



UNIVERSIDAD DE SEVILLA
Escuela Técnica Superior
de Ingeniería Informática
Departamento de Lenguajes
y Sistemas Informáticos

Procesamiento y Optimización de Consultas

*Grupo de Ingeniería del Software y Bases de Datos
Departamento de Lenguajes y Sistemas Informáticos
Universidad de Sevilla
septiembre 2016*



UNIVERSIDAD DE SEVILLA
Escuela Técnica Superior
de Ingeniería Informática
Departamento de Lenguajes
y Sistemas Informáticos

Procesamiento y Optimización de Consultas

- **Objetivos de este tema**
 - Conocer las fases del **procesado de consultas** en bases de datos relacionales.
 - Conocer los principales **factores** que afectan a la **eficiencia** de una consulta en bases de datos relacionales.
 - Ser capaz de escribir **consultas eficientes** a bases de datos relacionales aplicando simplificación de expresiones, equivalencias de álgebra relacional y heurísticas.

1. Procesado de consultas
2. Coste de una consulta
3. Simplificación de expresiones
4. Equivalencias en AR
5. Heurísticas de optimización



septiembre 2016

Introducción a la Ingeniería del Software y a los Sistemas de Información

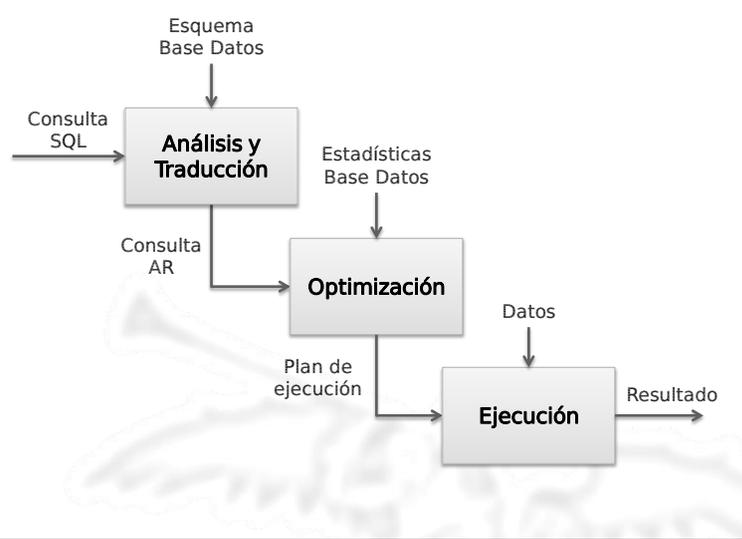
1



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Procesamiento y Optimización de Consultas

- Fases habituales del procesado de consultas



```

    graph LR
      SQL[Consulta SQL] --> AT[Análisis y Traducción]
      EBD[Esquema Base Datos] --> AT
      AT --> AR[Consulta AR]
      AR --> O[Optimización]
      EBD --> O
      O --> PE[Plan de ejecución]
      PE --> E[Ejecución]
      D[Datos] --> E
      E --> R[Resultado]
    
```

1. Procesado de consultas
2. Coste de una consulta
3. Simplificación de expresiones
4. Equivalencias en AR
5. Heurísticas de optimización

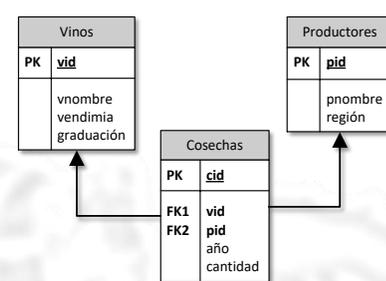
septiembre 2016 Introducción a la Ingeniería del Software y a los Sistemas de Información 2



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Procesamiento y Optimización de Consultas

- Ejemplo de procesado de consulta
 - Supongamos una base de datos relacional sobre vinos, productores y cosechas donde las estadísticas son las siguientes:
 - Vinos: 500 tuplas
 - Cosechas: 1.200 tuplas
 - $\sigma_{cantidad > 100}(Cosechas)$: 100 tuplas



```

    erDiagram
        Vinos ||--o{ Cosechas : "FK1"
        Cosechas ||--o{ Productores : "FK2"
        Vinos {
            string vid PK
            string vnombre
            int vendimia
            float graduación
        }
        Cosechas {
            string cid PK
            string vid FK1
            int año FK2
            int cantidad
        }
        Productores {
            string pid PK
            string pnombre
            string región
        }
    
```

1. Procesado de consultas
2. Coste de una consulta
3. Simplificación de expresiones
4. Equivalencias en AR
5. Heurísticas de optimización

septiembre 2016 Introducción a la Ingeniería del Software y a los Sistemas de Información 3



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Procesamiento y Optimización de Consultas

