Base de Datos I
Curso No. TWB22B

Fundamentos de DB2, Administración, Seguridad y Programación en Base de Datos

Unidad 1
Fundamentos de DB2

Objetivos del Aprendizaje

- Listar las capacidades de DB2
- Describir los diferentes tipos de DB2.
- Discutir sobre la arquitectura de DB2 UDB.
- Explicar las capacidades del servidor DB2 y el cliente DB2
- Explicar las funciones de los productos que acompañan a DB2.
- Discutir sobre cómo las herramientas de DB2 ayudan a los administradores de bases de datos.
- Describir las interfaces proporcionadas en DB2 para los programadores.

Capacidades de DB2

- DB2 Universal Database (DB2 UDB) es un Sistema Administrador de Bases de Datos Relacionales (RDBMS) robusto
- DB2 es un RDBMS versátil
- Está diseñado para manejar una gran variedad de datos
- Puede almacenar y administrar grandes volúmenes de datos
- Proporciona una buena seguridad para los datos almacenados y así como una alta disponibilidad de estos
- Proporciona soporte para diferentes plataformas y sistemas operativos

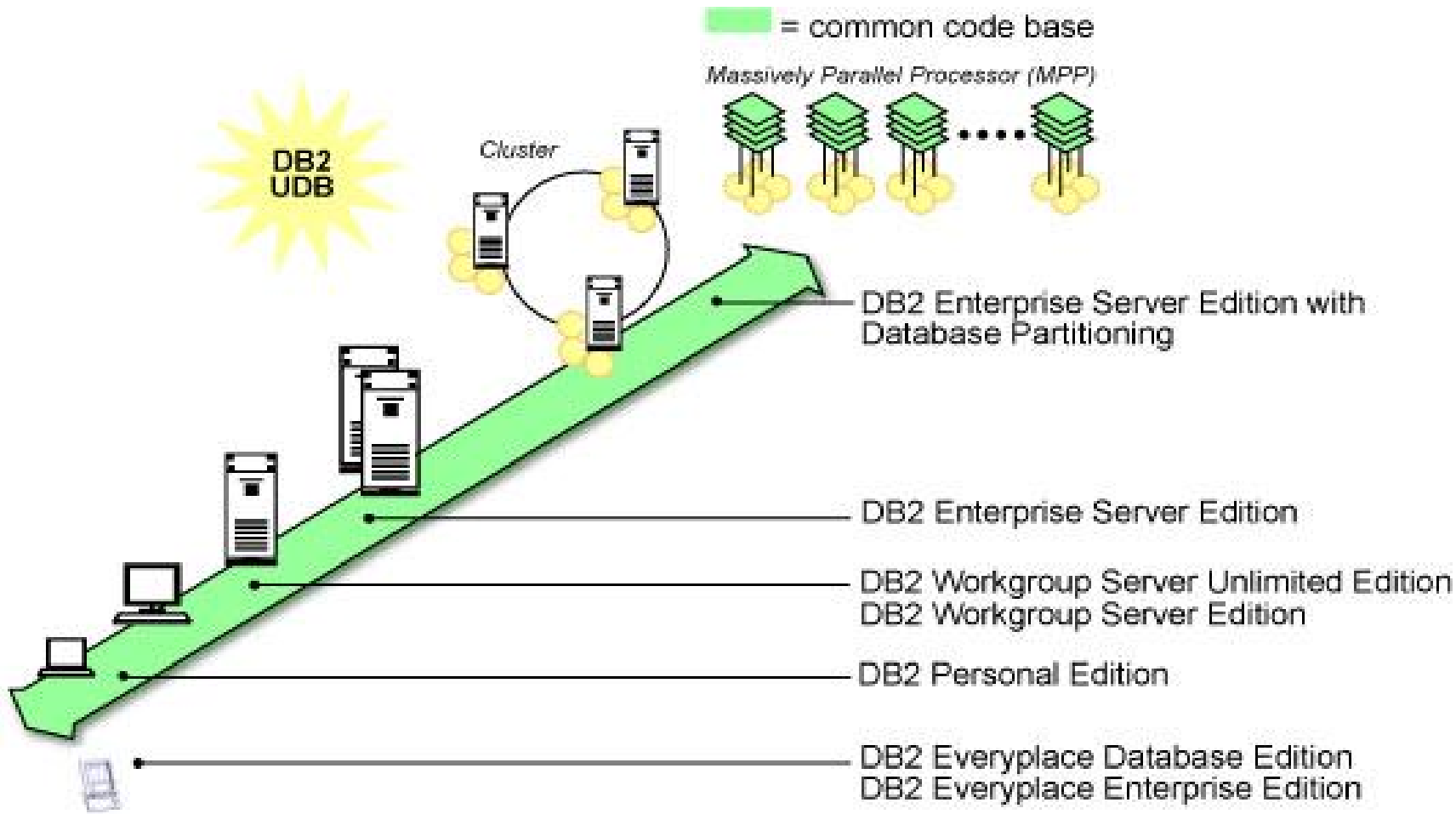
Ventajas de DB2

- Confiabilidad de los datos
- Alta disponibilidad de datos
- Escalabilidad
- Rendimiento de base de datos
- Rápido desarrollo de aplicaciones
- Capacidad para construir soluciones OLAP
- Soluciones de administración de bases de datos de siguiente generación

Familia de Productos DB2

- DB2 Everyplace
- DB2 Personal Edition (DB2 PE)
- DB2 Personal Developer’s Edition
- DB2 Universal Developer’s Edition
- DB2 Workgroup Server Edition
- DB2 Workgroup Server Unlimited Edition
- DB2 Enterprise Server Edition

