UNIVERSIDAD DE BUENOS AIRES

FACULTAD DE CIENCIAS EXACTAS Y NATURALES

DEPARTAMENTO DE COMPUTACIÓN

BASES DE DATOS
Procesamiento y Optimización de Consultas

GUÍA DE EJERCICIOS

---

1 Introductorios

1.1. Dado el siguiente esquema relacional:

MATERIA (CodMat, NombreMat, DuracionHs, CodCarrera)
CARRERA (CodCarrera, Nombre)

Y la siguiente consulta

```
SELECT M.NombreMat FROM MATERIA M, CARRERA C
    WHERE M.CodCarrera=C.CodCarrera and M.DuracionHs>50
```

Se pide

(a) Construir el árbol canónico de la consulta.

(b) Aplicar paso a paso las reglas de optimización heurística que se puedan para construir un árbol optimizado. Se deben justificar las decisiones para el armado del árbol optimizado. Analice los factores que podrían influir en la decisión de aplicar o no determinadas heurísticas

1.2. Dado el siguiente esquema relacional

PILOTO (cod_piloto, nombre, fecha_nacimiento, nacionalidad, campeonatos_ganados)

CARRERA (id_carrera, nombre_carrera, pais, fecha, cant_vueltas)

CORRIO_EN (cod_piloto, id_carrera, auto, posic_clasific, tiempo_clasific, posic_carrera, tiempo_carrera, cant_vueltas_carrera, tiempo_mejor_vuelta, Cant_paradas_boxes)

Dada la siguiente consulta para obtener los pilotos que ganaron carreras en el segundo semestre del 2010:

```
SELECT P.nombre
FROM PILOTO P, CARRERA C, CORRIO_EN E
WHERE P.cod_piloto=E.cod_piloto AND C.id_carrera=E.id_carrera
AND C.fecha >= '01/07/2010' AND C..fecha <= '31/12/2010'
AND E.posic_carrera=1
```

Se pide

(a) Construir el árbol canónico de la consulta

(b) Aplicar paso a paso las reglas de optimización heurística que se puedan para construir un árbol optimizado. Se deben justificar las decisiones para el armado del árbol optimizado. Analice los factores que podrían influir en la decisión de aplicar o no determinadas heurísticas.

1.3. Dada la siguiente relación $R(A, B, C, D, E)$ donde

$T_R = 5.000.000$
$FB_R = 10$

Asumiendo que A es una clave primaria de R, con valores correlativos entre 0 y 4.999.999 y que R está ordenado según A, para cada una de las siguientes consultas:

(a) `SELECT * FROM R WHERE a < 50.000`

(b) `SELECT * from R WHERE a = 50.000`

(c) `SELECT * FROM R WHERE a > 50.000 and a < 50.010`

(d) `SELECT * FROM R WHERE a <> 50.000`

Indicar y justificar cuál sería la mejor estrategia a aplicar:

- File scan sobre R
- Usar un índice árbol B+ clustered (altura 3) sobre el atributo A
- Usar un índice árbol B+ no clustered (altura 3) sobre el atributo A
- Usar un índice hash sobre el atributo A (máximo de 4 bloques por bucket)

1.4. Dada la siguiente relación **Cliente**(cid, nombre, edad) donde

$T_{Cliente} = 100.000$
$FB_{Cliente} = 10$
$PK = cid$
$I_{Cliente.Nombre} = 95.000$

Todos los campos tienen una longitud de **4 bytes**.

Los números de cliente son correlativos a partir de 1.

La edad de nuestros clientes varía entre los 18 y los 85 años

Existe un índice clustered sobre **cid**, uno **no clustered** (ambos árboles B+ de 3 niveles) sobre nombre, y otro **no clustered** (árbol B+ de 4 niveles) sobre (nombre,edad).

