BASES DE DATOS ACTIVAS

Docentes : Ing. Fernando Sato

A.S. Sebastian Trossero

Versión: 201904021230

RESUMEN

- Introducción
- Caracteristicas.
- Paradigma ECA.
- Modelo de Conocimiento.
- Modelo de Ejecución.
- Casos de Estudio.

BASES DE DATOS PASIVAS

Pasivas:

Todas las acciones sobre los datos resultan de invocaciones **explícitas** de los programas de aplicación.

(Por DML Insert – Update y Delete para los servidores SQL Relacionales).

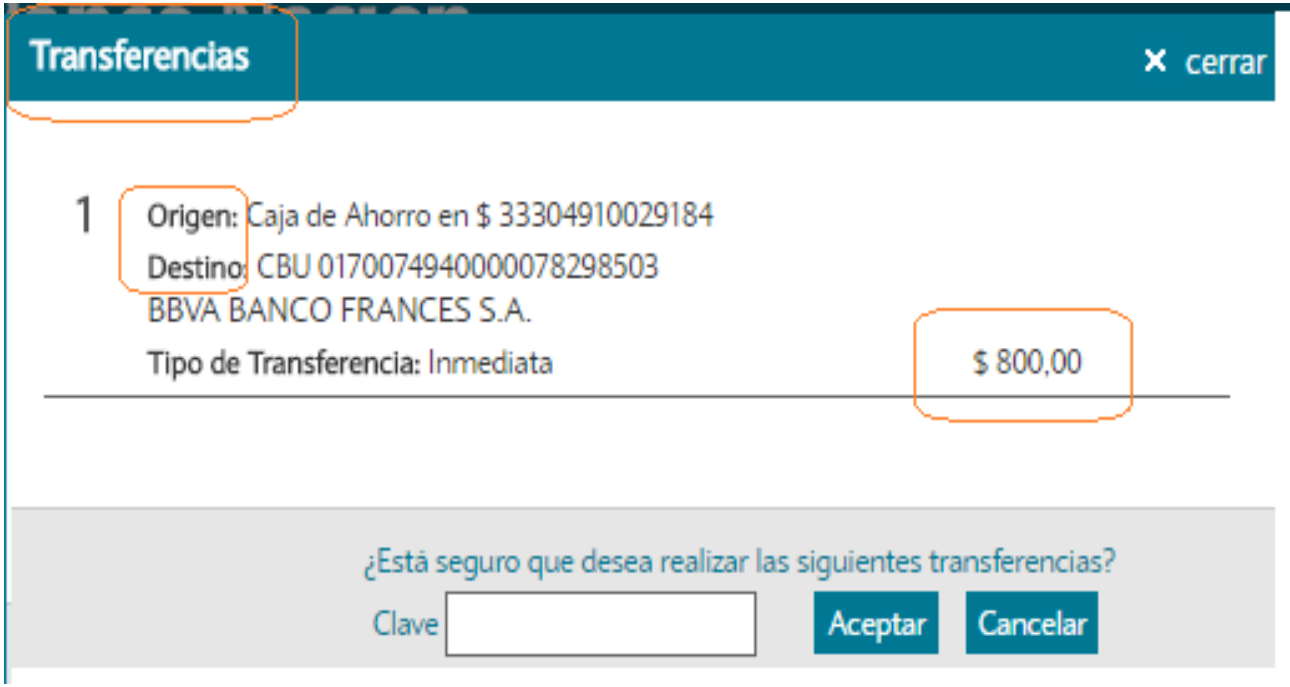
Caso de Análisis - Definición

El Banco K permite realizar una transacción denominada **TRANSFERENCIA** DE CUENTAS INTERNAS, extrayendo un **Importe** determinado de una cuenta **Origen (debito)**, y acreditandolo en una cuenta **Destino (credito).**

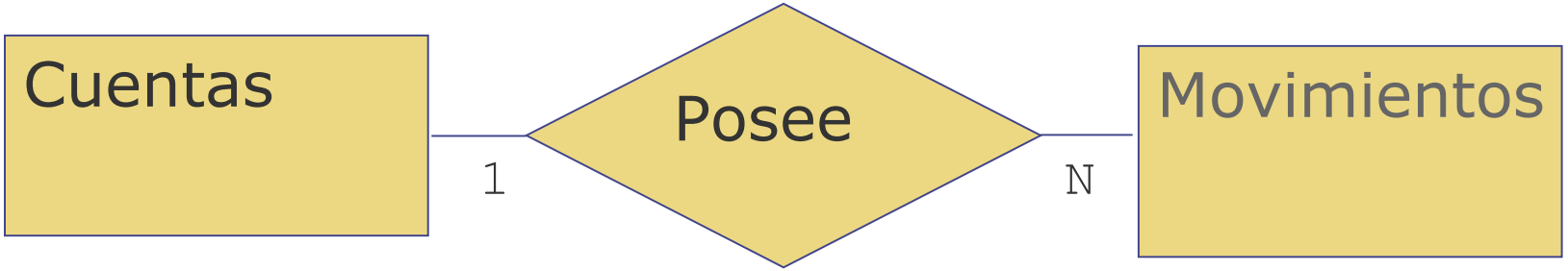
Este debito y credito se debe registrar en la tabla **MOVIMIENTOS**.

Ademas, todo **Movimiento** Debe ajustar el Atributo **Saldo** De la tabla **Cuentas**.

Regla de Negocio



Caso de Análisis – Esquemas BD



Numero
Nombre
Saldo

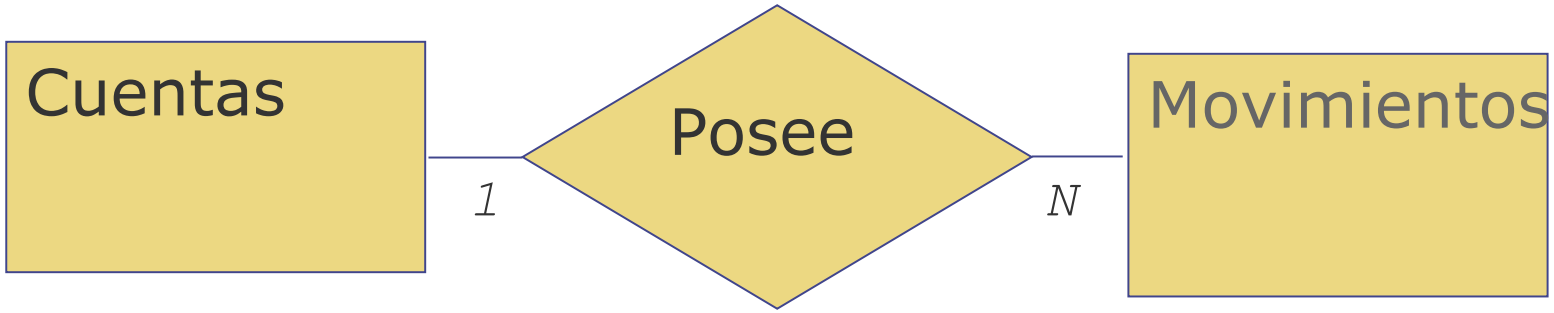
NumeroTransaccion
Fecha
CuentaNumero
DebitoCredito
Importe

Definiciones

```
create tabble cuentas (
 Numero int not null primary key,
 Nombre varchar(50),
 Saldo numeric(15,2)
 );
create domain debitocredito as
Char(1) check(value in ('D','C'));
```

```
create table movimientos (
 NumeroTransaccion int not null,
 Fecha Date,   CuentaNumero int,
 DebitoCredito debitocredito,
 Importe Numeric(15,2),
Foreign key (CuentaNumero)
References Cuentas(numero));
```

Caso de Análisis – Trazo Fino



Numero
Nombre
Saldo

NumeroTransaccion
Fecha
CuentaNumero
DebitoCredito
Importe

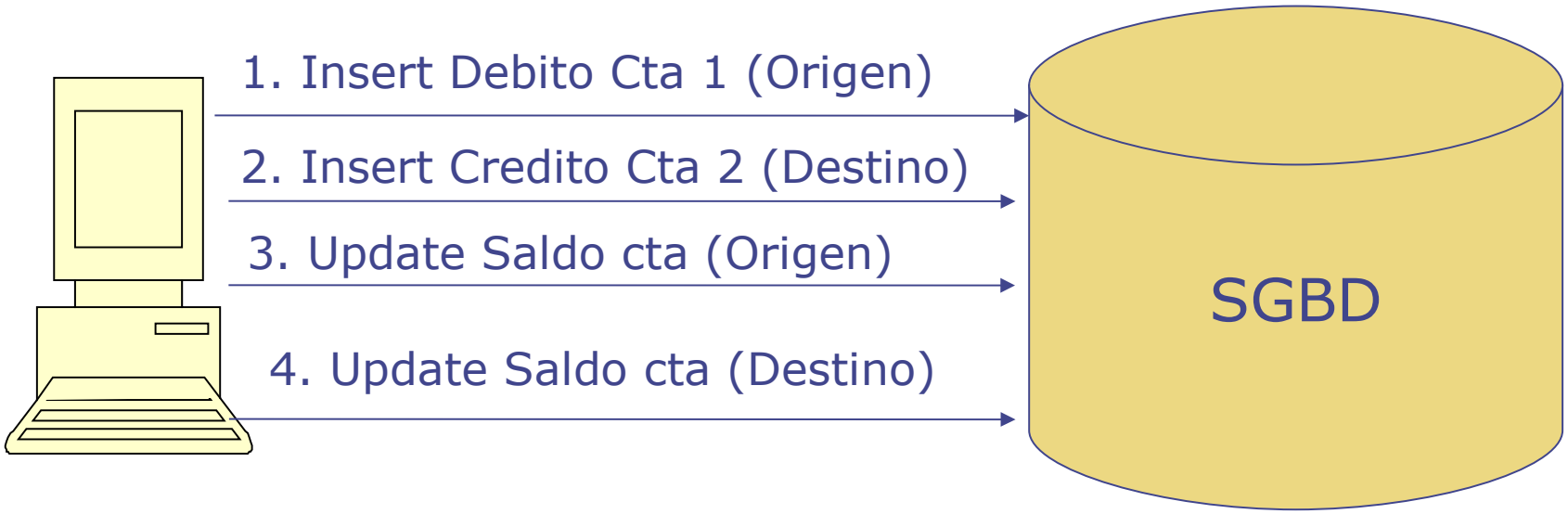
Regla de Negocio

Si insertamos un movimiento de DebitoCredito con 'D' = debito, el saldo de la cuenta asociada se decrementará en valor correspondiente a importe.