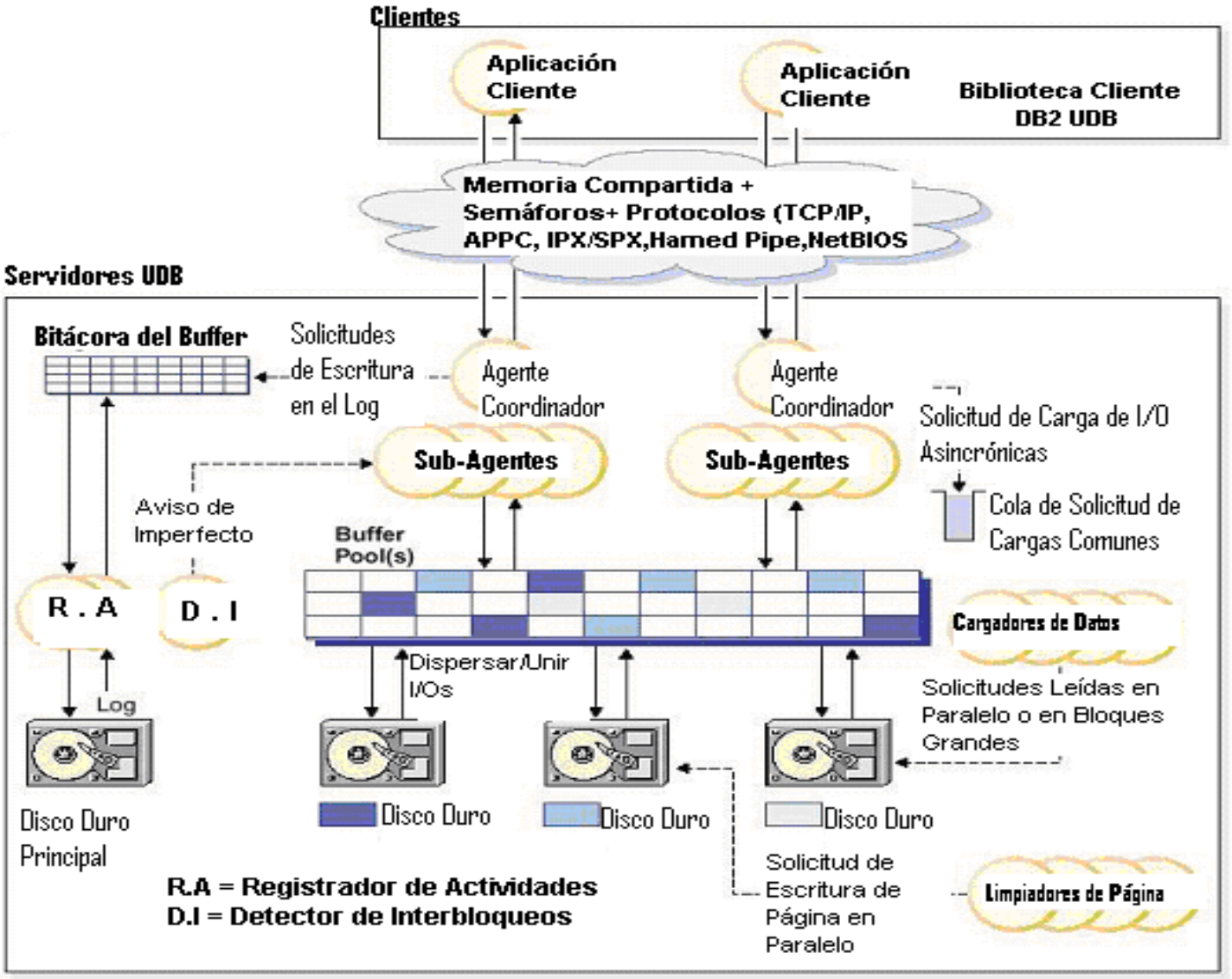
Familia de Productos DB2 … 2



Arquitectura de DB2

- Se basa en la arquitectura cliente-servidor
- Puede comunicarse con cualquier aplicación que soporte el protocolo de Arquitectura de Bases de Datos Relacionales Distribuidas (Distributed Relational Database Architecture - DRDA)
- DB2 soporta una variedad de protocolos
- Los clientes DB2 pueden comunicarse con el servidor incluso si están en diferentes plataformas

