

Vistas

Una tabla percibida por el usuario puede ser una tabla base o una vista.

Una tabla base es real (existe físicamente) mientras que una vista es una tabla virtual que no tiene existencia en sí misma sino que se deriva de una o más tablas bases subyacentes. No existe en sí pero ante el usuario parece existir.

```
Ejemplo : CREATE VIEW BUENOS_PROVEEDORES AS
          SELECT S#, SITUACION, CIUDAD
          FROM S
          WHERE SITUACION > 15;
```

Para el usuario es como si existiera una tabla llamada BUENOS_PROVEEDORES.

Las vistas son una ventana a través de la cual se ve la tabla real S. Esta es una ventana dinámica pues las modificaciones hechas en S se verán automáticamente.

Los usuarios pueden trabajar en las vistas como si fueran una tabla real.

```
Ejemplo : SELECT *
          FROM BUENOS_PROVEEDORES
          WHERE CIUDAD <> 'Londres';
```

El sistema convierte esta operación en una equivalente realizada sobre la tabla (ó tablas) subyacente:

```
SELECT *
FROM S
WHERE CIUDAD <> 'Londres'
      AND SITUACION > 15;
```

La conversión se hace combinando el SELECT DEL USUARIO con la proposición SELECT guardada en el catálogo cuando se definió la vista.

```
Ejemplo :
UPDATE BUENOS_PROVEEDORES
SET SITUACION = SITUACION + 10
WHERE CIUDAD = 'París';
```

Será convertida en:

```
UPDATE S
SET SITUACION = SITUACION + 10
WHERE CIUDAD = 'París'
      AND SITUACION > 15;
```

La inserción y eliminación se manejan en forma análoga.

Una subconsulta puede extraer datos de otra vista.

```
CREATE VIEW
PARTES_ROJAS_LONDRES
AS SELECT P#, PESO
```

FROM PARTESROJAS
WHERE CD = 'Londres';

Con opción de verificación las operaciones UPDATE e INSERT se verificarán para garantizar que toda fila modificada o insertada satisfaga la condición de la vista.

Sean las siguientes dos vistas sobre una misma tabla:

1. CREATE VIEW S#_CIUDAD
AS SELECT S#, CIUDAD FROM S;
2. CREATE VIEW SITUACION_CIUDAD
AS SELECT SITUACION, CIUDAD FROM S;

Se puede insertar un nuevo registro (S6, Roma), mediante la inserción real del registro (S6,NULL,NULL,Roma) en la vista 1.

Se puede eliminar un registro existente.

– Por ejemplo: (S1,Londres) eliminando en realidad (S1,Salazar,20,Londres)

Se puede modificar un campo.

– Por ejemplo: Londres por Roma para S# = 'S1'.

Si tratamos de insertar un nuevo registro (40, Roma) en vista 2.

– el sistema tendrá que insertar el registro

(NULL,NULL,40,Roma) en la tabla subyacente

– Esa operación fracasará pues los S# están definidos NOT NULL

Si tratamos de eliminar un registro existente.

– Por ej. (20,Londres), el sistema no sabe cual, porque no se ha especificado el nro. de proveedor, porque no es parte de la vista

Si se quiere modificar un campo

– por ej. (20,Londres) a (20, Roma), el sistema no sabe que registro modificar de la tabla subyacente.

Vista 1 incluye la clave primaria de la tabla subyacente mientras Vista 2 no la incluye. Una vista se puede actualizar si conserva la clave primaria.

Ventajas de las vistas:

* Permiten a los usuarios ver los mismos datos de distintas maneras al mismo tiempo

* Se simplifica la percepción del usuario : Muestra solo los datos que le interesan y no otros.

* Se cuenta con seguridad automática para datos ocultos (información no visible a través de una vista)

Seguridad e Integridad

Seguridad se refiere a la protección de los datos contra una visualización, alteración o destrucción no autorizada. Implica asegurar que los usuarios están autorizados para llevar a cabo lo que tratan de hacer.

Integridad se refiere a la exactitud o validez de los datos. Implica asegurar que lo que tratan de hacer es correcto.

El DBA se encargará de especificar restricciones: ¿quién puede tener acceso a qué?

- Un usuario dado tendrá diferentes derechos de acceso o autorizaciones sobre diferentes objetos de información.
- Diferentes usuarios pueden tener diferentes derechos sobre el mismo objeto.

SQL cuenta con dos características independientes para mantener la seguridad:

- El mecanismo de vistas.
- El subsistema de autorización, mediante el cual usuarios pueden conceder de manera selectiva y dinámica esos derechos a otros usuarios, y después revocar esos derechos si lo desean.

Las proposiciones de SQL del subsistema de autorización son GRANT (otorgar) y REVOKE (revocar). El sistema las graba en el catálogo en forma de restricciones de autorización.

El DBA tiene todos los derechos y puede concederlos a otros

- Si el administrador concede a otro usuario U el derecho de crear algún objeto (vista o tabla), entonces U automáticamente tiene todo tipo de derecho sobre ese objeto, incluyendo el derecho de conceder derechos.

Si el usuario U1 tiene el derecho de conceder cierta autorización A1 a otro usuario U2,

- tendrá también el derecho de concederla al usuario U2 “con la opción GRANT”

Si el usuario U1 emite ahora: REVOKE SELECT ON TABLE S FROM U2;

- La revocación se propagará y se revocarán todas automáticamente.