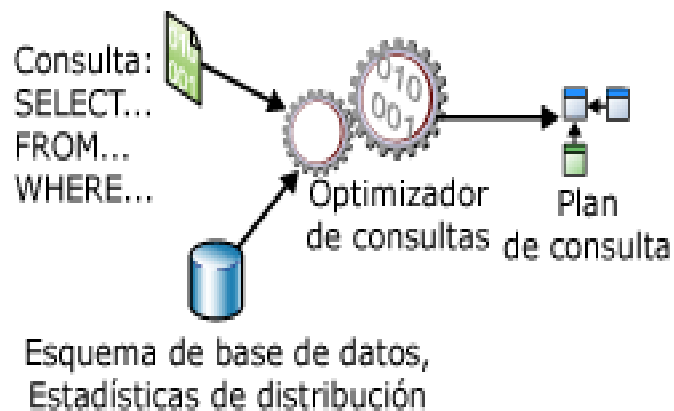


PROCESAMIENTO Y OPTIMIZACION DE CONSULTAS



FACULTAD de INGENIERÍA de
SISTEMAS E INFORMÁTICA

Procesamiento en SQL



Procesamiento en SQL

Una instrucción SELECT define únicamente los siguientes elementos:

- ☐ El formato del conjunto de resultados, este elemento se especifica principalmente en la lista de selección
- ☐ Sin embargo, también afectan a la forma final del conjunto de resultados otras cláusulas como ORDER BY y GROUP BY
- ☐ Las tablas que contienen los datos de origen. Esto se especifica en la cláusula FROM

Procesamiento en SQL

- ☐ ¿ Cómo se relacionan lógicamente las tablas para la instrucción SELECT ?

Esto se define en las especificaciones de combinación, que pueden aparecer en la cláusula WHERE

- ☐ Las condiciones que deben cumplir las filas de las tablas de origen para satisfacer los requisitos de la instrucción SELECT. Estas condiciones se especifican en las cláusulas WHERE y HAVING

Plan de Ejecución

- ❑ Un plan de ejecución es el conjunto de pasos que tiene que realizar el DBMS para ejecutar una consulta
- ❑ Un plan de ejecución de una consulta SQL es una definición de:
 - La secuencia en la que se tiene acceso a las tablas de origen
 - Los métodos que se utilizan para extraer los datos de cada tabla

Optimización

- ❑ El proceso de selección de un plan de ejecución entre varios planes posibles se conoce como optimización
- ❑ El optimizador de consultas es uno de los componentes más importantes de un sistema de base de datos SQL

Análisis de Consultas y Transacciones

Para cada consulta establecer:

- a. Las tablas a las que accederá
- b. Los atributos sobre los que se especificarán condiciones de selección (WHERE)
- c. Los atributos sobre los que se especificarán condiciones de reunión o de enlace de tablas
- d. Los atributos cuyos valores se obtendrá en la consulta

*Los atributos de los incisos **b** y **c** son candidatos a constituir índices (estructuras de acceso)*

Análisis de Consultas y Transacciones

Para cada transacción de actualización establecer:

- a. Las tablas que actualizará
 - b. El tipo de operación en cada tabla (insertar, modificar o eliminar)
 - c. Los campos sobre los que se especificarán condiciones de selección para operaciones de eliminación o modificación
 - d. Los campos cuyos valores alterará una operación de modificación
- ✓ *Los campos del inciso **c** son candidatos para índices*
- ✓ *Los campos del inciso **d** son candidatos a evitar en los índices, ya que su modificación requerirá la actualización de estas estructuras de acceso.*

AJUSTE DE INDICES

Objetivos:

- ◆ Evaluar dinámicamente los requerimientos, que pueden cambiar según época del año, día del mes o de la semana
- ◆ Reorganizar los índices para obtener mejor rendimiento

AJUSTE DE INDICES

- ◆ Ciertas consultas pueden tardar mucho en ejecutarse por falta de un índice apropiado
- ◆ Puede haber índices que no se utilicen
- ◆ Puede haber índices que originen trabajo adicional por estar definidos sobre atributos que sufren continuos cambios

AJUSTE DE CONSULTAS

Indicadores:

- ◆ Demasiados accesos al disco (por ejemplo una consulta de emparejamiento exacto que recorre una tabla completa)
- ◆ El plan de ejecución de consulta muestra que no se están usando los índices relevantes.

