

## EVIDENCIAS DE CLASE

```
CREATE USER UNIVERSISDAD IDENTIFIED BY U
GRANT CONNECT, RESOURCE, UNLIMITED TABLESPACE TO UNIVERSIDAD;
```

```
CREATE TABLE UNIVERSIDADES (
    ID_U VARCHAR(5) PRIMARY KEY,
    NOM_U VARCHAR(20) NOT NULL
);
```

```
INSERT INTO UNIVERSIDADES (ID_U, NOM_U)
VALUES ('U01', 'UTA');
```

```
CREATE TABLE EXTENSIONES (
    ID_EXT VARCHAR(5) PRIMARY KEY,
    NOMBRE VARCHAR(20) NOT NULL,
    ID_U_PER REFERENCES UNIVERSIDADES(ID_U)
);
```

```
INSERT INTO EXTENSIONES (ID_EXT, NOMBRE, ID_U_PER)
VALUES ('E01', 'Ingahurco', 'U01');
```

```
INSERT INTO EXTENSIONES (ID_EXT, NOMBRE, ID_U_PER)
VALUES ('E02', 'Huachi', 'U01');
```

```
INSERT INTO EXTENSIONES (ID_EXT, NOMBRE, ID_U_PER)
VALUES ('E03', 'Querochaca', 'U01');
```

```
CREATE TABLE CARRERAS (
    ID_CAR VARCHAR(5) PRIMARY KEY,
    NOM_CAR VARCHAR(20) NOT NULL,
    CUPOS_DISP NUMBER,
    ID_EXT_PER REFERENCES EXTENSIONES(ID_EXT)
);
```

```
INSERT INTO CARRERAS (ID_CAR, NOM_CAR, CUPOS_DISP, ID_EXT_PER)
VALUES ('C01', 'Ingeniería Sistemas', 20, 'E01');
```

```
INSERT INTO CARRERAS (ID_CAR, NOM_CAR, CUPOS_DISP, ID_EXT_PER)
VALUES ('C02', 'Medicina', 30, 'E02');
```

```
INSERT INTO CARRERAS (ID_CAR, NOM_CAR, CUPOS_DISP, ID_EXT_PER)
VALUES ('C03', 'Veterinaria', 25
, 'E03');
```

```
SELECT NOM_CAR
FROM CARRERAS
WHERE ID_EXT_PER='E01';
```

```
SELECT NOM_CAR
FROM CARRERAS
WHERE ID_EXT_PER='E02';
```

```
SELECT NOM_CAR
FROM CARRERAS
WHERE ID_EXT_PER='E03';
```

Según el nivel de soporte de la transparencia de distribución, pueden examinarse tres casos de consulta:

1. Caso 1: La base de datos soporta transparencia de fragmentación.
2. ALTER TABLE CARRERAS
3. ADD CUPOS\_DISP NUMBER;
4. UPDATE CARRERAS SET CUPOS\_DISP=20;
- 5.
6. UPDATE CARRERAS SET CUPOS\_DISP=10 WHERE ID\_CAR='C01';
- 7.
8. ALTER TABLE CARRERAS
9. MODIFY CUPOS\_DISP NUMBER NOT NULL;
- 10.
11. SELECT \* FROM CARRERAS
12. WHERE CUPOS\_DISP>=20;
13. Caso 2: La base de datos soporta transparencia de ubicacion.

## Transparencia de Distribución

- Fragmentacion
- Ubicacion
- Ubicacion Local

## Transparencia de Transacción

Una transacción es un proceso

---

08/09/2025

- preguntas de la semana 2

En UTA\_Distribuida, una consulta a global.extensiones devuelve filas de Huachi, Ingahurco y Querochaca sin que el usuario conozca dónde vive cada fragmento. ¿Qué tipo de transparencia es la principal en juego?

global.oferta\_carrera une (UNION ALL) las tablas site\_\*.oferta\_carrera y el usuario no escribe manualmente múltiples SELECT por sede. ¿Qué nivel de transparencia se ilustra?

Para simplificar consultas, se crean sinónimos carreras, extensiones, oferta que apuntan a las vistas globales; así el usuario no necesita prefijos de esquema. ¿Qué transparencia describe mejor este caso?

Una transacción inserta oferta en Querochaca y actualiza cupos en Huachi como una unidad atómica; si falla un sitio se revierte todo. ¿Qué transparencia se busca demostrar?

