Optimización

- Todas las operaciones relacionales tales como restringir, proyectar y juntar son operaciones *en el nivel de conjunto.* Como consecuencia, los lenguajes relacionales *no son de procedimientos*, en el sentido de que los usuarios especifican *el qué*, no *el cómo;* es decir, dicen lo que desean sin especificar un procedimiento para obtenerlo.

- El proceso de "navegar" por los datos a fin de satisfacer la petición del usuario es realizado automáticamente por el sistema, en vez de ser realizado en forma manual por el usuario. Por esta razón, en ocasiones se dice que los sistemas relacionales realizan una **navegación automática.**

- En contraste, en los sistemas no relacionales la navegación es generalmente responsabilidad del usuario.

Capacidad de Optimización

- La **capacidad de optimizacion,** el hecho de que las peticiones relaciónales sean optimizables es de hecho una *ventaja* de los sistemas relaciónales. Entonces, el propósito general del optimizador es seleccionar una estrategia eficiente para la evaluación de una expresión relacional dada.

- El término *optimización* es en cierta forma una exageración, ya que por lo general no hay ninguna garantía de que la estrategia de Implementación elegida sea en realidad *óptima* en cualquier sentido mensurable; de hecho podría serlo, pero generalmente lo único que se sabe con certeza es que la estrategia "optimizada" es una *mejora* de la versión original no optimizada.

Optimización Automática

La ventaja de la optimización automática no es únicamente que los usuarios no tienen que preocuparse por formular sus consultas de la mejor manera (es decir, por cómo formular las peticiones para obtener el mejor desempeño del sistema).

El hecho es que existe una posibilidad real de que el optimizador pueda hacerlo *mejor* que un usuario humano. Existen varias razones para esta situación, entre las que se encuentran las siguientes:

- 1. Un buen optimizador tendrá una gran cantidad de información disponible que por lo general no tienen los usuarios humanos; en particular conocerá determinada información **estadística** tal como:

  ■ La cantidad de valores en cada dominio.

  ■ La cantidad actual de tupias en cada varrel base.

  ■ La cantidad actual de valores distintos en cada atributo de cada varrel base.

  ■ La cantidad de veces que tales valores se dan en cada uno de esos atributos.

Por consecuencia, el optimizador deberá ser capaz de hacer una valoración más precisa de la eficiencia de cualquier estrategia dada para implementar una petición en particular, y por lo tanto será más probable que escoja la implementación más eficiente.

Optimización Automática

- 2. Además, si las estadísticas de la base de datos cambian con el tiempo, tal vez sea necesaria una selección de estrategia diferente; en otras palabras, es posible que se requiera una *reoptimización.* En un sistema relacional, la reoptimización es trivial, ya que simplemente involucra un reprocesamiento de la petición relacional original a cargo del optimizador del sistema. Por el contrario, en un sistema que no es relacional, la reoptimización involucra la reescritura del programa y es muy probable que no se realice nunca.

- 3. Por otra parte, el optimizador es un *programa* y es por definición mucho más paciente que un usuario humano típico. El optimizador es bastante capaz de considerar literalmente cientos de estrategias de Implementación diferentes para una petición dada, y en cambio es muy poco probable que un usuario humano llegue a considerar más de tres o cuatro (al menos a profundidad).

- 4. Por último, el optimizador puede ser considerado, en cierto sentido, como la personificación de las habilidades y servicios de "los mejores" programadores humanos. Como consecuencia, tiene el efecto de poner a disposición de *todos* esas habilidades y servicios, lo que significa que está poniendo a disposición de un amplio rango de usuarios, un conjunto de recursos en una forma eficiente y económica.

Procesamiento de Consultas

- Podemos identificar cuatro grandes etapas en el procesamiento de consultas, de la siguiente forma:

1. Convertir la consulta a su forma interna.
2. Convertirla a la forma canónica.
3. Seleccionar procedimientos candidatos de bajo nivel.
4. Generar planes de consulta y seleccionar el más barato.

Procesamiento de Consultas

Etapa 1: Convertir la consulta a su forma interna

- La primera etapa involucra la conversión de la consulta original en alguna representación interna que sea más adecuada para manejarla en la máquina, eliminando así consideraciones meramente externas (tales como los detalles de la sintaxis concreta del lenguaje de consulta que se está considerando) y allanando el camino para las etapas subsecuentes del proceso de optimización.

- *Nota:* El procesamiento de vistas (es decir, el proceso de reemplazar las referencias a las vistas por las expresiones de definición de vistas aplicables) también se realiza durante esta etapa.

