

BASES DE DATOS

The background of the slide features several overlapping, wavy lines in orange, light blue, and lime green, creating a dynamic and modern aesthetic.

CLASE 7

Optimización de consultas

- **Costo de ejecución** de una consulta
 - Costo de **acceso** a almacenamiento secundario
 - Buscar, leer y escribir bloques de datos que residen en disco
 - Costo de **almacenamiento**
 - Almacenar archivos intermedios que generan una estrategia de ejecución
 - Costo de **cómputo**
 - Costo de realizar las operaciones
 - Costo de **comunicación**
 - Costo de enviar la consulta y los resultados (sist. distribuidos)

Optimización de consultas

- **Costo de ejecución** de una consulta
- Dada una consulta hay varios métodos para llegar a la respuesta



Optimización de consultas

- **Pasos de la optimización**
 - **Analizador sintáctico (parser)** → convierte la consulta en una estructura tipo árbol
 - **Optimizador** → a partir de la expresión interna de la consulta obtiene una consulta equivalente **más eficiente**
 - Finalmente, se toma en cuenta el **estado actual de la BD** y los **índices definidos** para resolver la consulta

Optimización de consultas

- **Estado temporal** de una tabla de la BD
 - CT_{tabla1} \rightarrow cantidad de tuplas de **tabla1**
 - CB_{tabla1} \rightarrow tamaño en bytes de una tupla de **tabla1**
 - $CV_{(a, \text{tabla1})}$ \rightarrow cantidad de valores diferentes que existen de **a** en **tabla1**

Optimización de consultas

- Estimación de costo de consultas
 - Estadísticas de la BD $\rightarrow CT_R, CB_R, CV_{(a, R)}$
 - Son útiles para **decidir rápidamente el costo** de cada alternativa de solución y así elegir la mejor
 - Están **parcialmente** actualizadas
 - Ej. de costo de una consulta: **Producto cartesiano R x T**
 - Cantidad de tuplas = $CT_R * CT_T$
 - Cantidad de bytes por tupla = $CB_R + CB_T$

Optimización de consultas

- **Optimización lógica**
 - Expresiones equivalentes en Álgebra Relacional
 - El **AR** es un lenguaje procedural, por lo que cuando se escribe una consulta existe una **secuencia de resolución**
 - Se puede encontrar que algunas expresiones son **más eficientes** que otras equivalentes

Optimización de consultas

- Optimización lógica

- Selección

- Hacerlas lo antes posible

- $\sigma_p(R \bowtie T) \rightarrow \sigma_p(R) \bowtie \sigma_p(T)$

- $\sigma_p(R \cup T) \rightarrow \sigma_p(R) \cup \sigma_p(T)$

- $\sigma_p(R - T) \rightarrow \sigma_p(R) - \sigma_p(T)$

- Descomponer la condición

Optimización de consultas

- Optimización lógica

Alumnos

Idalumno	NombreAlumno	Dni	Idlocalidad	Idcarrera
1	García	23456876	1	1
2	Perez	17876234	2	1
3	Gomez	33009876	1	2
4	Pizarro	25678965	1	2
5	Castelli	14239034	2	1
6	Pettorutti	19023487	3	2
7	Montenzanti	23434564	2	3
8	Suarez	30212305	3	1

Localidades

Idlocalidad	NombreLocalidad
1	General Pico
2	La Plata
3	Tres Arroyos

Carreras

Idcarrera	NombreCarrera	Duración_años
1	Informatica	5
2	Quimica	3
3	Contador	4

Optimización de consultas

- Optimización lógica

- Ej1: obtener los DNI de alumnos de 'La Plata'

$\pi_{dni} (\sigma_{Localidades.nombre = "La Plata"} (Alumnos \bowtie Localidades))$

Alumnos \bowtie Localidades

Idalumno	NombreAlumno	DNI	Idlocalidad	Idcarrera	Idlocalidad	NombreLocalidad
1	García	23456876	1	1	1	General Pico
2	Perez	17876234	2	1	2	La Plata
3	Gomez	33009876	1	2	1	General Pico
4	Pizarro	25678965	1	2	1	General Pico
5	Castelli	14239034	2	1	2	La Plata
6	Pettorutti	19023487	3	2	3	Tres Arroyos
7	Montenzanti	23434564	2	3	2	La Plata
8	Suarez	30212305	3	1	3	Tres Arroyos

Optimización de consultas

- **Optimización lógica**

- Ej1: obtener los DNI de alumnos de 'La Plata'