Calcular el costo total de ejecución de las siguientes consultas

(a) `SELECT nombre FROM Cliente WHERE edad > 21`

(b) `SELECT nombre, edad FROM Cliente WHERE cid = 65865`

(c) `SELECT edad FROM Cliente where WHERE = "Juan Perez"`

Indicar y justificar cuál fue la estrategia a aplicada en cada inciso

1.5. Dada el los siguientes esquemas de relaciones:

JUGADOR (IdJugador, Nombre)
DESEMPEÑO (IdJugador,idPartido, Puntos)

Calcular el costo total de ejecución de la siguiente consulta:

```
SELECT j.Nombre, d.Puntos
FROM Jugador j, Desempeño d
WHERE j.IdJugador = d.IdJugador and d.Puntos > 15
```

Datos:

$T_{Jugador} = 1.000$
$T_{Desempeno} = 10.000$
Tamaño de bloque $= 10K$
$I_{Jugador.Nombre} = 500$
$I_{Desempeno.Puntos} = 50$

Todos los campos tienen una longitud de 5 bytes.

Existe un índice clustered (altura 3) sobre Jugador.IdJugador y ambas relaciones caben en memoria

No existen índices y la memoria disponible es de 8 bloques.

2 Avanzados

2.1. Dados los siguientes esquemas de relaciones

ELEMENTO (nroe, descripcion, precio)
PROVEE (nrop, nroe)
PROVEEDOR (nrop, nombre, domicilio, pcia)

Datos:

$T_{ELEMENTO} = 5.000$, $T_{PROVEE} = 80.000$. $T_{PROVEEDOR} = 200$
Longitud de campos: 32 bytes
Tamaño de bloque: 1024 bytes
Cantidad de bloques de memoria disponibles : 10
Hay un índice non-clustered de 2 niveles ($X = 2$) sobre $PROVEEDOR.nombre$
Sabiendo que:

- Juan Pérez provee el 5% de los elementos
- No hay dos proveedores con el mismo nombre
- La tercera parte de los elementos tiene precio superior a 100

Para la consulta

```
SELECT descripcion
FROM PROVEE PE, ELEMENTO E, PROVEEDOR P
WHERE precio >100 AND E.nroe = PE.nroe AND
PE.nrop = P.nrop AND nombre = 'Juan Perez';
```

Se pide:

(a) Construir el árbol canónico de la consulta.
(b) Construir el árbol optimizado aplicando paso a paso optimización heurística (tener en cuenta la cantidad de registros de las tablas).
(c) Calcular el costo de la consulta utilizando el árbol obtenido en b) y el índice dado.
(d) Si pudiera agregar un segundo índice para mejorar aún más el costo de la consulta. ¿Cuál elegiría? Justifique (no hace falta que recalcule el costo)

2.2. Dados los siguientes esquemas de relaciones

R (A, B)
S (A, C, D, E)
T (C, F)

Datos:
$T_R = 200$, $T_S = 10.000$. $T_T = 1.000$
Longitud de campos: 8 bytes
Tamaño de bloque: 512 bytes
Cantidad de bloques de memoria disponibles: 3

Atributos:

A, C: Claves primarias
B: Clave candidata
$I_{S.D} = 1.000$ con rango [1...1.000]
$I_{S.E} = 2$

$I_{T.F} = 10$ con rango [1...10]

Sabiendo que:
Cada valor de R.A está referenciado 50 veces en S.A
Cada valor de T.C está referenciado 10 veces en S.C Las claves primarias tienen índices clustered y las candidatas non-clustered (todos de altura 3).

