



*Procesamiento de  
Consultas*

*Parte I*

# Presentación



*Cuando realizamos operaciones en una base de datos, no solamente buscamos eficiencia y consistencia de los datos, sino también buscamos que los tiempos de respuesta sean óptimos y aceptables.*

*Muchas veces se generan bases de datos que inicialmente funcionan correctamente y luego de un período comienza a crecer en volumen de datos o bien aumenta la concurrencia y consecuentemente la performance comienza a degradarse.*

*Por lo cual, constantemente tratamos de optimizar las operaciones que se realizan en una base de datos, ya sea desde el diseño lógico y físico de la base de datos o desde la programación de los procesos que se ejecutan.*

*Un buen diseño, la correcta definición de sus estructuras y una eficiente programación de los procesos SQL, nos ayudarán a obtener una respuesta deseada de la base de datos.*



# Costos de Procesamiento

---

- **Costo de acceso al almacenamiento secundario:** *este es el costo de buscar, leer y escribir bloques de datos en algún medio de almacenamiento, generalmente disco.*
- **Costo de almacenamiento:** *es el costo de almacenar cualquier dato intermedio para poder ejecutar una consulta.*
- **Costo de cómputo:** *es el costo de ejecutar las operaciones de la consulta.*
- **Costo de uso de memoria:** *es el costo relacionado con el uso de la memoria para poder llevar a cabo la consulta.*
- **Costo de comunicación:** *es el costo de enviarlos datos de resultado de dónde se originó la consulta.*

