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.

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)

Donde: CORRIO_EN.cod_piloto referencia a PILOTO.cod_piloto
CORRIO_EN.id_carrera referencia a CARRERA.id_carrera

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.

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)

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 (costo de input + costo de output) 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 nombre = "Juan Perez"

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

5.

Dada la siguiente base de datos:

Jugador (IdJugador, Nombre)
Desempeño (IdJugador, IdPartido, Puntos)

Calcular el costo total (costo de input + costo de output) 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
$T_{Desempeño}$ = 10.000
Tamaño de bloque = 10 K

$I_{Jugador.Nombre}$ = 500
$I_{Desempeño.Puntos}$ = 50

Todos los campos tienen una longitud de 5 bytes.

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

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

6.

Dado lo siguiente:

Tablas	**#Registros**
ELEMENTO (nro_e, desc, precio)	5.000
PROVEE (nro_p, nro_e)	80.000
PROVEEDOR (nro_p, nombre, domicilio, pcia)	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

Y la consulta:

```
Q= SELECT desc
     FROM PROVEE PE, ELEMENTO E, PROVEEDOR P
   WHERE  precio >100 AND
          E.nro_e= PE.nro_e AND PE.nro_p= P.nro_p AND
          nombre= 'Juan Pérez';
```

(Descripción de los elementos de precio superior a $100 provistos por Juan Pérez)

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)

7.

Dado lo siguiente:

Tablas	**#Registros**
R (A, B)	200
S (A, C, D, E)	10.000

T (C, F)	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
$I_{S.F}$ = 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).

Y 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 Calcular el costo de la consulta utilizando el árbol obtenido en b) y el índice dado.
c) Si pudiera agregar otro índice para mejorar aún más el costo de la consulta. ¿Cuál elegiría? Justifique (no hace falta que recalcule el costo)

8.
Considere el siguiente esquema de base de datos

Departamento (CodDepartamento, Nombre, DNIGerente, GerenteDesde)
CodDepartamento es clave primaria
DNIGerente referencia a Empleado(DNI)

Empleado (DNI, Nombre, Apellido, FechaNacimiento, Dirección, Sexo, Sueldo, DNISupervisor, CodDepartamento)
DNI es clave primaria
DNISupervisor referencia a Empleado(DNI)
CodDepartamento referencia a Departamento(CodDepartamento)

Proyecto (CodProyecto, Nombre, Ubicación, CodDepartamento)
CodProyecto es clave primaria
CodDepartamento referencia a Departamento(CodDepartamento)

TrabajaEn(DNI, CodProyecto, Horas)
DNI referencia a Empleado(DNI)
CodProyecto referencia a Proyecto(CodProyecto)

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 queries.
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:
$T_{Departamento}$ = 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.

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

9.

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?

10.

Considere el siguiente esquema de base de datos :
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** (máxima cantidad de bloques por bucket = 4) y otro índice no clustered por **TipoCuenta** (altura 3). 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.
- Hay 50 bloques de memoria.

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-DEC-2009'
   AND    S.Saldo < 0
```

```
AND     C.NumCuenta = T.NumCuenta
AND     C.TipoCuenta = 5
AND     T.TipoTitular = 3
```

a. Obtenga los árboles canónico y optimizado, 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**.

11.
Dada una relación con un esquema **R** (A(4), B(4) C(4), D(8), E(8), F(8), G(8)) con:
T_R: 1500
Tamaño de bloque: 1024 bytes

Existen índices: un índice clustered sobre A e índices non-clustered (todos los árboles con altura 3) sobre B y CF con:
$I_{R.A}$: 1.500
$I_{R.B}$: 1.000 con rango de [1...1.000]
$I_{R.C}$: 9
$I_{R.F}$: 7 (posibles valores: a1, b1,..., g1)
$I_{R.CF}$: 54

Dada las consultas:

a) SELECT A, B
FROM R
WHERE B >=450 AND C= 1 AND F< 'g1'

b) SELECT A, B
FROM R
WHERE B >=250
AND (C= 1 OR (C= 2 AND F= 'f1'));

Analice los diferentes caminos de acceso y evalúe el costo de cada uno de los mismos, considerando una memoria de 10 bloques.

12.
Dada la siguiente. BD:

R1 (A1, A2): B= 10, I_{A2}=5, archivo *heap*, sin índices, dom(A1)={1,2,...,100}, dom(A2)={20,40,...,100}
R2 (A2, A3): B= 50, I_{A3}= 10, archivo *heap*, sin índices, dom(A2)={1,2, ...,500}, dom(A3)={50,100,...,500}
R3 (A3, A4): B= 100, I_{A4}= 50, archivo *sorted* por A3, índice *unclustered* sobre A4, dom(A3)={1,2,...,1000}, dom(A4)={20,40,...,1000}
R4 (A4, A5): B= 200, I_{A5}= 100, archivo *hashed* por A4 (como máximo 3 bloques por bucket), dom(A4)={1,2,...,2000}, dom(A5)={20,40,...,2000}
R5 (A5, A6): B= 500, I_{A6}= 200, archivo *sorted* por A6, sin índices, dom(A5)={1,2,...,5000}, dom(A6)={25,50,...,5000}
R6 (A6, A7): B= 1.000, I_{A7}= 500, índice *clustered* sobre A6, dom(A6)={1,2,...,10000}, dom(A7)={20,40,...,10000}

- En todos los casos se asume distribución uniforme
- El factor de bloqueo es de 10 tuplas por bloque (FB= 10)

- Los índices son del tipo árbol B+ con 3 niveles (X=3)
- La memoria disponible es de 3 bloques (M=3)
- Los métodos implementados en el SGBD para el operador junta son: *Block Nested Loops Join* (BNLJ), *Index Nested Loops Join* (INLJ) y *Sort-Merge Join* (SMJ)

Obtener el árbol optimizado, derivar un plan de ejecución eficiente y calcular los costos en accesos a bloques para cada una de las siguientes consultas:

a) SELECT A1, A3 FROM R1, R2 WHERE R1.A2=R2.A2;
b) SELECT * FROM R3 WHERE A3<=500 AND A4=120;
c) SELECT DISTINCT A3, A5 FROM R3, R4 WHERE R3.A4=R4.A4;