Ajuste de Consultas - Casos

1. Muchos optimizadores no usan índices en presencia de:
 - Expresiones aritméticas (SALARIO/365 > 10.50)
 - Comparaciones numéricas de campos de diferente tamaño y precisión (ACANT = BCANT donde ACANT es de tipo Integer y BCANT es Smallinteger)
 - Comparaciones con NULL (FECHA IS NULL)
 - Comparaciones de subcadenas (como (APELLIDO LIKE '%EZ'))

Ajuste de Consultas - Casos

2. Los índices no suelen usarse en consultas anidadas que utilizan IN:

```
SELECT NSS FROM EMPLEADO
WHERE DNO IN (SELECT DNUMERO
                FROM DEPARTAMENTO
                WHERE NSS_JEFE = '3334444')
```

Puede no utilizar el índice definido sobre DNO en EMPLEADO, mientras que la utilización de DNO = DNUMERO en la cláusula WHERE con una consulta de un solo bloque puede ocasionar que el índice sí se utilice

Ajuste de Consultas - Casos

3. Algunos ***DISTINCT*** pueden ser redundantes y podrían evitarse sin modificar el resultado

Un ***DISTINCT*** generalmente provoca una operación de clasificación y debe evitarse siempre que sea posible

Ajuste de Consultas - Casos

4. El uso innecesario de tablas temporales puede evitarse juntando varias consultas en una sola, *a menos que* la relación temporal sea necesaria para algún resultado intermedio

Ajuste de Consultas - Casos

5. En algunas situaciones en las que se usa consultas correlacionadas son útiles las tablas temporales

```
SELECT NSS
FROM EMPLEADO E
WHERE SALARIO = SELECT MAX(SALARIO)
FROM EMPLEADO AS M
WHERE M.DNO = E.DNO)
```

Esto tiene el peligro potencial de buscar en toda la tabla M EMPLEADO interna para cada tupla de E EMPLEADO externa.

Ajuste de Consultas - Casos

Para hacerlo más eficiente puede descomponerse en dos consultas, la primera de las cuales calcula el salario máximo de cada departamento:

```
SELECT MAX(SALARIO) AS SALARIO_MAYOR, DNO
INTO TEMP
FROM EMPLEADO
GROUP BY DNO;
```

```
SELECT NSS
FROM EMPLEADO, TEMP
WHERE SALARIO = SALARIO_MAYOR AND
EMPLEADO.DNO = TEMP.DNO
```

Ajuste de Consultas - Casos

- 6. De haber varias opciones posibles para la condición de reunión, elegir una que use un índice de agrupación (CLUSTER), y evitar aquellas que contengan comparaciones de cadenas:**

Si el campo NOMBRE es una clave candidata en EMPLEADO y en ALUMNO, es mejor usar

```
EMPLEADO.NSS = ALUMNO.NSS
```

como condición de reunión, en lugar de

```
EMPLEADO.NOMBRE = ALUMNO.NOMBRE
```

si NSS tiene un índice de agrupación en una o en ambas tablas.

Ajuste de Consultas - Casos

7. En algunos optimizadores de consultas el orden en el que aparecen las tablas en el FROM puede afectar el procesamiento de la reunión

- En esos casos debe cambiarse el orden para que procese primero las tablas con menos datos, y la más grande se use con el índice correspondiente

Ajuste de Consultas - Casos

8. Algunos optimizadores dan peores tiempos con consultas anidadas que con sus equivalentes no anidadas

```
SELECT NSS FROM EMPLEADO
WHERE DNO IN (SELECT DNUMERO
              FROM DEPARTAMENTO
              WHERE NSS_JEFE = '3334444')
```

La transformación de subconsultas correlacionadas puede llevar a que se creen tablas temporales

Ajuste de Consultas - Casos

9. Muchas aplicaciones se basan en vistas que definen los datos de interés para las aplicaciones

- A veces estas vistas pueden ser excesivas cuando la consulta puede realizarse directamente sobre la tabla base, en lugar de usar una vista que se ha definido sobre una reunión

Ajuste de Consultas - Casos

10. Una consulta con varias condiciones OR puede hacer que no se empleen los índices que existen

Puede rehacerse expresándose como una unión de consultas, cada una con una condición sobre un atributo que hace que se emplee el índice

```
SELECT NOMBRE, APELLIDO, SALARIO, EDAD  
FROM EMPLEADO  
WHERE EDAD > 45 OR SALARIO < 5000
```

Ajuste de Consultas - Casos

```
SELECT NOMBRE, APELLIDO, SALARIO, EDAD
FROM EMPLEADO
WHERE EDAD > 45
```

UNION

```
SELECT NOMBRE, APELLIDO, SALARIO, EDAD
FROM EMPLEADO
WHERE SALARIO < 5000
```

Puede usar los índices definidos sobre **SALARIO** y sobre **EDAD**

Ajuste de Consultas - Casos

11. Las condiciones WHERE pueden reescribirse de modo que se utilicen índices para varias columnas:

```
SELECT REGION, TIPO_PROD, MES, VENTAS
FROM ESTADISTICA_VENTAS
WHERE REGION = 3 AND ((TIPO_PROD BETWEEN 1 AND 3) OR
(TIPO_PROD BETWEEN 8 AND 10))
```

Puede usar un índice únicamente sobre REGION y debe buscar a través de todas las páginas hoja del índice un emparejamiento con TIPO_PROD.

