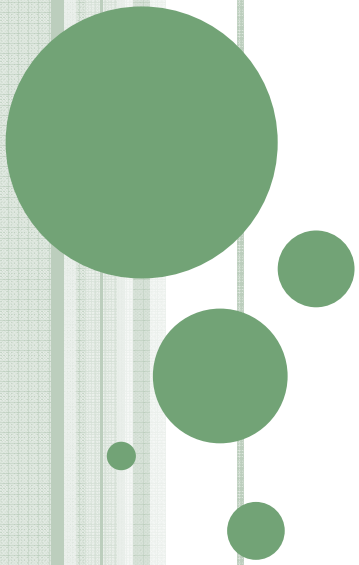
SISTEMAS GESTORES DE BASES DE DATOS OBJETO-RELACIONAL (SGDBOR)

"PROCEDIMIENTOS ALMACENADOS"

POR:

NELSON LÓPEZ

Skype: nelson_hidalgo



ÍNDICE

1. Introducción
2. Debilidades del Modelo Relacional
3. Clasificación de Michael Stonebreaker
4. Definición de SGBOR
   - Ventajas
   - Desventajas
5. Características de los SGBOR
6. Tipos de datos de gran tamaño
7. Tipos definidos por los usuarios
8. Constructores de Tipos (ROW y Arrays)
9. Procedimientos almacenados
   - Funciones
   - PA
   - Desencadenantes
10. Ventajas y Desventajas
11. Ejemplos de PA

INTRODUCCIÓN

Los sistemas relacionales de bases de datos soportan un pequeño conjunto fijo de tipos de tipos de datos (ej. Enteros, fechas, cadenas de caracteres) que son adecuados para los dominios de aplicaciones tradicionales.

Sin embargo en muchas aplicaciones, se deben manejar tipos de datos muchos más complejos (ej. El diseño y modelado asistido por computador CAD/CAM, repositorios multimedia, y gestión de documentos) generalmente esos datos complejos se han guardado en sistemas de archivos del disco duro o estructuras de datos especializadas, en vez de un SGBD.

Para dar soporte a esas aplicaciones, los SGBD deben admitir tipos de datos complejos. Los conceptos orientados a objetos han influido poderosamente en los esfuerzos por mejorar el soporte de las bases de datos, a los datos complejos y han llevado al desarrollo de sistemas de bases de datos orientadas a objetos.

Debilidades del modelo relacional

- ✓ Pobre representación de entidades del mundo real
- ✓ Dificultad de cambiar esquemas sin afectar las aplicaciones
- ✓ Sobrecarga semántica
- ✓ Número fijo de atributos y todos atómicos
- ✓ Operaciones limitadas
- ✓ Dificultad para manejar queries recursivas

STONEBREAKER's VIEW

Stonebreaker propuso una Matriz de cuatro-cuadrantes en el libro titulado "Objeto-relacional DBMS: la siguiente gran ola"

Esta es una clasificación de las aplicaciones de bases de datos y sistemas.