Arquitectura de Procesos de DB2



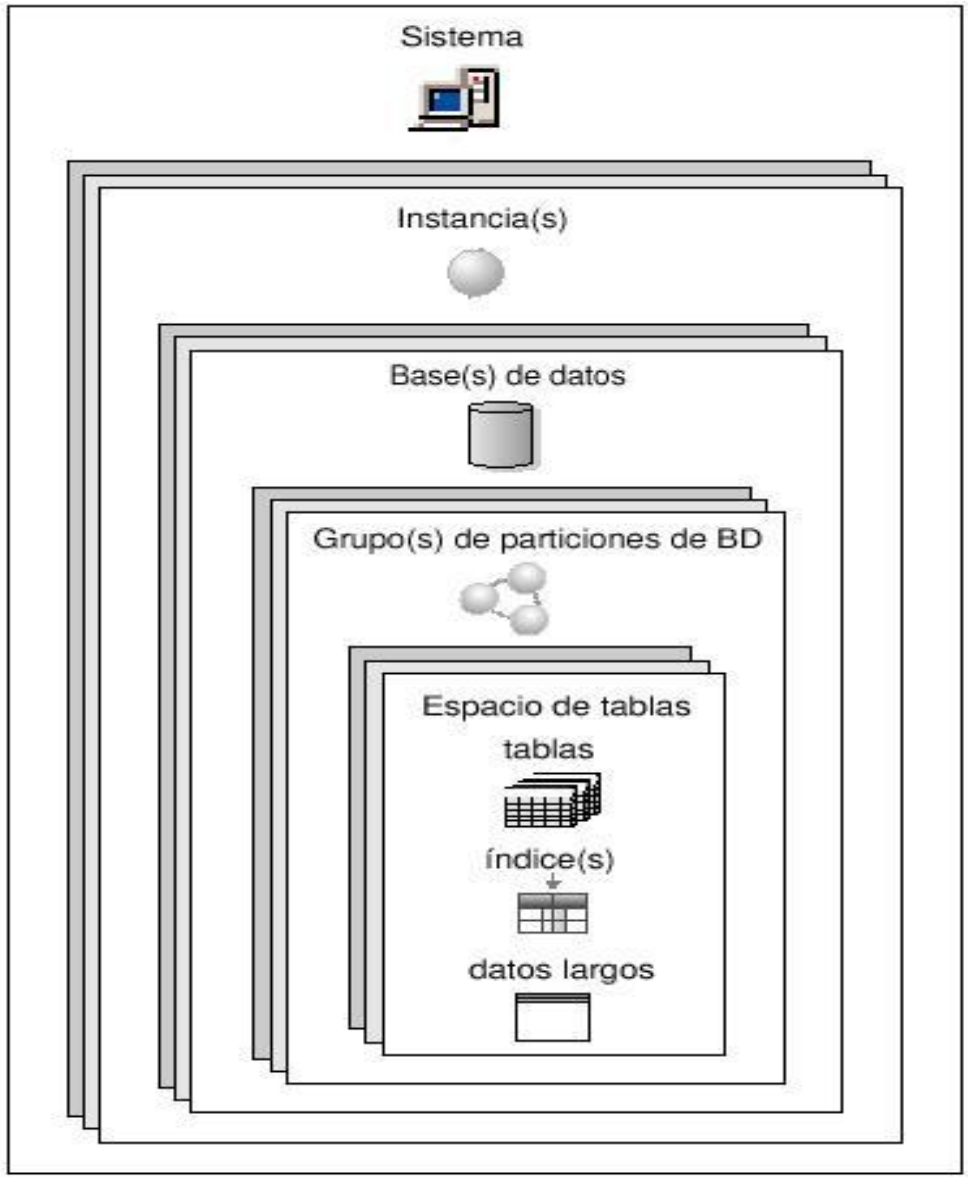
Diferentes Agentes DB2

- **Agente Lógico (Logical Agents)**
  - Responsable de facilitar interacciones con la base de datos
  - Un agente es asignado para cada aplicación conectada al administrador de la base de datos
  - Almacena la información requerida por la aplicación.
- **Agente Trabajador (Worker Agent)**
  - No está atado a alguna aplicación