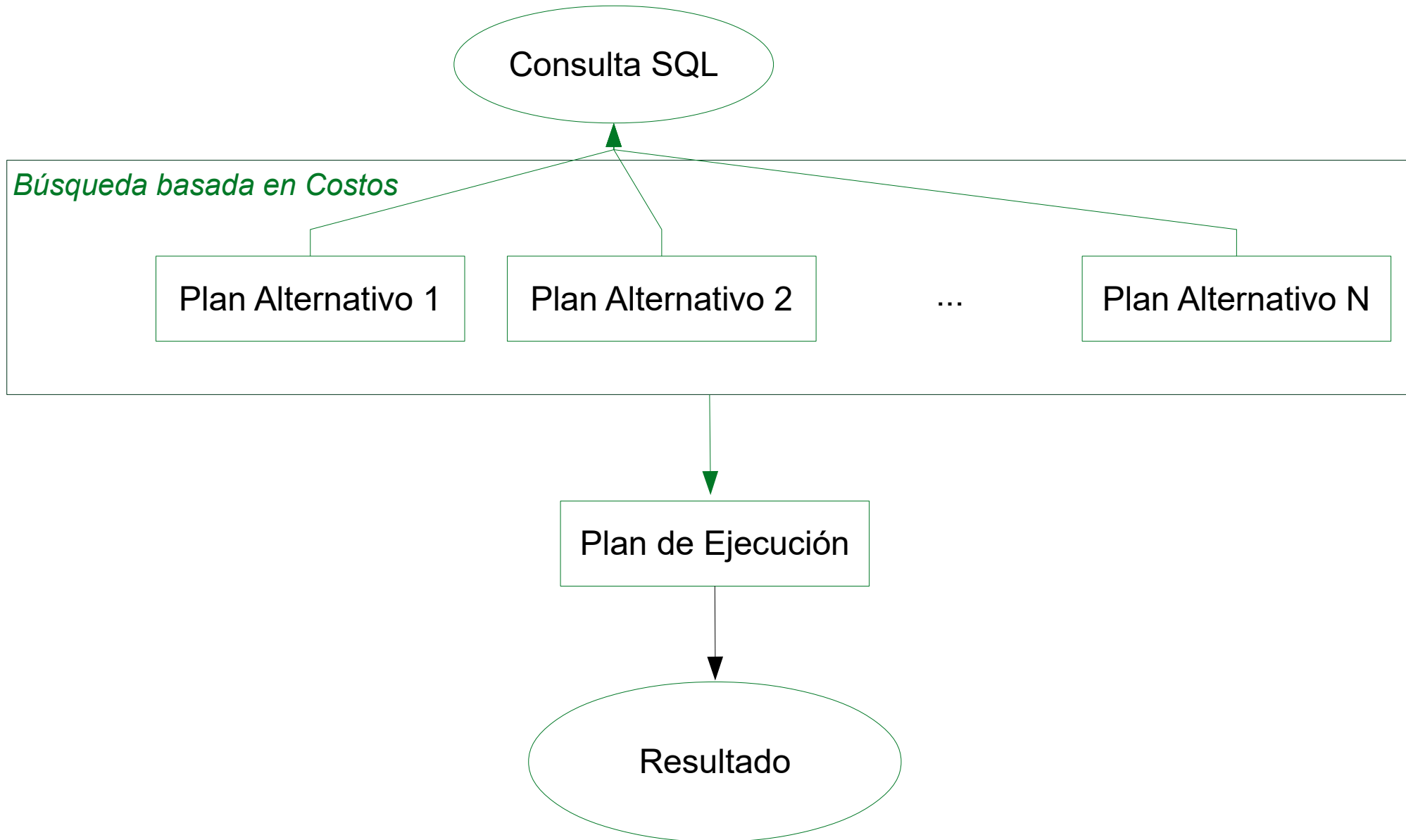
# Procesamiento de consultas



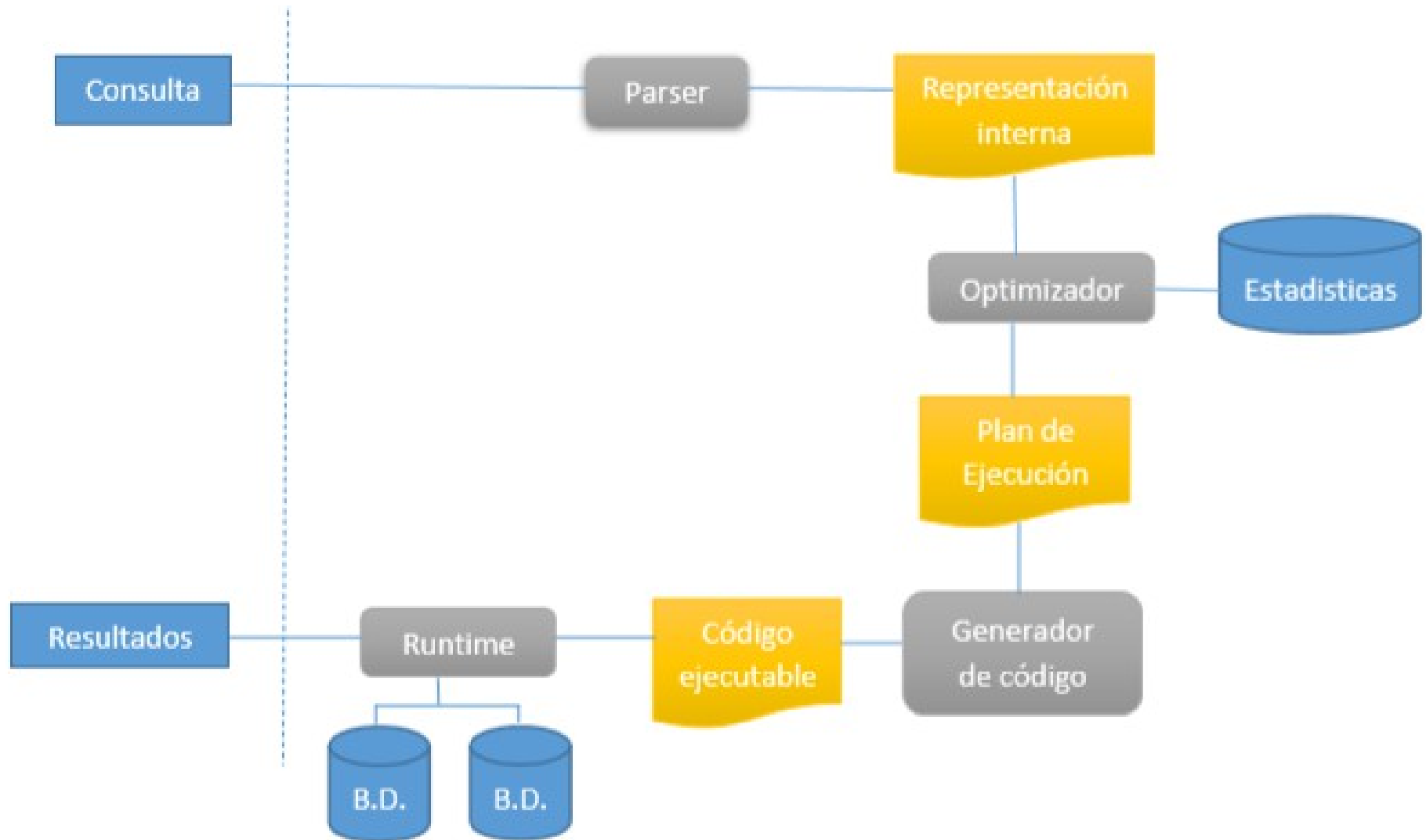
## *¿Cómo hace el SGBD para resolver una consulta?*

- *Se valida la sintáxis y semántica de la consulta SQL.*
- *La consulta validada es convertida a A.R.*
- *El AR obtenido se convierte en un árbol canónico, el cual llamaremos plan de ejecución. El mismo tendrá la forma más óptima para ejecutar la consulta.*
- *El SGBD determinará el algoritmo a utilizar en cada operador del árbol.*