Arquitectura ANSI-SPARC, esquema - estructura

- replications (consulta vista)
- diseño de tablas (transparente)
- script

nivel interno: describe estructura física de bd mediante ni

nivel conceptual: describe entidades, atributos, operaciones de usuario y restricciones

nivel externo: describe varios esquemas externos o vistas de usuario

## ANSI- SPARC EN SISTEMA DISTRIBUIDO

Para agregar lo aspectos de distribución se agregan 2 niveles de arquitectura

## HETEROGENEO

esquema global va todo tipo de transacciones



# Resumen Extendido: Bases de Datos Distribuidas (BDD)

## I. Definición y Componentes Fundamentales

Una **Base de Datos Distribuida (BDD)**, o *Distributed Database (DDB)*, es una base de datos tradicional que se encuentra dividida en diferentes partes, las cuales están **físicamente dispersas**<sup>1</sup>. Estas partes se acceden de forma lógica, al igual que una base de datos centralizada, a través de un **\*\*Sistema de Administración de Bases de Datos Distribuida (DDBMS)\*\***<sup>2</sup>.

Un **DDBMS** (*Distributed Database Managment System*) rige el almacenamiento y procesamiento de datos que están **lógicamente relacionados** a través de un sistema de computadoras interconectadas, distribuyendo las funciones de datos y procesamiento entre varios sitios<sup>3</sup>.

## Componentes de un DDBMS

El sistema opera gracias a la interacción de varios componentes<sup>4</sup>:

1. **Estaciones de Trabajo (Sitios y Nodos):** Los puntos físicos donde residen los datos y se procesan las transacciones.
2. **Componentes de Software y Hardware:** Los recursos informáticos necesarios en cada sitio.
3. **Medios de Comunicación:** La red que interconecta los sitios para el intercambio de datos y solicitudes.
4. **Procesador de Transacciones:** Gestiona las solicitudes de los usuarios. Cada uno puede acceder a datos en cualquier procesador de datos<sup>5</sup>.

5.

**\*\*Procesador de Datos:\*\*** Maneja todas las solicitudes de datos locales provenientes de cualquier procesador de transacciones<sup>6</sup>.

## Características de Operación de un DDBMS

Un DDBMS realiza una serie de actividades de forma transparente para el usuario<sup>7</sup>:

1. Recibe la solicitud de la aplicación<sup>8</sup>.
2. Valida, analiza y descompone la solicitud en operaciones matemáticas o lógicas (como seleccionar clientes con saldos específicos), pudiendo involucrar una o varias tablas<sup>9</sup>.
3. Descompone la solicitud en operaciones de I/O de disco<sup>10</sup>.
4. Busca, localiza, lee y valida los datos<sup>11</sup>.
5. Garantiza la **consistencia, seguridad e integridad** de los datos<sup>12</sup>.
6. Valida los datos según las condiciones especificadas en la solicitud<sup>13</sup>.
7. Presenta los datos seleccionados en el formato requerido<sup>14</sup>.

---

## II. Diseño y Fragmentación de la BDD

El diseño de una BDD se basa en las etapas de una base de datos centralizada: **Diseño Conceptual, Diseño Lógico y Diseño Físico**<sup>15</sup>. Sin embargo, el **Diseño Físico** en un entorno distribuido debe resolver tres problemas fundamentales (Rob, 2004)<sup>16</sup>:

1. Cómo dividir la base de datos en **fragmentos**<sup>17</sup>.
2. Qué fragmentos **replicar**<sup>18</sup>.
3. Dónde **localizar** (asignar) estos fragmentos y réplicas<sup>19</sup>.

La **Fragmentación y Replicación** gestionan la división y copia de datos, mientras que la **Colocación (Asignamiento)** se ocupa de la ubicación de los datos en la red<sup>20</sup>.

## Objetivos de la Distribución de Datos (Bell, 1992)

Los objetivos principales que guían la distribución de los datos son<sup>21</sup>:

1.

**\*\*Procesamiento Local:\*\*** Se busca **\*\*maximizar el procesamiento local\*\*** ubicando los datos lo más cerca posible de las aplicaciones que los utilizan<sup>22</sup>. Para ello, se evalúa el número de referencias locales y remotas para determinar la mejor localización de cada fragmento<sup>23</sup>.

2.