Si insertamos un movimiento de DebitoCredito con 'C' = crédito, el saldo de la cuenta asociada se incrementará en valor correspondiente a importe.

Caso de Análisis – Diseño - Problema

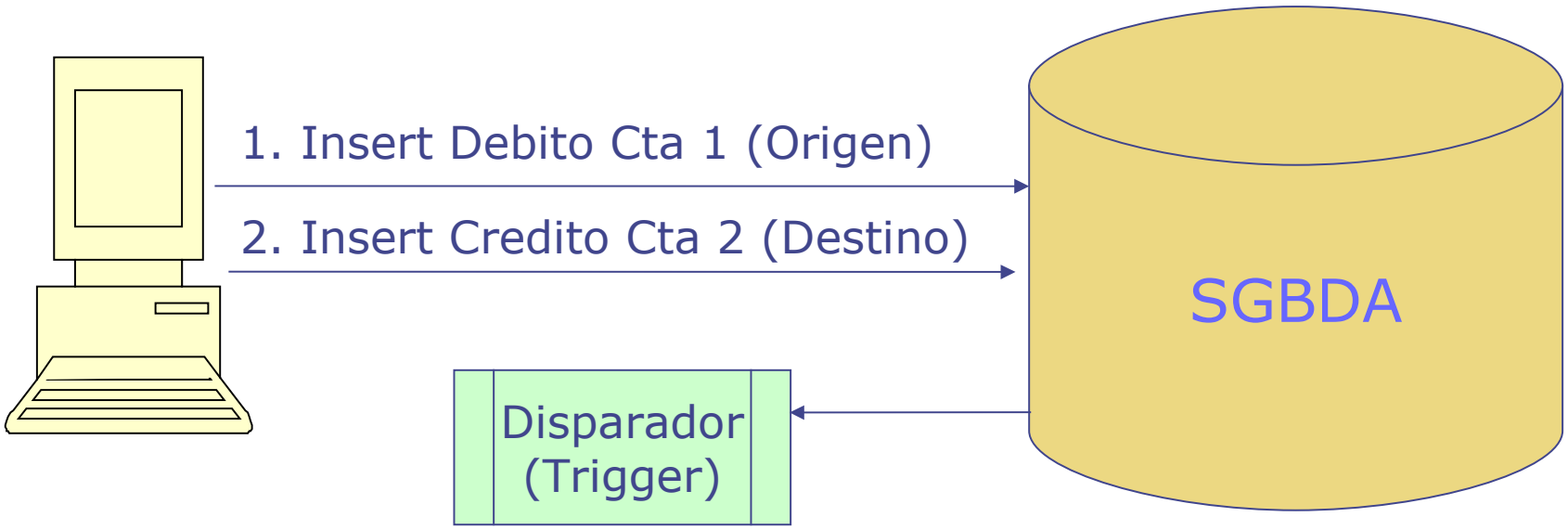
Transferencia bancaria entre cuentas.



Comportamiento Completamente PASIVO

Caso de Análisis – Diseño - Solución

Transferencia bancaria entre cuentas.



AUTOMATICO

3. Trigger actualiza saldo (Origen)
4. Trigger actualiza saldo (Destino)

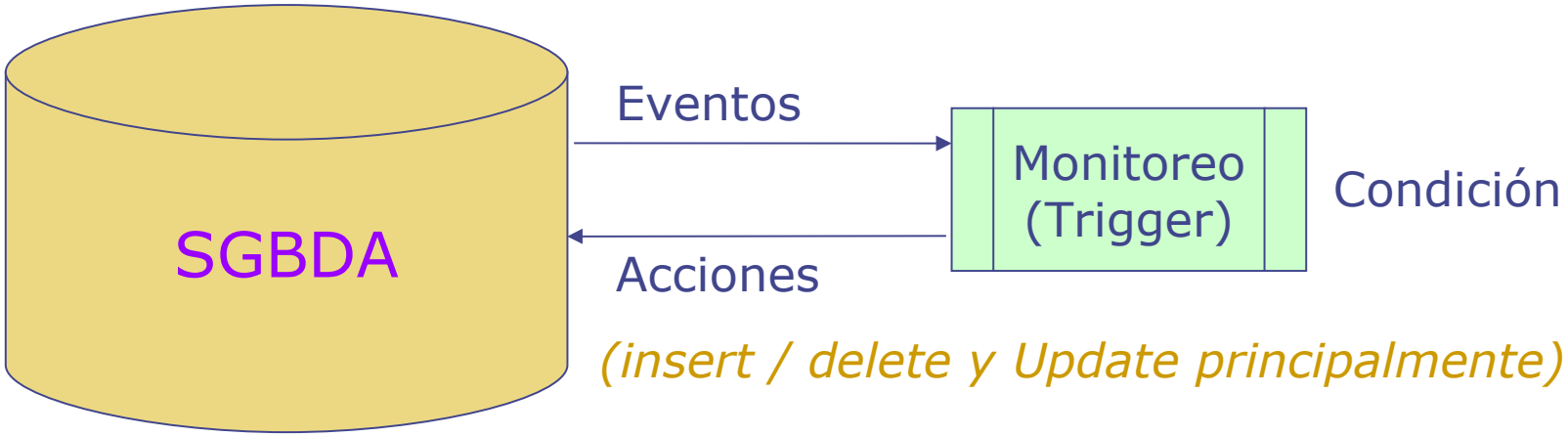
Comportamiento ACTIVO realizado por trigger de BDA

BASES DE DATOS ACTIVAS

Concepto

Una Base de Datos Activa es capaz de detectar situaciones de interés y de actuar en consecuencia.

Debe ser capaz de monitorear y reaccionar ante eventos de manera oportuna y eficiente.



BASES DE DATOS ACTIVAS

Características

Los SGBD Activos (SGBD**A**):

Proporcionan mecanismos para definir el **cuando** y el **qué** mediante la definición de Reglas Activas. (**Modelo de Conocimiento**).

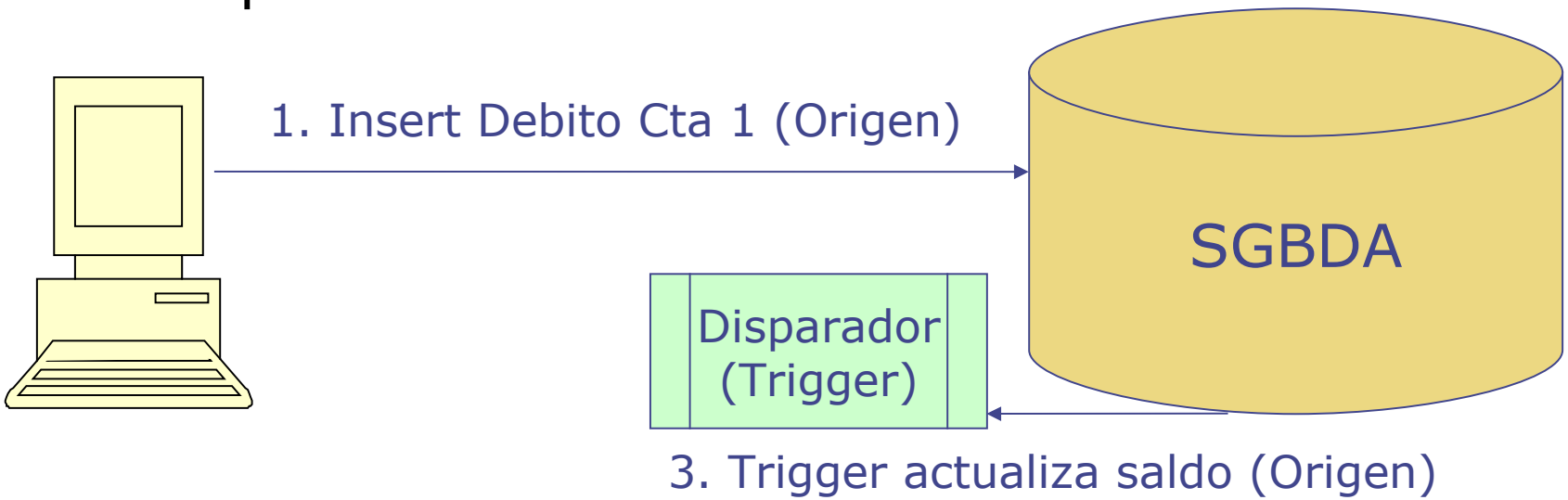
El **cuando** y el **qué**, definen una Regla Activa. (**Modelo de Ejecución**). ***Implementan -> Trigger.***

Opcionalmente realizan la evaluación de una condición que determina si se ejecuta el **que**.

BDA: *Deben tener la capacidad de almacenar las Reglas al igual que se almacenan definiciones de Tablas y Vistas.*

BASES DE DATOS ACTIVAS
Características (II)

Entonces → Estos conceptos aplicados al caso de Análisis queda.



Regla de Negocio → ***Modelo de Conocimiento*** → ***Regla Activa***

Si insertamos un movimiento de DebitoCredito con 'D' = debito, el saldo de la cuenta asociada se incrementará en valor correspondiente a importe.

Si insertamos un movimiento de DebitoCredito con 'C' = crédito, el saldo de la cuenta asociada se incrementará en valor correspondiente a importe.