  - Llevan a cabo solicitudes de acciones y tienen información para completar una acción en particular

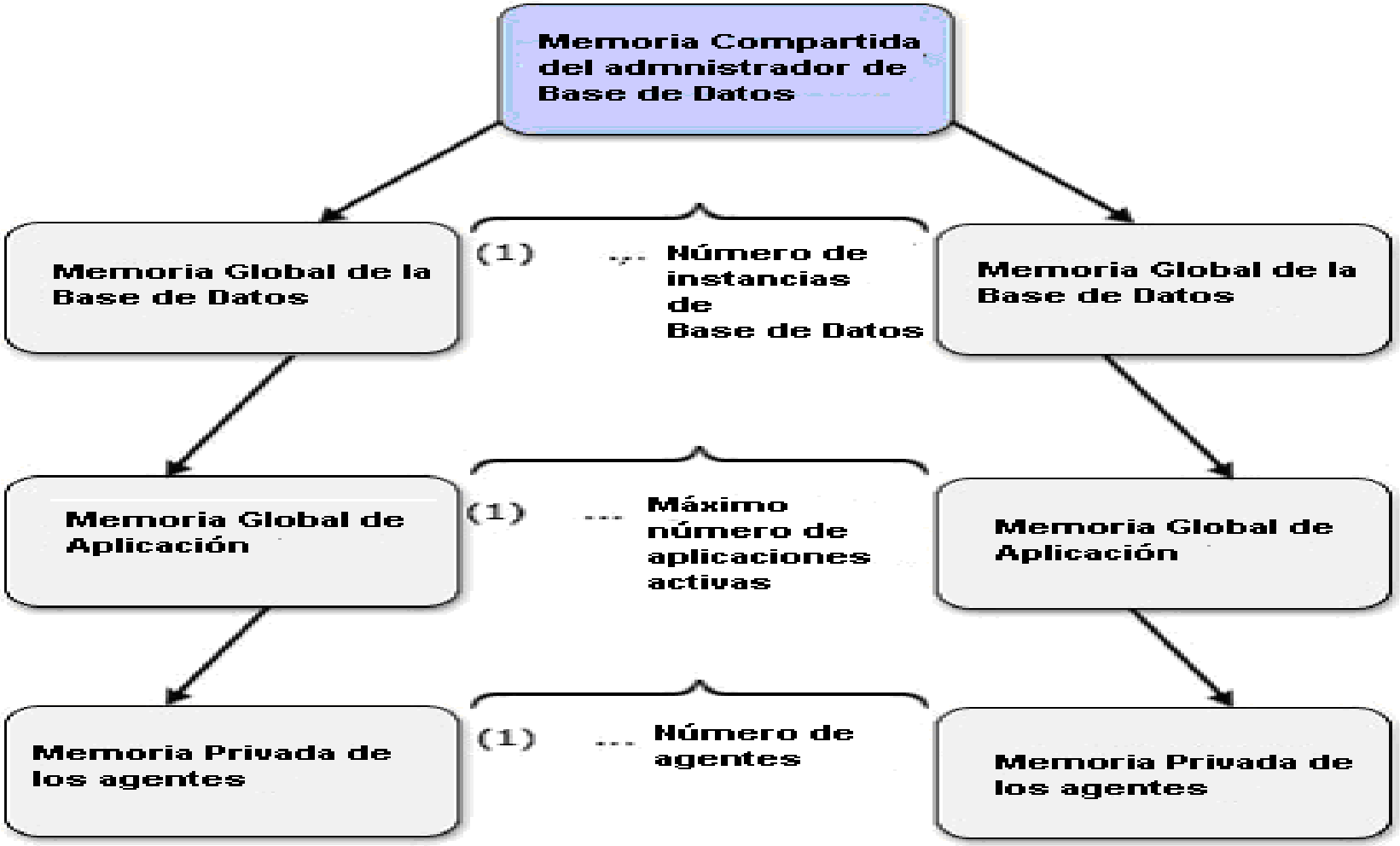
Diferentes Agentes DB2 ... 2

- **Agente Coordinador (Coordinator Agent)**
  - Es asignado a cada aplicación cliente
  - Coordina el procesamiento y se comunica con la aplicación cliente
  - Puede haber un conjunto de subagentes trabajando para lograr el requerimiento