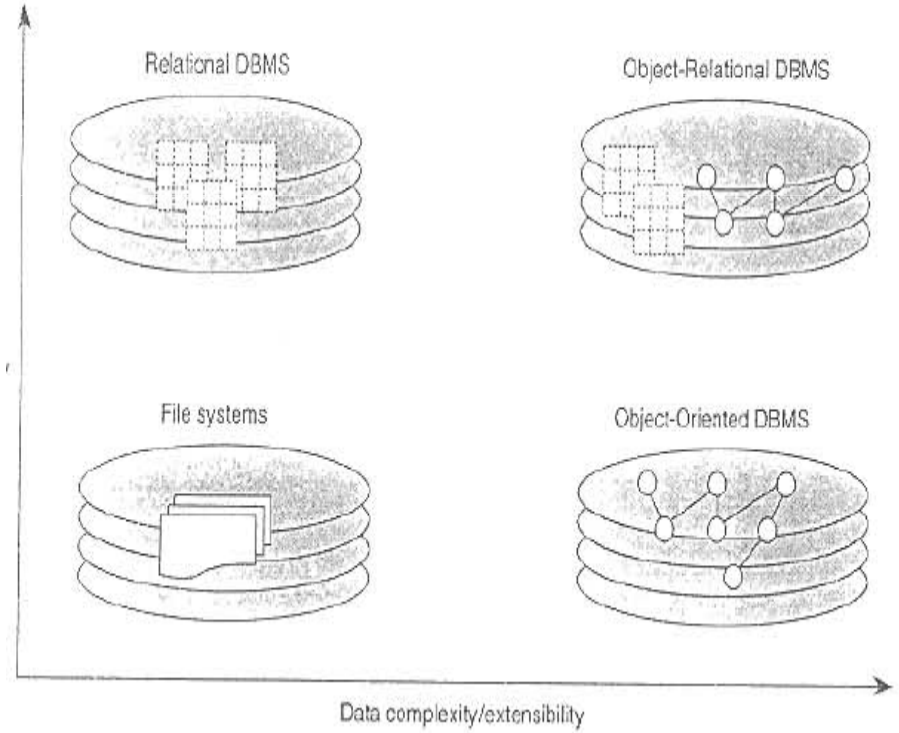
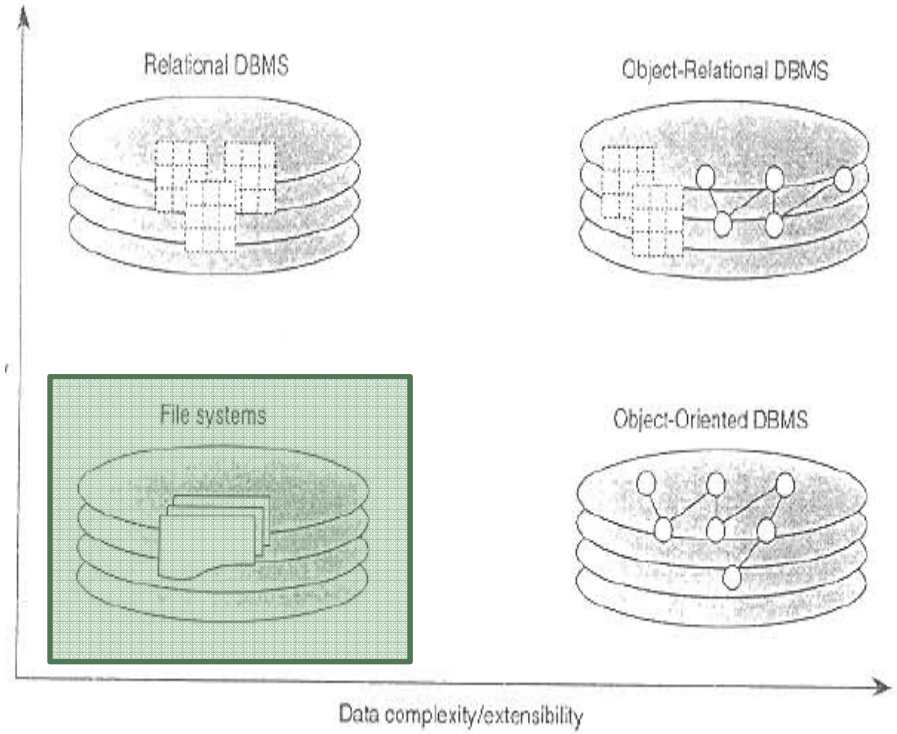


Imagen tomada de: [Conolly, Begs], Database System, A practical Approach To Design, Implementation, And Management

CUADRANTE INFERIOR IZQUIERDO

Aplicaciones que procesan datos simples y no requieren capacidad de consulta, por ejemplo, procesadores de texto (word, emacs)

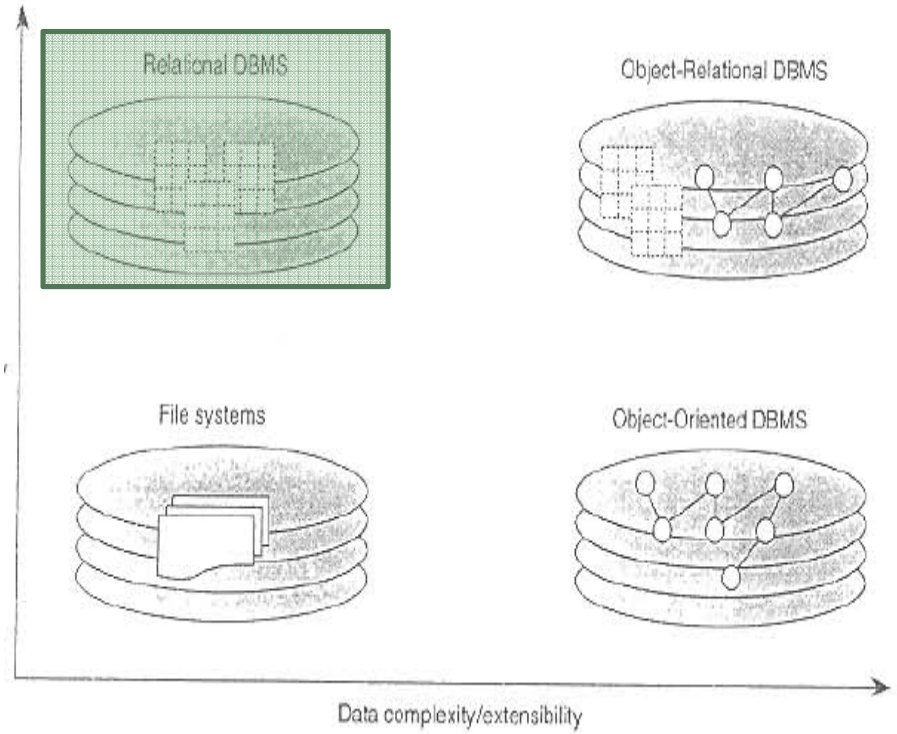
- ✓ La información tiene poca estructura interna.
- ✓ Las actualizaciones de documentos son relativamente infrecuentes.
- ✓ Los documentos son de tamaño modesto.
- ✓ Las consultas son simples cadenas o patrones de búsqueda.