**\*\*Distribución de la Carga de Trabajo:\*\*** Se aprovechan las capacidades de las computadoras en cada sitio para **\*\*maximizar el grado de paralelismo\*\*** en la ejecución de las aplicaciones<sup>24</sup>.

3.

**\*\*Costo de Almacenamiento y Disponibilidad:\*\*** Aunque se pueden tener sitios especializados en almacenamiento, el costo de almacenamiento no es tan relevante como los costos de CPU, I/O y transmisión<sup>25</sup>. La **\*\*disponibilidad\*\*** es el factor de diseño más crucial<sup>26</sup>.

## Estrategias de Fragmentación

La fragmentación es el **particionamiento de la información** para distribuir cada parte en los sitios de la red<sup>27</sup>. Permite dividir un objeto (base de datos o tabla) en dos o más segmentos<sup>28</sup>. La validez de una fragmentación se prueba verificando que la tabla original pueda **recrearse** mediante una combinación de **uniones y articulaciones** (UNION y JOIN) de sus partes fragmentadas<sup>29</sup>. La información de esta división se registra en el **\*\*Catálogo de Datos Distribuidos (DDC)\*\***<sup>30</sup>.

Las estrategias a nivel de tabla son<sup>31</sup>:

- 

**\*\*Fragmentación Horizontal:\*\*** Divide una relación en subconjuntos de **\*\*tuplas (filas)\*\***<sup>32</sup>. Cada fragmento tiene **\*\*filas únicas\*\*** y **\*\*los mismos atributos\*\***<sup>33</sup>. Es el equivalente a una sentencia `SELECT` con una cláusula `WHERE` en un solo atributo<sup>34343434</sup>.

- 

**\*\*Fragmentación Vertical:\*\*** Divide una relación en subconjuntos de **\*\*atributos (columnas)\*\***<sup>35</sup>. Cada fragmento tiene **\*\*columnas únicas\*\***, con la excepción de la **\*\*columna clave\*\***, que es común a todos los fragmentos para permitir su posterior unión<sup>36363636</sup>. Es el equivalente a una sentencia `SELECT columnal, column2 INTO Nueva\_Tabla FROM Tabla`<sup>37373737</sup>.

- 

**\*\*Fragmentación Mezclada:\*\*** Es una **\*\*combinación de ambas estrategias\*\***<sup>38</sup>. Una tabla se divide primero horizontalmente (filas) y luego cada subconjunto se divide verticalmente (columnas) para satisfacer diferentes requerimientos por ubicación y por departamento<sup>39393939</sup>.

---

## III. Clasificación de las BDD

Los sistemas de administración de base de datos se clasifican según cómo se soporta la distribución de los procesos y los datos<sup>40</sup>:

1.

**\*\*Proceso en un Solo Sitio y Datos en un Solo Sitio (SPSD):\*\*** Es el escenario de una base de datos centralizada con un solo DBMS anfitrión<sup>41</sup>.

2.

**\*\*Proceso en Múltiples Sitios y Datos en un Solo Sitio (MPSD):\*\***  
Permite el procesamiento en múltiples sitios (como en un Servidor de Archivos o varios DBMS de LAN), pero los datos residen en una única ubicación<sup>42</sup>.

3.

**\*\*DDBMS Cliente/Servidor Totalmente Distribuido:\*\*** Describe el escenario ideal de una BDD, donde tanto los procesos como los datos están en **\*\*múltiples sitios\*\***<sup>43</sup>.

A su vez, los sistemas totalmente distribuidos se clasifican en:

- 

**\*\*SMBDD Homogéneas:\*\*** Todos los sitios utilizan el **\*\*mismo tipo de hardware, sistema operativo y DBMS\*\***<sup>44</sup>. Se asemejan a un sistema centralizado, donde el esquema global es la unión de todas las descripciones de datos locales<sup>45</sup>.

- 

**\*\*SMBDD Heterogéneas:\*\*** Manejan **\*\*diferentes sistemas de administración de base de datos (DBMS)\*\*** en los nodos locales<sup>46</sup>. Estos son característicos de los sistemas de **\*\*multi-bases de datos\*\***, donde la integración se realiza mediante software, y existen usuarios locales y globales<sup>47</sup>.