- La pregunta obvia es: ¿en qué formalismo deberá estar basada la representación interna? Por supuesto, sin importar cuál formalismo se elija, éste deberá ser lo suficientemente amplio para representar todas las consultas posibles en el lenguaje de consulta externo. También deberá ser lo más neutral posible, en el sentido de que no deberá interferir con las selecciones subsecuentes. Por lo general, la forma interna seleccionada es algún tipo de **árbol de sintaxis abstracto** o **árbol de consulta.**

Procesamiento de Consultas

- **Etapa 2: Conversión a la forma canónica**

- En esta etapa, el optimizador realiza varias optimizaciones que son "garantizadas como buenas", sin tomar en cuenta los valores actuales de los datos ni las rutas de acceso físicas que existen en la base de datos almacenada. El punto es que los lenguajes relaciónales permiten generalmente que todas las consultas sean expresadas en diversas formas que son al menos superficialmente distintas.

- En realidad, el desempeño de una consulta no debe depender de la forma particular en que al usuario se le ocurra escribirla. Por lo tanto, el siguiente paso en el procesamiento de la consulta es convertir la representación interna en alguna **forma canónica** equivalente, con el objeto de eliminar esas diferencias superficiales y encontrar una representación que sea, en cierta forma, más eficiente que la original.

- *Una nota con relación a la "forma canónica":* La noción de *forma canónica* es fundamental en muchas ramas de las matemáticas y disciplinas relacionadas. Podemos definirla de la siguiente manera: Dado un conjunto de objetos *Q* (digamos consultas) y una noción de equivalencia entre esos objetos (digamos la noción de que las consultas *ql* y *q2* son equivalentes si y sólo si producen necesariamente el mismo resultado), decimos que el subconjunto C de *Q* es un **conjunto de formas canónicas** para *Q* bajo la definición establecida de equivalencia, si y sólo si cada objeto *q* en *Q* es equivalente a sólo un objeto *c* en *C.* Decimos que el objeto c es *Informa canónica* para el objeto *q.* Todas las propiedades "interesantes" aplicadas al objeto *q* también son aplicadas a su forma canónica c; por lo tanto, es suficiente estudiar sólo el pequeño conjunto C, y no el conjunto grande *Q,* para probar una variedad de resultados "interesantes".

Procesamiento de Consultas

Etapa 3: Selección de procedimientos candidatos de bajo nivel

- Una vez que convertimos la representación interna de la consulta en una forma más adecuada, debe decidir cómo ejecutar dicha consulta transformada. En esta etapa entran en juego consideraciones tales como la existencia de los índices u otras rutas de acceso físicas, la distribución de valores de datos, el agrupamiento físico de los datos almacenados, etcétera.

- Observe que en las etapas 1 y 2, no prestamos atención a estos asuntos. La estrategia básica es considerar a la expresión de consulta como la especificación de una serie de **operaciones de "bajo nivel"** (juntar, restringir, resumir, etcétera), con cierta interdependencia entre sí. Un ejemplo de tal interdependencia es el siguiente: para realizar una proyección, el código requerirá generalmente que sus tupias de entrada sean ordenadas en alguna secuencia que permita la eliminación de duplicados, y esto significa que la operación inmediata anterior de la serie debe producir sus tupias de salida en la misma secuencia.
- Ahora, para cada operación posible de bajo nivel el optimizador tendrá a su disposición un conjunto de **procedimientos de implementación** predefinidos

- Cada procedimiento tendrá también una **fórmula de costo** (con parámetros) asociada, que indica el costo de ejecutar ese procedimiento (generalmente en términos de E/S de disco, aunque algunos sistemas también toman en cuenta la utilización de la CPU y otros factores). Estas fórmulas de costo se usan en la etapa 4