CUADRANTE SUPERIOR IZQUIERDO

Aplicaciones que procesan datos simples y requieren capacidad de consulta compleja, por ejemplo, una aplicación empresarial típica requiere RDBMS.

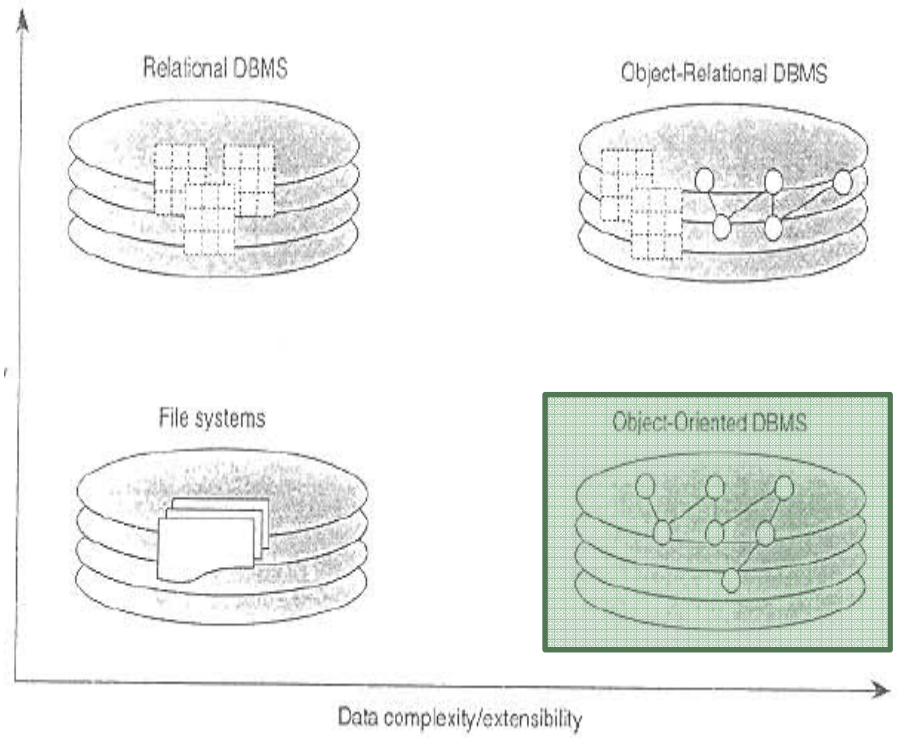
- ✓ La información tiene estructura sencilla y fija.
- ✓ La recopilación de información puede ser grande.
- ✓ El almacenamiento de información debe ser fiable.
- ✓ Las consultas son relativamente complejas.
- ✓ Las actualizaciones son frecuentes y la seguridad es vital.



CUADRANTE INFERIOR DERECHO

Aplicaciones que procesan datos complejos y no requieren capacidad de consulta ej. CAD application.

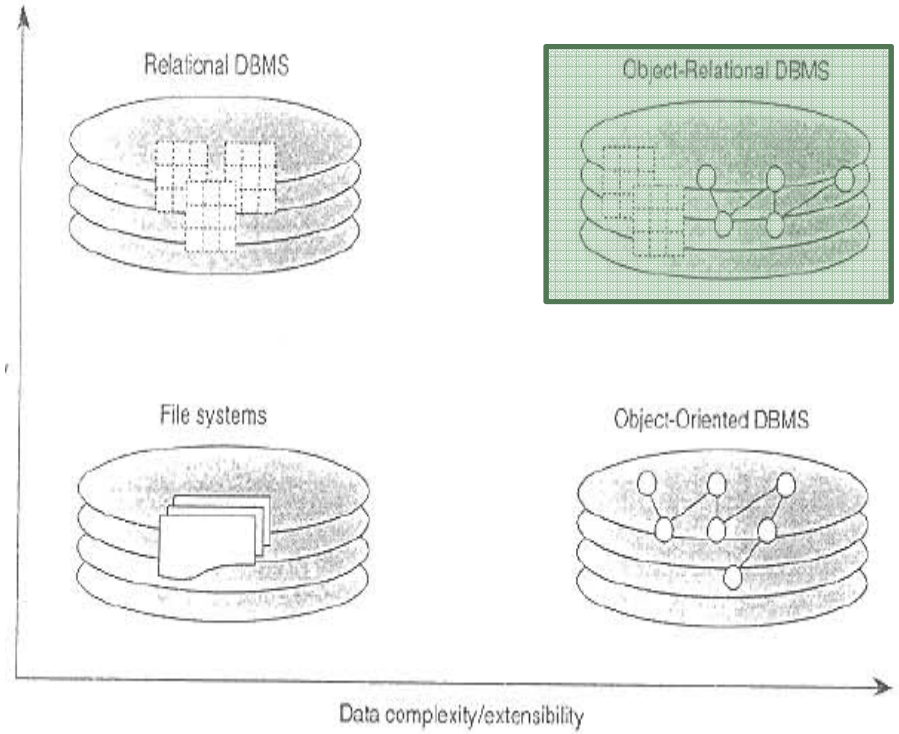
- ✓ La información tiene estructura compleja.
- ✓ Los análisis son complejos.
- ✓ La información es moderara en cantidad.
- ✓ Actualizaciones periódicas



