

UNIVERSIDAD NACIONAL DE LA MATANZA

DEPARTAMENTO DE INGENIERÍA
E INVESTIGACIONES TECNOLÓGICAS

INGENIERIA EN INFORMATICA

BASE DE DATOS

Práctica de Ejercicios de Optimización de consultas

Jefe de Cátedra: Ing. Verónica Ichazo

Docentes a cargo de curso:

Ing. Alfonso Palomares

Ing. Natalia Crespo

Ing. Guillermo Giannotti

Docentes a cargo de práctica:

Ing. Matías López

Ing. Fernando Ybarra

Ing. Javier Rebagliatti

Ing. Sebastián Deuteris

Ing. Hernán Jalil

2019

Referencias: **Clave Primaria**, Clave Foránea, **Clave Primaria y Foránea al mismo tiempo**

EJERCICIO 1

Dada la siguiente base de datos:

BANCO (**id(2)**, nombre (20), país(2)) 200 Bloques
CUENTA(monto(4), **idBanco(2)**, **idMoneda(2)**, **idPersona(2)**)
5.000 Bloques
MONEDA (**id(2)**, descripción(20), valorOro(4), valorPetroleo(4))
20 Bloques
OPERA (**idBanco(2)**, **idMoneda(2)**, cambioComp(4), cambioVta(4))
600 Bloques
PAÍS (**id(2)**, nombre(20)) 8 Bloques
PERSONA (**pasaporte**, codigofiscal, nombre(20), país(2)) 300.000 Bloques

Tamaño del bloque: 100Kb

Para las siguientes consultas, arme el árbol canónico y luego busque optimizar.

a)

```
SELECT b.nombre, m.descripcion, p.nombre
FROM banco b, moneda m, opera o, país p
WHERE b.id = o.idbanco AND b.pais = p.id AND o.idmoneda=m.id
AND o.cambioComp > 4.5 AND m.descripcion LIKE 'dolar%'
AND p.nombre IN ('ARG', 'URU', 'BRA')
```

b)

```
SELECT p.nombre, pp.pasaporte, pp.nombre, c.monto, b.nombre
FROM pais p, persona pp, cuenta c, banco b
WHERE b.id = c.idbanco AND p.nombre='ARG' AND pp.pais=p.id
AND c.idpersona=p.pasaporte AND p.pasaporte BETWEEN 15.000.00
AND 30.000.000
```

c)

```
SELECT c.monto, b.nombre, p.nombre, pp.pasaporte, pp.nombre
FROM pais p, persona pp, cuenta c, banco b
WHERE b.id = c.idbanco AND p.nombre='ARG' AND pp.pais=p.id
AND c.idpersona=p.pasaporte AND p.pasaporte BETWEEN 11.000.00
AND 45.000.000
```

d)

```
SELECT b.nombre, m.descripcion, p.nombre
FROM banco b, moneda m, opera o, país p
WHERE b.id = o.idbanco AND b.pais = p.id AND o.idmoneda=m.id
AND o.cambioComp > 4.5 AND m.descripcion LIKE 'dolar%'
AND p.nombre IN ('FRANCIA', 'UCRANIA', 'BRASIL')
```

EJERCICIO 2

Dada la siguiente base de datos:

Cliente (**idCliente(4)**, nombre(20), mail(20)) 200 Bloques

Venta (idCliente(4), idTorta(4), fecha(8)) 5000 Bloques
 Torta (idTorta(4), nombre(20), precio(4)) 5 Bloques
 Receta (idTorta(4), idIngrediente(4), cantidad(2)) 75 Bloques
 Ingrediente(idIngrediente(4), descripcion(20), peso(4)) 10 Bloques

Tamaño del bloque: 50Kb

Para las siguientes consultas, arme el árbol canónico y luego busque optimizar.

a)

```
SELECT c.nombre, c.mail
FROM cliente c JOIN venta v ON c.id_cliente = v.id_cliente
JOIN torta t ON t.id_torta = v.idTorta JOIN receta r ON
t.id_torta = r.id_torta
JOIN ingrediente i ON i.idIngrediente = r.idIngrediente
WHERE i.descripcion = "Chocolate"
AND v.fecha BETWEEN '01/01/2011' AND '31/12/2011';
```

b)

```
SELECT c.nombre, c.mail
FROM venta v, torta t, receta r, ingrediente i
WHERE v.id_torta = t.id_torta AND r.id_torta = t.id_torta
AND r.idIngrediente = i.idIngrediente
AND v.fecha > '12/01/2007'
AND t.precio IN (5, 10, 15)
AND i.peso = 55;
```

