Multibases de Datos

El alumno explicará los diferentes tipos y clasificaciones de base de datos distribuidas.

Multibases de Datos

- Un Sistema Multibases de Datos soporta operaciones en múltiples sistemas pre-existentes de bases de datos.

- "Cada sistema de base de datos componente es administrado por un Sistema Manejador de Bases de Datos (DBMS) componente. Un sistema de base de datos componente puede estar centralizado o distribuido y puede residir en la misma computadora o en varias computadoras conectadas por un sistema de comunicación" Sheth et.al (1990).

Multibases de datos

- Un Sistema Multibases de datos permite operaciones de acceso y modificación de datos entre diversos sistemas de bases de datos de manera concurrente y transaccional.

- La diferencia entre las bases de datos distribuidas y los sistemas Multibases de datos es la autonomía local. No se puede forzar a los esquemas locales pre-existentes a que se ajusten a un esquema estándar.

Clasificación por autonomía de SBD

- La autonomía local en los sistemas de múltiples bases de datos es que cada sistema manejador de base de datos retiene el control completo sobre los datos locales y su procesamiento. En este caso, ni los cambios globales, como la adición de otros componentes tienen efecto en los sistemas de base de datos locales.

- Si existe completa autonomía local, entonces los sistemas Multibases de datos proporcionan un servicio de solo lectura, porque los sistemas manejadores de base de datos no colaboran entre sí para procesar transacciones distribuidas.

Clasificación por autonomía de SBD

- En el caso de poca o nula autonomía, existe una sola imagen de la base de datos completa disponible a todos los usuarios. Todos los sitios trabajan juntos para completar una transacción.

- Se puede realizar la propagación de actualizaciones a lo largo de las bases de datos componentes con datos semánticamente equivalentes a través de un control a nivel distribuido y local por manejo de transacciones y control de concurrencia.

Clasificación por autonomía de SBD

- En el caso de semi autonomía, los sistemas de base de datos pueden operar independientemente, pero han decidido participar en una federación para compartir sus datos locales con otros sistemas de base de datos.

- El tipo de autonomía de los sistemas multibase de bases de datos determinará si pueden formar un Sistema Federado cuando éstos se integran para compartir información.

-

Sistemas Federados

- Los Sistemas de Base de Datos Federados se forman a partir de un conjunto de sistemas cooperativos pero autónomos de base de datos que se unen para poder compartir e intercambiar información.

- Una federación se refiere a un conjunto de base de datos que constituyen una base de datos federada.

- Los Sistemas Federados también son conocidos como Sistemas Multibases de datos autónomos y heterogéneos.

Sistemas no Federados

- Los Sistemas de Bases de Datos no Federados resultan de la integración de sistemas gestores de base de datos que no son autónomos.
- Todos los sistemas de base de datos están completamente integrados a un solo esquema global.
- Los sistemas Multibases de datos unificados lógicamente se parecen a un Sistema de Base de Datos Distribuido.
-

Clasificación por variedad de sistemas componentes

- **Multibases de datos homogéneas**
- **Multibases de datos heterogéneas**

Multibases de datos homogéneas

- Sistema multibase de datos homogéneo es aquel en donde el sistema manejador de base de datos de todas las bases de datos componentes es el mismo.

Multibases de datos heterogéneas

- Por tanto, cuando los manejadores de las bases de datos componentes son diferentes se trata de un Sistema multibase de datos heterogéneo.

Multibases de datos heterogéneas

- Existen dos casos de heterogeneidad a nivel Manejador de base de datos:

- a) El Sistema Multibase de datos está compuesto por diferentes Manejadores de base de Datos, pero pertenecen al mismo modelo de datos. Por ejemplo, Sybase y Oracle ofrecen Manejadores de Bases de Datos Relacionales.

- b) El Sistema Multibases de datos está compuesto por diferentes Manejadores de Base de Datos pertenecientes a diferentes modelos de datos, como el relacional y el objeto-relacional con Sybase Adaptive Server Enterprise y Versant Object Database por ejemplo.
-

Tipos de heterogeneidad

- Los tipos de heterogeneidad a nivel sistemas de base de datos pueden dividirse en:
  - Heterogeneidad sintáctica (lenguaje usado para representación de los objetos)
  - Heterogeneidad estructural (diferencias en tipos, estructuras de los elementos)
  - Heterogeneidad de Modelo/Representación (relacional, OO, etc.)
  - Heterogeneidad Semántica (diferencias en la semántica de los datos).

Heterogeneidad sintáctica


```
CREATE TABLE ACTOR (

    NOMBRE              VARCHAR2(100) NOT NULL,
    ACTOR_ID            NUMBER(10, 0) NOT NULL,
    APELLIDO_PATERNO    VARCHAR2(100) NOT NULL,
    APELLIDO_MATERNO    VARCHAR2(100) NOT NULL,
    TIPO_ACTOR          CHAR(2) NOT NULL,
    RFC                 VARCHAR2(100) NOT NULL,
    FECHA_NACIMIENTO    DATE NOT NULL,
    FECHA_INGRESO       DATE NOT NULL,
    CONSTRAINT PK1 PRIMARY KEY (ACTOR_ID)

)
```

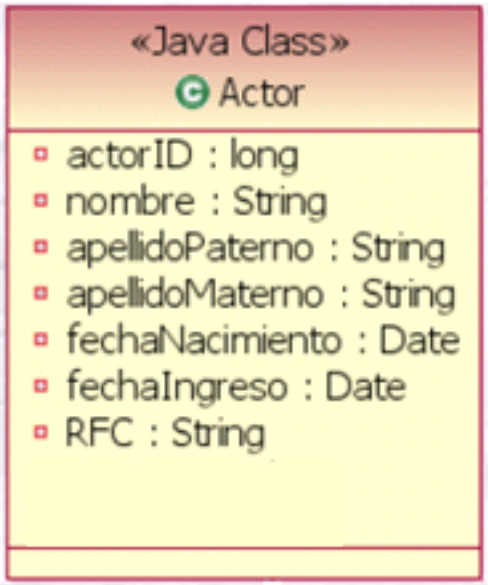
```
import java.util.Date;


public class Actor {
        private String nombre;
        private long actorID;
        private String apellidoPaterno;
        private String apellidoMaterno;
        private char tipoActor;
        private String RFC;
        private Date fechaNacimiento;
        private Date fechaIngreso;

        public long getActorID() {
            return actorID;
        }
        public void setActorID(long actorID) {
            this.actorID=actorID;
        }
        public String getNombre() {
            return nombre;
        }
        public void setNombre( String nombre) {
            this.nombre=nombre;
```