Ajuste de Consultas - Casos

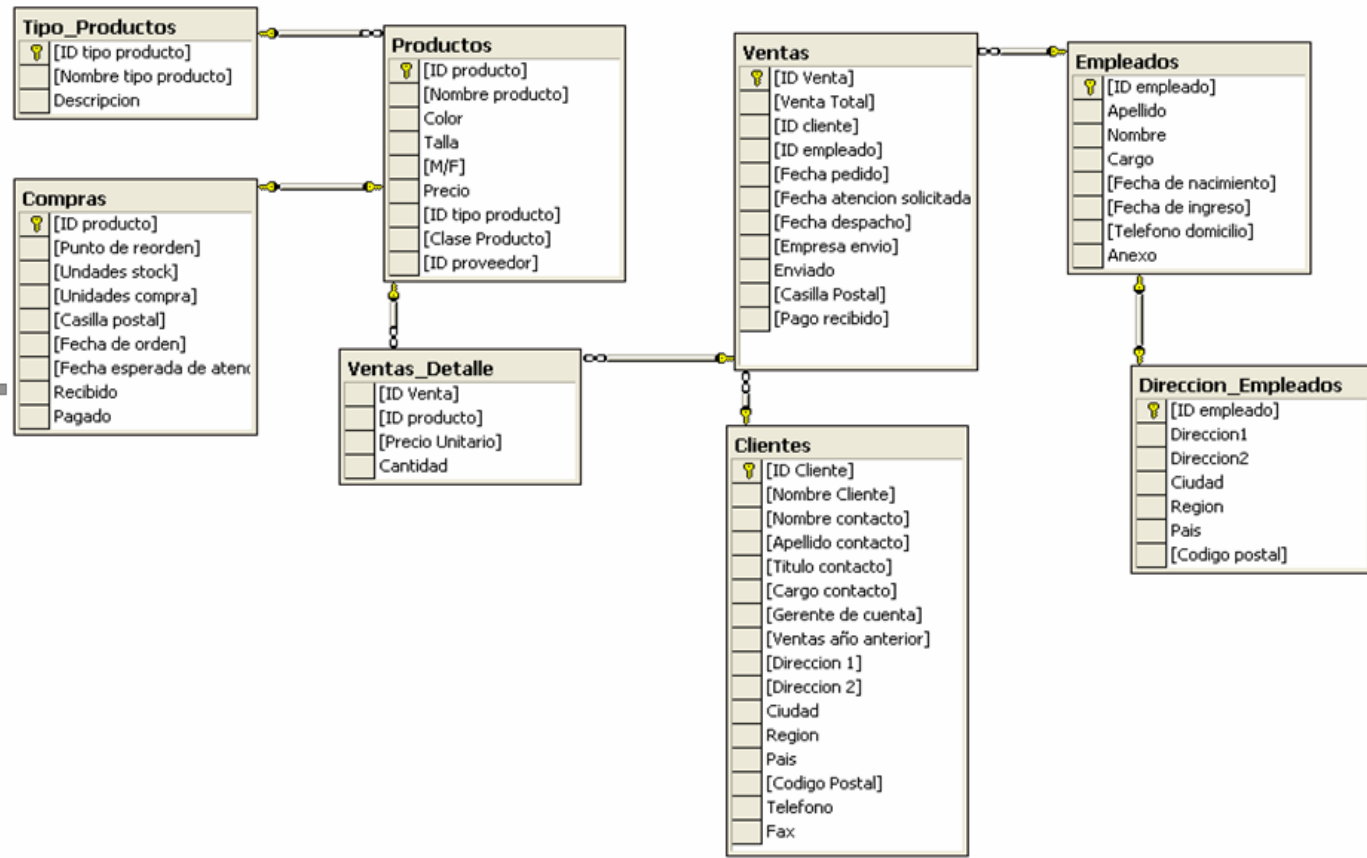
En cambio:

```
SELECT REGION, TIPO_PROD, MES, VENTAS  
FROM ESTADISTICA_VENTAS  
WHERE (REGION = 3 AND (TIPO_PROD BETWEEN 1 AND 3)) OR  
      (REGION = 3 AND (TIPO_PROD BETWEEN 8 AND 10))
```

Puede usar un índice compuesto sobre (REGION, TIPO_PROD) y trabajará mucho más eficientemente.

RESUMEN

- El diseño físico de la base de datos debe considerar la creación de estructuras de acceso adecuadas
- El mantenimiento y optimización de los índices es una labor permanente
- Se puede obtener mejoras dramáticas en el rendimiento de la BD optimizando el código de las consultas SQL, dejando la desnormalización como última alternativa, después aún de la mejora en hardware



FACULTAD de INGENIERÍA de
SISTEMAS E INFORMÁTICA