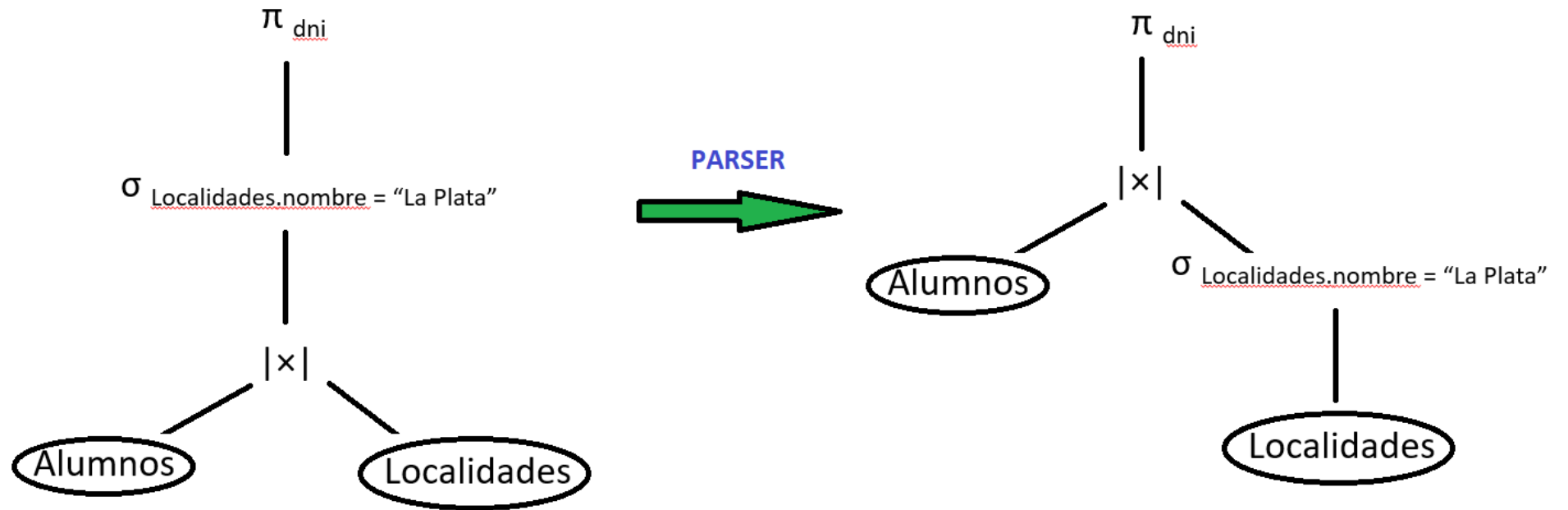
$\pi_{dni} ((\text{Alumnos} \mid \times \mid \sigma_{\text{nombre} = \text{"La Plata"}} (\text{Localidades}))$

- Solo genera 3 tuplas, contra las 8 de la consulta anterior
- La operación de selección debe realizarse antes de un producto, una unión o una diferencia

Optimización de consultas

- Optimización lógica

- Ej1: obtener los DNI de alumnos de 'La Plata'



Optimización de consultas

- **Optimización lógica**

- Ej2: nombre de alumnos con idLocalidad=1 e idCarrera=3

$\pi_{\text{nombre}} (\sigma_{\text{idCarrera}=3 \text{ AND } \text{idLocalidad}=1} (\text{Alumnos}))$

- Supongamos que la tabla ALUMNOS tiene 1000 tuplas
 - Se aplican ambas comparaciones resolviéndose 2000 expresiones lógicas
 - Luego se resuelve el operador AND entre los resultados → 1000 expresiones lógicas adicionales
 - TOTAL: **3000** expresiones lógicas

Optimización de consultas

- **Optimización lógica**

- Ej2: nombre de alumnos con idLocalidad=1 e idCarrera=3

$\pi_{\text{nombre}} (\sigma_{\text{idCarrera}=3} (\sigma_{\text{idLocalidad}=1} (\text{Alumnos})))$

- Supongamos que la tabla ALUMNOS tiene 1000 tuplas
 - Se resuelve la selección de localidad.
 - Peor caso: todos son de Idlocalidad = 1. Quedan 1000 tuplas
 - Caso esperable: un 20% son de esa localidad. Quedan 200 tuplas
 - Se resuelve la selección de carrera.
 - TOTAL peor caso: $1000 + 1000 = 2000$ expresiones lógicas
 - TOTAL caso esperable: $1000 + 200 = 1200$ expresiones lógicas

Optimización de consultas

- Optimización lógica

- Proyección

- Resolver las proyecciones lo antes posible

- Producto natural

- Resolver en pasos: $R \mid x \mid T \mid x \mid S \rightarrow R \mid x \mid (T \mid x \mid S)$
 - Orden: $R \mid x \mid T$ vs $T \mid x \mid R$

Optimización de consultas

- **Optimización lógica**

- ¿ $T1 \mid X \mid T2 = T2 \mid X \mid T1$?

- El atributo **A** en común entre ambas tablas es clave primaria en T1, y clave foránea en T2.

- $T1 = (\underline{A}, B, C, D)$ $T2 = (\underline{G}, F, A)$

- Resolver $T2 \mid x \mid T1 \rightarrow$ **más eficiente**

Optimización de consultas

- **Optimización lógica**

- ¿ $T1 \bowtie X T2 = T2 \bowtie X T1$?

- El atributo A en común entre ambas tablas no es clave primaria en ninguna de las tablas
 - $T1 = (\underline{B}, C, D, E, A)$ $T2 = (\underline{G}, F, A)$
- Es **más eficiente** resolver el producto sobre la tabla que tenga mayor cantidad de valores diferentes del atributo común (CV_A)
 - Si por ej. $CV_A(T2) > CV_A(T1)$ entonces $T1 \bowtie X T2$ es más eficiente

Optimización de consultas

- **Optimización física**
 - **Técnicas de acceso adecuadas**
 - Dispersión / Índices
 - **Procesadores paralelos**
 - Mayor capacidad de computo
 - **Redes**
 - Separar el trabajo entre distintas computadoras