Heterogeneidad estructural

Modelo Relacional

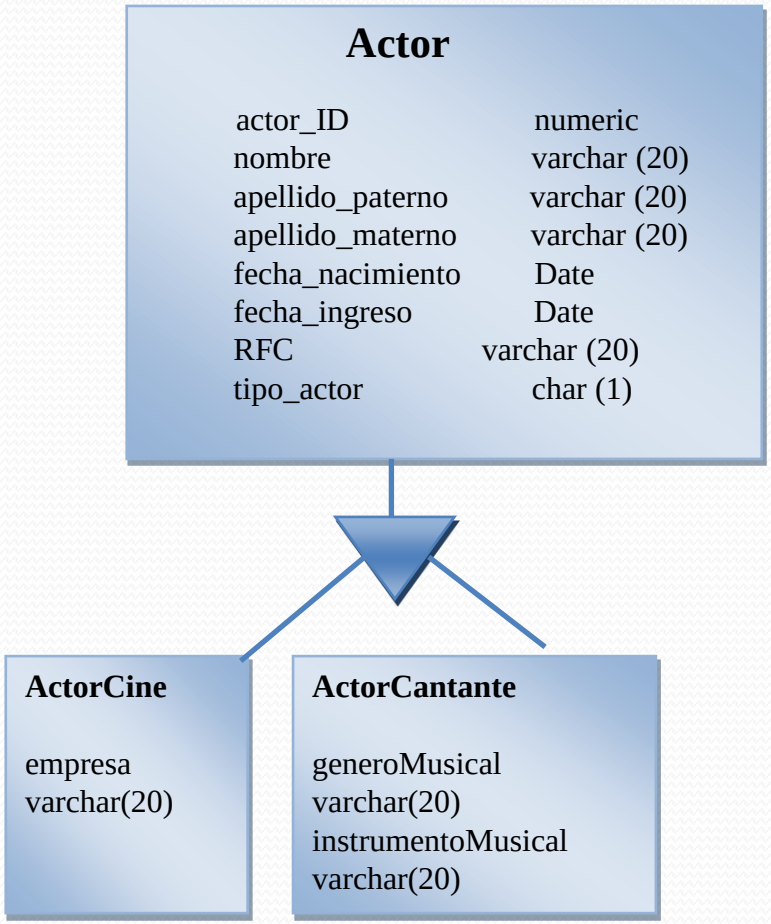
Actores						
actor_ID	**nombre**	**apellido_paterno**	**apellido_materno**	**fecha_nacimiento**	**fecha_ingreso**	**RFC**
PK						

Modelo Orientado a Objetos

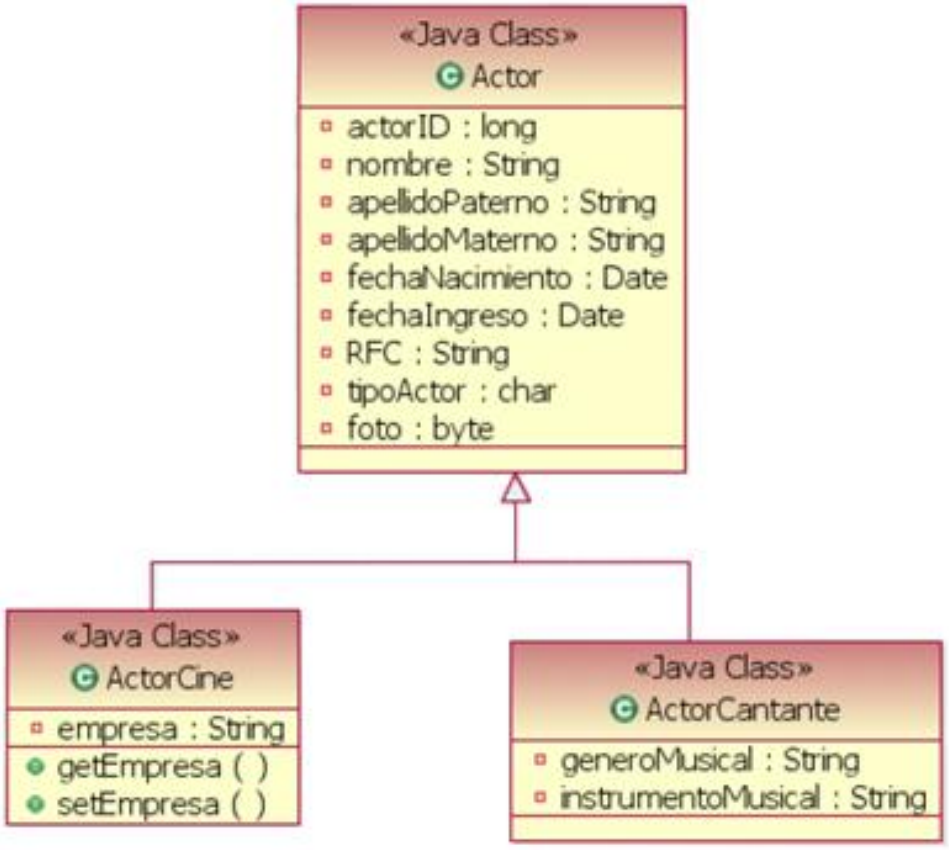


Heterogeneidad de Modelo /Representación

Modelo Relacional



Modelo Orientado a Objetos



Heterogeneidad Semántica

	Profesores								
Profesores_ID	Nombre	Apellido_Paterno	Apellido_Materno	RFC	Fecha_Nacimiento	Grado	Tipo	Fecha_Ingreso	Ingresos
PK									
0001	Juan	Robles	Pérez	ROPJ19021966	19/02/1966	Maestro	1	20-01-2010	$15 000
0002	Sara	Mendoza	Hernández	MEHS03051986	08/05/1986	Licenciado	2	16-12-2009	$10 000

DIFERENTE FORMA DE REPRESENTACIÓN

SINÓNIMOS

DATOS FALTANTES

DIFERENTE FORMA DE REPRESENTACIÓN

PRECISIÓN

Docentes									
Docentes_ID	Nombre	Apellido_Paterno	Apellido_Materno	CURP	Sexo	Fecha_Nacimiento	Grado	Tipo	Fecha_Ingreso	Sueldo
PK										
1	Ricardo	Rico	Martínez	RIMR09101970	M	09-Noviembre-1970	Doctor	A	20-01-2000	30,000.00
2	Carolina	Álvarez	Prado	ALPC20061980	F	20-Junio-1988	----------	B	10-08-1900	10,000.00

Heterogeneidades

- En los componentes del Modelo de datos:
  - Estructura, restricciones, lenguajes de consulta (sintaxis, esquema)
    - Ej. Heterogeneidad en la misma estructura relacional
      - Tamaño de la Relación (diferente grado o número de atributos)
    - Ej. Heterogeneidad en diferentes esquemas (orientado a objetos, relacional, etc.) Diferentes niveles de generalización (ESTUDIANTE, ESTUDIANTE_POSGRADO)
  - Diferentes niveles de agregación (VAGON, BUQUE)
- Implementación propia del manejador:
  - Manejo de transacciones, control de concurrencia, protocolos de respaldo y recuperación, diferencias de comportamiento (políticas de inserción/borrado) etc.
- Diseño lógico de base de datos

Heterogeneidad semántica

- Ocurre con la discrepancia en el significado, interpretación o intensión de uso de los datos. A continuación se mencionan algunos ejemplos de heterogeneidades:

- Conflictos de nombrado e.g. Homónimos, sinónimos (se usan diferentes nombres para representar el mismo concepto).