- Ejemplo de procesado de consulta
 - Queremos ejecutar la siguiente consulta en SQL:


```
select V.vnombre from V Vinos, C Cosechas
                    where V.vid = C.vid and C.cantidad > 100;
```
 - El primer paso es traducir la consulta a álgebra relacional.
 - **Opción 1:** join, selección, proyección

$$\Pi_{vnombre} (\underbrace{\underbrace{\sigma_{cantidad > 100} (Cosechas \bowtie Vinos)}_{1.200 \text{ tuplas}}}_{100 \text{ tuplas}})$$

septiembre 2016

Introducción a la Ingeniería del Software y a los Sistemas de Información

4



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Procesamiento y Optimización de Consultas

- Ejemplo de procesado de consulta
 - Queremos ejecutar la siguiente consulta en SQL:


```
select V.vnombre from V Vinos, C Cosechas
                    where V.vid = C.vid and C.cantidad > 100;
```
 - El primer paso es traducir la consulta a álgebra relacional.
 - **Opción 2:** selección, join, proyección

$$\Pi_{vnombre} (\underbrace{\underbrace{\sigma_{cantidad > 100} (Cosechas)}_{100 \text{ tuplas}} \bowtie Vinos)}_{100 \text{ tuplas}})$$

septiembre 2016

Introducción a la Ingeniería del Software y a los Sistemas de Información

5



UNIVERSIDAD DE SEVILLA
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Procesamiento y Optimización de Consultas

- Ejemplo de procesado de consulta
 - Queremos ejecutar la siguiente consulta en SQL:


```
select V.vnombre from V Vinos, C Cosechas
                    where V.vid = C.vid and C.cantidad > 100;
```
 - El primer paso es traducirla a álgebra relacional.
 - **Opción 3:** producto cartesiano, selección, proyección

$$\Pi_{vnombre} \left(\sigma_{\substack{Cosechas.vid=Vinos.vid \\ \wedge cantidad > 100}} (Cosechas \times Vinos) \right)$$

600.000 tuplas
100 tuplas

septiembre 2016

Introducción a la Ingeniería del Software y a los Sistemas de Información

6



UNIVERSIDAD DE SEVILLA
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Procesamiento y Optimización de Consultas

- Ejemplo de procesado de consulta
 - ¿Cuál de las tres consultas en álgebra relacional es más eficiente? ¿Cuál seleccionaría? ¿Por qué?

$$\Pi_{vnombre} \left(\sigma_{cantidad > 100} (Cosechas \bowtie Vinos) \right)$$

1.200 tuplas
100 tuplas

$$\Pi_{vnombre} \left(\left(\sigma_{cantidad > 100} Cosechas \right) \bowtie Vinos \right)$$

100 tuplas
100 tuplas

$$\Pi_{vnombre} \left(\sigma_{\substack{Cosechas.vid=Vinos.vid \\ \wedge cantidad > 100}} (Cosechas \times Vinos) \right)$$

600.000 tuplas
100 tuplas



septiembre 2016

Introducción a la Ingeniería del Software y a los Sistemas de Información

7



UNIVERSIDAD DE SEVILLA
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Procesamiento y Optimización de Consultas

- **Coste de una consulta**
 - El coste, en tiempo de ejecución, de una consulta a una base de datos depende de:
 - **Tiempo de acceso al sistema de E/S** (90%)
 - Tiempo de procesamiento de CPU (10%)
 - El tiempo de acceso al sistema de E/S depende de:
 - **Volumen de datos:** número y tamaño de las tuplas, tanto de las relaciones involucradas como de los resultados intermedios
 - **Organización física:** índices, *tablespaces*, ...
 - Tamaño de los **buffers en memoria** para almacenar los resultados intermedios



1. Procesado de consultas

2. Coste de una consulta

3. Simplificación de expresiones

4. Equivalencias en AR

5. Heurísticas de optimización

septiembre 2016

Introducción a la Ingeniería del Software y a los Sistemas de Información

8



UNIVERSIDAD DE SEVILLA
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Procesamiento y Optimización de Consultas

- **Para reducir el coste de una consulta...**
 - Reducir los accesos al sistema de E/S
 - **Reducir los resultados intermedios:** menos accesos y más probabilidad de que quepan en los buffers en memoria.*
 - **Seleccionar por atributos indexados:** accesos mucho más eficientes y búsquedas mucho más rápidas.
 - Reducir el procesamiento de CPU
 - **Simplificar expresiones de selección:** ahorran tiempo de procesamiento y, a veces, accesos al sistema de E/S.



1. Procesado de consultas

2. Coste de una consulta

3. Simplificación de expresiones

4. Equivalencias en AR

5. Heurísticas de optimización

septiembre 2016

Introducción a la Ingeniería del Software y a los Sistemas de Información

9

* Si los resultados intermedios no caben en los buffers en memoria, deben almacenarse en disco.