BASES DE DATOS ACTIVAS
Características (III)

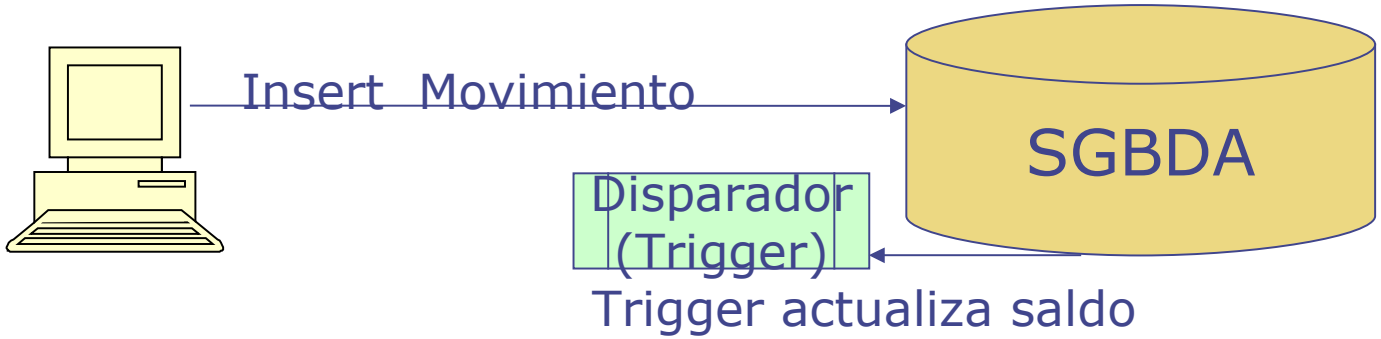
Definición de Regla Formal

Una Regla Activa es un conjunto de acciones que el SGBD ejecuta cuando se produce un evento determinado y se "da" una condición determinada.

Mediante estas reglas se puede:

- *Hacer respetar reglas de integridad*
- *Generar datos derivados propagando actualizaciones*
- *Grabar Pistas de auditorías y manejo de versiones*
- *Implementar Vistas Materializadas*
- *Implementar reglas de negocios en general*

BASES DE DATOS ACTIVAS
Características (IV)



Comportamiento Activo = **CUANDO** + **QUÉ**

Cuando se inserte un Movimiento entonces **se actualiza el saldo** de la Cuenta asociada.

Todo **Movimiento** Debe ajustar el atributo **Saldo** de la tabla **Cuentas**.

Nota:
*En este caso no se presenta condición, **se da siempre.***

BASES DE DATOS ACTIVAS
Características (V)

¿y que pasa con la **Independencia** y La **Consistencia**?

BASES DE DATOS ACTIVAS
Características (V')

Independencia y Consistencia

Una Base de Datos Activa tiene mayor independencia y consistencia en sus datos que una Pasiva, porque en la misma se encuentran definidas las reglas de integridad que se deben cumplir, de esta manera las aplicaciones son más independientes porque deben realizar menos "controles" sobre ellos y un cambio en estas no implica la propagación del cambio en las aplicaciones.

Pensemos una solución multiplataforma para el caso de uso de transferencias bancarias antes planteado.

BASES DE DATOS ACTIVAS
Características (VI)

Triggers

Todos los eventos causados por las transacciones de las Bases de Datos son monitoreados por el DBMS y automáticamente dispara acciones (triggers) asociados a estos.

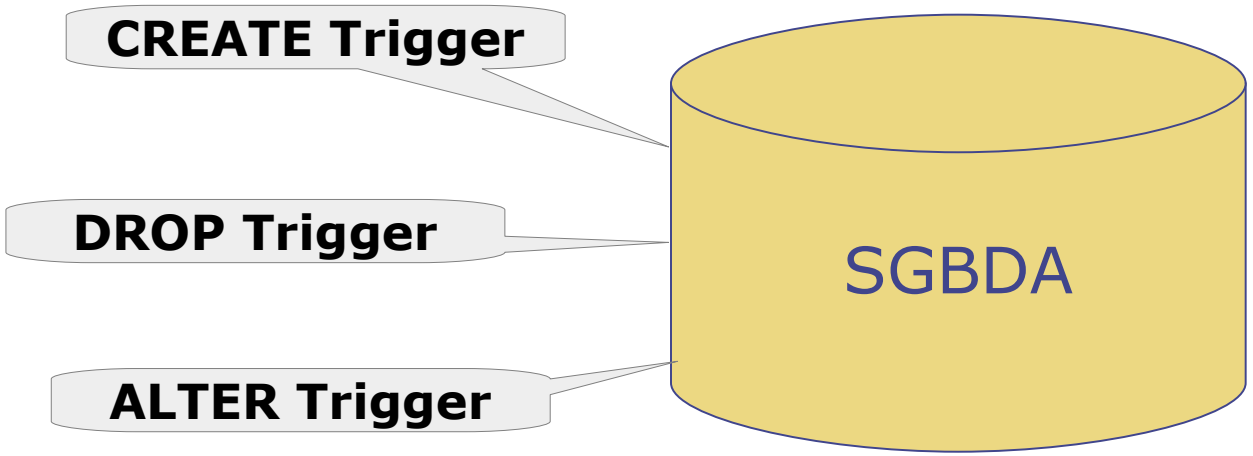
Efecto: semántica activa codificada en el schema y disponible para todas las aplicaciones.

Concretamente:
Las reglas activas se implementan con triggers en todos los productos.

BASES DE DATOS ACTIVAS
Características (VII)

Resumen SGBA

Un sistema de bases de datos activas es un sistema de gestión de bases de datos (SGBD) que contiene la capacidad de definición, almacenamiento y gestión de reglas activas (Triggers).



RESUMEN

Introducción

Caracteristicas.

Paradigma ECA.

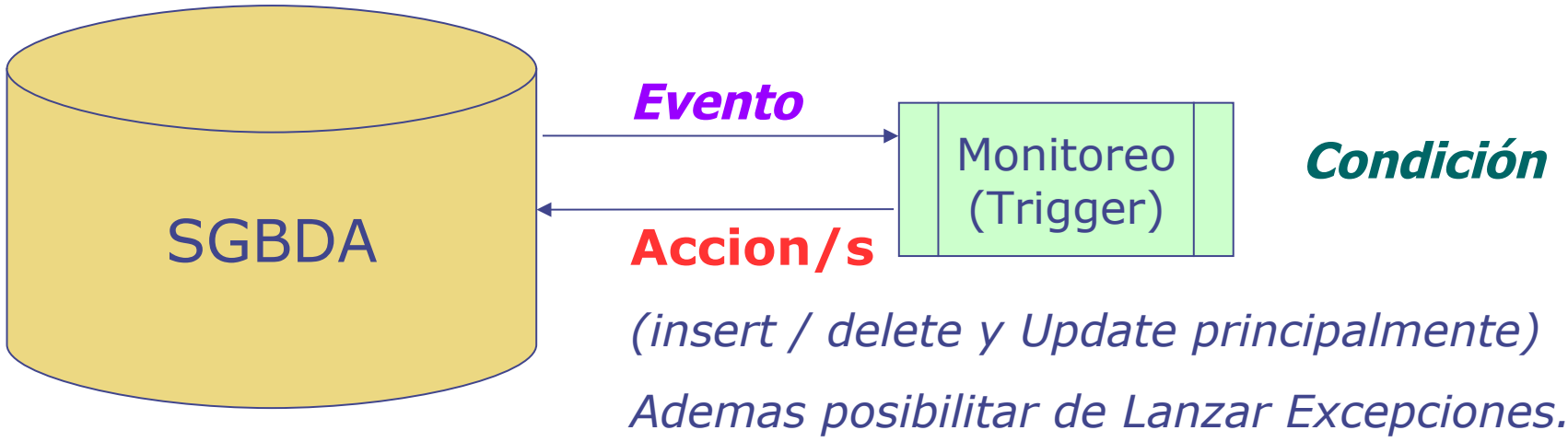
Modelo de Conocimiento

Modelo de Ejecución

Casos de Estudio.

BASES DE DATOS ACTIVAS
Paradigma ECA

Este paradigma se denomina ***Evento*-*Condición*-Acción** porque reacciona a los eventos que ocurren sobre los datos, evaluando condiciones que dependen de los datos y ejecutando una acción cuando esta es verdadera

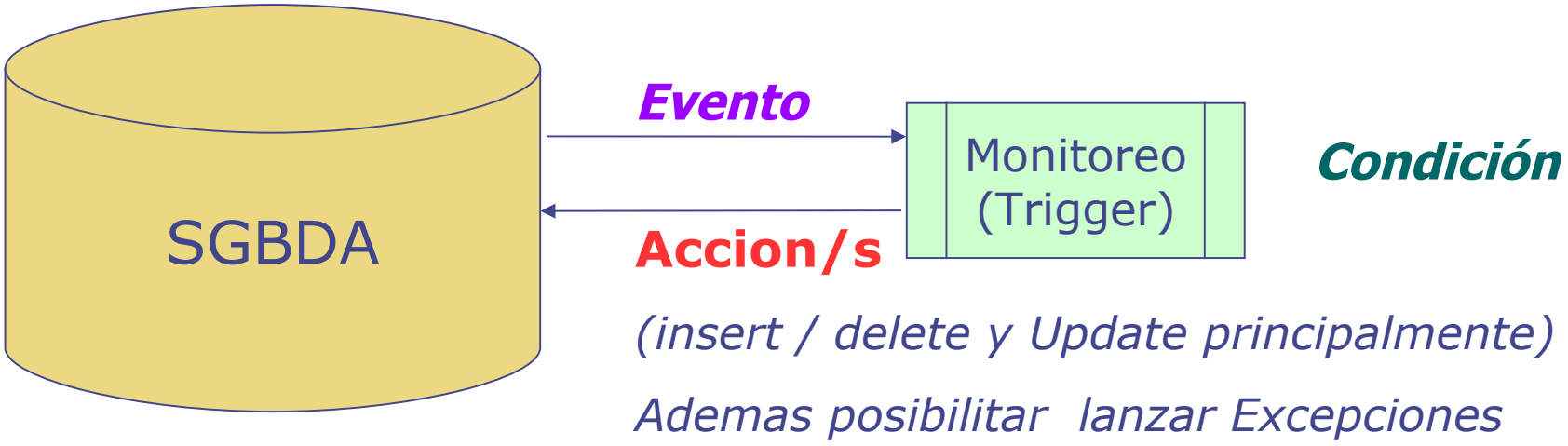


BASES DE DATOS ACTIVAS

Paradigma ***Evento-Condición-Acción***

Eventos Posibles

- DML que cambian el estado de la BD (inserciones, actualizaciones o borrado)
- Consultas
- Por cronograma (Viernes a las 5 hs.)
- Específicas de una aplicación (externas)



BASES DE DATOS ACTIVAS

Paradigma ***Evento-Condición-Acción***

Dupla {Tiempo Ocurrencia /Tipos de Eventos}

Un evento consiste como una pareja

`{<Tiempo Ocurrencia(Toc)>,<Tipo de evento>}`

Donde:

- `Toc (tiempo de ocurrencia)` corresponde al punto en el tiempo cuando ocurre dicho tipo de evento
- `Tipo de evento` es la descripción o especificación del evento a detectar.