Diseño de Almacenamiento DB2



Arquitectura de Memoria de DB2



Servidores DB2

- DB2 Enterprise Server Edition (DB2 ESE)
- DB2 Workgroup Server Edition (DB2 WSE)

Clientes DB2

- Run-Time Client
- Administration Client
- Application Development Client

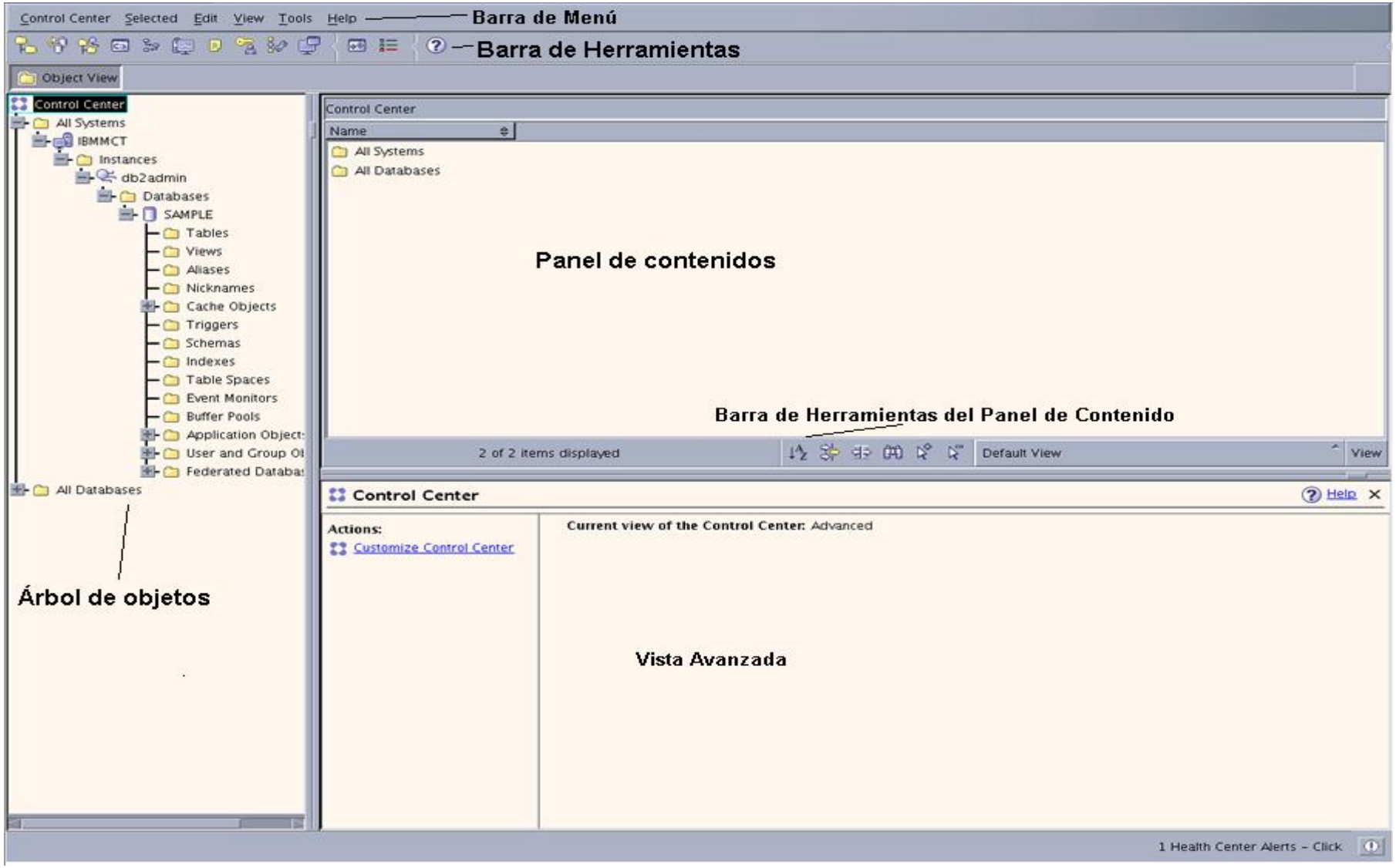
Otros Productos DB2

- DB2 Data Links File Manager
- DB2 Connect
- DB2 Connect Personal Edition
- DB2 Connect Enterprise Edition
- DB2 Connect DB2 Connect Unlimited Edition
- UDB Extenders
- DB2 XML Extendido (DB2 XML Extender
- DB2 Net Search Extender
- DB2 Spatial Extender
- DB2 Audio, Video e Image Extender
- DB2 Text Extender

Tareas a ser Realizadas por el Administrador

- Iniciar y parar la base de datos DB2.
- Configurar Instancias y Base de Datos.
- Obtener copias de respaldo (back-up) y restaurar (restore).
- Crear tablas.
- Crear índices.
- Monitorear eventos de la base de datos y tomar medidas correctivas.
- Monitorear y mejorar el rendimiento de la base de datos.
- Establecer alertas del sistema.