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

1. Procesado de consultas
2. Coste de una consulta
3. Simplificación de expresiones
4. Equivalencias en AR
5. Heurísticas de optimización

Procesamiento y Optimización de Consultas

- Ejemplo de simplificación de expresiones


```

                select región from Productores
                where nombre = 'Osborne' → o
                or not ( región = 'Jerez' ) → j
                and ( región = 'Jerez' or región = 'La Mancha' ) → m
                and not ( región = 'La Mancha' );
            
```
- Simplificando la condición del **where**...

$$o \vee \bar{j} \wedge (j \vee m) \wedge \bar{m} \equiv o \vee \underbrace{((\bar{j} \wedge j) \vee (\bar{j} \wedge m))}_{\text{falso}} \wedge \bar{m} \equiv$$

$$o \vee (\bar{j} \wedge m) \wedge \bar{m} \equiv o \vee \underbrace{((\bar{j} \wedge \bar{m}) \wedge \underbrace{(m \wedge \bar{m})}_{\text{falso}}))}_{\text{falso}} \equiv$$

10

septiembre 2016 Introducción a la Ingeniería del Software y a los Sistemas de Información



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

1. Procesado de consultas
2. Coste de una consulta
3. Simplificación de expresiones
4. Equivalencias en AR
5. Heurísticas de optimización

Procesamiento y Optimización de Consultas

- Ejemplo de simplificación de expresiones


```

                select región from Productores
                where nombre = 'Osborne'
                or not ( región = 'Jerez' )
                and ( región = 'Jerez' or región = 'La Mancha' )
                and not ( región = 'La Mancha' );
            
```



```

                select región from Productores
                where nombre = 'Osborne';
            
```



11

septiembre 2016 Introducción a la Ingeniería del Software y a los Sistemas de Información



UNIVERSIDAD DE SEVILLA
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Procesamiento y Optimización de Consultas

- Algunas equivalencias lógicas
 - **Conjunciones y disyunciones básicas**
 - $a \wedge \neg a \equiv \text{falso}$
 - $a \vee \neg a \equiv \text{cierto}$
 - $a \wedge \text{falso} \equiv \text{falso}$
 - $a \vee \text{falso} \equiv a$
 - $a \wedge \text{cierto} \equiv a$
 - $a \vee \text{cierto} \equiv \text{cierto}$
 - **Distribución**
 - $a \wedge (b \vee c) \equiv (a \wedge b) \vee (a \wedge c)$
 - $a \vee (b \wedge c) \equiv (a \vee b) \wedge (a \vee c)$

1. Procesado de consultas

2. Coste de una consulta

3. Simplificación de expresiones

4. Equivalencias en AR

5. Heurísticas de optimización

septiembre 2016

Introducción a la Ingeniería del Software y a los Sistemas de Información

12



UNIVERSIDAD DE SEVILLA
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Procesamiento y Optimización de Consultas

- Algunas equivalencias lógicas
 - **Leyes de De Morgan**
 - $\neg (a \wedge b) \equiv \neg a \vee \neg b$
 - $\neg (a \vee b) \equiv \neg a \wedge \neg b$
 - **Implicación**
 - $a \rightarrow b \equiv \neg a \vee b$
 - **Cuantificadores**
 - $\forall x \cdot P(x) \equiv \neg \exists x \cdot \neg P(x)$
 - $\forall x \cdot \neg P(x) \equiv \exists x \cdot P(x)$

1. Procesado de consultas

2. Coste de una consulta

3. Simplificación de expresiones

4. Equivalencias en AR

5. Heurísticas de optimización

septiembre 2016

Introducción a la Ingeniería del Software y a los Sistemas de Información

13



UNIVERSIDAD DE SEVILLA
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Procesamiento y Optimización de Consultas

- **Equivalencias en álgebra relacional***
 - **Cascada de selecciones**
 - $\sigma_{c_1 \wedge c_2}(R) \equiv \sigma_{c_1}(\sigma_{c_2}(R))$
 - **Cascada de proyecciones ($a_1 \subseteq a_2 \subseteq \dots \subseteq a_n$)**
 - $\Pi_{a_1}(\Pi_{a_2}(\dots \Pi_{a_n}(R))) \equiv \Pi_{a_1}(R)$
 - **Asociatividad de joins**
 - $(R \bowtie S) \bowtie T \equiv R \bowtie (S \bowtie T)$

1. Procesado de consultas
2. Coste de una consulta
3. Simplificación de expresiones
4. Equivalencias en AR
5. Heurísticas de optimización

septiembre 2016 Introducción a la Ingeniería del Software y a los Sistemas de Información 14



UNIVERSIDAD DE SEVILLA
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Procesamiento y Optimización de Consultas

- **Equivalencias en álgebra relacional***
 - **Distribución de la selección sobre el join**
 - $\sigma_{c_R \wedge c_S}(R \bowtie S) \equiv \sigma_{c_R}(R) \bowtie \sigma_{c_S}(S)$
 - Donde c_R sólo incluye atributos de R y c_S sólo de S .
 - **Distribución de la proyección sobre el join**
 - $\Pi_{A_R \cup A_S}(R \bowtie S) \equiv \Pi_{A_R \cup A_S}(\Pi_{A_R \cup A_{\bowtie}}(R) \bowtie \Pi_{A_S \cup A_{\bowtie}}(S))$
 - Donde A_R sólo incluye atributos de R , A_S sólo de S y A_{\bowtie} los atributos sobre los que se realiza el join.