# Plan de Ejecución



# Modelo de Procesamiento



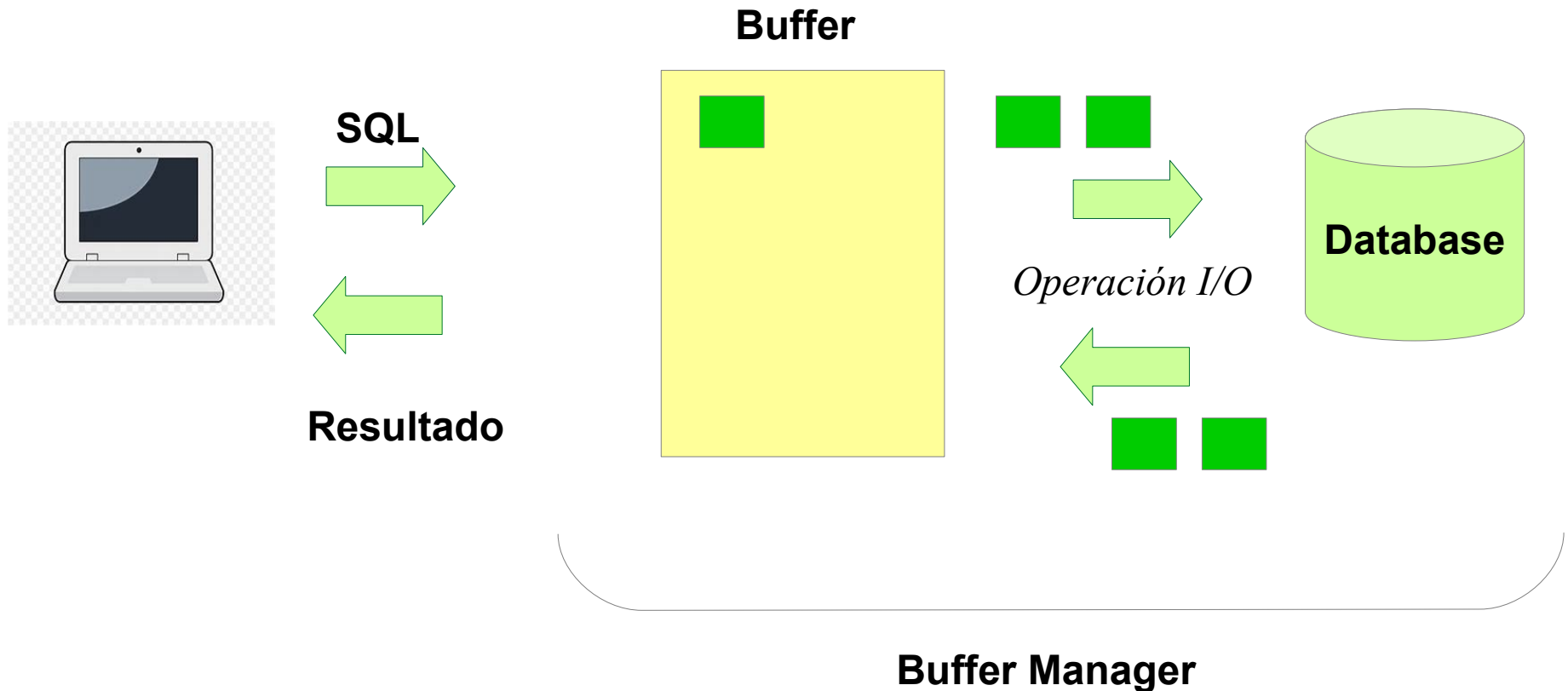
# Unidad de medida de Costos

---



- El costo se mide en Accesos a Bloques.
- El bloque es la mínima unidad de almacenamiento y lectura.
- El bloque puede medir 4 Kb, 8 Kb, 16 Kb, 32 Kb, 64 Kb, por ejemplo.
- Cada vez que el SGBD accede a disco, lee o escribe N bloques.

# Lectura y escritura de datos







# Plan de Ejecución

---

La evaluación del plan de ejecución consiste en verificar:

- Cuál es orden de las operaciones a realizar
- Qué algoritmos utilizará para cada una de las operaciones
- Qué caminos de acceso utilizará para cada una de las tablas/vistas
- Qué datos de entrada se tiene para la evaluación más adecuada del plan (estadísticas).



# Plan de Ejecución: Ejemplo

- **Modelo:**

*Departamento (IDDpto, DDesc, Responsable, Edificio)*  
*Carrera (IDCarrera, CDesc, IDCDpto)*

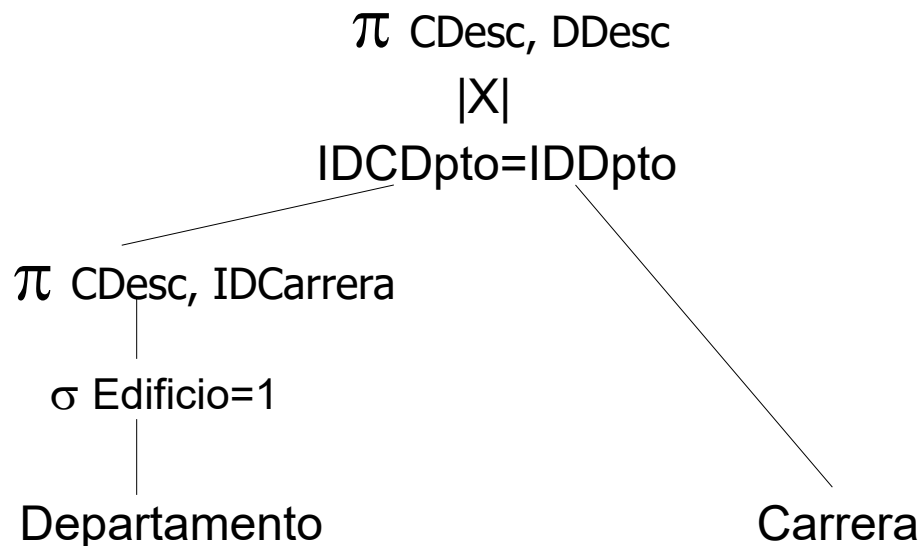
- **Consulta:**

*Select C.CDesc as Carrera, D.DDesc as Departamento*  
*From Departamento D, Carrera C*  
*Where C.IDCDpto=D.IDDpto and Edificio=1*

- **Consulta A.R.:**

$\pi$  Cdesc, Ddesc ( $\sigma$  IDCDpto=IDDpto and Edificio=1 (Departamento X Carrera))

- **Plan de Ejecución:**

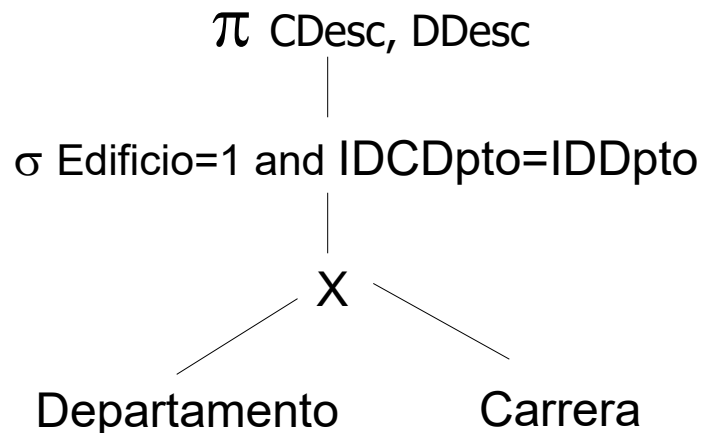




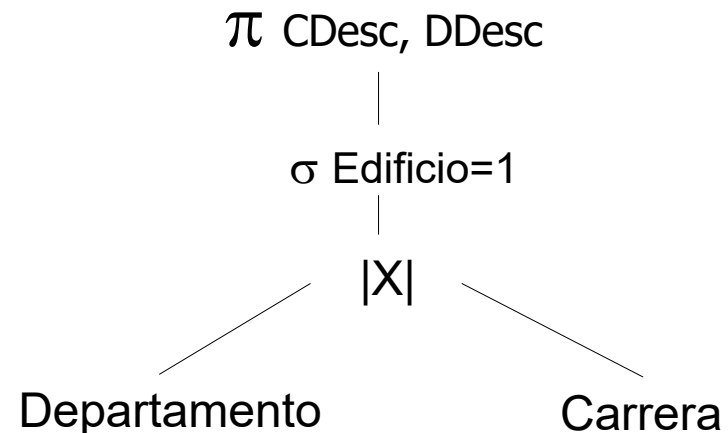
# Plan de Ejecución: Ejemplo

$\pi$  CDesc, DDesc ( $\sigma$  IDCDpto=IDDpto and Edificio=1 (Departamento X Carrera))

