

Objetos de la base
de datos
Firebird

Los objetos de las bases de datos a las que nos conectamos, son:

- Dominios (Domains)
- Tablas (Tables)
- Vistas (Views)
- Procedimientos (Procedures)
- Disparadores (Triggers)
- Generadores (Generators)
- Excepciones (Exceptions)
- Funciones definidas por el usuarios (UDFs)
- Roles
- Tablas del sistema

Dominios

En Firebird se llama dominio a los tipos de datos creados por el programador, no son los tipos de datos estándar sino los creados por el programador para cubrir sus propias necesidades.

Un dominio siempre está basado en un tipo de datos estándar pero limita ese tipo de dato estándar, obteniéndose así un nuevo tipo de dato más acorde a lo que el programador necesita.

Ejemplos

```
CREATE DOMAIN D_BOOLEAN  
AS CHAR(1) CHARACTER SET NONE CHECK (VALUE = 'V' OR VALUE = 'F');
```

```
CREATE DOMAIN D_FECHA_INICIAL  
AS DATE CHECK (VALUE >= '2012/10/15');
```

```
CREATE DOMAIN D_PRECIO1  
AS DECIMAL(10, 2) CHECK (VALUE > 0);
```

```
CREATE DOMAIN D_CALIFICACIONES  
AS SMALLINT CHECK (VALUE >= 0 AND VALUE <= 100);
```

Excepciones

- En Firebird una excepción es un mensaje que el Servidor le envía al Cliente para informarle que ocurrió un error.
- Cuando se envía una excepción el servidor **Firebird** detiene lo que estaba haciendo y regresa al nivel anterior.
- Las excepciones se usan en los *procedimientos* y en los *disparadores*.

Ejemplo

```
CREATE EXCEPTION ID_EXISTE 'id existente';
```

Roles

- Un rol se define a nivel base de datos.
- Un usuario esta definido a nivel servidor y disponible para cualquier base de datos con sus roles y permisos adecuados.
- Se puede implementar reglas de seguridad sin utilizar roles, sólo con usuarios definidos en el servidor. Se crea un usuario con su password y se le asigna sus permisos en los diferentes objetos de la base de datos, pero si existen muchos usuarios se debe realizar esta tarea para todos ellos. Un rol se define en la base de datos y contiene una serie de permisos para ese rol, luego se relaciona ese rol con los usuarios del mismo perfil.

Roles

Ejemplo

```
create user romer password 'romer'
```

```
create rol rolselect
```

```
grant select on tabla1 to rolselect
```

```
grant rolselect to romer
```

Generadores

En Firebird son variables públicas e internas.

Una variable pública se puede usar en cualquier `SELECT`, *stored procedure* o *trigger*.

Una variable Interna significa que fuera de la Base de Datos no existe.

Aunque se puede cambiar el valor de un generador, lo correcto es que el servidor Firebird lo cambie.

Ejemplo

```
CREATE GENERATOR GEN_CLIENTE_ID;
```

```
SET GENERATOR GEN_CLIENTE_ID TO 0;
```

```
DROP GENERATOR GEN_CLIENTE_ID;
```

Generadores

Función `GEN_ID()`

Incrementa un generador y retorna el valor generado.

Ejemplo

```
NEW.ID = GEN_ID(GEN_CLIENTE_ID,1);
```


Disparadores

Los triggers (disparador en castellano) son rutinas, que se ejecutan en forma automática cuando se cumple alguna de estas condiciones:

- Se está insertando una nueva fila
 - Se está borrando una fila
 - Se está actualizando una fila

Los triggers se “disparan” automáticamente:

- Antes de la inserción, borrado, actualización
- Después de la inserción, borrado, actualización

Se usan para

- Asegurarnos que en la tabla se guarden datos válidos.
- Para actualizar una tabla cuyos datos dependen de otra tabla

Stored Procedures

En los stored procedures y en los triggers se procesan los datos.

Los stored procedures (procedimientos almacenados) son los equivalentes a las rutinas, procedimientos, o funciones disponibles en casi todos los lenguajes de programación , allí se utilizan los datos que se encuentran en las tablas, se realizan operaciones aritméticas o lógicas sobre esos datos y se devuelve el resultado de ese procesamiento.

Los stored procedures pueden recibir cero, uno, o varios parámetros de entrada y devolver cero, uno, o varios parámetros de salida.

En **Firebird** hay dos clases de *stored procedures*: Ejecutables y Seleccionables.

Vistas

En el **Firebird** se llaman *vistas* a las consultas (SELECTs) que se guardan dentro de la Base de Datos.

Las ventajas son:

- Mayor rapidez para obtener los resultados
- Mayor seguridad en los datos
- Confianza, el SELECT que escribiste es correcto y funciona.
- La mayor rapidez se debe a que al estar las *vistas* compiladas dentro de la Base de Datos son mucho más rápidas que los SELECTs equivalentes.
- La mayor seguridad se debe a que si a los usuarios solamente se les otorgan derechos sobre las *vistas* pero no sobre las tablas usadas en esas *vistas*, dichos usuarios no podrán ni ver ni alterar el contenido de esas *tablas*, solamente verán lo que en la vista se les permite que vean y nada más.
- Las *vistas* se usan exactamente igual que el uso de las tablas dentro de los SELECTs, para el **Firebird** una *vista* es lo mismo que una *tabla*.

Funciones definidas por el usuario UDF

Las funciones en FireBird son conocidas como funciones UDF (o Funciones Definidas por el Usuario). FireBird permite, crear nuestras propias funciones, si tenemos los conocimientos necesarios para ello.

Una UDF es una función escrita en cualquier lenguaje de programación y compilada dentro de una librería compartida (*.so en Linux, *.dll en Windows), que puede ser usada en consultas SQL, procedimientos almacenados y triggers.

Programación de Triggers

Casos típicos

- Implementación Generator sobre tabla para generar ID en forma automática.
 - Implementación Tipo-Subtipo mutuamente exclusivos.
 - Implementación de cardinalidad máxima.
- Implementación Entidad Débil-Entidad Fuerte con discriminador ascendente consecutivo.
 - Implementación de log de actividad sobre una tabla
 - El usuario o la aplicación interactúa directamente con las tablas de la DB
- El usuario o la aplicación no interactúa directamente con las tablas de la DB