EJERCICIO 3

Dada la siguiente base de datos:

Sandwich(id(2), nombre(20), precio(2)) 50 Tuplas
 Proveedor (nro(2), razonSocial(20), tel(8), dir(20),
codProv(2)) 200 Tuplas
 Provincia (cod(2), descripción(20)) 20 Tuplas
 Ingrediente (cod(2), nombre(20)) 100 Tuplas
 CompuestoPor (sandwichId(2), ingredienteCod(2)) 4000
 Tuplas
 Provee (ingredienteCod(2), proveedorNro(2), precio(2)) 18000
 Tuplas

Tamaño del bloque: 100Kb

Para las siguientes consultas, arme el árbol canónico y luego busque optimizar.

a)

```
SELECT s.nombre, i.nombre, p.razon_social
FROM Provee pe, compuestoPor cp, Proveedor p, Sandwich s,
ingrediente i
WHERE pe.proveedor_nro=p.nro AND p.telefono > 5000000 AND
pe.ingredienteCod = i.cod AND i.nombre = "tomate" AND
cp.ingredienteCod = pe.ingredienteCod AND cp.sandwichId = s.id
```

```
and s.precio < pe.precio
```

b)

```
SELECT p.razon_social, i.nombre, p.dirección, pr.descripcion
FROM ingrediente i NATURAL JOIN proveedor p NATURAL JOIN provee
pe NATURAL JOIN provincia pr
WHERE i.cod = pe.ingredienteCod AND pe.proveedorNro = p.cod AND
p.codProv = pr.cod
AND pe.precio IN (10, 20)
```

EJERCICIO 4

Dada la siguiente base de datos:

```
Chofer (Legajo(8), NyA(18), FechaNac(8), Dir(24), Loc(20),
Tel(16), Sueldo(6))
tuplas 330
Vehículo (Patente(8), Marca(16), Modelo(20), Año(4))
tuplas 160
Asignado (Legajo(8), Patente(8), FechaDesde(8), FechaHasta(8))
tuplas 1200
```

Tamaño del bloque: 100Kb

Para las siguientes consultas, arme el árbol canónico y luego busque optimizar.

a)

```
Select C.Legajo, C.Sueldo, V.Patente, A.FechaDesde,
A.FechaHasta
From Chofer C, Vehículo V, Asignado A
Where C.legajo = A.legajo AND V.Patente = A.Patente AND
C.Localidad NOT IN ('LaFerrere') AND C.sueldo = 4500
```

b)

```
Select c.legajo, NyA, v.Marca, v1.Marca
From Chofer C, Vehículo V, Asignado A, Asignado B, Vehículo V1
Where C.legajo = A.legajo AND V.Patente = A.Patente AND
C.legajo = B.legajo AND V1.Patente = B.patente
AND V.patente != V1.patente
```

EJERCICIO 5

Dada la siguiente base de datos:

```
Cliente (codCli(4), nomCli(32), tel(12), fecAlta(8), ciudad(18) )
1800 tuplas
Factura ( tipoFac(1), nroFac(6), codCli(4), fechaEmision(8),
          fechaVto(8), importe(6), nroSuc(3), nroVend(4) )
6000 tuplas
```

Para las siguientes consultas, arme el árbol canónico y luego busque optimizar.

a)

```
SELECT * FROM Factura WHERE ( tipoFac = 'A' AND importe > 100 ) OR ( tipoFac = 'B' AND nroSuc = 5 )
```

b)

```
SELECT c.codCli, c.nomCli, c.tel, f.fechaEmision, f.importe
FROM Factura f INNER JOIN Cliente c ON f.codCli = c.codCli
WHERE f.nroSuc IN (4,6,7)
```

EJERCICIO 6

```
ALUMNO ( dni, nombre, apellido, fecha_nac, teléfono, email )
MATERIA( cod, descripcion, codCarrera )
CURSA ( dniAlumno, codMat, nroAula, anioLectivo )
AULA ( nro, capacidadMaxima, codSector )
```

La tabla Alumno ocupa 100 bloques. La tabla Materia ocupa 6 bloques. La tabla Cursa ocupa 30 bloques.

La tabla Aula ocupa 9 bloques.