- Conflictos de dominio o conflictos de representación de datos e.g. se usan diferentes valores para representar el mismo concepto.
- Escala de Datos: Diferentes Unidades –conversión fija (kilogramos/libras)
- Escala de Datos: Diferentes Unidades-conversión variante con el tiempo
  - (£, $ tipos de cambio)

Heterogeneidad semántica

- Conflictos en la Representación de datos:
  (precio con/sin impuesto), (teléfono con /sin código de área),
  Diferente cadena/mismo significado: femenino/masculino o F/M)
  Diferentes Formatos de Cadena (DD/MM/YY o DD-MM-YYYY)

- Conflictos en la Precisión de datos: (peso 2 kg./2.25 kg.)
  Se usan los mismos valores de datos de dominios de diferentes cardinalidades para los mismos datos.
- Conflictos en lo que significa el dato
  - (teléfono trabajo o de casa)
- Contexto en el cual el dato es capturado
  - (mediciones que dependen de ciertas condiciones)
- Diferentes en el nivel de abstracción
  - (calificaciones pueden ser representadas de manera diferente {5..10}/{A..D}

Heterogeneidad por semántica

- Conflictos de metadatos e.g. los mismos conceptos son representados a nivel de esquema y a nivel de instancia
- Conflictos de datos e.g. atributos faltantes
- Conflictos de esquema e.g. conflicto entre tablas, incluyendo conflictos de datos, de nombrado, etc.
- Conflictos de manejo de información faltante, conflictos de valores por omisión.

- Para poder crear un esquema federado, se debe resolver heterogeneidad antes de integrar los esquemas de base de datos componentes.

Schema matching & Schema mapping

- El problema de la heterogeneidad ha sido un obstáculo muy importante para la implementación de Sistemas Federados de Bases de Datos.
- Este problema lo tratan de resolver la correspondencia de esquemas y el mapeo de esquemas.
- A veces schema matching y schema mapping se usan de indistintamente, sin embargo existe diferencia.

Correspondencia de esquemas (Schema matching)

- El schema matching trata de hacer “corresponder” esquemas que son semánticamente equivalentes, pero difieren en el nombrado de sus partes (modelo de datos).
- Schema matching trata de identificar pares de objetos que estan semanticamente relacionados. En muchos casos la correspondencia de esquemas es un paso dentro de el mapeo de esquemas.

Mapeo de Esquemas (schema mapping)

- El mapeo de esquemas trata de encontrar una transformación de datos que, dada una instancia de un esquema, producirá una instancia que este conforme al esquema destino (modelo canónico), mientras se preserva la información apropiada del contenido de la fuente.

- El mapeo por pares entre n atributos resulta en reglas de mapeo que indican la equivalencia.
- Una forma de implementar esto último es proporcionar un esquema global que comprenda las partes relevantes de todos esquemas participantes y proporcione los mapeos en la forma de vistas de base de datos.
- Para esto se tienen dos soluciones principales:
- Global as View (GaV): el esquema global esta definido en términos de los esquemas subyacentes.
- Local as View (LaV): Los esquemas locales están definidos en términos del esquema global

Autonomía

- Ya se había comentado que la diferencia fundamental entre los MDBS y los FDBS es el concepto de autonomía.
- Primero se explican los aspectos de autonomía para las bases de datos componentes y como se abarca cuando un sistema de BDs componente participa en la federación, existen cuatro tipos de autonomía:
  - Autonomía de diseño
  - Autonomía de comunicación
  - Autonomía de ejecución
  - Autonomía de asociación

Autonomía

- Autonomía de Diseño, la cual se refiere a la habilidad de escoger el diseño sin importar los datos, el lenguaje de consulta, la conceptualización o la funcionalidad de la implementación del sistema.

- Las Heterogeneidades en un FDBS son principalmente debidos a la autonomía en el diseño.

Autonomía

- Autonomía de Comunicación se refiere a la operación general de los DBMSs para comunicarse con otro DBMS o no.

- Autonomía de ejecución permite a un DBMS componente controlar las operaciones requeridas a través de operaciones locales y externas.

Autonomía

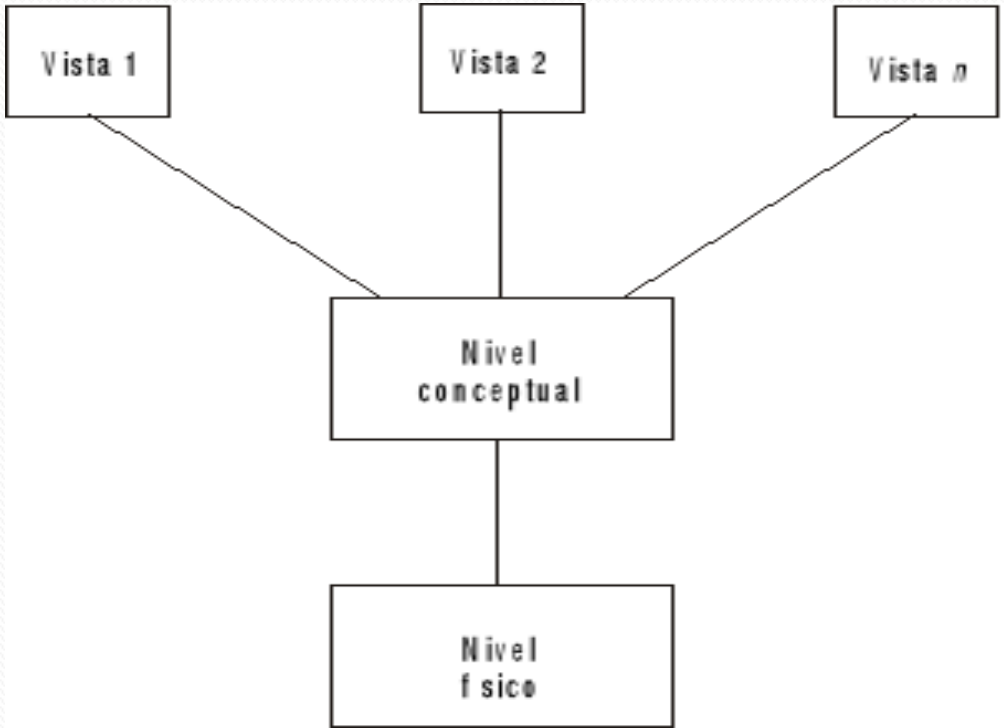
- Autonomía de asociación permite a un DBS componente desasociarse de la federación, lo que significa que un FDBS puede operar independientemente de cualquier DBS.

Control de Concurrencia

- Los requerimientos de heterogeneidad y la autonomía representan un reto para el logro de la serializabilidad (secuencialidad) global dentro del control de concurrencia en un FDBS.

Arquitectura 3 niveles para Sistemas de Base de Datos

- (Repaso)
- El Comité ANSI-SPARC (American National Standard Institute - Standards Planning and Requirements Committee) propuso una arquitectura de tres niveles para los sistemas de bases de datos.