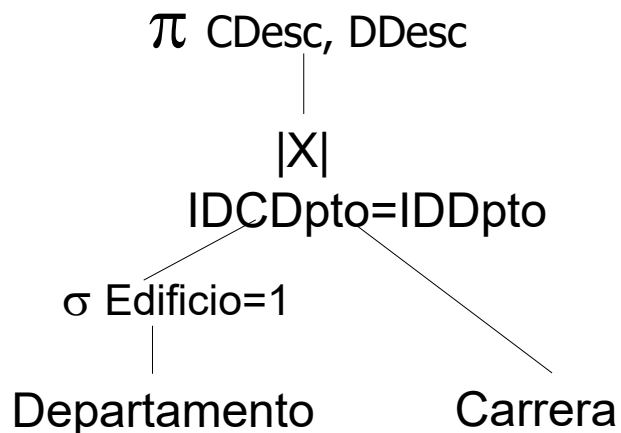
## ● Plan de Ejecución Paso 1:



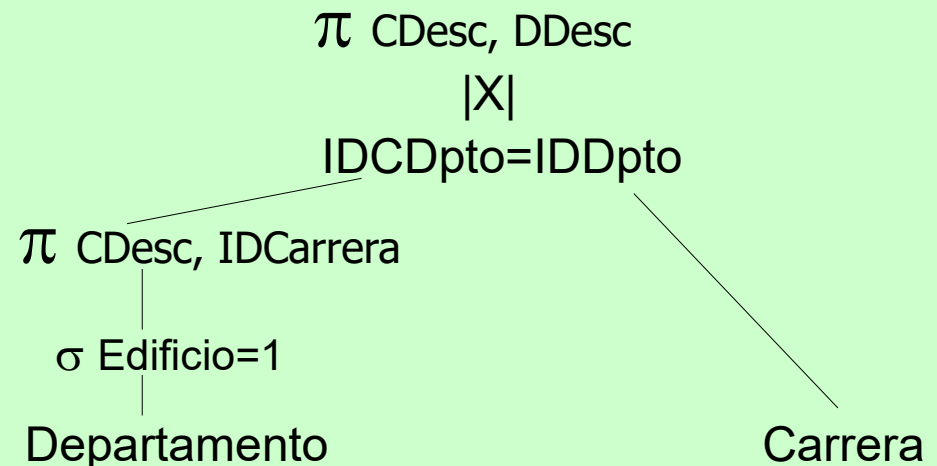
## ● Plan de Ejecución Paso 2:



## ● Plan de Ejecución Paso 3:



## ● Plan de Ejecución Paso 4 (OPTIMIZADO):





# Heurísticas

*Son reglas que, generalmente, ayudarán a optimizar las consultas y le servirán al SGBD para tomar la decisión en la confección del plan de ejecución:*

- *Ejecutar operaciones de selección tan pronto como sea posible.*
- *Ejecutar primero las selecciones más restrictivas (las que producen menor cantidad de tuplas).*
- *Utilizar juntas en lugar de productos cartesianos, si fuera posible.*
- *Ejecutar las operaciones de proyección tan pronto como sea posible.*

# Transformación de expresiones algebraicas



- Propiedad Conmutativa:

Sea  $\otimes$  un operador binario, es conmutativo si  $A \otimes B = B \otimes A$ , para todo  $A$  y  $B$ .

Operador	Evaluación	¿Conmutativo?
→ Unión	$A \cup B = B \cup A$	SI
→ Intersección	$A \cap B = B \cap A$	SI
→ Diferencia	$A - B \neq B - A$	NO
→ Producto	$A \times B = B \times A$	SI
→ Junta	$A \bowtie B = B \bowtie A$	SI
→ Cociente	$A \% B \neq B \% A$	NO
→ Selección	$\sigma_{c_1}(\sigma_{c_2}(A)) = \sigma_{c_2}(\sigma_{c_1}(A))$	SI
→ Proyección	$\pi_a(\pi_b(A)) = \pi_b(\pi_a(A))$	Sólo si $b \subseteq a$

# Transformación de expresiones algebraicas



- **Propiedad Asociativa:**

*Sea  $\otimes$  un operador binario, es asociativo si  $A \otimes (B \otimes C) = (A \otimes B) \otimes C$ , para todo  $A$ ,  $B$  y  $C$ .*

Operador	Evaluación	¿Asociativo?
→ Unión	$A \cup (B \cup C) = (A \cup B) \cup C$	SI
→ Intersección	$A \cap (B \cap C) = (A \cap B) \cap C$	SI
→ Diferencia	$A - (B - C) \neq (A - B) - C$	NO
→ Producto	$A \times (B \times C) = (A \times B) \times C$	SI
→ Junta	$A  x  (B  x  C) = (A  x  B)  x  C$	SI
→ Cociente	$A \% (B \% C) \neq (A \% B) \% C$	NO

# Transformación de expresiones algebraicas



- **Propiedad Idempotencia:**

*Sea  $\otimes$  un operador binario, es idempotente si  $A \otimes A = A$ , para todo  $A$ .*

Operador	Evaluación	¿Idempotente?
→ Unión	$A \cup A = A$	SI
→ Intersección	$A \cap A = A$	SI
→ Diferencia	$A - A \neq A$	NO
→ Producto	$A \times A \neq A$	NO
→ Junta	$A \parallel A = A$	SI
→ Cociente	$A \% A \neq A$	NO

# Transformación de expresiones algebraicas



- **Propiedad Distributiva:**

Sea  $\otimes$  y  $\oplus$  operadores binarios, es distributivo si  $A \otimes (B \oplus C) = (A \otimes B) \oplus (A \otimes C)$ , para todo  $A$ ,  $B$  y  $C$ .

