WHERE g.id = r.idGrupo AND r.idTeatro = t.id
AND b.TeatroID = t.id AND t.fechCreación > '2014-01-01'
AND (v.monto = 500 OR v.fecha > t.fechaCreación)

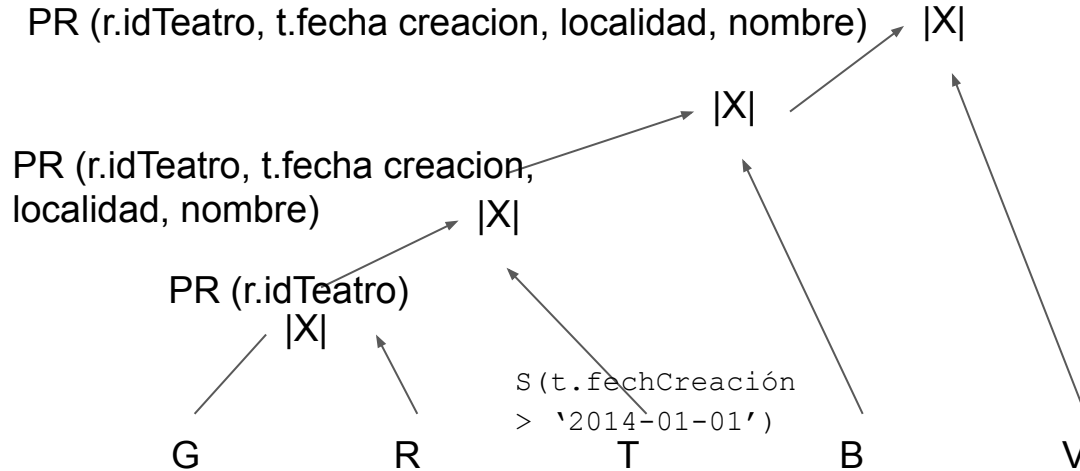
```

PROYECCION (T.NOMBRE, T.LOCALIDAD)

5to - Bajo las proyecciones

SELECCION

g.id = r.idGrupo AND r.idTeatro = t.id
AND b.TeatroID = t.id AND t.fechCreación
> '2014-01-01'



Dada la siguiente base de datos:

ELECTRODOMESTICO (nro_serie, modelo, version, cuit_fabricante, cod_punto_venta, cod_categoria)

FABRICANTE (cuit, razonSocial, telefono, direccion)

PUNTO_DE_VENTA (codigo, nombre, es_hipermercado, es_online)

CATEGORIA (codigo, descripcion)



La tabla ELECTRODOMESTICO ocupa 520 bloques.

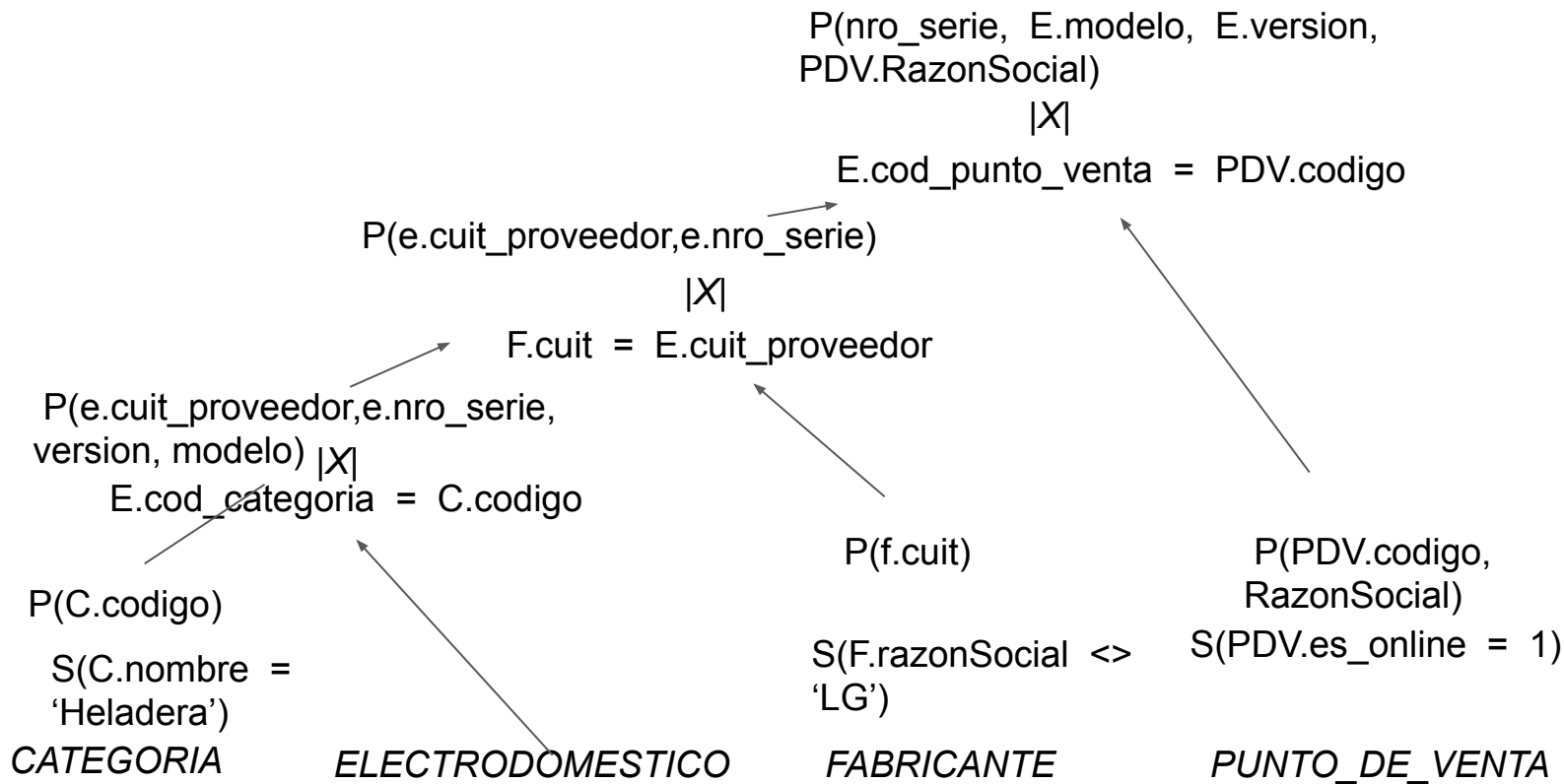
La tabla FABRICANTE ocupa 190 bloques.

La tabla PUNTO_DE_VENTA ocupa 260 bloques.

La tabla CATEGORIA ocupa 100 bloques.

Para la siguiente consulta SQL, **armar** el árbol optimizado. **Justifique** el orden elegido para las juntas.