Arquitectura 3 niveles para Sistemas de Base de Datos

- El objetivo de la arquitectura de tres niveles es el de separar los programas de aplicación de la base de datos física. (esquemas externo, conceptual e interno)
- Es decir, la independencia lógica radica en modificar el esquema conceptual independientemente de los esquemas externos ni los programas de aplicación que no se refieran a los cambios al esquema.
- Por otro lado, la independencia física es la capacidad de alterar el esquema interno sin necesidad de reflejar estos cambios en el esquema conceptual ni en el esquema externo.

Arquitectura 5 niveles para Sistemas de Base de Datos Federadas

- Sin embargo, la arquitectura de base de datos de tres capas no soporta las dimensiones de distribución, heterogeneidad y autonomía de los sistemas federados.

- Por tanto en Sheth et.al (1990) definieron la Arquitectura de 5 niveles que se menciona a continuación:

Arquitectura 5 niveles para Sistemas de Base de Datos Federadas

- (esquemas local, componente, exportación, federado y externo)
- 1.- El Esquema Local está expresado en el modelo de datos nativo de cada sistema de base de datos componente.

- 2.- El Esquema Componente se deriva de la traducción de los esquemas locales a un modelo de datos llamado Modelo de Datos Canónico Común (el modelo de datos que se usará como estándar).

- 3.- El Esquema de Exportación es un subconjunto de un esquema componente que está disponible a la federación. Este puede incluir información de control de acceso para uso de algún usuario de la federación.

Arquitectura 5 niveles para Sistemas de Base de Datos Federadas

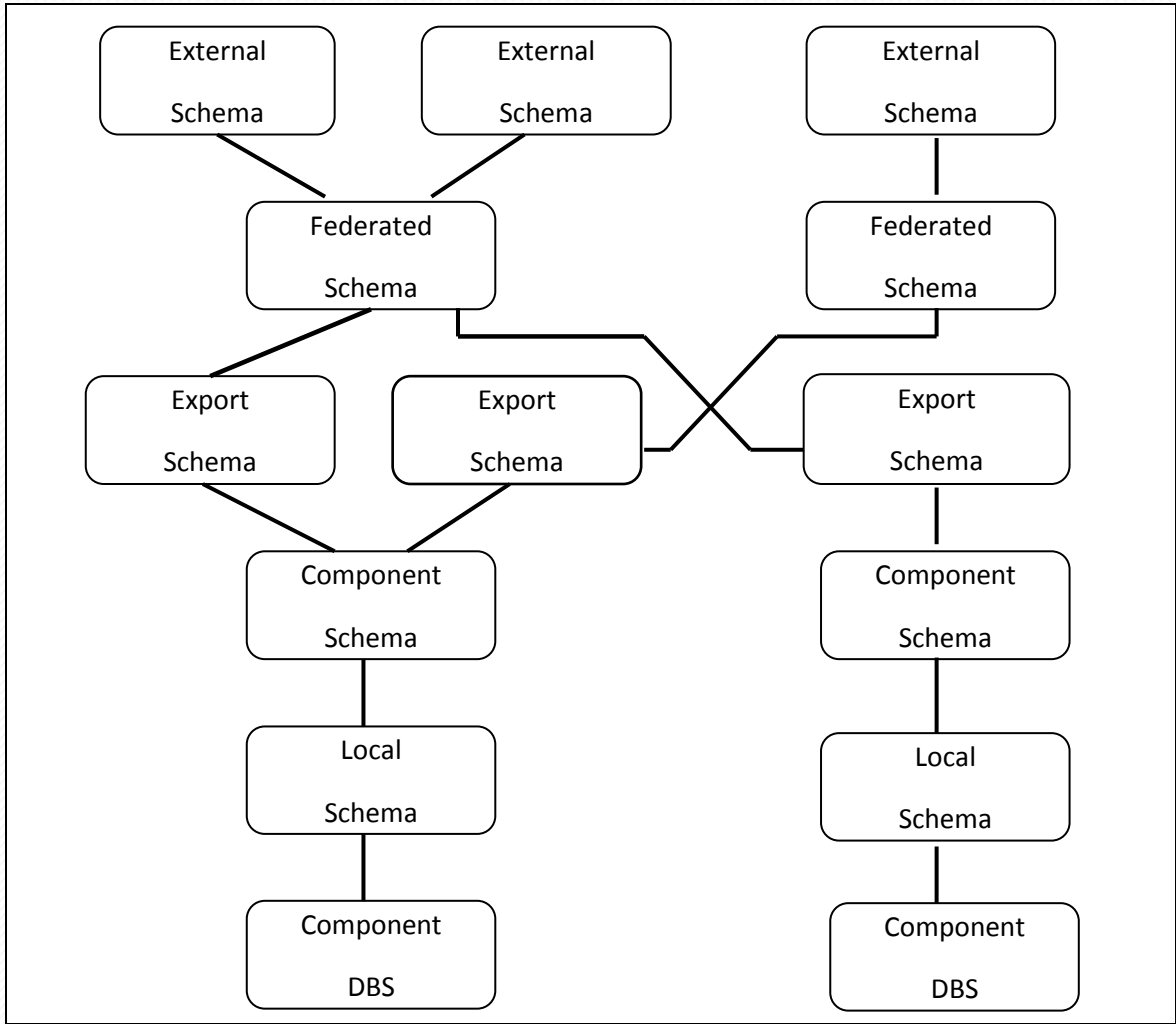
- El Esquema Federado se forma de la integración de varios esquemas de exportación o modelos de datos, conocido también como Esquema Global o Esquema de Importación.

- El Esquema Externo es un subconjunto de información de un esquema federado que es relevante a una clase o aplicación de usuario.

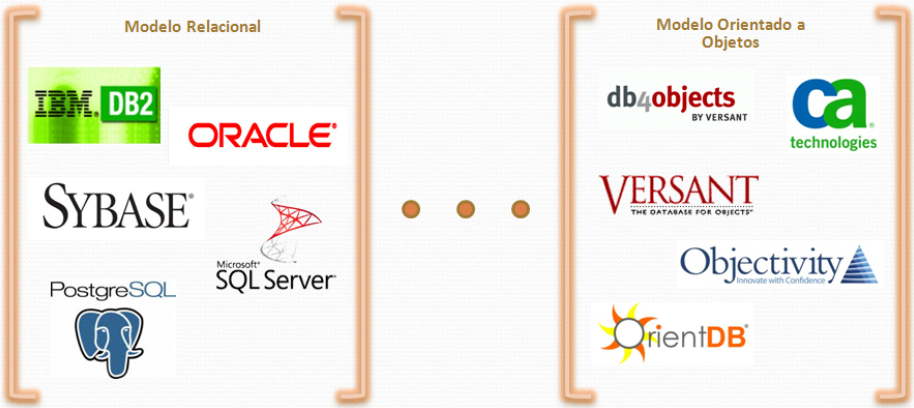
Arquitectura 5 niveles para Sistemas de Base de Datos Federadas

- Al desarrollarse el esquema federado que integra múltiples esquemas de exportación, se proporciona transparencia de localización, replicación, y distribución.