Operador	Evaluación	¿Distributiva?
→ <b>Proyección y Junta</b>	$\pi_{ab}(A \mid X \mid B) = \pi_a(A \mid X) \mid \pi_b(B)$	SI (se deben incluir los atributos de la junta)
→ <b>Selección y Junta</b>	$\sigma_{ca \text{ and } cb}(A \mid X \mid B) = \sigma_{ca}(A \mid X) \mid \sigma_{cb}(B)$	SI (sólo si ambas condiciones corresponden a cada Relación)
→ ...		



# Transformación de expresiones algebraicas



- **Propiedad Cascada:**

*Una condición de selección conjuntiva puede descomponerse en una cascada de operaciones individuales o viceversa.*

Operador	Evaluación	¿Cascada?
→ Selección	$\sigma_{ca \text{ and } cb}(A) = \sigma_{ca}(\sigma_{cb}(A))$ $\sigma_{ca}(\sigma_{cb}(A)) = \sigma_{ca \text{ and } cb}(A)$	SI

# Arbol canónico



- **Modelo:**

*Departamento (IDDpto, DDesc, Responsable, Edificio)*

*Carrera (IDCarrera, CDesc, IDCDpto)*

*Docente (Legajo, Nombre, Apellido, FechaIngreso, mail, IDDCarrera)*

- **Consulta:**

*Select C.CDesc as Carrera, D.DDesc as Departamento, DO.mail*

*From Departamento D, Carrera C, Docente DO*

*Where C.IDCDpto=D.IDDpto and DO.IDDCarrera=C.IDCarrera*

*and D.Edificio=1*

*and DO.Apellido like 'Perez%' and DO.FechaIngreso>='2015-01-01'*

- **Consulta A.R.:**

$\pi$  Cdesc, Ddesc, mail (

$\sigma$  IDCDpto=IDDpto and IDDCarrera=IDCarrera and Edificio=1 and Apellido like 'Perez%' and FechaIngreso>='2015-01-01' (Departamento X Carrera X Docente))

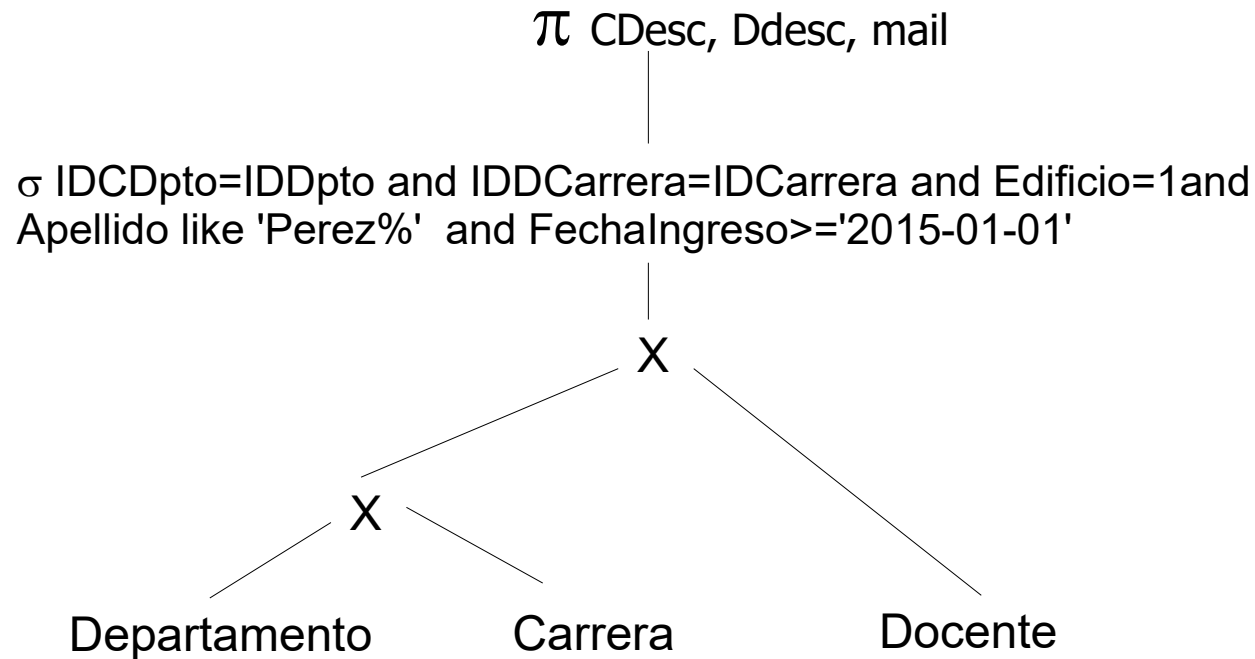


# Arbol canónico: Diseño

## ● Consulta A.R.:

$\pi$  Cdesc, Ddesc, mail (

$\sigma$  IDCDpto=IDDpto and IDDCarrera=IDCarrera and Edificio=1 and Apellido like 'Perez%' and FechaIngreso>='2015-01-01' (Departamento X Carrera X Docente))