CUADRANTE SUPERIOR DERECHO

Las aplicaciones que procesan datos complejos y requieren capacidad de consulta compleja, por ejemplo, un archivo de datos de imagen requiere ORDBMS.

- ✓ La información tiene estructura compleja.
- ✓ La información puede incluir tipos especiales (imágenes, información espacial)
- ✓ Los análisis son complejos.
- ✓ La información es amplia en cantidad.
- ✓ Las consultas son importantes
- ✓ Actualizaciones son periódicas



CARACTERÍSTICAS DE LOS SGBOR

✓ **Nuevos Tipos de Datos** que permiten gestionar aplicaciones más complejas con una gran riqueza de dominios (imagen, voz, sueldo, etc.)

✓ **Nuevas operaciones** que permiten gestionar el comportamiento de los Tipos de Datos

✓ **Mayor capacidad expresiva** para los conceptos y asociaciones complejas

✓**Reusabilidad**, propio de la Orientación a Objetos. Se pueden compartir diversas bibliotecas de clases ya existentes

✓**Integración de lenguajes**: relacional, de objetos, XML, etc. En un solo Lenguaje

✓**Nuevas Consultas** con mayor capacidad consultiva (consultas anidadas, recursivas, almacenadas, pre-fabricadas), etc.

TIPOS DE GRAN TAMAÑO

SQL: 99 incluye un tipo de datos nuevo denominado objeto de gran tamaño o LOB, con dos variantes denominadas BLOB (objeto grande binario) y CLOB (objeto de gran tamaño de caracteres), estos tipos de gran tamaño poseen las siguientes características.

✓ Gran capacidad de almacenamiento (hasta GigaBytes).

✓Útiles para el almacenamiento de imágenes, sonido, texto formateado y otras necesidades multimedia de las aplicaciones actuales.

✓Se mantienen en la base de datos, no en ficheros externos.

✓El tamaño se puede indicar en la definición.

EJEMPLO CREACIÓN DE UNA TABLA CON OBJETOS DE GRAN TAMAÑO

```
Create table ‘peliculas’
(
   id_pelicula int(10) not null,
   nom_pelicula varchar(25) not null,
   sipnosis longtext,
   demo_pelicula blob,
   primary key(id_pelicula)
)
```

Ejemplo hecho en MySql, una tabla películas con objetos de gran tamaño (Autor: Nelson López)

USER DEFINED TYPES (UDT)

Los tipos definidos por el usuario están divididos en dos subcategorías: tipos distintos y tipos estructurados.

Tipos distintos

Es el mas simple de ambos, se basa en el renombrado de tipos existentes:

- ✓Comparten la misma representación
- ✓Tienen distinto comportamiento
- ✓El tipo distinto no es comparable con el tipo fuente

Operaciones definidas sobre los tipos distintos:

- ✓Operadores de comparación
- ✓Métodos y funciones
- ✓No existe herencia entre tipos distintos (FINAL)

EJEMPLO DEL TIPO DISTINTO

```
CREATE TYPE tipoSala
      AS CHAR(10) FINAL;

CREATE TYPE tipoMetros
      AS INTEGER FINAL;

CREATE TYPE tipoMetrosCuad
      AS INTEGER FINAL;

CREATE TABLE Sala (
      IdSala       tipoSala,
      LongSala     tipoMetros,
      AnchoSala    tipoMetros,
      AreaSala     tipoMetroscuad,
      PerimSala    tipoMetros));
```

```
UPDATE Sala
SET AreaSala=LongSala;
```

Error

```
UPDATE Sala
SET AnchoSala=LongSala;
```

Correcto

Fuente de la imagen: *www.kybele.etsii.urjc.es/.../**BD/2008-2009**/.../%5BBD-**2008-2009**%5D**Tema3-ModeloOR**.pdf*

TIPOS ESTRUCTURADOS

Se pueden usar como:

- ✓**Tipos de columnas:**
  - Permiten modelar atributos de las entidades, como atributos compuestos, sin existencia propia. Ej: dirección (calle, ciudad, provincia)
  - Permiten mejorar el soporte para objetos multimedia como texto, imagen, vídeo, series de tiempo, puntos, líneas, etc.