- Las transparencias se manejan a través de los mapeos entre los esquemas federados y los esquemas de exportación.

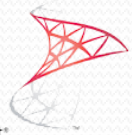


COMPONENT DBS



COMPONENT DBS

Modelo Relacional

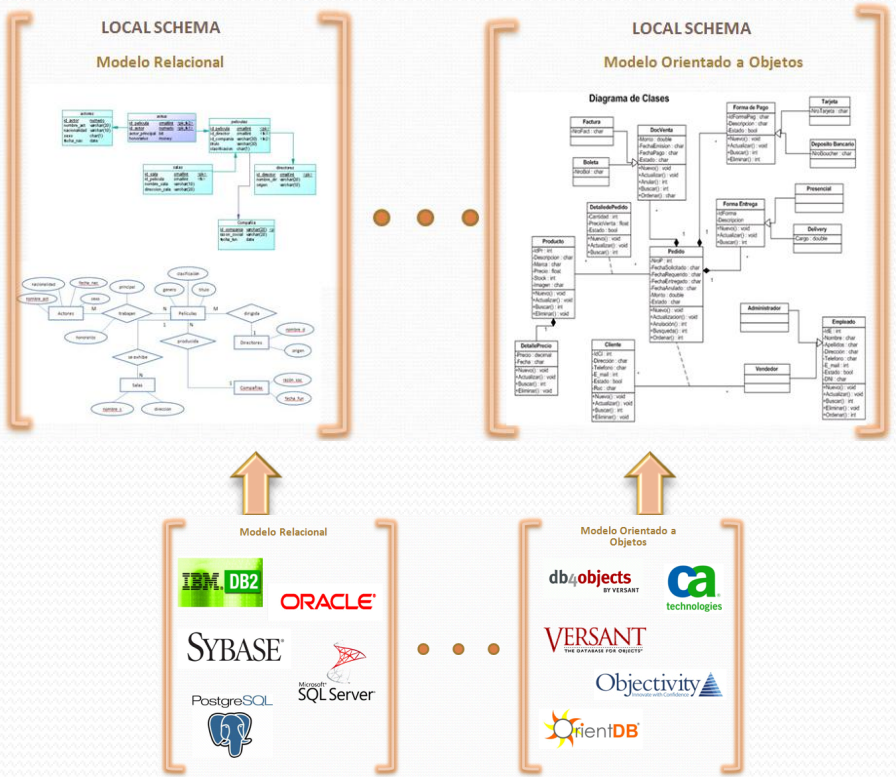


Microsoft SQL Server

Modelo Orientado a Objetos

ESQUEMA LOCAL : Modelo de Datos Nativo

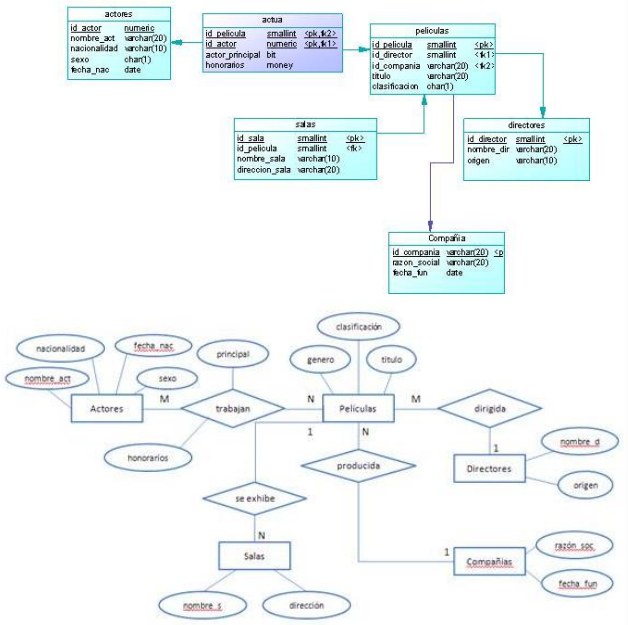
COMPONENT DBS



ESQUEMA LOCAL : Modelo de Datos Nativo

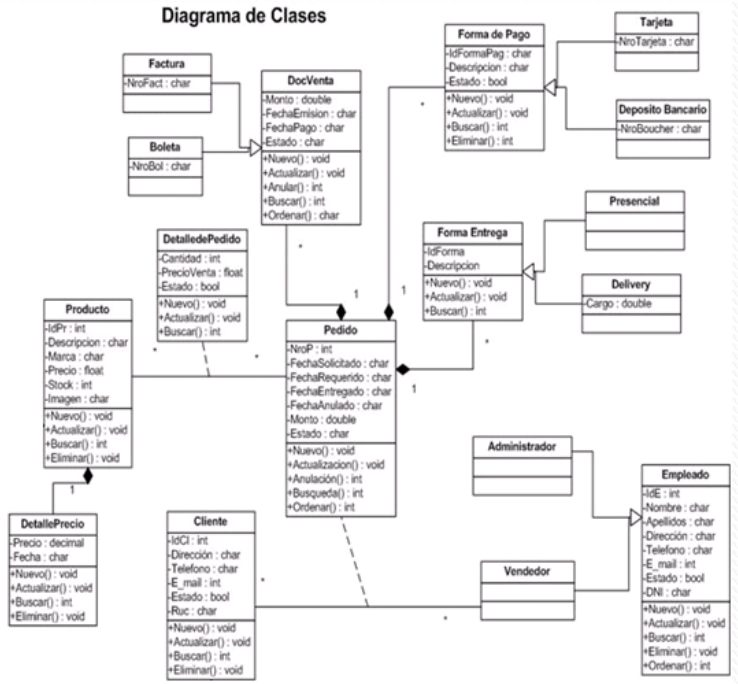
LOCAL SCHEMA

Modelo Relacional



LOCAL SCHEMA

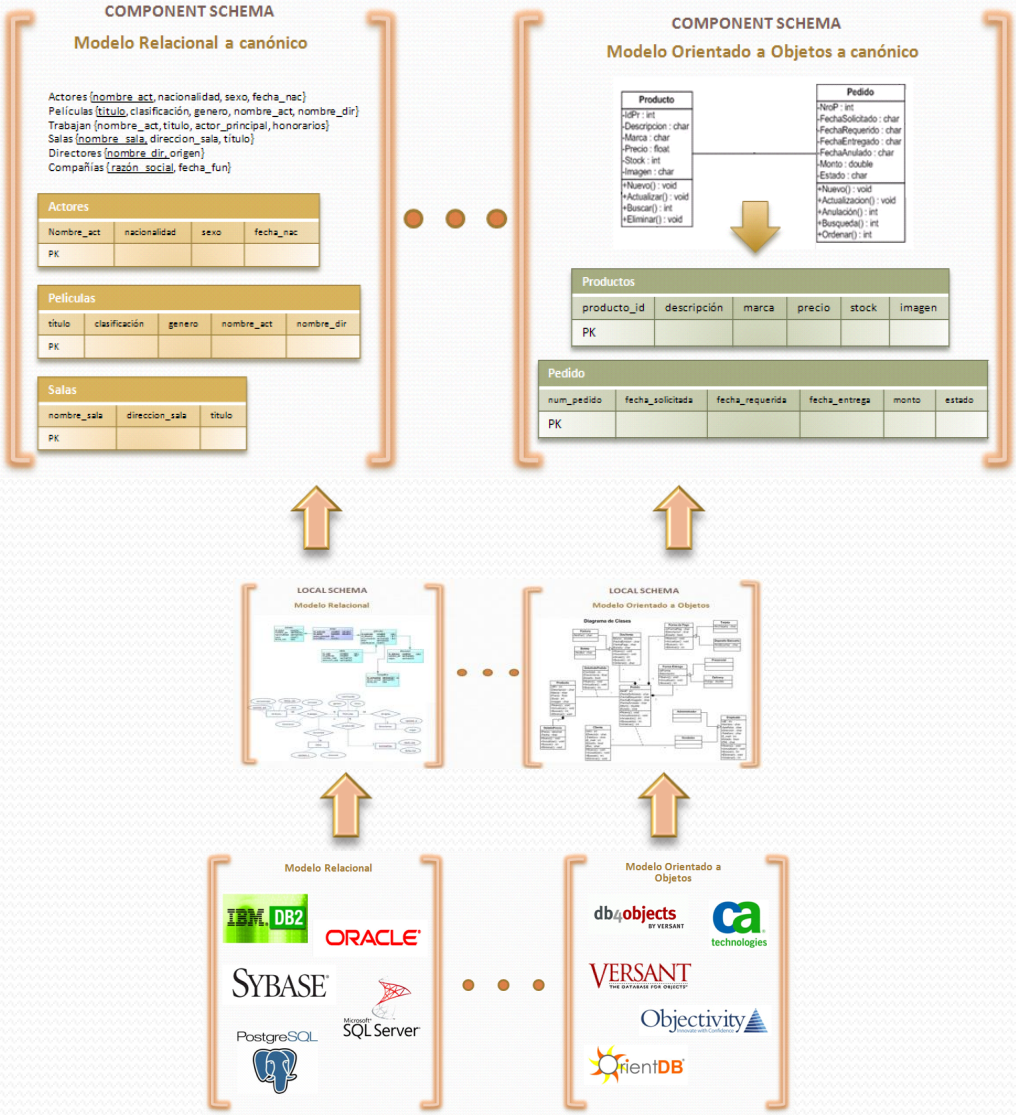
Modelo Orientado a Objetos



ESQUEMA COMPONENTE : Modelo de Datos Canónico

ESQUEMA LOCAL : Modelo de Datos Nativo

COMPONENT DBS



ESQUEMA COMPONENTE : Modelo de Datos Canónico

COMPONENT SCHEMA