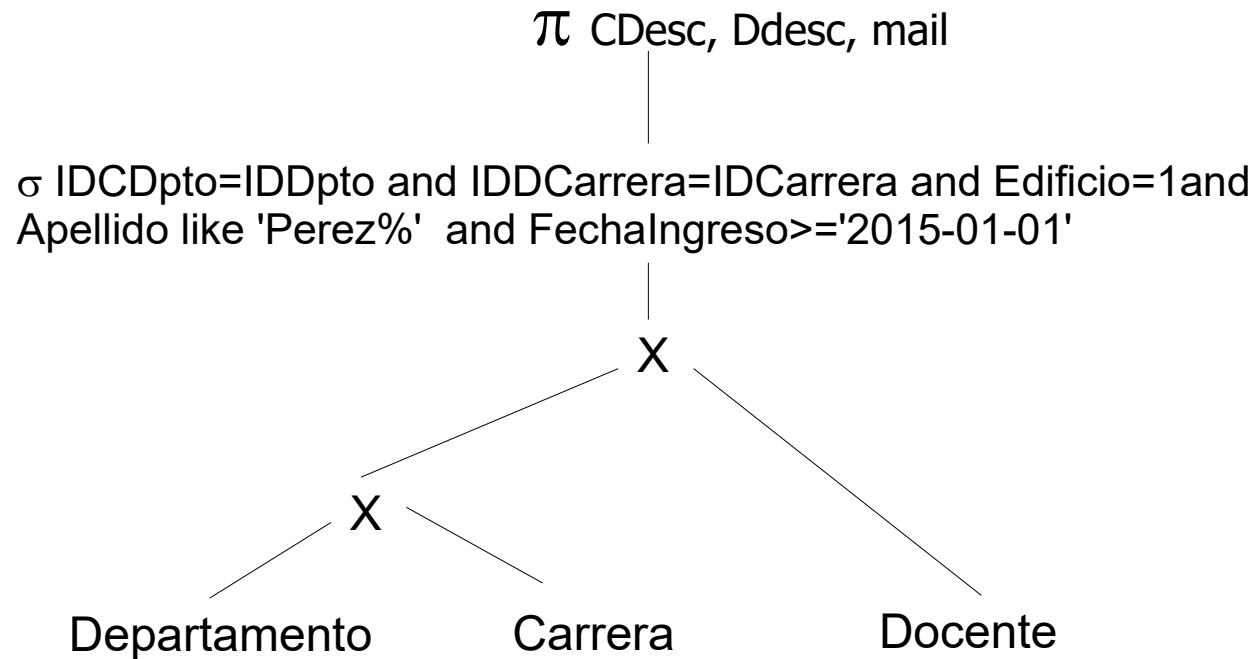


# Arbol canónico: Diseño

## ● Consulta A.R.:

$\pi$  Cdesc, Ddesc, mail (

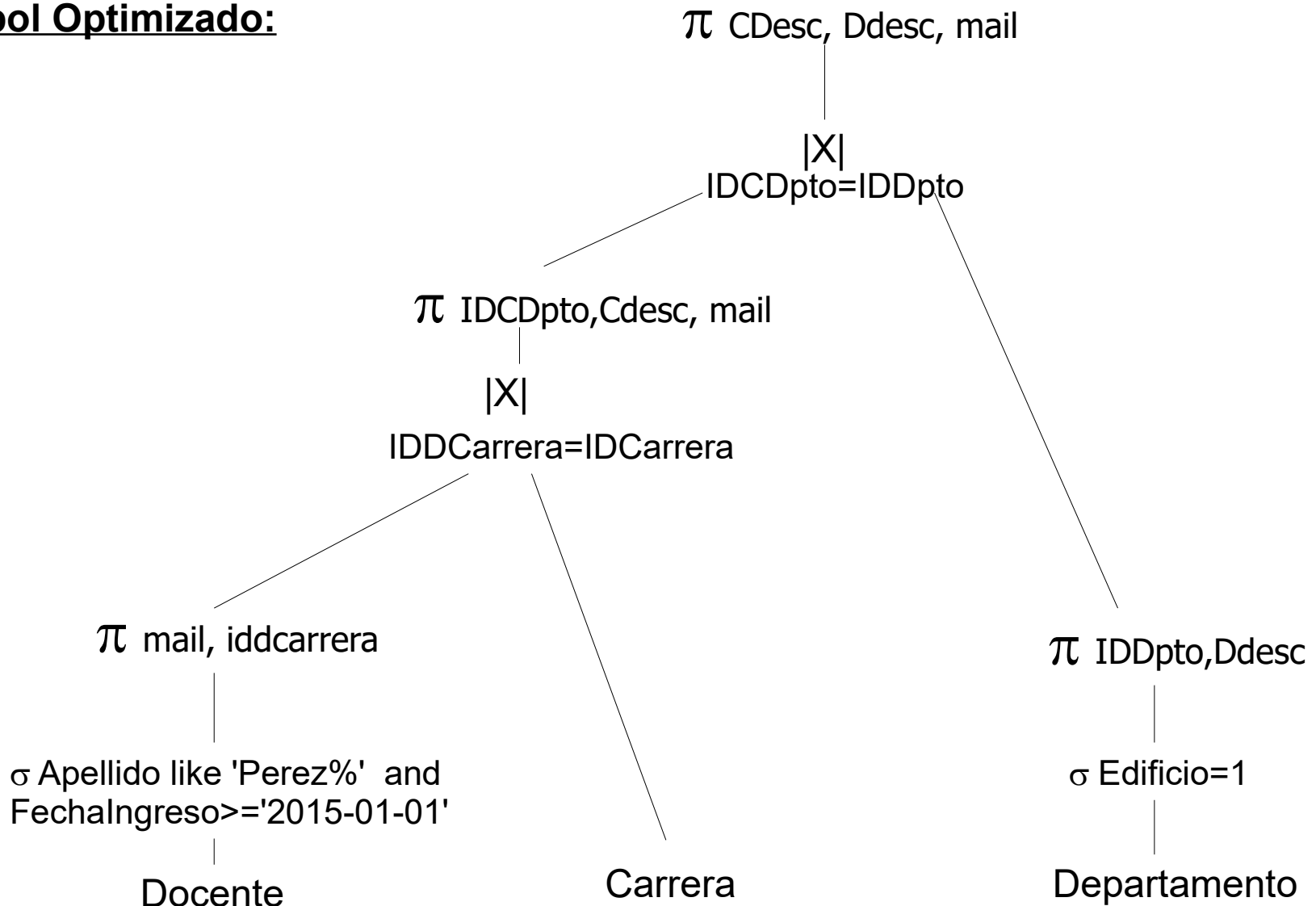
$\sigma$  IDCDpto=IDDpto and IDDCarrera=IDCarrera and Edificio=1 and Apellido like 'Perez%' and FechaIngreso>='2015-01-01' (Departamento X Carrera X Docente))