Centro de Control

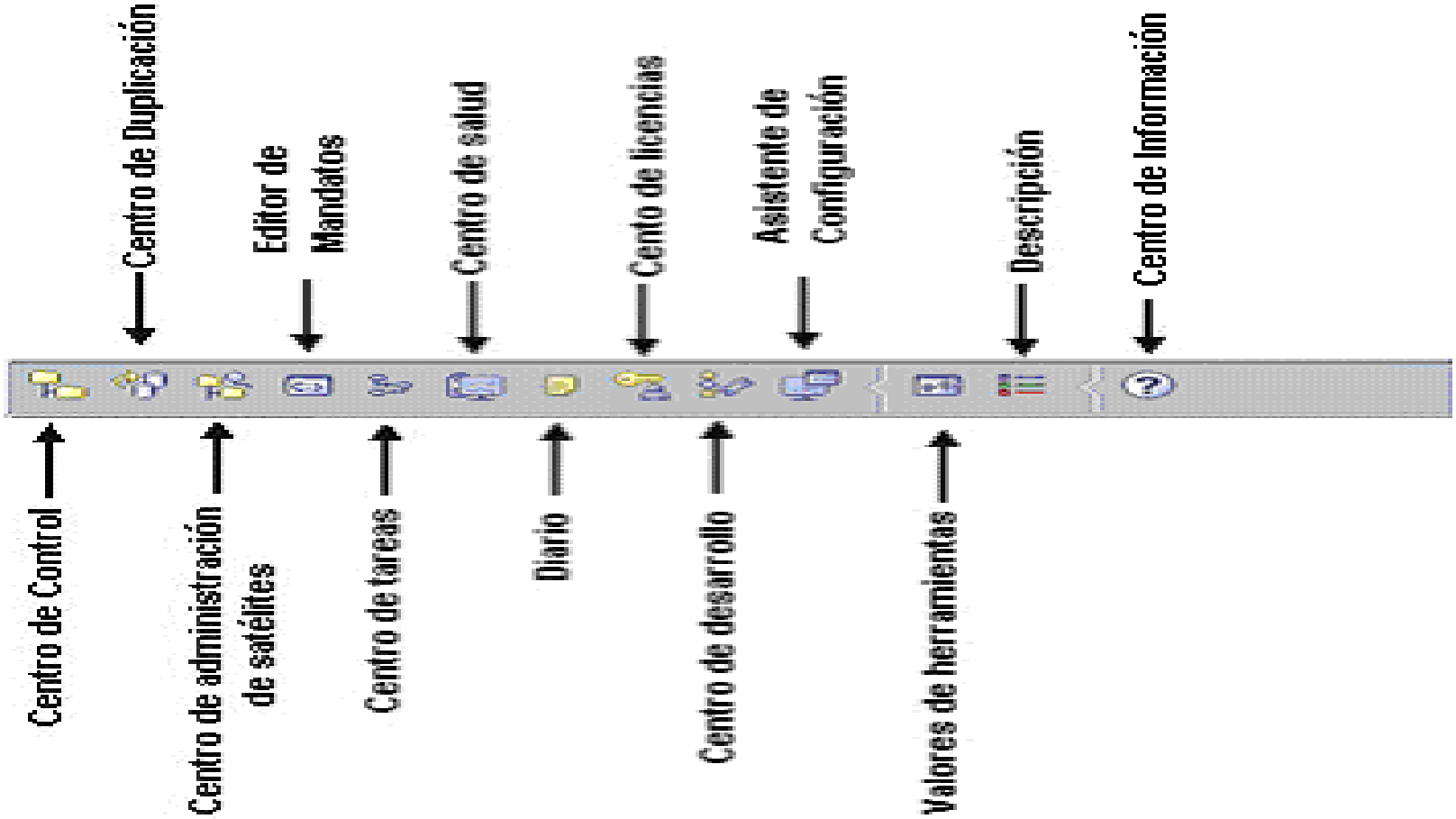


Barra de Menú

- Contiene elementos con los cuales el administrador puede manipular los objetos de la base de datos
- Los elementos del menú permiten además al administrador acceder a otras herramientas