Modelo Relacional a canónico

Actores {nombre_act, nacionalidad, sexo, fecha_nac}
Películas {titulo, clasificación, genero, nombre_act, nombre_dir}
Trabajan {nombre_act, titulo, actor_principal, honorarios}
Salas {nombre_sala, direccion_sala, título}
Directores {nombre_dir, origen}
Compañías {razón_social, fecha_fun}

Actores

Nombre_act	nacionalidad	sexo	fecha_nac
PK			

Peliculas

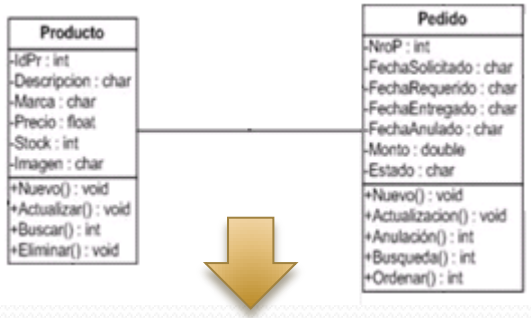
título	clasificación	genero	nombre_act	nombre_dir
PK				

Salas

nombre_sala	direccion_sala	titulo
PK		

COMPONENT SCHEMA

Modelo Orientado a Objetos a canónico



Productos

producto_id	descripción	marca	precio	stock	imagen
PK					

Pedido

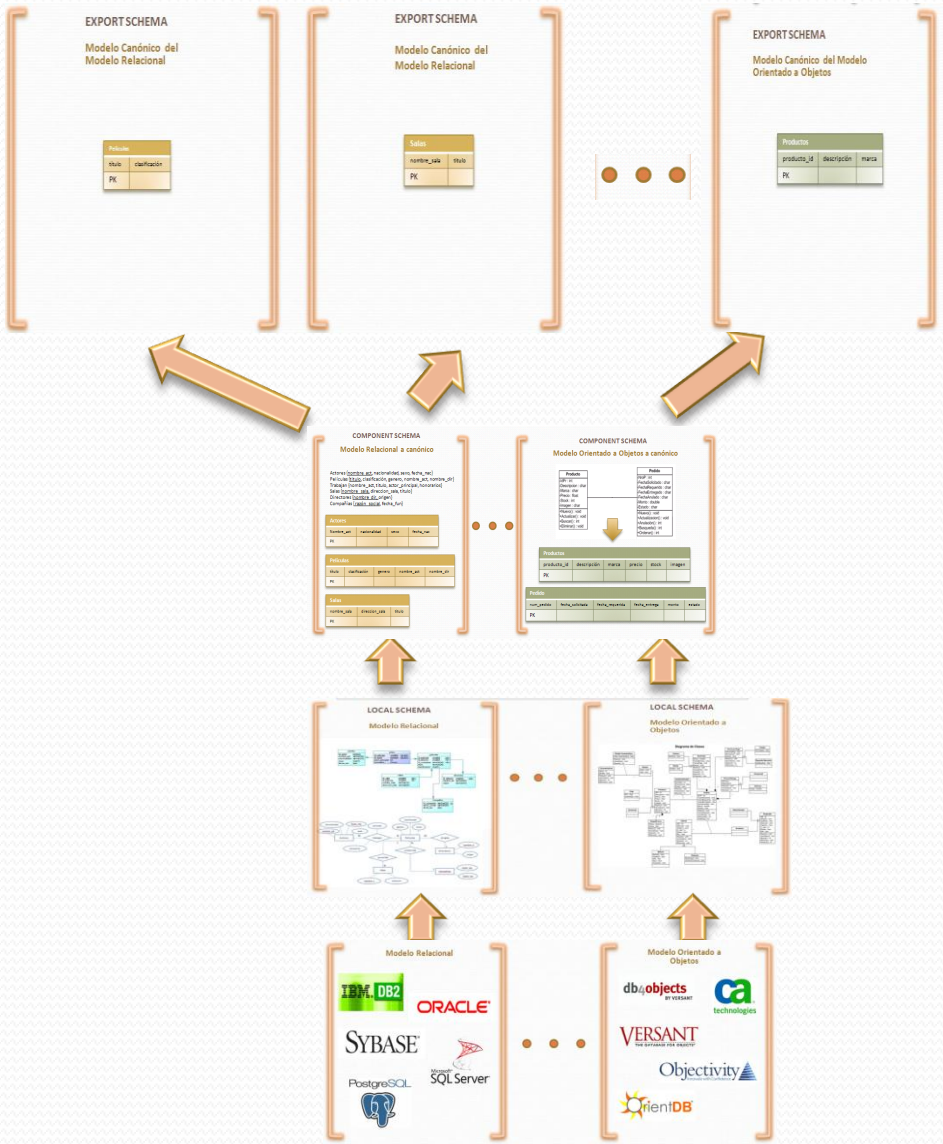
num_pedido	fecha_solicitada	fecha_requerida	fecha_entrega	monto	estado
PK					