Ejemplos: *{AFTER, INSERT} : {despues, Insert}*

BASES DE DATOS ACTIVAS

Paradigma ***Evento-Condición-Acción***

Los tipos de eventos pueden ser situaciones dentro de la base de datos o sucesos en el ambiente.

Tipos de Eventos → BD Relacionales, BD O.Objetos

Bases de datos relacionales:

- Insert
- Delete
- Update

Bases de datos orientadas a objeto:

- Creación de objetos
- Borrado de objetos
- Llamado a métodos

El tiempo también puede ser considerado un tipo de evento interesante tal como un punto absoluto en el tiempo ("9 de julio de 2013") o puntos relativos o periódicos en el tiempo ("cada viernes a las 14:30 hs.").

BASES DE DATOS ACTIVAS
Eventos Simples y Compuestos

Evento Simple:

La mayoría de las veces las reglas se limitan a monitorear un sólo evento.

Eventos compuestos:

En lenguajes + desarrollados se pueden definir eventos complejos desde los mas simples por medio del cálculo (conjunción, disyunción, negación, precedencia entre eventos).

Eventos Compuestos → Reglas Polivalentes:

En casos particulares las reglas monitorean una colección de eventos que se corresponden con una regla disyuntiva se disparan si alguno de los eventos ocurre: INSERT OR DELETE OR UPDATE.

Para estos casos es de suma utilidad detectar que evento disparó la regla/trigger. *"UNA REGLA POLIVALENTE monitorea varios eventos."*

BASES DE DATOS ACTIVAS

Paradigma ***Evento-Condición-Acción***

Historia de Eventos

Se denomina historia de eventos al conjunto de todos los eventos sucedidos en el tiempo. La historia inicia desde el momento en que se define el primer tipo de evento en la base de datos.

BASES DE DATOS ACTIVAS
Reglas Activas- Tiempo Ocurrencia

Inmediatas

Las acciones son ejecutadas

BEFORE → Antes de ...

AFTER → Después de ...

INSTEAD OF → En lugar de ...

respecto del evento que es monitoreado

Diferidas

La acción de la regla se ejecuta al final de la transacción, similar a AFTER, pero no necesariamente inmediatamente después.

BASES DE DATOS ACTIVAS

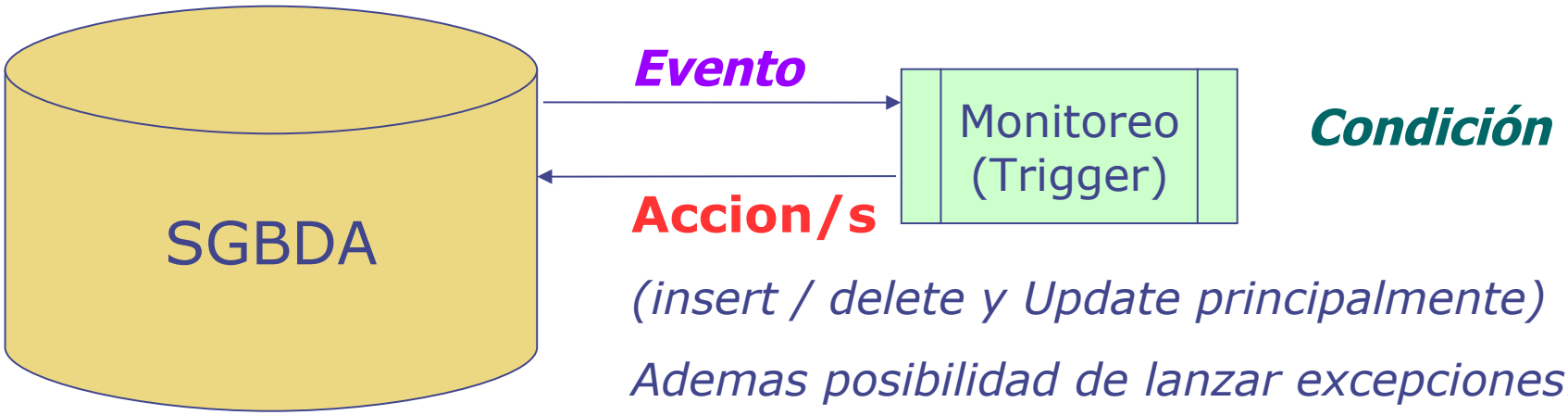
Paradigma ***Evento-Condición-Acción***

Condiciones

Predicados de BD

Consultas: los predicados se evalúan y arrojan, VERDADERO o FALSO.

Unicamente se activa (enjutará acción) si este resultado es VERDADERO.



(insert / delete y Update principalmente)

Ademas posibilidad de lanzar excepciones.

BASES DE DATOS ACTIVAS

Paradigma ***Evento-Condición-Acción***

Acciones

- *programas de manipulación de datos arbitrarios (SP / Funciones)*
- *comandos transaccionales (DMLs) en si.*
- *comandos de activación de reglas (activación, desactivación)*
- *llamadas a procedimientos externos (Funciones/Script Externas)*

(insert / delete y Update principalmente)

Ademas de posibilidad de lanzar excepciones.

BASES DE DATOS ACTIVAS

Paradigma ***Evento-Condición-Acción***

Las **acciones** se dividen en:

-Acciones Externas: Se dan cuando son especificadas por aplicaciones, por ejemplo enviar un correo electrónico (*email*), imprimir una orden de compra, activar un dispositivo hardware.

- Acciones Internas: Son acciones de la base de datos, como un *insert, update, delete.*

BASES DE DATOS ACTIVAS

Caso de Estudio: Punto de Reposición

Si el stock de un producto baja de un cierto valor estipulado solicitar provisión. **Para implementarlo:**

- Toda aplicación que modifique el stock de algún producto hay que añadir código que compruebe si se baja del umbral para solicitar provisión.
  - La semántica está distribuida por las aplicaciones.
  - Posiblemente es una fuente de errores.

Posible Solución

Realizando un programa que periódicamente "sondee" todas las condiciones `(¿stock[i] < puntoPedido[i]?)`

- Frecuencia de sondeo alta --> INEFICIENCIA
- Frecuencia de sondeo baja --> INCONSISTENCIAS

BASES DE DATOS ACTIVAS

Caso de Estudio: Punto de Reposición

Si quisiéramos pasar este caso a un esquema activo:

Cuales Serían Cada Elemento:

BASES DE DATOS ACTIVAS

Caso de Estudio: Punto de Reposición

Elementos:

Modificación del Stock

Enviar un Pedido al Proveedor

¿stock < puntoPedido?

BASES DE DATOS ACTIVAS

Caso de Estudio: Punto de Reposición

Pseudo Código Genérico:

```
ON evento IF condición THEN acción/s
```

Si se Produce Evento y se cumple una determinada condición en BD

Se realiza una acción.

Pseudo Código Especifico

```
ON Update  Producto(Stock)
    IF Producto.stock < Producto.PuntoPedido
    THEN Insert into Pedidos ….
```

¿Concretamente como se implementaría?

BASES DE DATOS ACTIVAS

Caso de Estudio: Punto de Reposición

Código PLpgSql:


```
CREATE TRIGGER trgReposicionPedidos
 AFTER UPDATE  OF stock ON Productos
 FOR EACH ROW
 WHEN (new.stock < new.PuntoPedido)
 Execute Procedure fxEnviarPedidos();

CREATE FUNCTION fxEnviarPedidos() RETURNS TRIGGER AS $$
  DECLARE
  BEGIN
 Insert into Pedidos (productoCodigo, fecha, cantidadPedida)
 values (new.codigo, current_date, new.PuntoPedido * 3);
   RETURN Null;
  END;
 $$
Language PLpgSql;
```

BASES DE DATOS ACTIVAS
Reglas Activas – Granularidad

Las Reglas Activas se pueden "colgar":

a nivel de instancia

Afectando tuplas o filas individuales de una tabla o vista.

a nivel de instrucción

Las DMLs ejecutadas se consideran como eventos

BASES DE DATOS ACTIVAS
Reglas Activas – Valores de Transición

Los **Valores de transición** se pueden considerar como variables locales *que describen los cambios de* estado realizados por una transacción:

a nivel de instancia

Los cambios afectan una tupla u objeto (variables NEW y OLD)

a nivel de instrucción

Valores de transición colectado en tablas (INSERTED, DELETED)

Las actualizaciones son tratadas de modo:

- **explícito:** OLD-UPDATED, NEW-UPDATED
- **implícito:** DELETED, INSERTED

Nota: *Estos conceptos son implementados en forma parcial o total por los SGBA en sus objetos triggers.*

BASES DE DATOS ACTIVAS

Reglas Activas – Selección de Reglas

Conflicto:

Varias Reglas pueden ser disparadas al mismo tiempo, esto da lugar a un (conjunto conflicto).