# Árbol canónico: Diseño



- Árbol Optimizado:





# Tipos de Costos

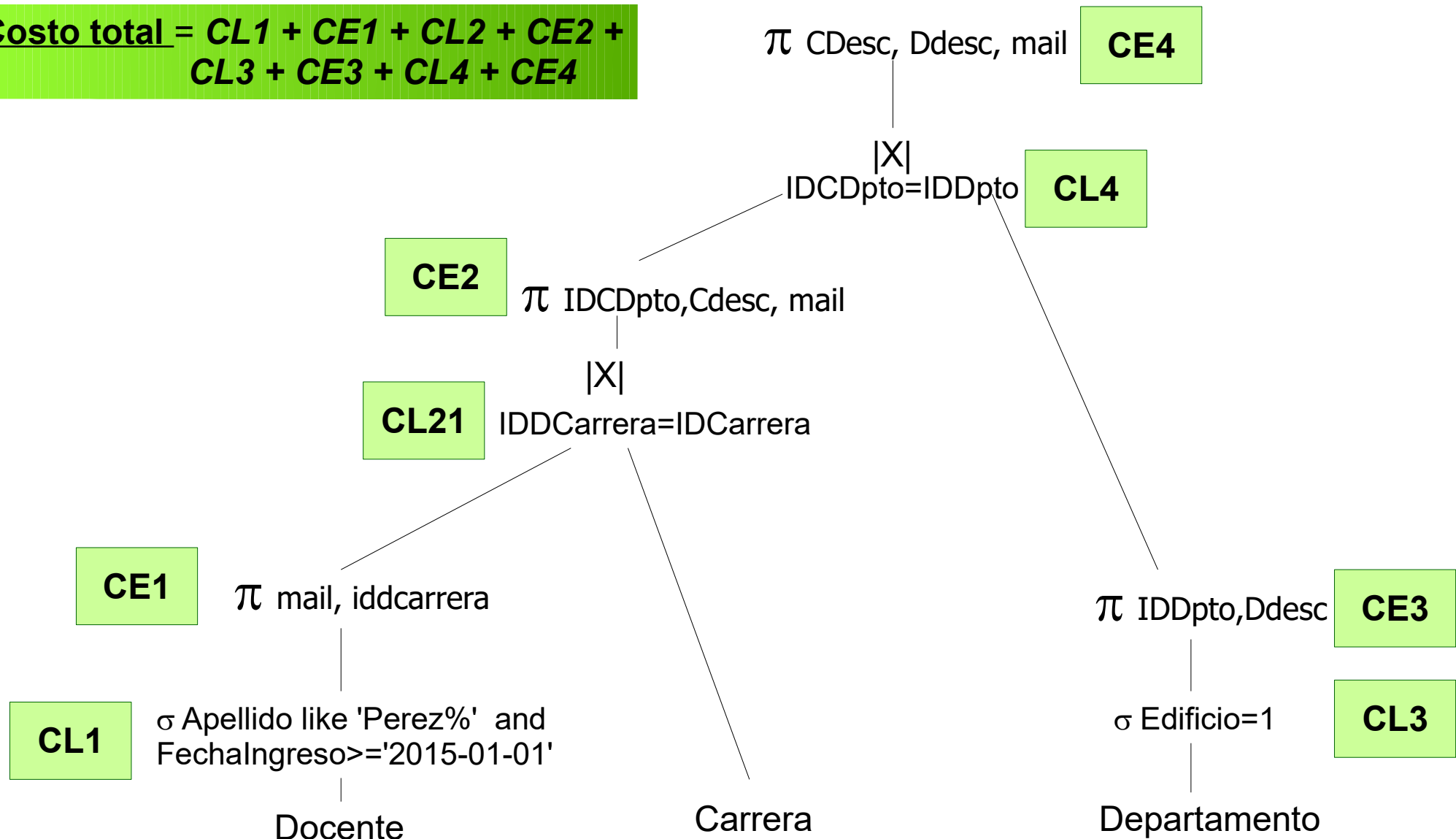
---

- **Costo de Escritura:** *Cuando escribimos bloques a disco.*
- **Costo de Lectura:** *Cuando leemos bloques del disco.*
- **Costo total =** *Costo de Lectura + Costo de Escritura*



# Arbol canónico: Diseño

**Costo total = CL1 + CE1 + CL2 + CE2 + CL3 + CE3 + CL4 + CE4**



# Variables de la estimación de costos



- **Tr:** *Cantidad de tuplas de la relación R.*
- **Br:** *Cantidad de bloques de la relación R.*
- **FBr** = *Factor de Bloqueo. Indica la cantidad de tuplas por bloque de la relación R.*
- **V(A,r):** *Variabilidad o Imagen del atributo A en la instancia de relación R. Es la cantidad de valores distintos que tiene A en r(R).*
- **Lr:** *Longitud máxima de una tupla de la relación R.*
- **CL:** *Costo de lectura.*
- **CE:** *Costo de Escritura.*
- **M:** *Cantidad de bloques de memoria disponibles.*
- **TB:** *Tamaño del bloque*





*Las estadísticas son objetos binarios que contienen información estadísticas sobre la distribución de los valores de una o más columnas de una tabla dentro de la base de datos. El Optimizador utiliza esas estadísticas para estimar la cardinalidad del resultado de una consulta.*

Parte de la información que almacena una estadística es:

- $n_r$ : número de tuplas de la relación R
- $b_r$ : número de bloques de R
- $l_r$ : tamaño de la tupla de R
- $f_r$ : factor de bloqueo de R, es decir, el número de tuplas de R que entran en un bloque.
- $\sigma_A = V(r)$  = número de filas que satisfacen la condición ( $n_r/V(A,r)$ ). Lo ideal sería que este valor sea 1.
- Mínimo valor
- Máximo valor
- Cantidad de valores nulos



# Estadísticas: Creación

- **Configuración:**

- ➔ **Auto\_Update\_Stats:** *Permite actualizar automáticamente las estadísticas cuando el motor lo determine en base a los cambios sufridos en los objetos.*
- ➔ **Auto\_Create\_Stats:** *Permite crear automáticamente estadísticas que el motor considera necesario que permitan resolver mejor la consulta.*

- **Sintaxis Simple:**

```
CREATE STATISTICS nombreStats ON {table|view} (column  
[ ,...n ]) [WITH FULLSCAN]
```

- **Ejemplo Simple:**

```
CREATE STATISTICS ST_CLIENTE ON CLIENTE (NRO_CLIENTE)
```

# Estadísticas: Mantenimiento



- **Configuración:**

- ➔ **Auto\_Update\_Stats:** *Si esta opción se encuentra habilitada, el propio motor se encargará de la actualización de las estadísticas, según el algoritmo que cada SGBD tenga diseñado.*

- **Sintaxis Simple:**

```
UPDATE STATISTICS tabla [({nombre_stats | nombre_index})]  
[WITH {FULLSCAN | SAMPLE number PERCENT}]
```

- **Ejemplo Simple:**

```
UPDATE STATISTICS CLIENTE WITH FULLSCAN  
UPDATE STATISTICS CLIENTE (ST_NROCLIENTE) WITH SAMPLE 10 PERCENT
```

# Indices



- Son estructuras generadas sobre una clave de búsqueda para una tabla ó vista.
- Se elegirán sobre qué atributos se realizarán los índices.
- Se podrán crear múltiples índices sobre la misma tabla/vista y cada índice contendrá uno o varios atributos.
- La entrada del índice tendrá un apuntador que dirá dónde buscar la fila en la tabla, es decir, los índices tienen un puntero al registro que le dirá en qué bloque se encuentra y cuál es el offset dentro de ese bloque.





# Indices: Clasificación

- **Índices Ordenados:** *Son estructuras de índices que se encuentran previamente ordenados. Son utilizados en atributos que tengan una inserción en forma secuencial.*
- **Índices Hash:** *Son estructuras de índices que no se encuentran ordenados sino que se basan en una función hash para poder distribuir los valores.*

Los índices ordenados poseen una subclasificación:

- **Índices densos:** *En este caso cada valor de cada fila aparece dentro del índice.*
- **Índices dispersos:** *En el índice se posee una entrada sólo para algunos valores de las filas. Se busca la clave de acceso más cercana al valor buscado y luego se recorre a partir de ese valor hasta encontrar el deseado, dado que están ordenados.*



# Indices: Creación

- **Sintaxis Simple:**

```
CREATE [UNIQUE] [CLUSTERED|NONCLUSTERED] INDEX  
<nombredelindice> ON <tabla|vista> (campo1, campo2 ...,campoN)
```

- **Ejemplo Simple:**

```
CREATE CLUSTERED INDEX  IDX1_ALUMNO ON ALUMNO (FECHAINGRESO)
```

```
CREATE NONCLUSTERED INDEX  IDX2_ALUMNO ON ALUMNO (DOC, TIPODOC)
```

```
CREATE UNIQUE INDEX  IDX3_CLIENTE ON CLIENTE (CUIT)
```



# Indices: Mantenimiento

---

- **Sintaxis Simple:**

```
ALTER INDEX <nombredelindice> ON <nombredelatabla> REBUILD
```

- **Ejemplo Simple:**

```
ALTER INDEX idx1_ITEMFACTURA ON ITEMFACTURA REBUILD
```



# Indices: Consejos I

---

- Es conveniente crear índices en las tablas que tengan un volumen que amerite realizarlo.
- Las claves primarias siempre tendrán un índice asociado, por lo cual no se deben generar índices de iguales atributos a la clave, ya que serían redundantes.
- Se deben verificar los diferentes WHEREs que se emplean en las consultas, ya que frecuentemente los predicados son los candidatos a la creación de los índices.
- Los campos que se utilicen en los ORDER BY o en los GROUP BY deben ser evaluados como candidatos a índices, aunque se trata de evitar el uso excesivo de dichas sentencias en las consultas.





# Indices: Consejos II

- Los índices que generan para resolver condiciones de comparación  $>$  y  $<$ , es conveniente crearlo cuando los predicados sean de  $>$  ó de  $>$  and  $<$ , pero no cuando sea únicamente de  $<$ .
- Los valores nulos no están indexados, por lo cual es conveniente evitar el uso de dichos valores en las tablas. En muchas oportunidades, se puede utilizar un valor por default para evitarlos.
- Se debe tomar recaudos en aquellas tablas que sufran alteraciones constantes, ya que cada vez que una tabla actualiza sus datos, también debe actualizar todos los índices asociados y este tiempo aumentará en la medida que se agreguen más índices.



# Indices: Consejos III

- El uso de un índice para un predicado, debería arrojar un pequeño porcentaje de filas de la tabla. Sino, consideramos que el índice no tiene una buena variabilidad en muchas situaciones resultará más ineficiente utilizar el índice que acceder directamente a la tabla.
- Los índices deben estar actualizados. Por lo cual, el motor de base de datos estará modificándolo en cada operación de eliminación, inserción o actualización que se realice. Esto provocará tiempos extras en cada una de las operaciones. Si la tabla posee muchas alteraciones, se deberá evitar el uso excesivo de índices para no perjudicar la performance de las operaciones.
- Siempre será necesario verificar que los índices que se han generado estén siendo utilizados por el motor de base de datos. Podría ser que el motor opte por no utilizarlo, y estaríamos perjudicando a las modificaciones sin beneficiar a las consultas.

# Consultas

---



**Gracias.**