ESQUEMA DE EXPORTACIÓN :
Subconjunto de un esquema componente

ESQUEMA COMPONENTE :
Modelo de Datos Canónico

ESQUEMA LOCAL :
Modelo de Datos Nativo

COMPONENT DBS



ESQUEMA DE EXPORTACION : Subconjunto de un esquema componente (MR)

EXPORT SCHEMA

Modelo Canónico del Modelo Relacional

EXPORT SCHEMA

Modelo Canónico del Modelo Relacional

ESQUEMA DE EXPORTACION : Subconjunto de un esquema componente (MOO)

EXPORT SCHEMA

Modelo Canónico del Modelo Orientado a Objetos

Productos			Productos			ductos		
producto_id	descripción	ma	producto_id	descripción	marca	io	stock	imagen
PK			PK					

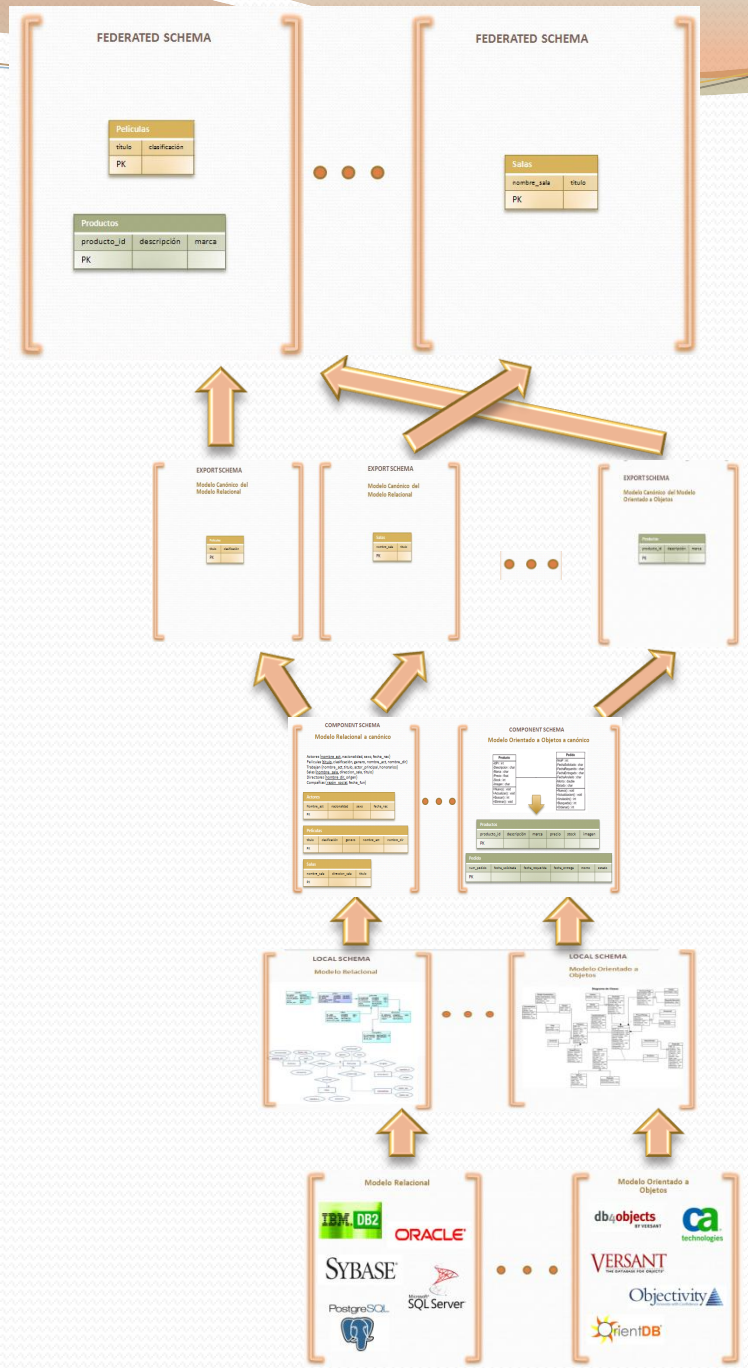
ESQUEMA DE FEDERADO :
Esquema Global o Esquema de Importación