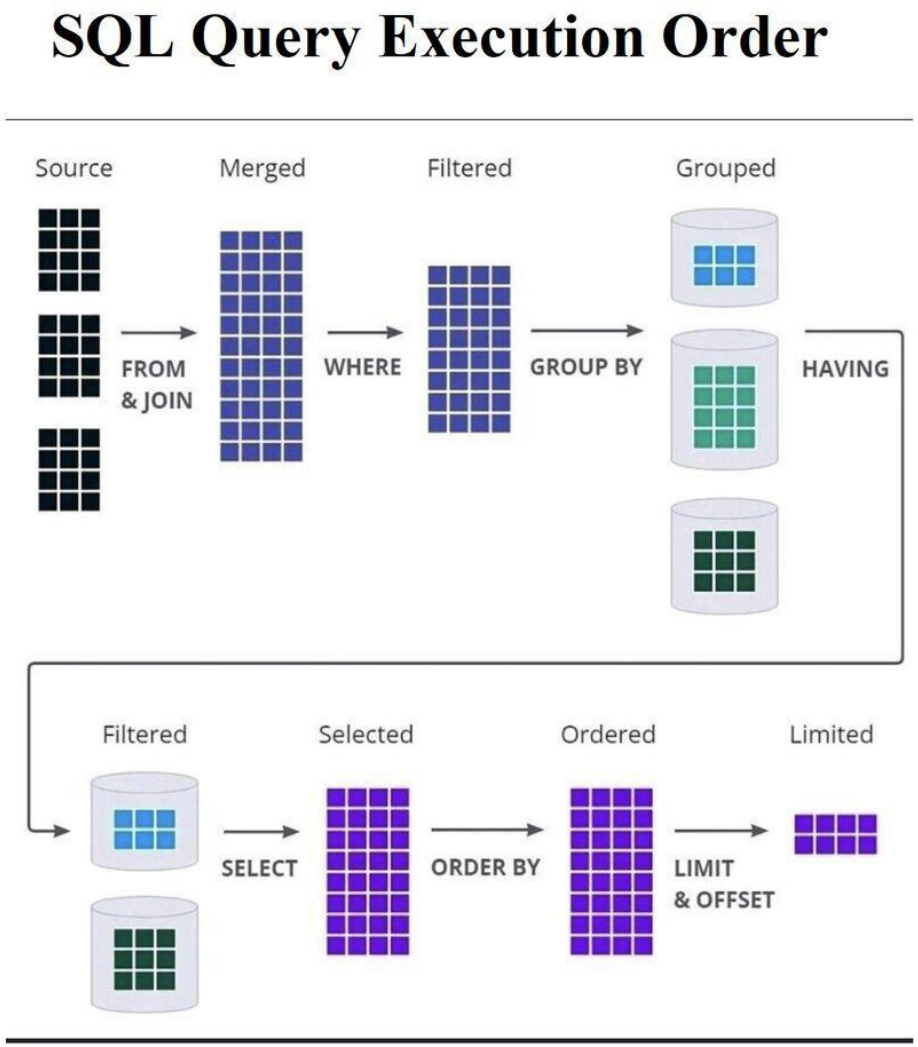
- Por lo tanto, si utilizamos la información del catálogo referente al estado actual de la base de datos (existencia de índices, cardinalidades actuales, etcétera) y usamos también la información de interdependencia que mencioné anteriormente, el optimizador seleccionará uno o más procedimientos candidatos para la implementación de cada una de las operaciones de bajo nivel en la expresión de consulta. En ocasiones, a este proceso se le llama **selección de ruta de acceso.**

Procesamiento de Consultas

- **Etapa 4: Generación de los planes de consulta y selección del más barato**

- La última etapa del proceso de optimización involucra la construcción de un conjunto de **planes de consulta** candidatos, seguida de una selección del mejor de esos planes (es decir, el más barato). Cada plan de consulta es construido por medio de la combinación de una serie de procedimientos de implementación candidatos; uno de estos procedimientos para cada una de las operaciones debajo nivel en la consulta. Observe que generalmente existirán muchos planes posibles (tal vez demasiados) para una consulta dada. De hecho, en la práctica tal vez no sea buena idea generar todos los planes posibles, ya que en combinación habrá muchos y la tarea de seleccionar al más barato bien puede llegar a ser excesivamente cara por sí misma; por lo tanto, es muy necesaria —aunque no esencial— alguna técnica para mantener dentro de límites razonables al conjunto generado. Por lo general, al hecho de "mantener el conjunto dentro de límites" se le llama *reducción del espacio de búsqueda*, ya que puede ser considerado como la reducción —a proporciones manejables— del rango ("espacio") de posibilidades queel optimizador debe examinar ("buscar"). Naturalmente, la selección del plan más barato requiere de un método para asignar un costo a cualquier plan dado. Por supuesto, el costo de un plan dado es básicamente la simple suma de los costos de los procedimientos individuales que conforman ese plan y por lo tanto, lo que el optimizador tiene que hacer es evaluar las fórmulas de costo de esos procedimientos individuales.
- El problema es que esas fórmulas de costo dependerán del tamaño de las relaciones a procesar, y debido a que casi todas —con excepción de las consultas más sencillas— involucran la generación de resultados intermedios durante la ejecución, el optimizador tendrá que estimar el tamaño de esos resultados intermedios para evaluar las fórmulas. Por desgracia, esos tamaños tienden a ser muy dependientes de los valores de datos actuales. En consecuencia, la estimación precisa de los costos puede ser un problema difícil.

Procesamiento de Consultas



Optimización Semántica

La optimización semántica puede definirse como el proceso de transformar una consulta específica en otra cualitativamente diferente, pero que sin embargo garantiza producir el mismo resultado original gracias a que está garantizado que los datos satisfacen una determinada restricción de integridad.

Es importante comprender que en principio *cualquier restricción de integridad* puede ser usada en la optimización semántica (la técnica no está limitada a restricciones referenciales).

Prueba Corta
8:40 – 8:50