- ✓**Tipos de fila:**
  - Permiten modelar entidades con relaciones y comportamiento Ej: empleado, departamento, alumno...
  - Permiten mejorar el soporte para objetos de negocio, con relaciones y comportamiento

Tipos Estructurados

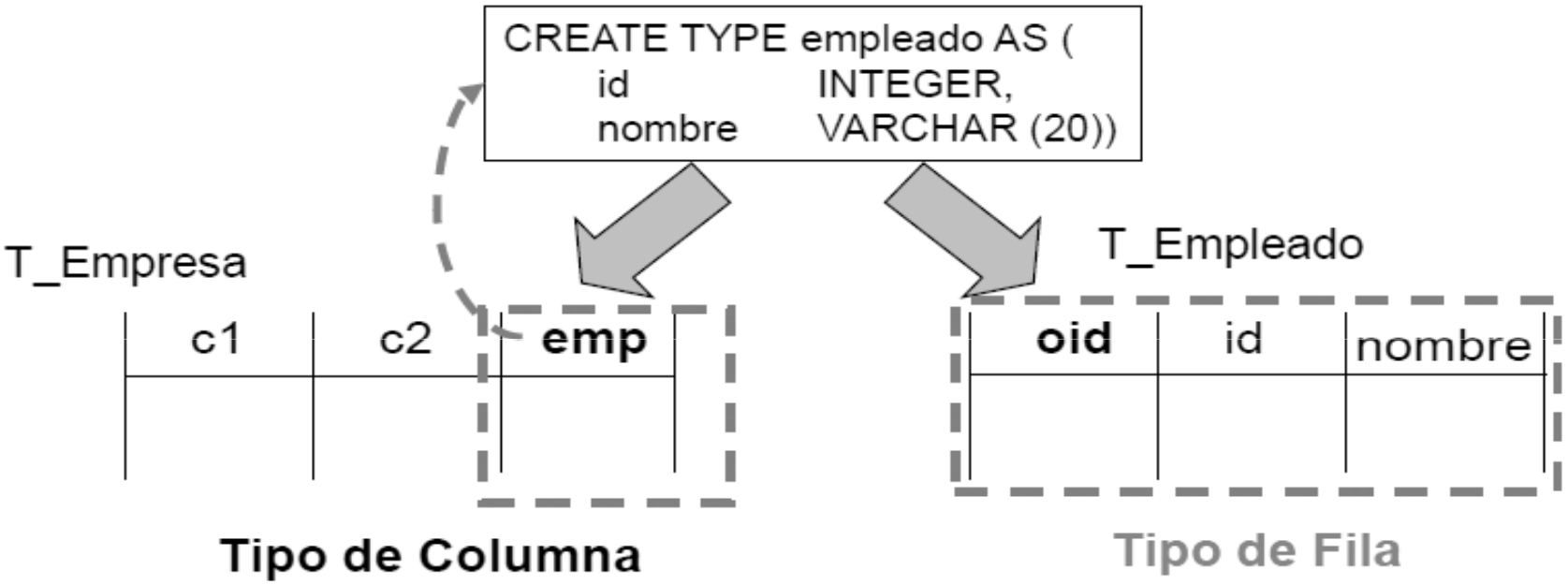


Imagen tomada de: *www.kybele.etsii.urjc.es/.../**BD**/**2008**-**2009**/.../%5BBD-**2008**-**2009**%5D**Tema3**-**ModeloOR**.pdf*

CONSTRUCTORES DE TIPOS

✓Permiten definir tipos nuevos con estructuras internas

✓Se denominan tipos estructurados.

✓Conduce más allá del modelo relacional, ya que el valor de estos campos ya no tiene que ser necesariamente atómico:

Principalmente son de dos tipos:

✓ ROW(n1, t1, ….,Nn,tn). Tipo que representa una fila, o tupla, de n campos con los campos n1,….,Nn de los tipos t1,….tn, respectivamente.

✓ ARRAY [i] tipo que representa un ARRAY de hasta i elementos del tipo base

Ejemplo:

```
CREATE TABLE libro (
    título VARCHAR (200),
    id_libro INTEGER,
    autores VARCHAR (15) ARRAY [20],
    resumen CLOB (32K),
    texto_libro CLOB (20M),
    película BLOB (2G));

INSERT INTO libro (título, id_libro, autores)
VALUES ("A guide to the SQL Standard", 15, ['Date', 'Darwen'])

SELECT id, autores[1] AS nombre
FROM libro
```

Ejemplo tomando de: [Costilla], |Curso de Bases de Datos (BSDT) 2005-06 [En linea] http://sinbad.dit.upm.es/docencia/grado/curso0607/BDOR%20Segunda%20Parte%20transparencias-Modelo%20de%20Datos%20O-R%20Nov%202006.pdf

```
CREATE TABLE empleado (
        id_emp INTEGER,
        nombre ROW (
                        nombre_de_pila VARCHAR(30),
                        apellido VARCHAR(30) ),
        dirección ROW (
                        calle VARCHAR(50),
                        ciudad VARCHAR(30),
                        provincia CHAR(2) ),
        sueldo REAL );

SELECT E.nombre.apellido
FROM empleado E
```

