

LENGUAJES DE CONSULTA ORIENTADOS A OBJETOS

Los lenguajes de consulta constituyen una funcionalidad importante de los SGBDOO. El usuario puede recuperar los datos especificando simplemente las condiciones que estos datos deben cumplir.

De manera general, estos lenguajes deben ser:

- De alto nivel, de modo que se puedan formular fácilmente operaciones complejas;
- Declarativos, esto es, que el foco de atención esté dirigido hacia QUÉ se quiere y no hacia CÓMO se obtiene;
- Eficientes: las construcciones deben permitir optimizaciones;
- Independientes de la aplicación, de modo que se puedan usar sobre cualquier esquema de bases de datos.

Características de los lenguajes de consulta orientados a objetos

Acceso a los objetos

- dos vías para el acceso a datos, en contraposición con los SGBD relacionales que sólo existe una vía de acceso a datos.
 - vía navegacional, el sistema accede al objeto directamente y navega a través de los objetos a los que se refieren sus atributos.
 - como segunda vía, tenemos la basada en los lenguajes de consulta tipo SQL, aunque, por supuesto, estas dos vías de acceso se pueden usar de forma complementaria.
- tipo de igualdad que se va a utilizar: que determina la semántica y la estrategia en la ejecución de operaciones: unión, diferencia, intersección y eliminación de duplicados.

Jerarquías de agregación

- condiciones que aparecen en una consulta se imponen sobre los atributos anidados en la jerarquía del objeto en cuestión, es una ventaja que se pueda navegar con facilidad a través de esta estructura.
- Consulta, tal como:

Encontrar todos los proyectos que tienen al menos una tarea cuyo jefe es un investigador especializado en bases de datos.

$$\{v \mid (\exists w)(\exists g)(\exists i) \text{Proyecto}(w) \wedge \text{Tarea}(g) \wedge \text{Investigador}(i) \\ \wedge (w = v) \wedge g \in w.\text{PlanDeTrabajo} \wedge g.\text{Jefe} = i \\ \wedge i.\text{Especializacion} = \text{"BD"}\}$$

- introducción explícita de dos variables g e i , y de dos predicados de reunión.
- utilizar expresiones de camino dentro del lenguaje, de modo que las reuniones que se apliquen para obtener una de las componentes de un objeto se puedan formular fácilmente, y así simplificar sensiblemente las expresiones de consulta.

Reunión (Join)

- reunión implícita (implicit join): se deriva de la estructura jerárquica de los objetos.
- reunión explícita (explicit join): reunión que explícitamente compare dos objetos utilizando la igualdad por identidad o por valor. La consulta anterior se expresaría entonces más sucintamente:

$$\{v \mid (\exists w)(\exists g)(\exists i) P(w) \wedge (w = v) \\ \wedge (\text{"DB"} \in w.\text{PlanDeTrabajo}.\text{Jefe}.\text{Especializacion})\}$$

- predicado sobre un atributo anidado se llama predicado anidado o complejo.
- atributo anidado en una consulta se denota por una expresión de camino.
- Forma diferente de ver las reuniones implícitas y explícitas:
 - consultas de un solo operando: recuperan instancias (ejemplares) de una sola clase o de una subclase de esta clase.
 - consultas de múltiples operandos: representan operaciones que se pueden comparar con las reuniones del modelo relacional y con operaciones sobre conjuntos.

Jerarquías de herencia

- aspectos importantes concernientes a la jerarquía de herencia.
 - Uno es que se puede consultar una clase o una clase y todas sus subclases. En la mayoría de los lenguajes de los sistemas existentes, por ejemplo ORION, ambas opciones son posibles.
- cada clase puede tener sus propios atributos, así como aquellos que hereda, entonces es útil tener la capacidad de expresar consultas mediante predicados alternativos con la forma siguiente:

If class(x)=C1 then pred1
If class(x)=C2 then pred2
.....
If class(x)=Cn then predn

Donde x es una variable que denota un objeto, y class (x) es un mensaje especial, que, cuando se aplica a un objeto, retorna la clase a la que pertenece. Los predicados alternativos son útiles donde las consultas contienen otros predicados sobre atributos comunes de la jerarquía de herencia.

Consultas recursivas

- construcciones para expresar consultas recursivas.
- Los objetos que son relevantes para muchas aplicaciones avanzadas son modelados de manera natural por la recursividad.

Ciclos de la jerarquía de clases

La relación de agregación no define una jerarquía en el sentido estricto de la palabra, pero admite la existencia de ciclos. En algunos sistemas se puede definir una clase directamente en términos de sí misma.

Una rama en una jerarquía de clases es cíclica si contiene dos clases, Ci y Cj, que cumplen las condiciones siguientes:

- Cj es el dominio (indirecto) de un atributo de Ci, y Ci es el dominio de un atributo de Cj.
- Cj es el dominio (indirecto) de un atributo de Ci, y una superclase o subclase de Ci es el dominio de un atributo de Cj.