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

```
SELECT alu.dni, alu.apellido, aul.*, mat.descripcion
FROM Alumno alu, Cursa cur, Aula aul, Materia mat
WHERE p.id_proy = a.id_proy
AND a.legajo = e.legajo
AND a.cod_cargo = c.cod_cargo
AND a.fecha_desde >= '01-01-2013'
AND c.cod_categoria > 3
```

EJERCICIO 7

Dada la relación R (a(4), b(12), c(4)) y la siguiente fórmula de Costo de Escritura:

$$\text{Costo Escritura} = \frac{1}{32} * (1 + T_R)$$

Se pide:

Indique a cuál de las siguientes consultas SQL corresponde la fórmula dada.

- a) SELECT a, b FROM R WHERE a=10 AND c>50
- b) SELECT * FROM R WHERE a>10 OR c>50
- c) SELECT a, c FROM R WHERE a=10 OR c>50
- d) SELECT b FROM R WHERE a>10 AND c>50

Tamaño del bloque = 128 bytes

Nota: Los valores que se muestran entre paréntesis, junto a cada atributo de la relación, indican su tamaño en bytes.

EJERCICIO 8

Dada la siguiente fórmula de Costo Total (inputs + outputs):

$$\text{Costo Total} = Br + \frac{Tr}{2 * V(B, R) * FBr} + Bs + \frac{Br}{2 * V(B, R)} + \frac{Br}{2 * V(B, R)}$$

Sabiendo que:

- No hay índices
- $Bs < Memoria < Br$
- $R(\underline{A}, B, \underline{C})$ y $S(\underline{C}, E)$

Se pide: Indique a cuál de las siguientes consultas SQL corresponde la fórmula dada.

- a) `SELECT * FROM R, S WHERE R.C=S.C AND R.A=100 AND R.B=200`
- b) `SELECT S.* FROM R, S WHERE R.C=S.C AND R.A=100 OR R.B=200`
- c) `SELECT R.* FROM R, S WHERE R.C=S.C AND R.A=100 OR R.B>200`
- d) `SELECT S.* FROM R, S WHERE R.C=S.C AND R.A=100 AND R.B>200`
- e) `SELECT R.* FROM R, S WHERE R.C=S.C AND R.A>100 AND R.B=200`

EJERCICIO 10

Dada la siguiente base de datos:

GaleríaDeArte(<u>id</u> , nombre, disponible, calle, nro, localidad)	100 Tuplas
Obra(<u>id</u> , nombre, material, <u>idTipo</u> , <u>idAutor</u>)	1000 Tuplas
TipoDeObra(<u>id</u> , descripción)	10 Tuplas
Temática(<u>id</u> , descripción)	10 Tuplas
Exposición(<u>idGaleria</u> , <u>idObra</u> , <u>idTematica</u> , fecha, sala)	5500 Tuplas
Autor(<u>id</u> , nya, fech_nacimiento)	50 Tuplas

Arme el árbol canónico de la siguiente query y luego optimice con la heurística.

```
SELECT O.nombre, A.nya, T.descripcion, E.fecha, TO.descripcion
FROM exposicion E, tematica T, autor A, tipoDeObra TO, obra O
WHERE T.id = E.id_tematica AND A.id = E.id_autor
AND TO.id = O.id_tipo AND O.id = E.id_obra
AND T.descripcion = 'ABSTRACTO' AND A.nya = 'PIET MONDRIAN'
```

EJERCICIO 11

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 = 2017
```

EJERCICIO 12

Utilizando las tablas del ejercicio anterior indique que tipo de hints se utiliza en el siguiente script. Modifique el mismo para forzar Table Scan en la tabla PREDIO y hacer una lectura de los registros de la tabla LIBRO ignorando los bloqueos exclusivos de datos.

```
UPDATE Feria_Libro WITH (ROWLOCK)
SET bonificacion = 30
FROM Feria_Libro
INNER JOIN Predio ON Predio.cod_predio = Feria_Libro.cod_predio
INNER MERGE JOIN Edicion ON Edicion.cod_libro =
Feria_Libro.cod_libro AND Edicion.nro = Feria_Libro.nro
INNER JOIN Libro ON Libro.cod_libro = Edicion.cod_libro
WHERE Predio.ciudad = 'Rosario' AND FL.autor_firma = 'si'
AND L.cantidad_capitulos >= 8
```