Para la consulta

```
SELECT B, E FROM S, R, T
WHERE R.A = S.A AND S.C = T.C AND
      B = 17 AND  F >= 5 AND  D <= 250
```

Se pide:

(a) Construir el árbol canónico de la consulta.
(b) Construir el árbol optimizado aplicando paso a paso optimización heurística (tener en cuenta la cantidad de registros de las tablas).
(c) Calcular el costo de la consulta utilizando el árbol obtenido en b) y los índices dados.
(d) Si pudiera agregar un segundo índice para mejorar aún más el costo de la consulta. ¿Cuál elegiría? Justifique (no hace falta que recalcule el costo)

2.3. Considere el siguiente esquema de base de datos relacional:

Departamento (CodDepartamento, Nombre, DNIGerente, GerenteDesde)
Empleado (DNI, Nombre, Apellido, FechaNacimiento, Dirección, Sexo, Sueldo, DNISupervisor, CodDepartamento)
Proyecto (CodProyecto,Nombre, Ubicación, CodDepartamento)
TrabajaEn (DNI, CodProyecto, Horas)

Y las siguientes consultas:

(a)
```
SELECT e.Nombre, e.Apellido, e.Direccion
FROM Empleado e, Departamento d
WHERE d.nombre = 'Research'
      AND e.CodDepartamento = d.CodDepartamento
```

(b)
```
SELECT e.Nombre, e.Apellido, s.Nombre, s.Apellido
 FROM Empleado e, Empleado s
 WHERE E.DNISupervisor = e.DNI
```

(c)
```
SELECT e.Nombre, d.Nombre, p.Nombre
FROM Empleado e, Departamento d, Proyecto p
WHERE E.CodDepartamento = d.CodDepartamento
     AND D.CodDepartamento = p.CodDepartamento
     AND P.CodDepartamento > 9000
```

(a) Escriba el árbol canónico para cada una de las consultas.
(b) Optimice los árboles obtenidos en el paso anterior aplicando los distintos criterios de optimización heurística
(c) Considerando los siguientes datos sobre la implementación física de las tablas y sus índices:
Tamaño de bloque: 4096 bytes
Tamaño de los campos: 8 bytes
Cantidad de bloques de memoria disponibles: 50

- **Tabla Empleado**:
  $T_{Empleado} = 10.000$
  Índice clustered sobre el campo Sueldo: Niveles X = 3,
  Índice non-clustered sobre el campo clave DNI: X = 4
  Índice non-clustered sobre el campo Sexo: X = 1

- **Tabla Departamento**:
  $T_{TDepartamento} = 125$
  Índice clustered sobre el campo CodDepartamento: X = 1
  Índice non-clustered sobre el campo DNIGerente: X = 2
  $T_{Departamento.nombre}$: 125
- **Tabla Proyecto**:
  $T_{Proyecto} = 34$
  Índice clustered sobre el campo CodProyecto: X = 1
- **Tabla TrabajaEn**:
  $T_{TrabajaEn} = 25.500$
  La tabla se encuentra ordenada según su clave primaria.

i. Calcule los costos (en términos de cantidad de accesos a bloques) asociados a los árboles obtenidos en los puntos (a) y (b).
ii. Compare los resultados obtenidos para los árboles canónicos con respecto a los árboles optimizados utilizando heurísticas.

2.4. Dada una implementación del esquema relacional **A**(a,c), **B**(b,c), **R**(a,b), donde:
R.a referencia a **A.a** y **R.b** referencia a **B.b**
Todos los campos tienen una longitud de 256 B
El tamaño de bloque es de 1024 B
La memoria disponible es de 3 bloques
Considere la siguiente consulta SQL:

```
SELECT A.c, B.c FROM R, A, B
WHERE R.a = A.a AND R.b = B.b
      AND A.a = A.c AND B.b = B.c
```

en una instancia donde: - A y B tienen 2.000 registros
- R tiene 400.000 registros
- Cada tupla de A aparece referenciada 200 veces en R
- Cada tupla de B aparece referenciada 200 veces en R
- El 10% de A satisface A.a = A.c
- El 1% de B satisface A.b = A.c

(a) Obtenga el árbol canónico y optimícelo.
(b) Asumiendo que las tablas NO TIENEN INDICES, calcular el costo del árbol optimizado.
(c) Si ahora suponemos que R tiene un índice clustered de 3 niveles sobre (a, b), en ese orden. ¿Se modifica el costo calculado en el punto (b)? ¿Existe ahora algún plan alternativo (o sea otro árbol) con un costo menor?