ESQUEMA DE EXPORTACIÓN :
Subconjunto de un esquema componente

ESQUEMA COMPONENTE :
Modelo de Datos Nativo

ESQUEMA LOCAL :
Modelo de Datos Nativo

COMPONENT DBS



ESQUEMA FEDERADO: Esquema Global o Esquema de Importación

FEDERATED SCHEMA

Peliculas	
título	clasificación
PK	

Productos		
producto_id	descripción	marca
PK		

FEDERATED SCHEMA

Salas	
nombre_sala	titulo
PK	

ESQUEMA EXTERNO:
Subconjunto de información de un esquema federado

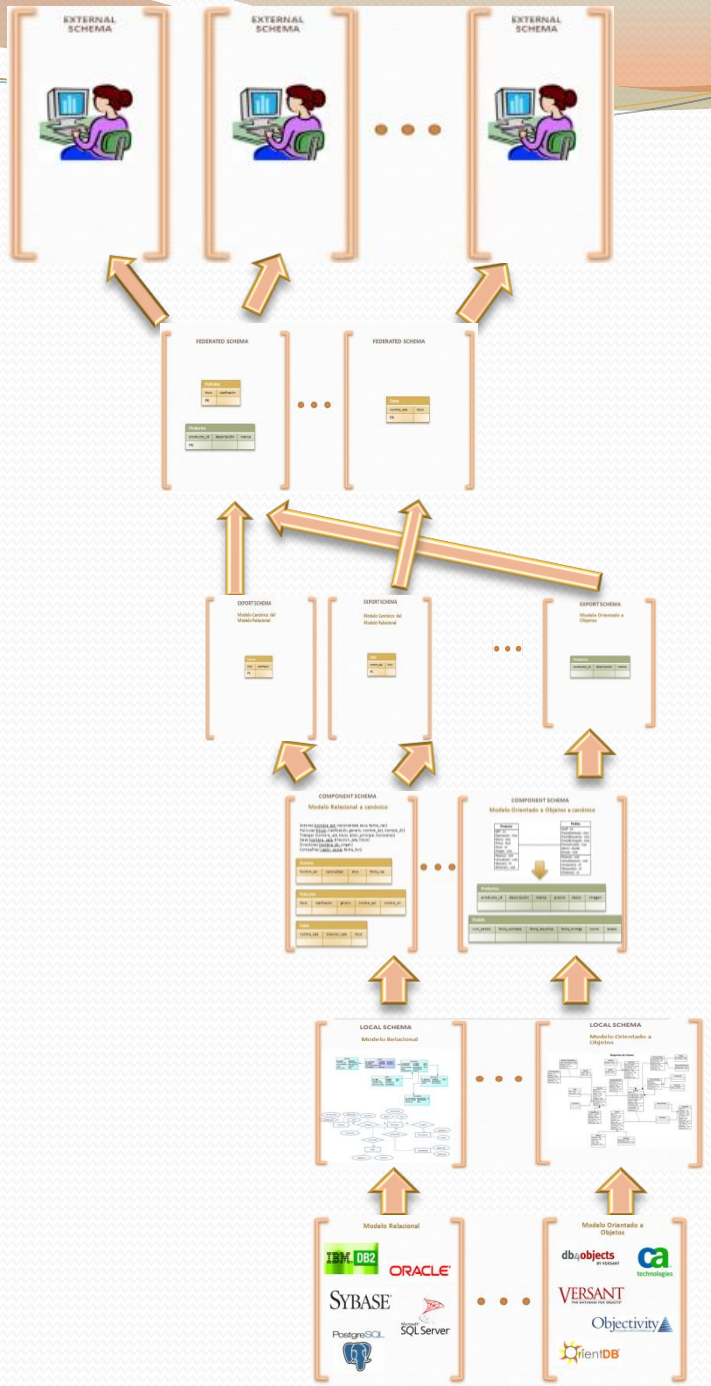
ESQUEMA DE FEDERADO :
Esquema Global o Esquema de Importación

ESQUEMA DE EXPORTACIÓN :
Subconjunto de un esquema componente

ESQUEMA COMPONENTE :
Modelo de Datos Nativo

ESQUEMA LOCAL :
Modelo de Datos Nativo

COMPONENT DBS



ESQUEMA DE EXTERNO : Subconjunto de información de un esquema federado

EXTERNAL SCHEMA

EXTERNAL SCHEMA

EXTERNAL SCHEMA

Diversos requerimientos dentro de los Sistemas Federados

- A pesar de que esta arquitectura representa el estado del arte en integración de datos, existe una desventaja, de nombre “IT imposed look and feel”.

- Los modernos usuarios de datos demandan control acerca de la presentación de los datos, sus necesidades tienen conflictos con los intereses de la integración de datos (de un nivel de abstracción menor a uno mayor).

Clasificación de los Sistemas Federados según su administración

- Los sistemas de base de datos federados se dividen en ligeramente acoplados y fuertemente acoplados dependiendo de cómo se administre la federación.

Sistemas federados ligeramente acoplados

- Los sistemas ligeramente acoplados requieren que las bases de datos componentes construyan su propio esquema federado.

- Un usuario típicamente puede accesar otro sistema de base de datos componente usando un lenguaje multibase de datos, sin embargo esto elimina la transparencia de ubicación, porque obliga al usuario a tener conocimiento directo del esquema federado.

Sistemas federados ligeramente acoplados

- En los sistemas ligeramente acoplados, cada usuario de la federación es el administrador de su propio esquema federado.
- El usuario es responsable de entender la semántica y de resolver heterogeneidades semánticas. Por tanto, él puede especificar de manera precisa las relaciones y mapeos entre los objetos en los esquemas de exportación.

Sistemas federados ligeramente acoplados

- El usuario de la federación busca en el conjunto de esquemas de exportación cuál de ellas contiene los datos que quiere acceder, y define un esquema federado importando los objetos del esquema de exportación que desea.
- El usuario nombra y almacena el esquema federado bajo la su cuenta.
- Por tanto, no existe un proceso predeterminado para controlar la creación de un esquema federado, puede ser creado, administrado y eliminado en el momento que se requiera.

Sistemas federados ligeramente acoplados

- Se pueden soportar múltiples semánticas dado que diferentes usuarios mantienen diferentes mapeos de sus esquemas federados hacia los esquemas de exportación.

- Este tipo de sistemas son más convenientes para integrar un gran número de bases de datos autónomas de sólo lectura accesibles en redes de comunicación.

- Por otro lado, no son muy convenientes para operaciones de actualización porque pueden degradar la integridad de los datos, debido a las diversas interpretaciones semánticas y al proceso de definición de vista, dado que las transformaciones de actualización de vistas son frecuentemente no determinadas.

Sistemas federados fuertemente acoplados

- Los sistemas federados fuertemente acoplados consisten de sistemas componentes que usan procesos independientes para construir y publicar un esquema federado integrado.

Sistemas federados fuertemente acoplados

- En los Sistemas Federados Fuertemente acoplados existe un solo administrador lógico de base de datos para la federación, quien a través de el proceso de integración de esquemas obtendrá mayor conocimiento de las base de datos componentes.

- Los Esquemas de exportación son creados por negociación entre los administradores de la base de datos componentes y el administrador de la federación.

Sistemas federados fuertemente acoplados

- El administrador de la Federación puede leer los esquemas componentes para negociar su acceso, es más fácil de soportar actualizaciones donde los administradores definen cuidadosamente los mapeos.

Sistemas federados fuertemente acoplados

- Los esquemas externos son creados por negociación entre el usuario y el administrador de la federación, donde éste último es quien tiene la autoridad.

- Es muy importante, que se establezcan protocolos de negociación, reglas de negocio y/o restricciones para creación y modificación de esquemas federados.

- Un esquema federado dado que es producto de la integración de esquemas de base de datos, evoluciona de forma gradual y controlada.

Sistemas federados fuertemente acoplados

- En este tipo de sistemas federados pueden ocurrir múltiples interpretaciones semánticas con múltiples federaciones pero pueden ser resueltas al momento de la creación del esquema de la federación a través de la integración de esquemas.

- La terminología descrita anteriormente es principalmente académica/ investigación.

- Pero, ¿ qué hay en la industria???
- ¿ Cómo se llaman? Que diferencias existen?
- ¿ Qué arquitecturas se implementan mas fácilmente?
- ¿ Bajo qué condiciones se implementa una u otra?
- ¿Cuál es la diferencia entre integración y centralización?

Tarea

- 1.- Investigar que es Integration Information
- 2.- Investigar las arquitecturas, características, aplicaciones, de los siguientes términos:
  - Datawarehouse (Bodegas de datos)
  - Enterprise Aplication Integration (Sistemas de Integración de Aplicaciones empresariales)
  - Enterprise Information Integration (Sistemas de Integracion de información empresarial)

3.- Investigar que softwares existen en el mercado que ayudan a la implementación de este tipo de sistemas.