Ejemplo tomando de: [Costilla], |Curso de Bases de Datos (BSDT) 2005-06 [En linea] http://sinbad.dit.upm.es/docencia/grado/curso0607/BDOR%20Segunda%20Parte%20transparencias-Modelo%20de%20Datos%20O-R%20Nov%202006.pdf

FUNCIONES Y PROCEDIMIENTOS

✓SQL:1999 permite: la definición de funciones, procedimientos y métodos.

✓Se pueden definir mediante el componente procedimental de SQL:1999 o mediante un lenguaje de programación como Java, C o C++.

Algunos sistemas de bases de datos soportan sus propios lenguajes procedimentales, tales como:

✓PL/SQL en Oracle y TransactSQL en SQL Server de Microsoft.

Éstos incorporan una parte procedimental parecida a SQL:1999, pero hay diferencias en la sintaxis y la semántica.

SQL:1999 SOPORTA INSTRUCCIONES WHILE, FOR Y REPEAT

```
declare n integer default 0;
while n < 10 do
        set n = n +1;
end while
repeat
        set n = n-1;
until n = 0
end repeat
```

```
declare n integer default 0;
for r as
        select saldo from cuenta
        where nombre-sucursal = 'Navacerrada'
do
        set n = n + r.saldo;
end for
```

Imagen tomada de: [Silberschatz, Korth, Sudarshan], FUNDAMENTOS DE BASES DE DATOS 4 Edition

SQL:1999 INCLUYE INSTRUCCIONES IF-THEN-ELSE CON ESTA SINTAXIS

```
if r.saldo < 1000
        then set p = p + r.saldo
elseif r.saldo < 5000
        then set m = m + r.saldo
else set g = g + r.saldo
end if
```

Imagen tomada de: [Silberschatz, Korth, Sudarshan], FUNDAMENTOS DE BASES DE DATOS 4 Edition

PROCEDIMIENTOS ALMACENADOS

Las nuevas capacidades de procedimientos almacenados (funciones y triggers) Mediante un uso racional, dan lugar a aplicaciones más robustas, fiables y eficientes.

Sin embargo, el uso inapropiado, o procedimientos almacenados mal construidos, puede llevar a aplicaciones que funcionan mal, a un mantenimiento difícil o con poca confiabilidad.

PRINCIPALES TIPOS DE PROCEDIMIENTOS ALMACENADOS

- **Stored Procedure**
- **Funciones**
- **Triggers**

Procedimientos almacenados

✓Un procedimiento es un programa dentro de la base de datos que ejecuta una acción o conjunto de acciones especificas.

✓Un procedimiento tiene un nombre, un conjunto de parámetros (opcional) y un bloque de código.
Ejemplo:

```
CREATE PROCEDURE spu_addCliente (
      @nombre varchar(100),
      @apellido1 varchar(100),
      @apellido2 varchar(100),
      @nifCif varchar(20),
      @fxNaciento datetime )
AS
      INSERT INTO CLIENTES
      (nombre, apellido1, apellido2, nifcif, fxnacimiento) VALUES
      (@nombre, @apellido1, @apellido2, @nifCif, @fxNaciento)
```

Fuente ejemplo: http://www.devjoker.com/contenidos/Tutorial-de-Transact-SQL/228/Introducci%C3%B3n-a-Transact-SQL.aspx

Funciones

Son similares a los procedimientos almacenados, pero el resultado de la ejecución, devuelve un único valor.

Ejemplo:

```
CREATE FUNCTION dbo.Factorial ( @iNumber int )
RETURNS INT
AS
BEGIN
        DECLARE @i int
        IF @iNumber <= 1
           SET @i = 1
        ELSE
            SET @i = @iNumber * dbo.Factorial( @iNumber - 1 )
        RETURN (@i)
END
```

Fuente ejemplo: http://www.devjoker.com/contenidos/Tutorial-de-Transact-SQL/228/Introducci%C3%B3n-a-Transact-SQL.aspx