Esta definición implica cuatro tipos diferentes de ciclos, como se muestra en la Figura 1, dos de los cuales, los tipos n-s y s-s, pueden considerarse casi ciclos, dado que no son ciclos en el sentido usual de la palabra. Supongamos que la rama tiene n nodos, y que cada nodo puede ser la raíz de una jerarquía de herencia. A continuación se dan las definiciones de los cuatro tipos de ciclo:

1. Un ciclo tipo n es un ciclo que consta de $n > 1$ nodos en la rama
2. Un ciclo tipo n-s es un casi ciclo que corresponde a un ciclo tipo n. Este consta de $n > 0$ nodos en la rama y de una superclase de uno de los n nodos.
3. Un ciclo tipo s es un ciclo que consta de un solo nodo.
4. Un ciclo tipo s-s es un casi ciclo que corresponde a un ciclo tipo s. Éste consta de una clase y de su superclase o subclase.

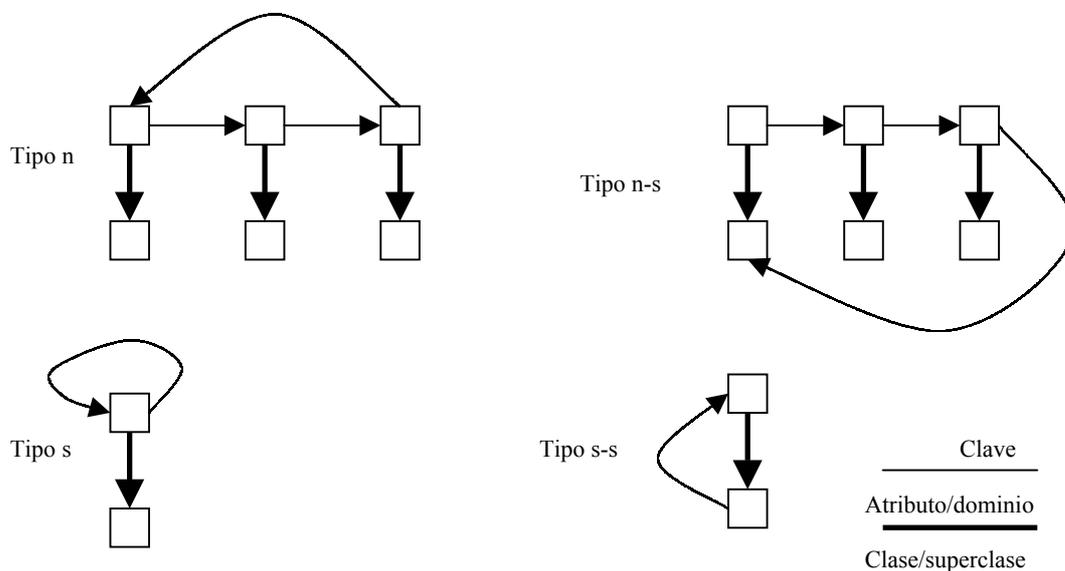


Figura 1. Posibles ramificaciones cíclicas

Métodos

- Un atributo derivado tiene una función comparable con la de un atributo, pero, mientras un atributo contiene un valor, un método es un procedimiento que calcula un valor sobre la base de los valores de los atributos de un objeto o de otros objetos de la base de datos. En este último caso, ejecutar un método se traduce en invocar a otros métodos sobre otros objetos.

Si un método, por ejemplo un atributo derivado, retorna objetos (valores primitivos o identificadores de objetos), un método predicado tiene una función similar, pero retorna los valores lógicos True o False.

- Un método predicado hace comparaciones entre objetos, y puede, por tanto, formar parte de las expresiones lógicas que determinan qué objetos satisfacen la consulta.

Resultados de una consulta

- El resultado de una consulta es una relación y, por tanto, goza de la flexibilidad de la construcción relacional. Por tanto, las consultas se pueden anidar y el resultado de una consulta se puede dar como operando de otra consulta.
- Puede ser costoso aplicar el mismo principio a consultas sobre objetos.
- El resultado de una consulta puede ser un objeto, o un conjunto de objetos, de una clase que aún no exista en el esquema de la base de datos, y que se debe definir por la consulta. Este enfoque resulta

costoso, por lo que algunos lenguajes imponen restricciones sobre las operaciones sobre clases.

- Una restricción común es limitar las proyecciones exigiendo que se deban recuperar o bien todos los atributos de cada objeto seleccionado (es decir, el objeto como tal) o bien recuperar un solo atributo. De esta manera, el resultado de una consulta será un conjunto de objetos de una clase existente.
- Otra solución: considerar los resultados como un conjunto de instancias de una clase general, la cual acepta todos los objetos y cuyos únicos métodos son aquellos para imprimir los objetos o mostrarlos por pantalla. Pero esto impide que los objetos puedan ser utilizados otra vez, para otras manipulaciones, limitando la idea de las consultas anidadas. La solución más restrictiva, es decir, la que permite sólo formas limitadas de proyección, es una solución adoptada por ORION.