1. Procesado de consultas
2. Coste de una consulta
3. Simplificación de expresiones
4. Equivalencias en AR
5. Heurísticas de optimización

septiembre 2016 Introducción a la Ingeniería del Software y a los Sistemas de Información 15



UNIVERSIDAD DE SEVILLA
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Procesamiento y Optimización de Consultas

- **Equivalencias en álgebra relacional***
 - **Distribución de la proyección sobre la selección**
 - $\Pi_A(\sigma_c(R)) \equiv \Pi_A(\sigma_c(\Pi_{A \cup A_c}(R)))$
 - Donde A_c son los atributos de R que se usan en la condición de la selección.
 - **Distribución de la selección sobre \cup , \cap y $-$**
 - $\sigma_c(R \cup S) \equiv \sigma_c(R) \cup \sigma_c(S)$ igual para \cap y $-$
 - **Distribución de la proyección sobre la unión**
 - $\Pi_A(R \cup S) \equiv \Pi_A(R) \cup \Pi_A(S)$

1. Procesado de consultas
2. Coste de una consulta
3. Simplificación de expresiones
4. Equivalencias en AR
5. Heurísticas de optimización

septiembre 2016 Introducción a la Ingeniería del Software y a los Sistemas de Información 16



UNIVERSIDAD DE SEVILLA
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Procesamiento y Optimización de Consultas

- **Heurísticas de optimización de consultas**
 - 1. Realizar las selecciones tan pronto como sea posible**
 - Reduce el número de tuplas de los resultados intermedios.
 - 2. Realizar las selecciones sobre atributos indexados antes que sobre los no indexados**
 - Reduce las operaciones de E/S al usar los índices y se ejecutan mucho más rápido. Aplicar la cascada de selecciones si es necesario.
 - 3. Realizar las proyecciones tan pronto como sea posible**
 - Reduce el número de atributos de los resultados intermedios.

1. Procesado de consultas
2. Coste de una consulta
3. Simplificación de expresiones
4. Equivalencias en AR
5. Heurísticas de optimización

septiembre 2016 Introducción a la Ingeniería del Software y a los Sistemas de Información 17



UNIVERSIDAD DE SEVILLA
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Procesamiento y Optimización de Consultas

- **Heurísticas de optimización de consultas**
 - 4. Realizar las operaciones de selección y join más restrictivas primero***
 - Reduce el número de tuplas de los resultados intermedios. Aplicar asociatividad si es necesario.
 - 5. Eliminar proyecciones redundantes**
 - Reduce el número de atributos de los resultados intermedios. Aplicar la cascada de proyecciones.
 - 6. Usar DISTINCT sólo cuando sea imprescindible**
 - Evita tener que comparar resultados intermedios para detectar duplicados y eliminarlos del resultado.

* En las que intervengan un menor número de tuplas.

1. Procesado de consultas

2. Coste de una consulta

3. Simplificación de expresiones

4. Equivalencias en AR

5. Heurísticas de optimización

septiembre 2016

Introducción a la Ingeniería del Software y a los Sistemas de Información

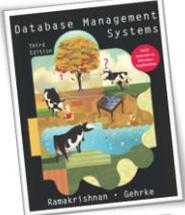
18



UNIVERSIDAD DE SEVILLA
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Procesamiento y Optimización de Consultas

- **Bibliografía**
 - R. Elmasri, S. Navathe, **Fundamentos de Sistemas de Bases de Datos** (5ª edición). Ed. Addison-Wesley, 2007.
 - Capítulo 15.
 - R. Ramakrishnan, J. Gehrke, **Database Management Systems** (3ª edición). Ed. McGraw-Hill, 2003.
 - Capítulos 12 y 15.
 - A. Silberschatz *et al.*, **Fundamentos de Bases de Datos** (5ª edición). Ed. McGraw-Hill, 2006.
 - Capítulos 13 y 14.


1. Procesado de consultas

2. Coste de una consulta

3. Simplificación de expresiones

4. Equivalencias en AR

5. Heurísticas de optimización

septiembre 2016

Introducción a la Ingeniería del Software y a los Sistemas de Información

19