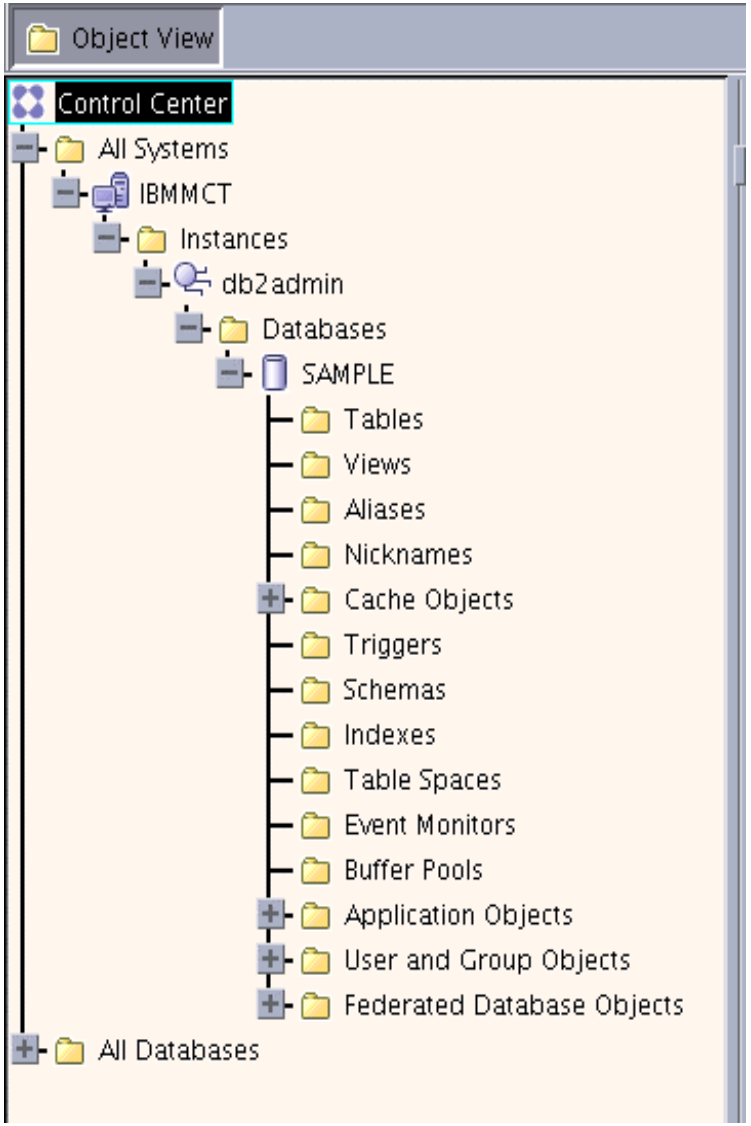
Barra de Herramientas

- Contiene iconos que ayudan al administrador a acceder a una variedad de herramientas



Árbol de Objetos

- Permite al administrador navegar por los objetos de la base de datos
- Permite manipular varios objetos de la base de datos