```
SELECT E.nro_serie, E.modelo, E.version, PDV.RazonSocial
FROM      Fabricante AS F, Electrodomestico AS E, Punto_de_Venta AS PDV, Categoria AS
C
WHERE F.cuit = E.cuit_proveedor
      AND E.cod_punto_venta = PDV.codigo
      AND E.cod_categoria = C.codigo
      AND C.nombre = 'Heladera'
      AND F.razonSocial <> 'LG'
      AND PDV.es_online = 1
```



5:00

Calcular el tamaño en bloques de cada tabla:

LLAMADA

$$Llla = 8 + 8 + 6 + 10 + 12 + 4 + 6 = 54 \text{ bytes}$$

$$FBlla = B / Llla = 512 / 54 = 9$$

$$Blla = Tlla / FBlla = 27000 / 9 = \mathbf{3000 \text{ bloques}}$$

CLIENTE

$$Lcli = 10 + 48 + 8 + 64 + 32 + 32 = 194 \text{ bytes}$$

$$FBcli = B / Lcli = 512 / 194 = 2$$

$$Bcli = Tcli / FBcli = 1000 / 2 = \mathbf{500 \text{ bloques}}$$

PLAN

$$Lpla = 4 + 32 + 6 + 6 = 48 \text{ bytes}$$

$$FBpla = B / Lpla = 512 / 48 = 10$$

$$Bpla = Tpla / FBpla = 600 / 10 = \mathbf{60 \text{ bloques}}$$



Para ello, se necesita calcular el tamaño en bloques de cada tabla:

$L_{lla} = 8 + 8 + 6 + 10 + 12 + 4 + 6 = 54$ bytes $L_{cli} = 10 + 48 + 8 + 64 + 32 + 32 = 194$ bytes

$$FBIIa = B / LIIa = 512 / 54 = 9 \qquad FBcli = B / Lcli = 512 / 194 = 2$$
$$Blla = Tlla / FBlla = 27000 / 9 = \mathbf{3000 \text{ blocques}}$$

$L_{pla} = 4 + 32 + 6 + 6 = 48$ bytes

$$FB_{pla} = B / L_{pla} = 512 / 48 = 10$$
$$B_{pla} = T_{pla} / FB_{pla} = 600 / 10 = 60 \text{ blocques}$$


Hints

Hints en SQL Server

Son agregados a un comando SQL que indican que debe ejecutarse de manera diferente a la predeterminada por el motor.

Existen 4 tipos de hints diferentes:

- Join Hints
- Index Hints
- Lock Hints
- Processing Hints:

Join Hints:

HASH JOIN:

```
SELECT title_id, pub_name, title FROM titles
```

```
INNER HASH JOIN publishers ON titles.pub_id = publishers.pub_id
```

MERGE JOIN:

```
SELECT title_id, pub_name, title FROM titles
```

```
INNER MERGE JOIN publishers ON titles.pub_id = publishers.pub_id
```

LOOP JOIN:

```
SELECT title_id, pub_name, title FROM titles
```

```
INNER LOOP JOIN publishers ON titles.pub_id = publishers.pub_id
```



Index Hints:

Se utiliza cuando queremos forzar el uso de un índice en particular, para optimizar la consulta. **El plan de ejecución generado por el SQL Server suele ser el más óptimo.**

Ejemplo para forzar un Table Scan:

SELECT * FROM authors **WITH (INDEX(0))** // El valor 0 fuerza un Table Scan.

Ejemplo para forzar el uso del Clustered Index:

SELECT * FROM authors **WITH (INDEX(1))** // El valor 1, se fuerza el uso del índice Clustered de la tabla.

Ejemplo para forzar el uso de un Non Clustered Index:

SELECT * FROM authors **WITH (INDEX(NOMBRE_DEL_INDICE))**

Lock Hints:

Este tipo de hints especifican que tipo de lockeo se debe efectuar en una operación.

Existen varios hints de este tipo, pero el más usado es el NOLOCK y ROWLOCK.

NOLOCK: (equivalente al READUNCOMMITTED): Se usa en la sentencia SELECT. Indica al motor que ignore los bloqueos exclusivos de datos, lo que suele llamarse "lectura sucia". Con esto ganamos mayor performance y escalabilidad

Ejemplo:

```
SELECT COUNT(*) FROM Usuarios WITH (NOLOCK) INNER JOIN MenuUsuario WITH (NOLOCK) ON Usuarios.UsuarioID = MenuUsuario.UsuarioID
```

ROWLOCK: Especifica que se apliquen bloqueos de fila cuando normalmente se aplicarían bloqueos de página o de tabla. Aplica solo a sentencias UPDATE, DELETE e INSERT. También, como en el caso del NOLOCK, este hint sirve para ganar mayor performance en entornos muy concurrentes.

Ejemplo:

```
UPDATE Usuarios WITH (ROWLOCK) SET UsuarioID = 20 WHERE UsuarioID = 1
```

Dada la siguiente consulta en lenguaje SQL, indique al menos 3 mejoras que ayudarían a mejorar el costo:

```
SELECT *  
FROM PRODUCTO P  
      LEFT JOIN TIPOPRODUCTO TP ON P.CodTipo = TP.Cod  
WHERE year(P.FechaIngreso)=2015  
      and TP.Cod in (1,10,100)
```

Dada la siguiente consulta en lenguaje SQL, indique al menos 3 mejoras que ayudarían a mejorar el costo:

```
SELECT *  
FROM PRODUCTO P  
      LEFT JOIN TIPOPRODUCTO TP ON P.CodTipo = TP.Cod  
WHERE year(P.FechaIngreso)=2015  
      and TP.Cod in (1,10,100)
```

1. Cambiar LEFT JOIN x INNER JOIN, ya que la condición del where inhabilita el LEFT
2. Generar un índice por FechaIngreso en Producto
3. Cambiar la función year por el rango e fechas >= <=
4. No utilizar *, ya que traería todos los campos de ambas tablas, indicar qué campos se requieren.

Indicar que tipo de hints se utiliza en la siguiente consulta SQL. Modificar la misma para *forzar el uso del índice cluster en la tabla SELECCION* y *hacer una lectura de los registros de la tabla DIRIGE ignorando los bloqueos exclusivos de datos*.

```
SELECT S.pais, DT.nombre, DT.apellido, M.deporte, D.cant_victorias
FROM Dirige AS D
INNER JOIN Seleccion AS S ON S.cod_seleccion = D.cod_seleccion
INNER HASH JOIN Mundial AS M ON M.cod_mundial = D.cod_mundial
INNER JOIN Director_Tecnico AS DT WITH (INDEX (0))
    ON DT.cod_dt = D.cod_dt
WHERE DT.fecha_nacimiento BETWEEN '19900101' AND '19991231'
AND M.anio = 2021
```

```

SELECT S.pais, DT.nombre, DT.apellido, M.deporte, D.cant_victorias
FROM Dirige AS D INNER JOIN Seleccion AS S ON S.cod_seleccion = D.cod_seleccion
INNER HASH JOIN Mundial AS M ON M.cod_mundial = D.cod_mundial
INNER JOIN Director_Tecnico AS DT WITH (INDEX (0)) ON DT.cod_dt = D.cod_dt
WHERE DT.fecha_nacimiento BETWEEN '19900101' AND '19991231' AND M.anio = 2021

```

Se utiliza un JOIN HINT del tipo **HASH JOIN** en la junta entre las tablas DIRIGE y MUNDIAL.

Se utiliza un INDEX HINT para forzar el **NO uso de índices** sobre la tabla DIRECTOR_TECNICO, por consiguiente se hará sobre la misma un **TABLE SCAN**.

Consulta modificada para utilizar el índice cluster en la tabla SELECCION y leer los registros de DIRIGE por más que estén bloqueados.

```

SELECT          S.pais, DT.nombre, DT.apellido, M.deporte, D.cant_victorias
FROM            Dirige AS D WITH (NOLOCK)
INNER JOIN      Seleccion AS S WITH (INDEX(1))
               ON      S.cod_seleccion = D.cod_seleccion
INNER HASH JOIN Mundial AS M
               ON      M.cod_mundial = D.cod_mundial
INNER JOIN      Director_Tecnico AS DT WITH (INDEX(0))
               ON      DT.cod_dt = D.cod_dt
WHERE           DT.fecha_nacimiento BETWEEN '19600101' AND '19691231'
               AND      M.anio = 2021

```

MUCHAS GRACIAS