Las estrategias para seleccionar las reglas del conjunto conflicto se relacionan con prioridades:

- *Orden Total con Prioridad Numérica o Alfabética (Determinista)*
- *Orden Parcial con Prioridad Numérica o Alfabética (Determinista)*
- *Selección no Determinista*

Modelos Repetibles

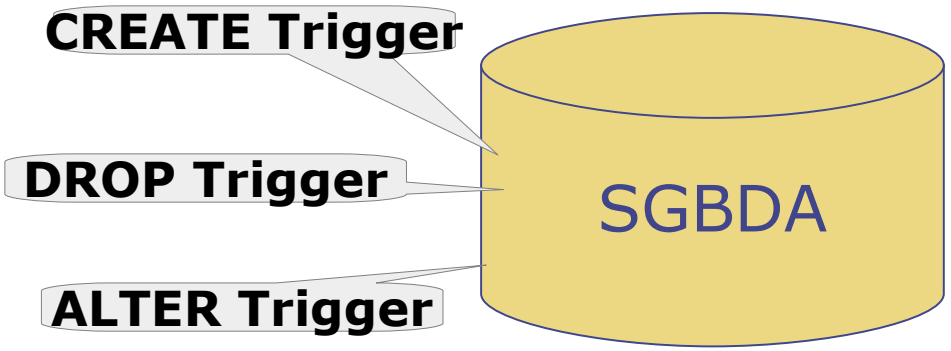
Aseguran que dos ejecuciones de la misma transacción sobre el mismo estado de la BD y el mismo conjunto de triggers produce la misma secuencia de Ejecución.

*Nota: Los Modelos de Selección Deterministas son **Repetibles.***

BASES DE DATOS ACTIVAS
Manipulación de Reglas

La manipulación de Reglas se controla con sentencias especificas:

Sentencias DDL: Create Alter Drop



Nota: Tenga en cuenta que la activación y desactivación con ALTER es peligrosa en un SBDA en producción.

BASES DE DATOS ACTIVAS
Reglas Activas – Clasificación

Las Reglas Activas se pueden clasificar de la siguiente manera:

- **Propagación de Actualizaciones** para manejo de la redundancia generalmente.
- **Validadoras** de la consistencia
- **Registración de Pistas para Auditar**
- **Versionado** de filas
- **Mantenimiento de Vistas Materializadas** de filas
- **Restauradoras** de la consistencia
- **Manejo de Valores por Defecto** de columnas y/o filas.

BASES DE DATOS ACTIVAS

Reglas Activas – Clasificación

REGLAS DE PROPAGACIÓN DE ACTUALIZACIONES

En un sistema de base de datos, su diseño se modela con la aplicación de relaciones **normalizadas**.

La aplicación de esta metodología en forma pura evita la redundancia, minimiza errores, sin embargo, usualmente algunos casos tienen baja perfomance.

Una alternativa que une ambos conceptos es modelar datos redundantes, por ejemplo de datos acumulados, pero con esta redundancia controlada por reglas activas.

El ejemplo mas claro de esto es el caso de análisis de transferencia bancaria.

Estas reglas se han denominado reglas de administración de la redundancia controlada. EJ: **Transferencia Bancaria**

BASES DE DATOS ACTIVAS

Reglas Activas – Clasificación

REGLAS VALIDADORAS

Los sistemas de base de datos actuales permiten realizar validaciones simples en forma declaratativa, por ejemplo la column-constraint "NOT NULL" o "UNIQUE", etc.

Pero determinadas operaciones complejas no se pueden realizar en forma declarativa, como por ejemplo, un control como el siguiente que plantea si un salario asignado a un puesto esta entre su valor máximo y mínimo, para ese puesto.

Ej: Ver pagina siguiente →

BASES DE DATOS ACTIVAS
Reglas Activas – Clasificación

REGLAS VALIDADORAS (II)

```
CREATE TRIGGER CompruebaSalario
BEFORE INSERT OR UPDATE OF Salario, Puesto ON Empleado
FOR EACH ROW
DECLARE
                minsal NUMBRER;
                maxsal NUMBER;
                Salario_Fuera_Rango EXCEPTION;
BEGIN
        SELECT minsal, maxsal INTO minsal, maxsal
        FROM Salarios
        WHERE Puesto = :NEW.Puesto;
        IF (:NEW.Salario < minsal OR :NEW.Salario >
maxsal)
        THEN RAISE Salario_Fuera_Rango;
END IF;
```

BASES DE DATOS ACTIVAS
Reglas Activas – Clasificación

REGLAS VALIDADORAS (II) – Otra alternativa

```
CREATE TRIGGER CompruebaSalario2
BEFORE INSERT OR UPDATE OF Salario, Puesto ON Empleado
WHEN NOT EXISTS SELECT * FROM Salarios
        WHERE Puesto = :NEW.Puesto AND NEW.salario
         BETWEEN minsal AND maxsal
  FOR EACH ROW
BEGIN
       RAISE Salario_Fuera_Rango;
END;
```

BASES DE DATOS ACTIVAS
Reglas Activas – Clasificación

REGISTRACION DE PISTAS PARA AUDITAR

Las transacciones activas en el SGBD tiene asociado el usuario que "disparo" la misma (el usuario de conexión), por lo que es posible controlar la ejecución de la misma de acuerdo al usuario, además la posibilidad de saber el usuario nos permite mantener información necesaria para auditorias.

Ej. Incorporar el Usuario y la Fecha Actual al momento de realizar un UPDATE.

```
CREATE TRIGGER Actualizo_Clientes FOR Clientes
BEFORE  UPDATE ON Clientes AS
BEGIN
          New.UltMod_Usuario      = CURRENT_USER;
          New.UltMod_Fechahora    = CURRENT_TIMESTAMP;
END
```

BASES DE DATOS ACTIVAS
Reglas Activas – Clasificación

VERSIONADO DE FILAS

En todo sistema de aplicación existen un conjunto de tablas visibles que, mas allá de las restricciones DCL para su actualización, es conveniente registrar `quien, a que hora, que función realizo y su estado anterior y posterior.`

Por ejemplo, si tuviéramos una tabla llamada Salario con

```
Create Table Salarios (
Puesto integer not null primary key,
NombrePuesto Varchar(50),
Minsal numeric(15,2),
Maxsal numeric(15,2)
);
```

Agregando una tabla de historia con datos de salario y los sig:

```
Usuario varchar(32), fechatxn timestamp, funcion varchar(32));
```

BASES DE DATOS ACTIVAS

Reglas Activas – Clasificación

MANTENIMIENTO DE VISTAS MATERIALIZADAS

A diferencia de las vistas "normales" una vista materializada almacena físicamente los datos resultantes de ejecutar la consulta definida en la vista.

Este tipo de vistas materializadas realizan una carga inicial de los datos cuando se definen y posteriormente con una frecuencia establecida se actualizan los datos de la misma.

Con la utilización de vistas materializadas logramos aumentar el rendimiento de las consultas SQL además de ser un método de optimización a nivel físico en modelos de datos muy complejos y/o con muchos datos.

Una vez definida una vista materializada uno de los problemas que nos encontramos es el de la actualización de los datos. Para resolver este problema se usan Reglas Activas.

BASES DE DATOS ACTIVAS
Reglas Activas – Clasificación

REGLAS RESTAURADORAS DE LA CONSISTENCIA

En un modelo de base de datos las reglas de negocio se modelan como restricciones de este. Estas restricciones representan las propiedades del mundo real.

Cuando se intenta actualizar la base de datos y se presenta una restricción generalmente se rechaza la transacción que la ha provocado, devolviendo la base de datos al estado anterior a su ejecución.

Esta solución usualmente es poco satisfactoria. Una alternativa a este comportamiento consiste en que el sistema modifique el estado inconsistente, tomando una **decisión determinista**, de forma que se repare la violación provocada por la transacción de usuario respetando los cambios propuestos por ésta.

Estas reglas se han denominado reglas restauradoras y eligen el **menor de los males**.

BASES DE DATOS ACTIVAS

Reglas Activas – Clasificación

REGLAS DE MANEJO DE VALORES POR DEFECTO

Existen un sin numero de situaciones que requieren que una atributo de una fila asuma un valor por defecto establecido por una regla de negocio, como casos concretos podemos plantear:

- Administración de atributos autoincrementales, ej nro correlativo de socios, nro de remito de venta, etc.
- Fecha del Sistema, Ejemplo fecha de registración, fecha de devolución de libro, etc.
- Usuario que realizo la transacción (utilizando un usuario del sistema de aplicación como usuario de la base de datos), etc.

Estas reglas se han denominado reglas de administración de valores iniciales o valores por defecto de atributos o propiedades de las clases y tablas.

BASES DE DATOS ACTIVAS
MANIFIESTO DE LOS SGA

CARACTERISTICAS

Características de los SGBDA

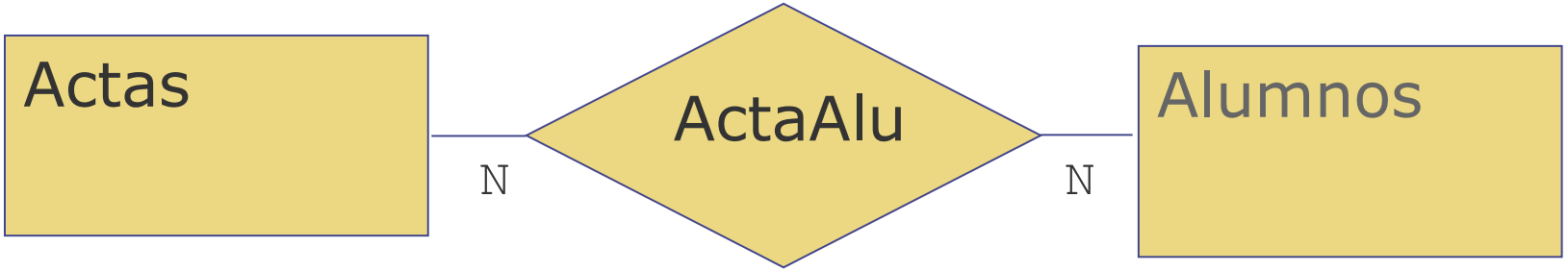
Un SGBDA es un SGBD.
Un SGBDA tiene un modelo de reglas ECA.
Un SGBDA debe soportar la gestión de reglas.
Un SGBDA debe gestionar la historia de eventos.
Un SGBDA debe implementar resolución de conflictos.