## Ventajas y Problemas de los DDBMS

El uso de DDBMS ofrece beneficios significativos<sup>48</sup>:

- Los datos se localizan cerca de donde son más demandados (máxima demanda)<sup>49</sup>.
- Acceso y procesamiento de los datos más rápido<sup>50</sup>.
- Facilita el **crecimiento** y la **independencia del procesador**<sup>51</sup>.
- 

**\*\*Comunicaciones mejoradas\*\*** y costos de operación reducidos<sup>52</sup>.

- Menos peligro de **falla en un solo punto** (mayor tolerancia a fallos)<sup>53</sup>.

No obstante, las BDD presentan complejidades<sup>54</sup>:

- El **rendimiento** puede afectarse por la carga de trabajo<sup>55</sup>.
  - La **confiabilidad** se reduce debido a la complejidad de los componentes (red, transacciones, replicación)<sup>56</sup>.
  - Mayores gastos de **construcción y mantenimiento**<sup>57</sup>.
  - Mayor complejidad en el **diseño e implementación** del sistema<sup>58</sup>.
-

## IV. Niveles de Transparencia en un DDBMS

La **Transparencia** es un concepto fundamental en las BDD, que se define como la separación de la semántica de alto nivel del sistema de los aspectos de bajo nivel relacionados con su implementación<sup>59</sup>. Su propósito es permitir que el usuario interactúe con el sistema como si fuera el único que utiliza la base de datos centralizada<sup>60</sup>.

Los principales niveles de transparencia son<sup>61</sup>:

### 1. Transparencia de Distribución

Permite manejar una base de datos físicamente dispersa como si fuera centralizada<sup>62</sup>. Se divide en tres subniveles (Rob, 2004)<sup>63</sup>:

- **\*\*Transparencia de Fragmentación:\*\*** El usuario no necesita saber que una tabla ha sido dividida en fragmentos (ej. Fragmentos E1, E2, E3 de la tabla `EMPLEADO` por ubicación)<sup>64</sup><sup>64</sup>. El usuario puede simplemente hacer un `SELECT \* FROM EMPLEADO`<sup>65</sup>.

- **\*\*Transparencia de Ubicación:\*\*** El usuario debe especificar los nombres de los fragmentos (`E1`, `E2`, etc.), pero no la ubicación física de la red (nodo) donde residen<sup>66</sup>.

- **\*\*Transparencia de Ubicación Local:\*\*** El usuario debe especificar tanto el fragmento como la ubicación exacta (el nodo) donde se encuentra<sup>67</sup>.

### 2. Transparencia de Transacción

Permite que una transacción (secuencia de operaciones atómicas) actualice datos en varios sitios de la red, garantizando que esta se realice o complete en su totalidad, o sea abortada, para **mantener la integridad de la base de datos**<sup>68</sup>. Una **Transacción Distribuida** actualiza y solicita datos de varios sitios remotos en una red<sup>69</sup>.

### 3. Transparencia de Replicación

Se refiere a que, si existen copias (**réplicas**) de objetos de la base de datos, su existencia debe ser **controlada por el sistema**, no por el usuario<sup>70</sup>.

### 4. Transparencia de Falla

Permite que el sistema **continúe operando** en caso de una falla de nodo. Las funciones que se perdieron debido a la falla serán recuperadas por otro nodo de la red<sup>71</sup>.

## 5. Transparencia de Desempeño

Permite que, cuando los objetos están fragmentados, el sistema maneje la **conversión de consultas** de usuario (definidas sobre relaciones globales) a consultas definidas sobre fragmentos<sup>72</sup>. También se ocupa de **mezclar las respuestas** de las consultas fragmentadas para obtener una respuesta global<sup>73</sup>.

Esto implica la **Optimización de Consultas**, cuyo objetivo es reducir al mínimo el costo total asociado con la ejecución de una solicitud<sup>74</sup>. Los costos a reducir incluyen<sup>75</sup>:

- Costo de **tiempo de acceso** (E/S).
- Costo de **comunicación** (transmisión de datos entre nodos).
- Costo de **tiempo de CPU** (sobrecarga de procesamiento).

Los algoritmos de optimización se basan en: **seleccionar el orden de ejecución óptimo** y **seleccionar los sitios a ser accedidos** para reducir al mínimo los costos de comunicación<sup>76</sup>.

## 6. Transparencia de Heterogeneidad

Permite la **integración de varios sistemas de administración de bases de datos locales diferentes** (relacional, de red, jerárquicos, multimedia) conforme a un esquema global común<sup>77</sup>. Es una característica esencial en los sistemas multi-bases de datos.