2.5. Considere el siguiente esquema de base de datos relacional:

Cuentas (NumCuenta, FechaAlta, TipoCuenta)
Saldos (NumCuenta, FechaSaldo, Saldo)
Titulares (NumCuenta, ApellidoyNombre, TipoTitular)

Existe un índice primario para cada una de las claves de las tablas. Además, Cuentas posee un índice hash por el campo **FechaAlta** y otro índice no clustered por **TipoCuenta**. La base de datos contiene información desde el 1 de marzo de 2004.

- **Cuentas** : Se dan de alta 60 cuentas por día en promedio. Existen 10 tipos distintos de cuentas
- **Saldos** : Los saldos se registran una vez por quincena para todas las cuentas. Uno de cada 100 saldos es negativo
- **Titulares** : 300.000 registros. Existen 3 tipos distintos de titular.
- 1 Bloque = 4096 bytes. Todos los campos miden 256 bytes

Se tiene la siguiente sentencia SQL:

```
SELECT  C.NumCuenta, S.Saldo, T.ApellidoyNombre
FROM Cuentas C, Saldos S, Titulares T
WHERE C.NumCuenta = S.NumCuenta
AND C.FechaAlta >= '01-DIC-2009'
AND S.Saldo < 0
AND C.NumCuenta = T.NumCuenta
AND C.TipoCuenta = 5
AND T.TipoTitular = 3
```

(a) Obtenga el árbol canónico y optimícelo, , justificando las decisiones tomadas.
(b) Calcule el costo de ejecución del plan optimizado obtenido en el paso anterior.
(c) Evalúe el costo de ejecución de un plan alternativo en donde puedan aprovecharse (en caso de ser posible) los índices sobre **Cuentas**, **Titulares** y **Saldos**.

3 Ejercicios de Parcial

3.1. Una empresa de comercio electrónico, cuenta con las siguientes relaciones:

ARTICULO (idArticulo, idVendedor, descripcion, categoria, antigüedad)
VENDEDOR (idVendedor, nombre, reputación, direccion, telefono)
PUBLICACION (idPublicacion, idArticulo, vendido, precio)

La tabla **ARTICULO** guarda la información de los articulos que se ofrecen. Tiene la siguiente información: identificador del articulo, identificador del vendedor, una descripción ampliada del artículo, la categoría del artículo, por ejemplo, "Electrodoméstico", y la antigüedad del artículo, que es un valor entre nuevo (0), casi nuevo (1), mediano uso (2), viejo (3) .

La tabla **VENDEDOR**, guarda el identificador del vendedor, su nombre, dirección, teléfono, y su reputación, que es un número entero entre 1 y 10.

La tabla **PUBLICACION** guarda el identificador de la publicación, el identificador del artículo, si el artículo está vendido (vendido=TRUE) o está aún en venta (vendido=FALSE), y el precio de venta.

Se desea optimizar la siguiente consulta:

```
SELECT P.idPublicacion, P.precio, A.idArticulo
FROM VENDEDOR V, PUBLICACION P, ARTICULO A
WHERE V.idVendedor = A.idVendedor AND A.idArticulo = P.idArticulo AND P.precio < 500
AND V.reputacion > 6 AND V.reputacion < 10 AND A.antiguedad < 2
AND V.vendido = FALSE
```

Se sabe adicionalmente que:

- Los atributos idX son numéricos enteros. Los atributos antigüedad y reputación son números enteros. El atributo vendido, es booleano. Cantidad de bytes que ocupa cada atributo: idX (4), nombre (60), descripción (800), categoría (60), antigüedad (4), reputación (4), dirección (50), teléfono (50), vendido (4), precio (8).
- $T_{VENDEDOR} = 1.000.000$; $T_{PUBLICACION} = 20.000.000$; $T_{ARTICULO} = 10.000.000$.
- El 60% de los artículos vale menos que 500.
- El 30% de las publicaciones corresponde a artículos que están actualmente en venta.
- Se dispone de los índices: **I1**: índice B+ unclustered sobre idArticulo en ARTICULO; **I2**: índice B+ unclustered sobre idVendedor en VENDEDOR; **I3**: indice Hash sobre reputación en VENDEDOR; **I4**: Indice Hash sobre vendido en PUBLICACION; **I5**: índice B+ clustered sobre antigüedad en ARTICULO.
- Los índices B+ tienen una altura $X = 3$. En los índices Hash se asume un máximo de 5 bloques por bucket. Asuma que los punteros a tupla que se necesiten recorrer en los índices B+ entran en una hoja, y que una hoja entra en un bloque.
- Tamaño de un bloque $L_{bloque} = 2048$ bytes
- Cantidad de bloques en memoria: 5
- Si no hay especificación, suponga distribución uniforme.

Proponga un plan de ejecución optimizado para la consulta dada, indicando solamente el árbol final. El plan de ejecución propuesto debe ser muy cercano al plan de menor costo posible. Justifique su plan. Justifique también porque decide utilizar o no utilizar cada uno de los índices disponibles.

3.2. Una empresa de cursos online, cuenta con las siguientes relaciones:

CURSO (idCurso, nombre, descripción)
ALUMNO (idAlumno, nombre, curriculum, nacionalidad, edad)
NOTAS (idCurso, idAlumno, nota)

La tabla **CURSO** guarda los cursos que se dictan. Tiene la información del id del curso, del nombre del curso, y la descripción, que es un texto con una descripción ampliada del curso. La tabla **ALUMNO** guarda el id del alumno, el nombre del alumno, su nacionalidad, su edad, y su curriculum vitae. La tabla **NOTAS** guarda el id del curso, el id del alumno, y la nota que sacó el alumno en dicho curso, que es un número float entre 0 y 10.

Se desea optimizar la siguiente consulta:

```
SELECT A.idAlumno, N.nota, C.idCurso
FROM CURSO C, ALUMNO A, NOTAS N
WHERE A.IdAlumno = N.idAlumno AND C.idCurso = N.idCurso
AND N.nota > 9 AND C.nombre = "Machine Learning" AND A.nacionalidad = "Argentina"
```

Se sabe adicionalmente que:

- Los atributos idX son numéricos enteros. Nota es numérico flotante. Cantidad de bytes que ocupa cada atributo: idX (4), nombre (80), nacionalidad (80), descripción (500), curriculum (1200), edad (4), nota(8).
- $T_{ALUMNOS} = 100.000$; $T_{CURSOS} = 300$; $T_{NOTAS} = 400.000$.
- Hay 195 países en el mundo.
- Se dispone de los índices: **I1**: Indice Hash sobre nota en NOTAS; **I2**: índice B+ unclustered sobre nombre en CURSO; **I3**: indice Hash sobre nacionalidad en ALUMNO; **I4**: índice B+ clustered sobre nacionalidad en ALUMNO.
- Los índices B+ tienen una altura $X = 6$. En los índices Hash se asume un máximo de 3 bloques por bucket. Asuma que los punteros a tupla que se necesiten recorrer en los índices B+ entran en una hoja, y que una hoja entra en un bloque.
- Tamaño de un bloque $L_{bloque} = 2048$ bytes
- Cantidad de bloques en memoria: 5
- Si no hay especificación, suponga distribución uniforme.

(a) Proponga un plan de ejecución optimizado para la consulta dada, indicando sólamente el árbol final. Justifique cada una de las decisiones que tome para su propuesta de plan de ejecución. Justifique también porque decide utilizar o no utilizar cada uno de los índices disponibles.

(b) Calcule el costo del plan del ítem a).