Características de ejecución de reglas ECA

Un SGBDA tiene un modelo de ejecución.
Un SGBDA debe ofrecer diferentes modelos de acoplamiento.

BASES DE DATOS ACTIVAS

Espacio de Discusión



Actas	ActaAlu	Alumnos
Numero	ActaNumero	LIbretaUniversitaria
NombreMateria	LibretaUniversitaria	Nombre
Fecha	Nota	
CantidadInscriptos		
CantidadAprobados		

Planteo

Dada las siguientes relaciones que representan el acta de alumnos, analice como administraría con reglas activas la redundancia controlada dada por Cantidad de Inscriptos y Cantidad de Aprobados.

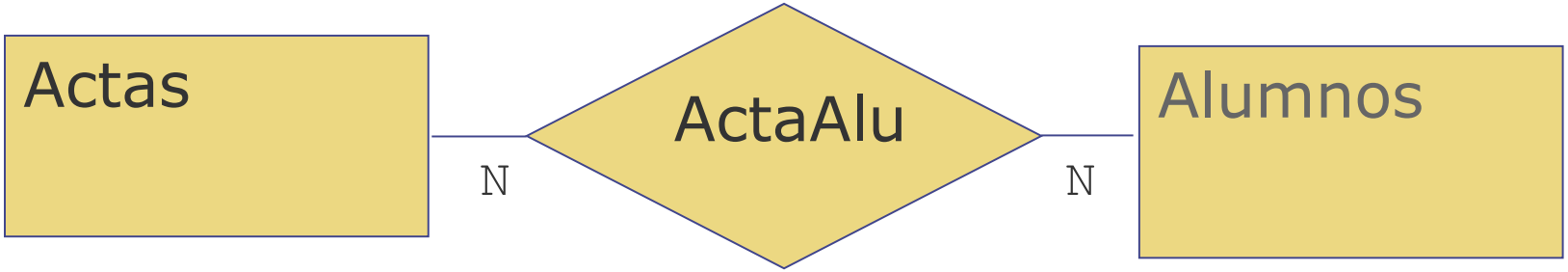
Se pide

a) Determine como reemplazaría `cantidadInscriptos` y `cantidadaprobados` usando sql, en que objeto de BD lo implementaría..

b) Que reglas plantearía para cada relación?

BASES DE DATOS ACTIVAS

Espacio de Discusión



Actas	ActaAlu	Alumnos
Numero	ActaNumero	LIbretaUniversitaria
NombreMateria	LibretaUniversitaria	Nombre
Fecha	Nota	
CantidadInscriptos		
CantidadAprobados		

c) Analice cada regla en cuanto a:

- Eventos (tiempo de ocurrencia, tipo de evento)
- Granularidad
- Tipo de Trigger
- Tabla
- Condición
- Acciones en sql

d) Describa cada regla y las acciones con sus palabras.

e) Intente realizar los triggers.

BASES DE DATOS ACTIVAS
Modelo de Ejecución - Concepto

MODELO DE EJECUCION

Se realiza un seguimiento de la situación y gestiona el comportamiento activo.

Especifica cómo un conjunto de reglas es tratado en tiempo de ejecución, se encarga de:

- realizar el seguimiento de la situación
- gestionar el comportamiento activo.

BASES DE DATOS ACTIVAS

Modelo de Ejecución - Concepto

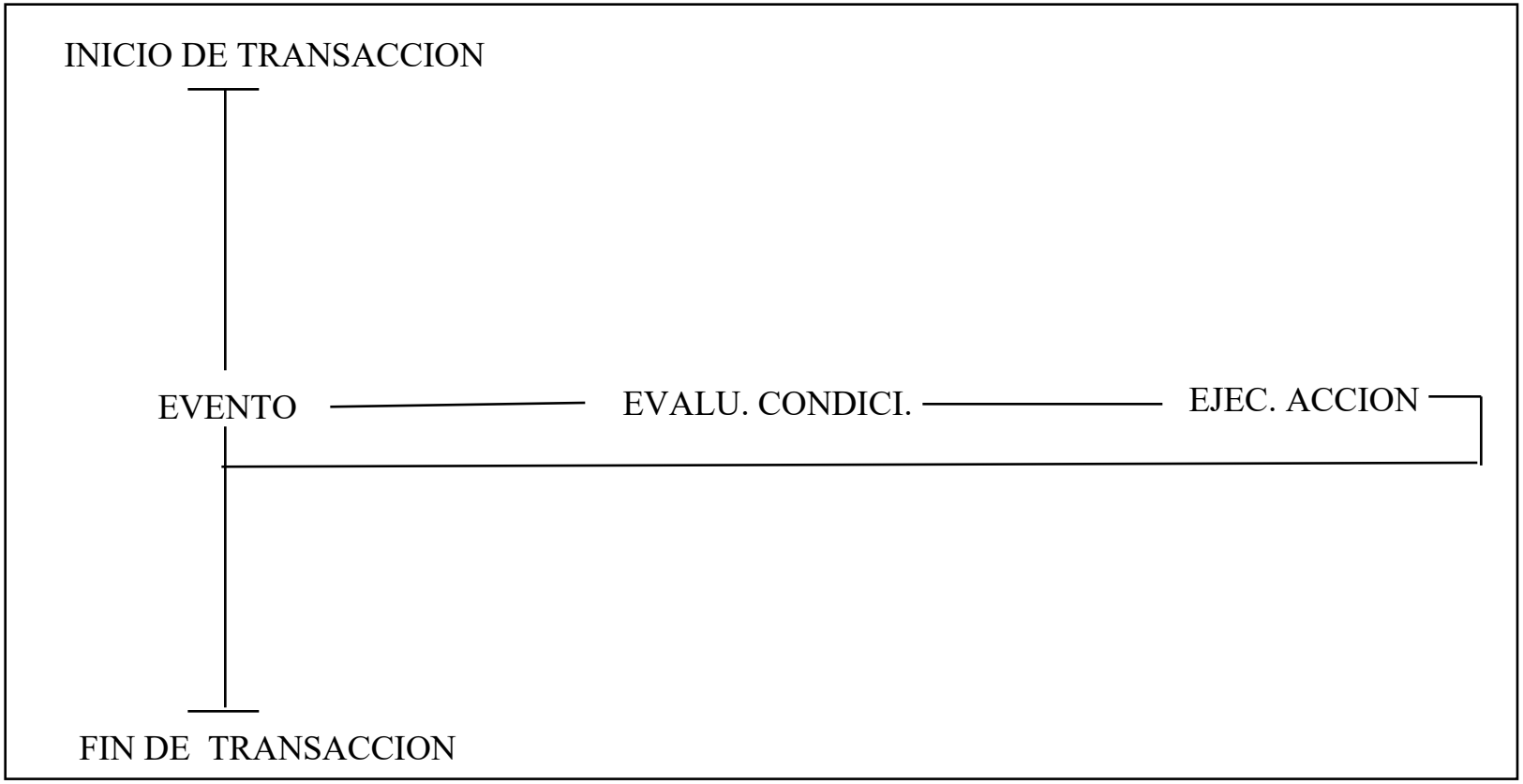
MODELO DE EJECUCION

- **Señalización:** En la señalización se detectan las ocurrencias de los eventos.
- **Activación:** en la activación se toman los eventos producidos y se registran las reglas correspondientes.
- **Planificación:** en la planificación se indica cómo se procesa el conjunto de reglas.
- **Evaluación:** Se selecciona una regla activada y se evalúa su condición.
- **Ejecución:** la ejecución es la que lleva a cabo las acciones de las reglas escogidas.