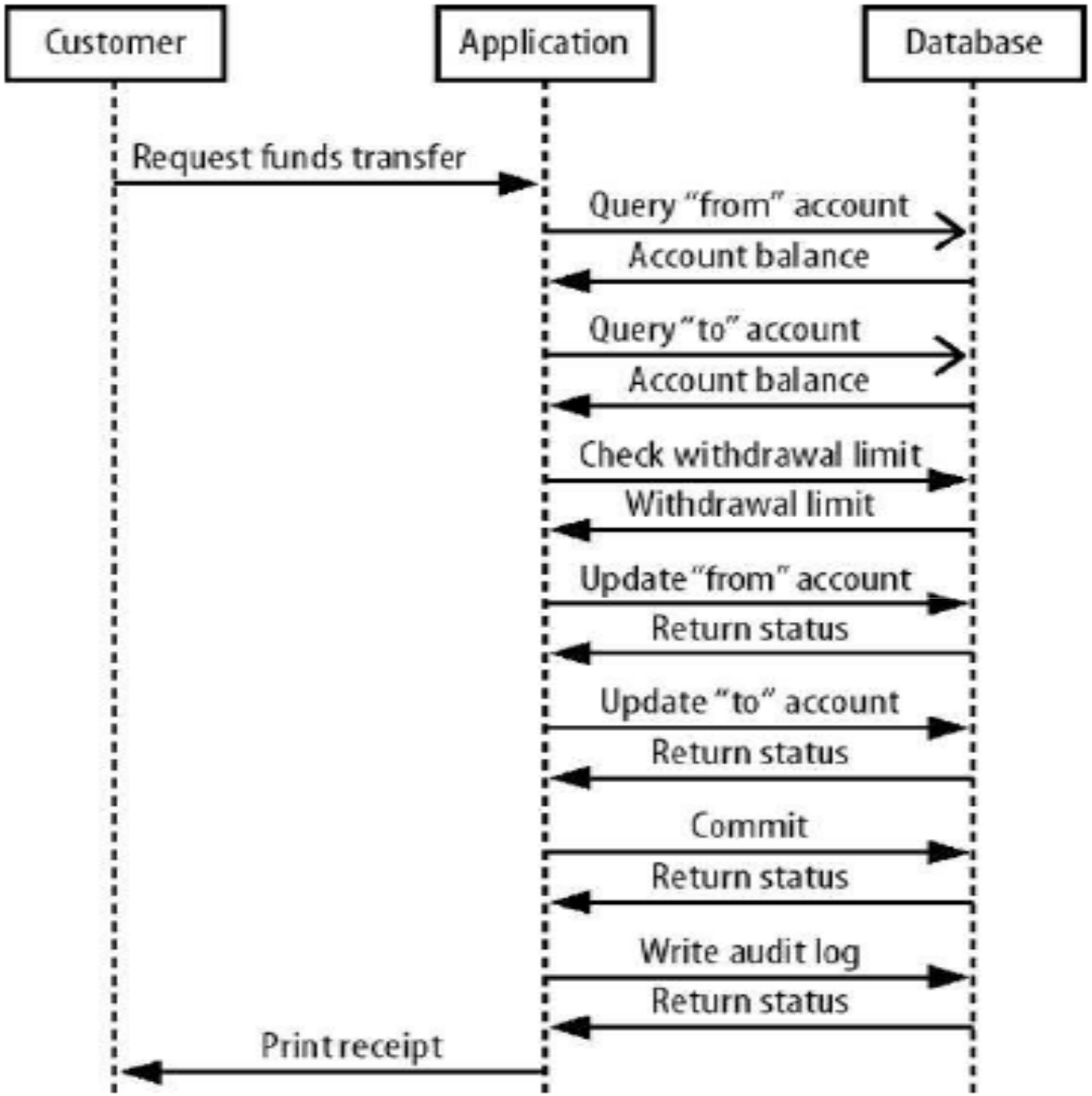
Triggers

Es una clase especial de procedimiento almacenado que se ejecuta automáticamente cuando se produce un evento en el servidor de bases de datos.

Ejemplo:

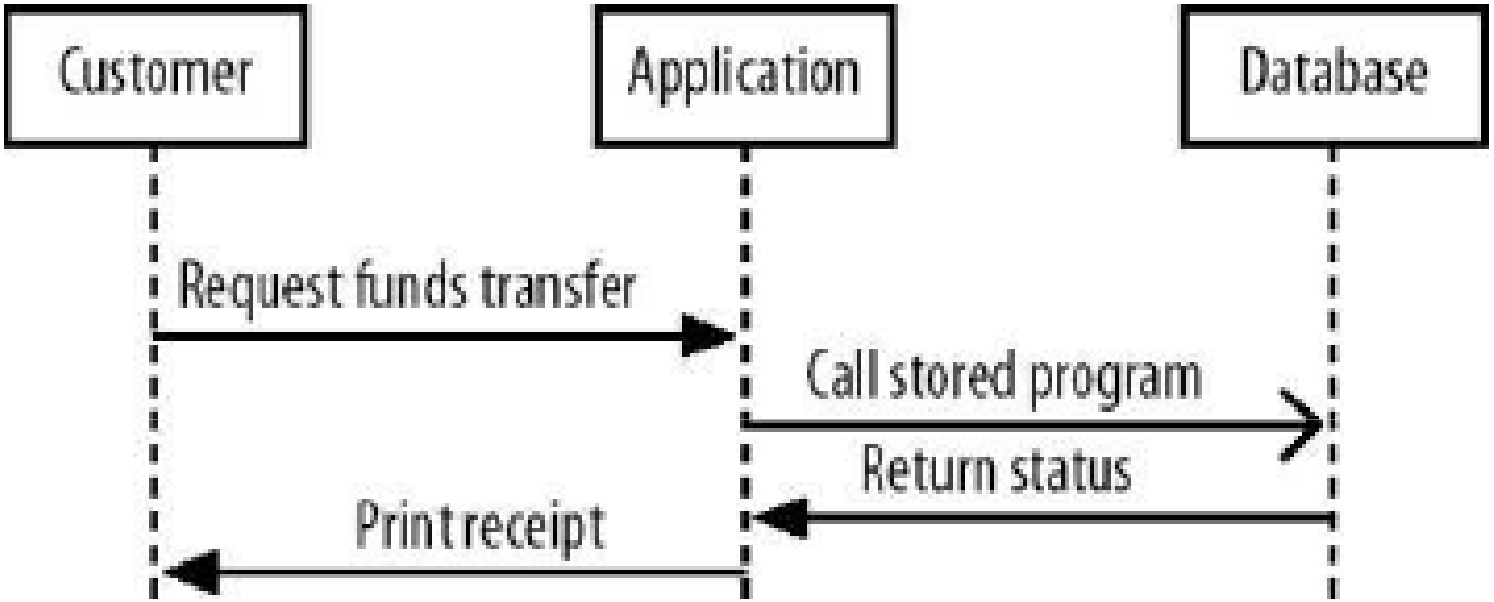
```
CREATE TRIGGER TR_SEGURIDAD
ON DATABASE FOR DROP_TABLE, ALTER_TABLE
AS
BEGIN
    RAISERROR ('No está permitido borrar ni modificar tablas !' , 16, 1)
    ROLLBACK TRANSACTION
END
```

Fuente ejemplo: http://www.devjoker.com/contenidos/Tutorial-de-Transact-SQL/228/Introducci%C3%B3n-a-Transact-SQL.aspx



Network round trips without a stored program

Imagen tomada de: [Feuerstein] [Harrison] MySQL Stored Procedure Programming, 2006



Network round trips involving a stored program

Imagen tomada de: [Feuerstein] [Harrison] MySQL Stored Procedure Programming, 2006

¿VENTAJAS?

- Sólo consumen el tiempo de la obtención de datos
- Permiten implementar mejores mecanismos de autorización de acceso
- Compartir funcionalidades entre diferentes aplicaciones
- Reduce el trafico en la red
- Seguridad
- A menudo se puede mejorar la portabilidad de código de la aplicación al mover lógica a programas almacenados.

tomado de: [Feuerstein] [Harrison] MySQL Stored Procedure Programming, 2006

¿DESVENTAJAS?

- El uso PA puede conducir a la fragmentación de la aplicación, lo que dificulta realizar un seguimiento de los errores de diseño o errores de aplicación.
- Migración
- Pueden ser más difíciles depurar
- Más ORM por ejemplo, J2EE CMP y hibernación no pueden explotar sin problemas PA.
- Diferencias de sintaxis

tomado de: [Feuerstein] [Harrison] MySQL Stored Procedure Programming, 2006

Según Daniel Seara

- **Ejecutar cualquier sentencia SQL significa**
  - Control de sintaxis
  - Control de validez de los objetos implicados
  - Compilación
  - Cálculo del plan de ejecución (Query Plan)
  - Ejecución y obtención de resultados

- **Ejecutar cualquier procedimiento almacenado significa**
  - Ejecución y obtención de resultados
  - (el resto se realiza al guardar el PA en la base de datos)

BIBLIOGRAFÍA

- **Database Systems** - A Practical Approach to Design, Implementation and Management, by Thomas Connolly and Carolyn Begg.
- **Database Management Systems** by **Raghu Ramakrishnan, Johannes Gehrke.**
- **Fundamentos de Bases de datos,** by Abraham Silberschatz.
- **MySQL Stored Procedure Programming** By Steven Feuerstein, Guy Harrison.
- **Beginning SQL Server 2008 for Developers**, by Robin Dewson
- **Estandar SQL99, Curso de bases de datos**, By Carmen costilla
- ***ORDBMS – Introduction,*** *S*pring 2008 – By Farnoush Banaei-Kashani
- **Carmen Costilla, Curso de Bases de Datos (BSDT) 2005-06** [En linea] http://sinbad.dit.upm.es/docencia/grado/curso0607/BDOR%20Segunda%20Parte%20transparencias-Modelo%20de%20Datos%20O-R%20Nov%202006.pdf
- **ModeloOR,***www.kybele.etsii.urjc.es/.../**BD/2008-2009**/.../%5BBD-**2008-2009**%5D**Tema3-ModeloOR**.pdf*
- ***Daniel Seara, Acceso Avanzado a Datos, DCE 5 Estrellas, Microsoft***