BASES DE DATOS ACTIVAS

Modelo de Acoplamiento Inmediato

MODELO DE ACOPLAMIENTO INMEDIATO



BASES DE DATOS ACTIVAS
Modelo de Acoplamiento Inmediato

MODELO DE ACOPLAMIENTO INMEDIATO

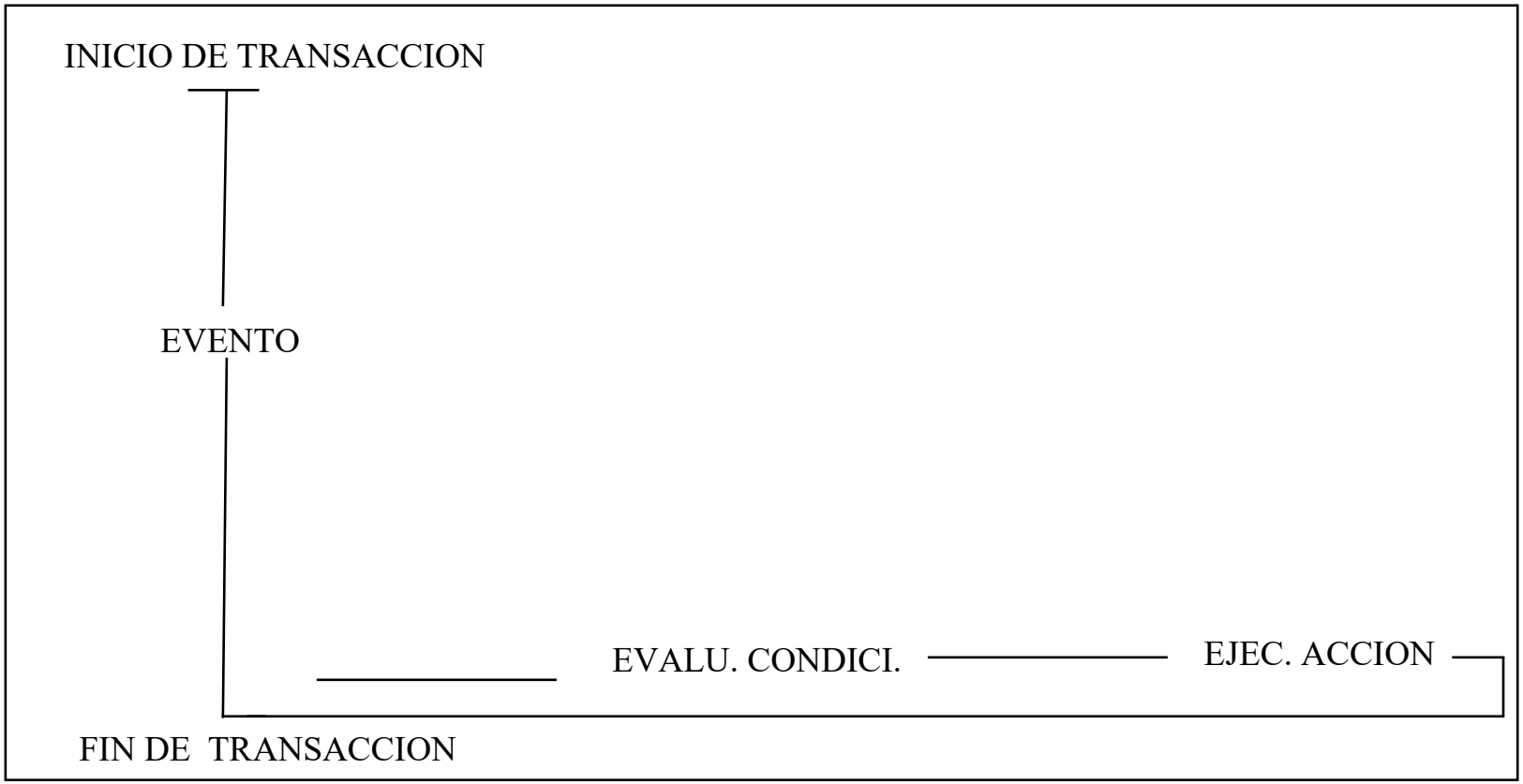
En este caso la condición es evaluada (verificada) inmediatamente después del evento.

Los modos inmediatos del acoplador se pueden utilizar, por ejemplo, para hacer cumplir trangresiones de la seguridad (reglas validadoras) o actualizar otras tablas asociadas (propagación de actualizaciones).

BASES DE DATOS ACTIVAS

Modelo de Acoplamiento Diferido

MODELO DE ACOPLAMIENTO DIFERIDO



BASES DE DATOS ACTIVAS
Modelo de Acoplamiento Diferido

MODELO DE ACOPLAMIENTO DIFERIDO

En este caso la condición se evalúa (verifica) dentro de la misma transacción que el evento que disparo la regla, pero no necesariamente despues de cada fila actualizada.

BASES DE DATOS ACTIVAS
Modelos Desacoplados

MODELO DE ACOPLAMIENTO DESACOPLADO

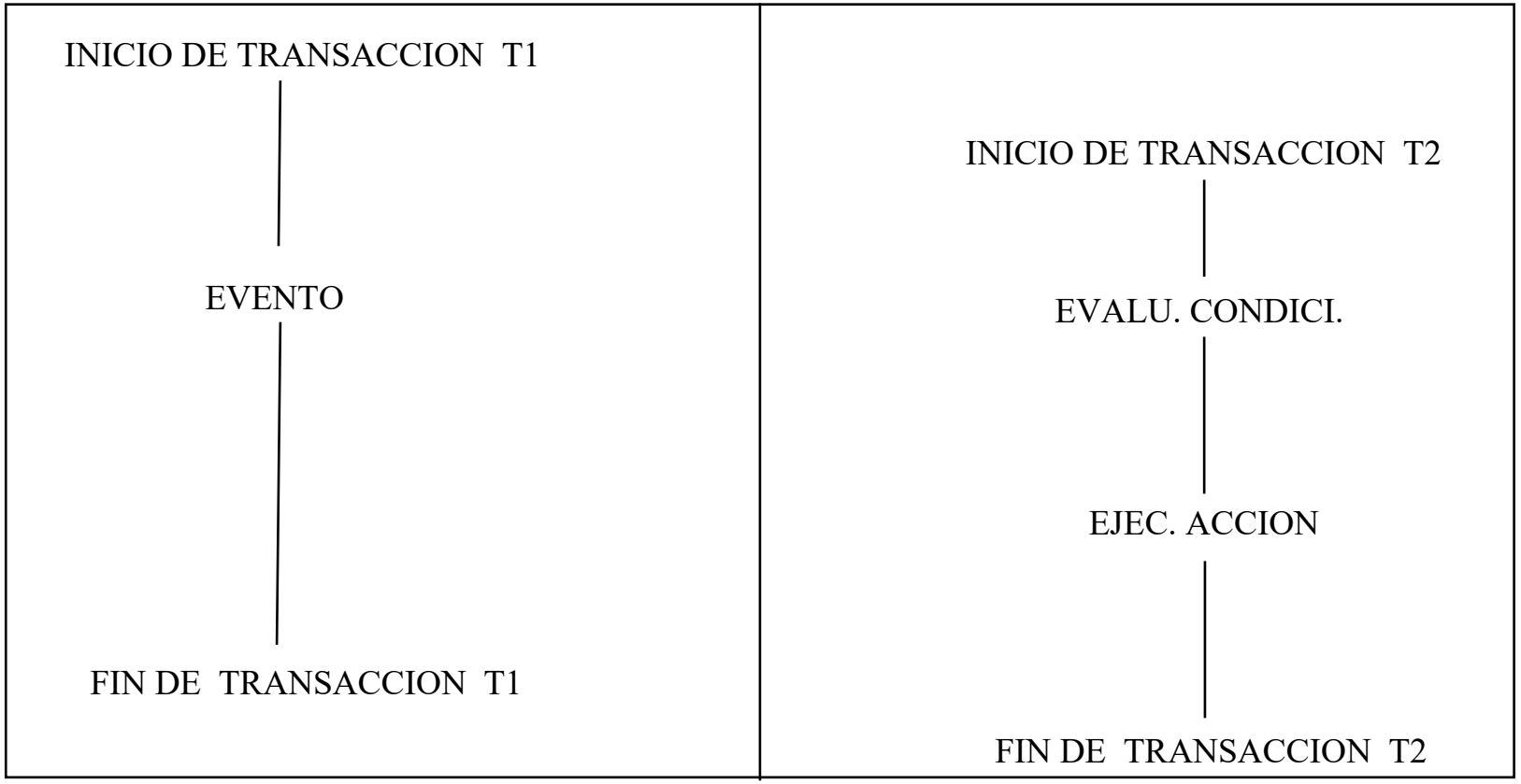
Independiente: este se produce cuando la condición, se evalúa (verifica), en una transacción diferente en la que se produjo el evento.

Dependiente: se produce cuando la condición, se evalúa(verifica), en una transacción diferente en la que se evalúa el evento; pero en este caso la ejecución es dependiente de la grabación (commit) de la transacción en la que el evento tiene lugar o en la que se evalúa la condición.

BASES DE DATOS ACTIVAS

Modelos Desacoplados

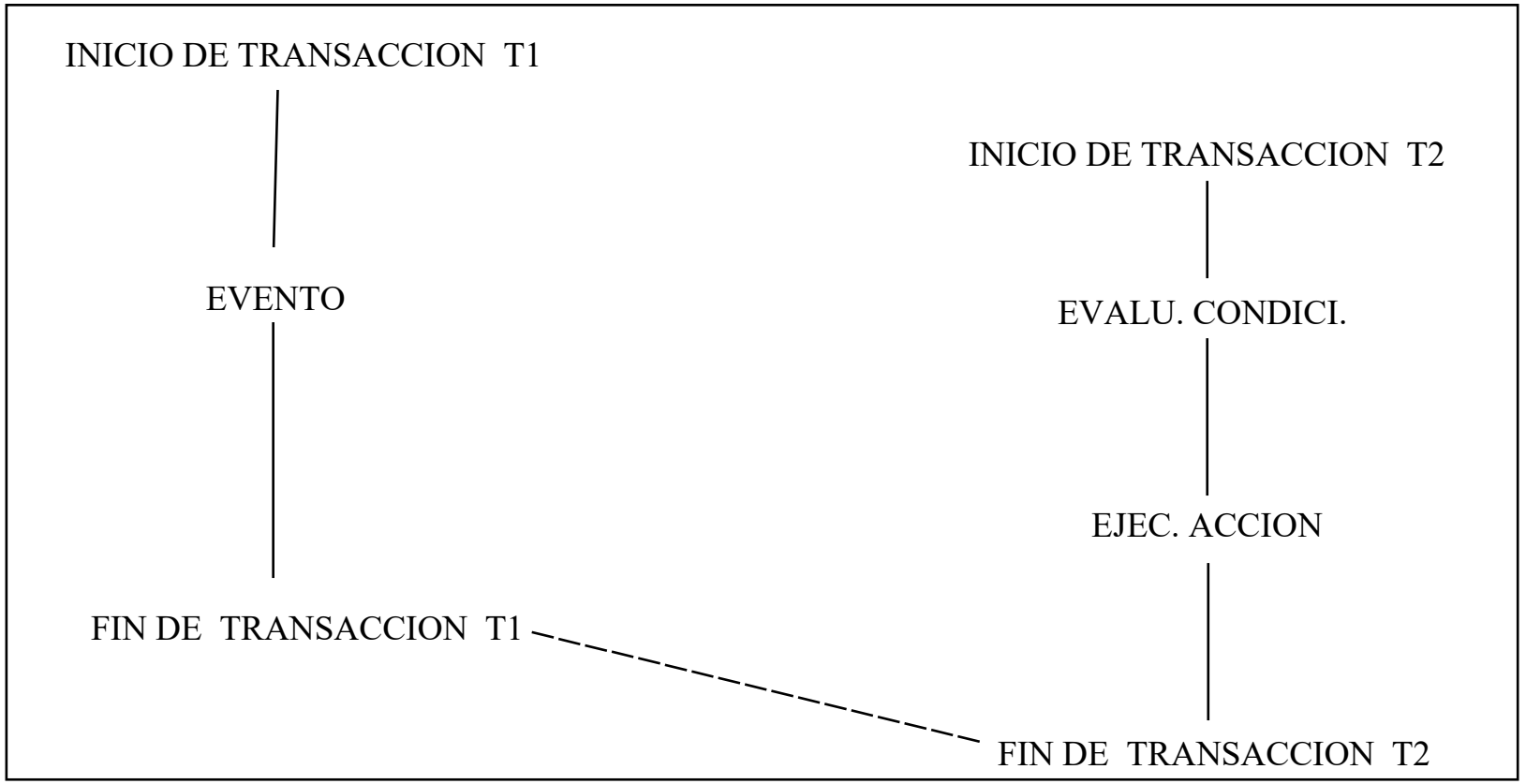
MODELO DE ACOPLAMIENTO DESACOPLADO INDEPENDIENTE



BASES DE DATOS ACTIVAS

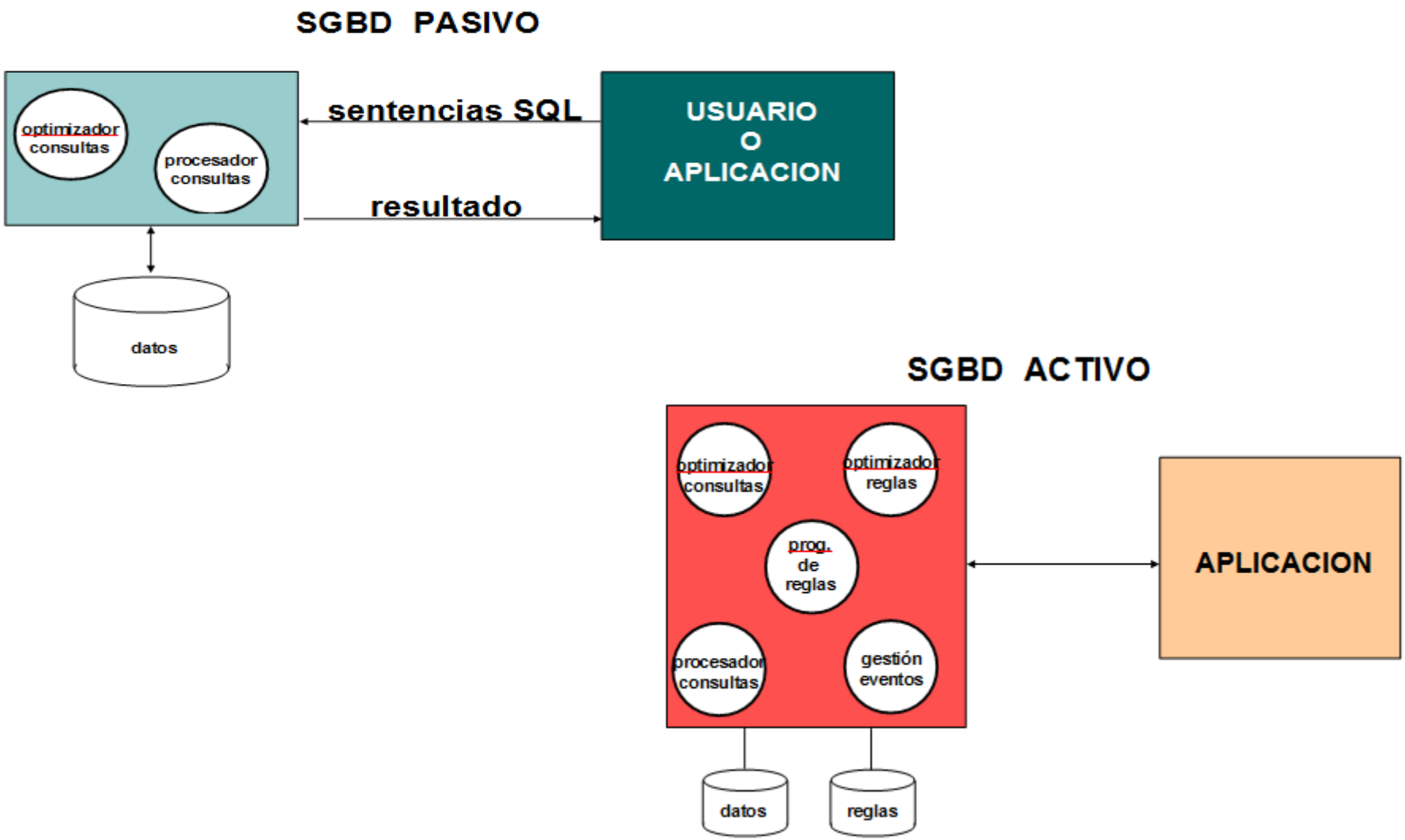
Modelos Desacoplados

MODELO DE ACOPLAMIENTO DESACOPLADO DEPENDIENTE



BASES DE DATOS ACTIVAS

Modulos de BDA y Pasiva



BASES DE DATOS ACTIVAS
Modulos de BDA y Pasiva

En la figura anterior se muestran las partes esenciales de un sistema de administración de bases de datos activa. En la parte inferior vemos una representación del lugar donde se guardan los datos y las reglas almacenadas.

-Aplicación: sistema con las necesidades particulares del negocio, el cuál se complementa con la BDA.

-Procesador de consulta: se encarga de convertir una consulta o manipulación de la base de datos, que puede estar expresada en un nivel muy alto (ejemplo, como consulta en SQL), en un serie de peticiones de datos almacenados.

BASES DE DATOS ACTIVAS
Modulos de BDA y Pasiva

-Optimizador de Reglas: se encarga de que todas las operaciones se realicen de la manera apropiada. En concreto, la ejecución apropiada requiere aplicar las propiedades ACID, abreviatura de los cuatro requisitos principales de la ejecución: **Atomicidad, Consistencia, Aislamiento, Durabilidad.**

-Optimizador de Consultas: es el encargado de seleccionar un buen plan de consulta o sea una serie de peticiones al sistema de almacenamiento que las atenderá.

BASES DE DATOS ACTIVAS
Modulos de BDA y Pasiva

-Programación de Reglas: se encarga de la programación de las reglas de la base de datos. Estas reglas son programadas (planificadas) en la base de datos para mantener la consistencia y la integridad de los datos.

-Gestor de Eventos: encargado de la detección de los eventos (ej. Insert, Update, Delete).

BASES DE DATOS ACTIVAS

Aportes que realiza

VENTAJAS DE UNA BD ACTIVA

Centralización de la Información y comportamiento: esto permite un mejor mantenimiento, ya que las reglas son almacenadas dentro de la Base de Datos, si es necesaria alguna modificación se hace sólo una vez en la BD en lugar de hacerlo en cada Programa. También permite una mayor Productividad, ya que los programas se simplifican.

BASES DE DATOS ACTIVAS

Aportes que realiza

VENTAJAS DE UNA BD ACTIVA

Encapsulamiento de Procedimientos: esto permite una mayor productividad ya que se pueden normalizar los procesos, sacando factor común de cierta lógica de los programas que se almacena una sola vez de forma centralizada. Permite la reutilización del código, ya que esta disponible cada vez que se necesite.

Minimiza tambien la transferencia de información.

BASES DE DATOS ACTIVAS
Problemas

DESVENTAJAS DE UNA BD ACTIVA

Standarización Escasa: Si bien el standart SQL 2003 define reglas generales para el encabezado el lenguaje interno de programación del trigger DEPENDE de cada implementación.

BASES DE DATOS ACTIVAS
Caso de Estudio

Planteo

En un sistema de RRHH se desea automatizar una regla de negocio que compruebe el salario máximo presupuestado y el mínimo posible para cada puesto de una empresa. Las tablas dispuestas por el responsable técnico del sistema son:

Empleado(legajo, nombre, salario, puesto)
Puesto FK con SalariosPorPuesto.puesto
SalariosPorPuesto(puesto, nombre, minsal, maxsal)

Pedido

- Se pide responda para la regla en cuestión:
  (a)Tabla
  (b)Eventos (tiempo de ocurrencia, tipo de evento)
  (c)Granularidad
  (d)Condición
  (e)Tipo de Trigger
  (f)Acciones en sql

BASES DE DATOS ACTIVAS

Caso de Estudio

Ej: de Implementación

```
CREATE TRIGGER CompruebaSalario
BEFORE(b) INSERT(b) OR UPDATE OF(b) Salario, Puesto ON Empleado(a)
FOR EACH ROW(c)
DECLARE
        minsal NUMBRER;
        maxsal NUMBER;
        Salario_Fuera_Rango EXCEPTION;
BEGIN
    SELECT minsal, maxsal INTO minsal, maxsal
    FROM SalariosPorPuesto
    WHERE Puesto = :NEW.Puesto;
    IF (:NEW.Salario < minsal OR :NEW.Salario > maxsal)
      THEN RAISE Salario_Fuera_Rango;
    END IF;
END
```

(d): Sin condición (referida a WHEN).
(e): Tipo de Trigger – Validación compleja.
(f): Desde Begin_a_End.

BASES DE DATOS ACTIVAS

Caso de Estudio

Ej: de Implementación

```
CREATE TRIGGER CompruebaSalario
BEFORE(b) INSERT(b) OR UPDATE OF(b) Salario, Puesto ON Empleado(a)
FOR EACH ROW(c)
DECLARE
        minsal NUMBRER;
        maxsal NUMBER;
        Salario_Fuera_Rango EXCEPTION;
BEGIN
    SELECT minsal, maxsal INTO minsal, maxsal
    FROM SalariosPorPuesto
    WHERE Puesto = :NEW.Puesto;
    IF (:NEW.Salario < minsal OR :NEW.Salario > maxsal)
      THEN RAISE Salario_Fuera_Rango;
    END IF;
END
```

(d): Sin condición (referida a WHEN).
(e): Tipo de Trigger – Validación compleja.
(f): Desde Begin_a_End.