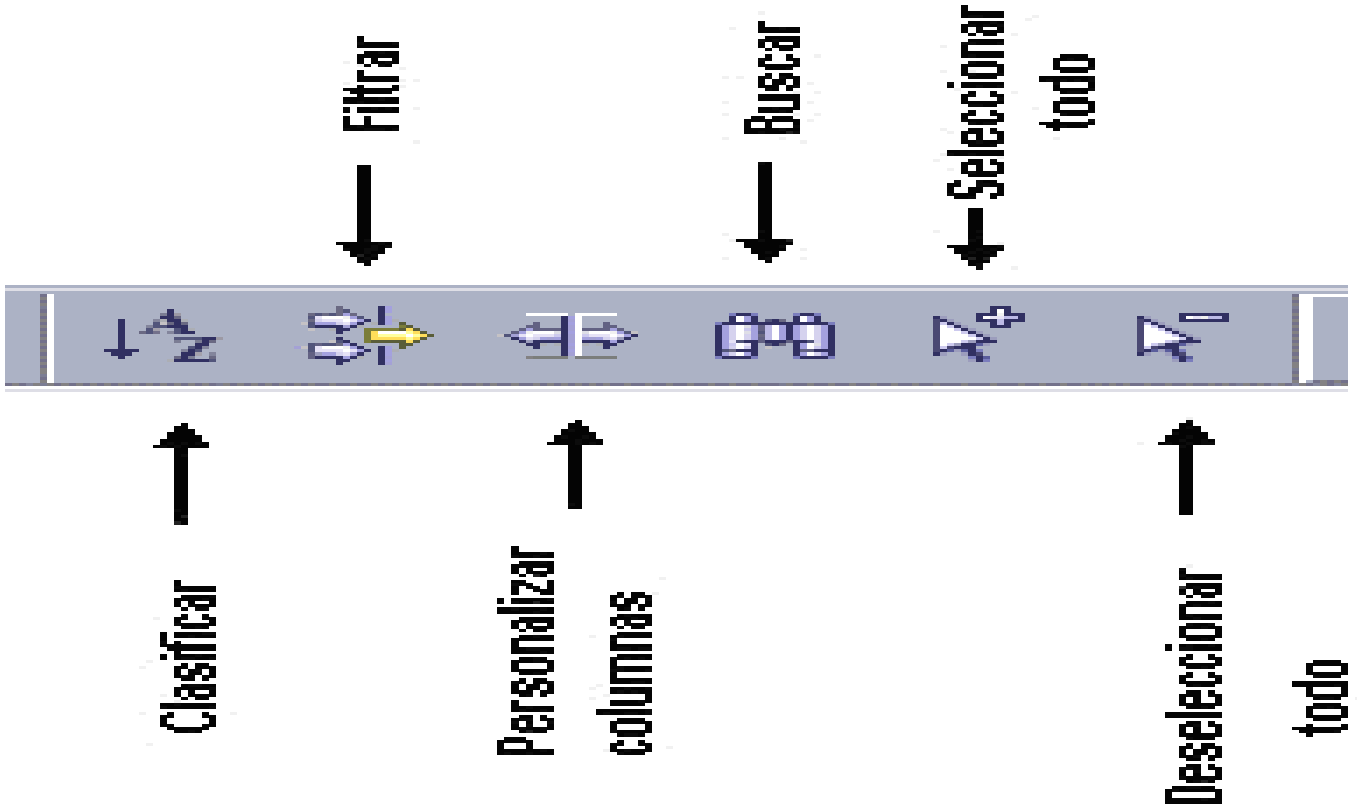


Panel de Contenidos

- Muestra los contenidos del objeto seleccionado en el árbol de objetos
- Permite al administrador trabajar con los objetos del sistema y de la base de datos

Barra de Herramientas del Panel de Contenidos

- Tiene herramientas especiales para realizar operaciones en el panel de contenidos tales como ordenamiento, filtrado, personalización de columnas y otras funcionalidades de edición



Menú de Herramientas

- Centro de Duplicación (Replication Center)
- Centro de administración de satélites (Satellite Administration Center)
- Editor de mandatos (Command Editor)
- Centro de tareas (Task Center)
- Centro de salud (Health Center)
- Diario (Journal Center)
- Centro de licencias (License Center)
- Centro de desarrollo (Development Center)
- Centro de información (Information Center)
- Asistente de Configuración (Configuration Assistant)

Resumen

- Se presentaron las funciones de DB2
- Se describieron los diferentes tipos de DB2
- Se discutió sobre la arquitectura de DB2 UDB
- Se explicaron las capacidades del servidor DB2 y el cliente DB2
- Se explicaron las funciones de los productos que acompañan a DB2
- Se discutió sobre cómo las herramientas de DB2 ayudan a los administradores de bases de datos
- Se describieron las interfaces proporcionadas en DB2 para los programadores

Unidad 2
Laboratorio de Fundamentos de DB2

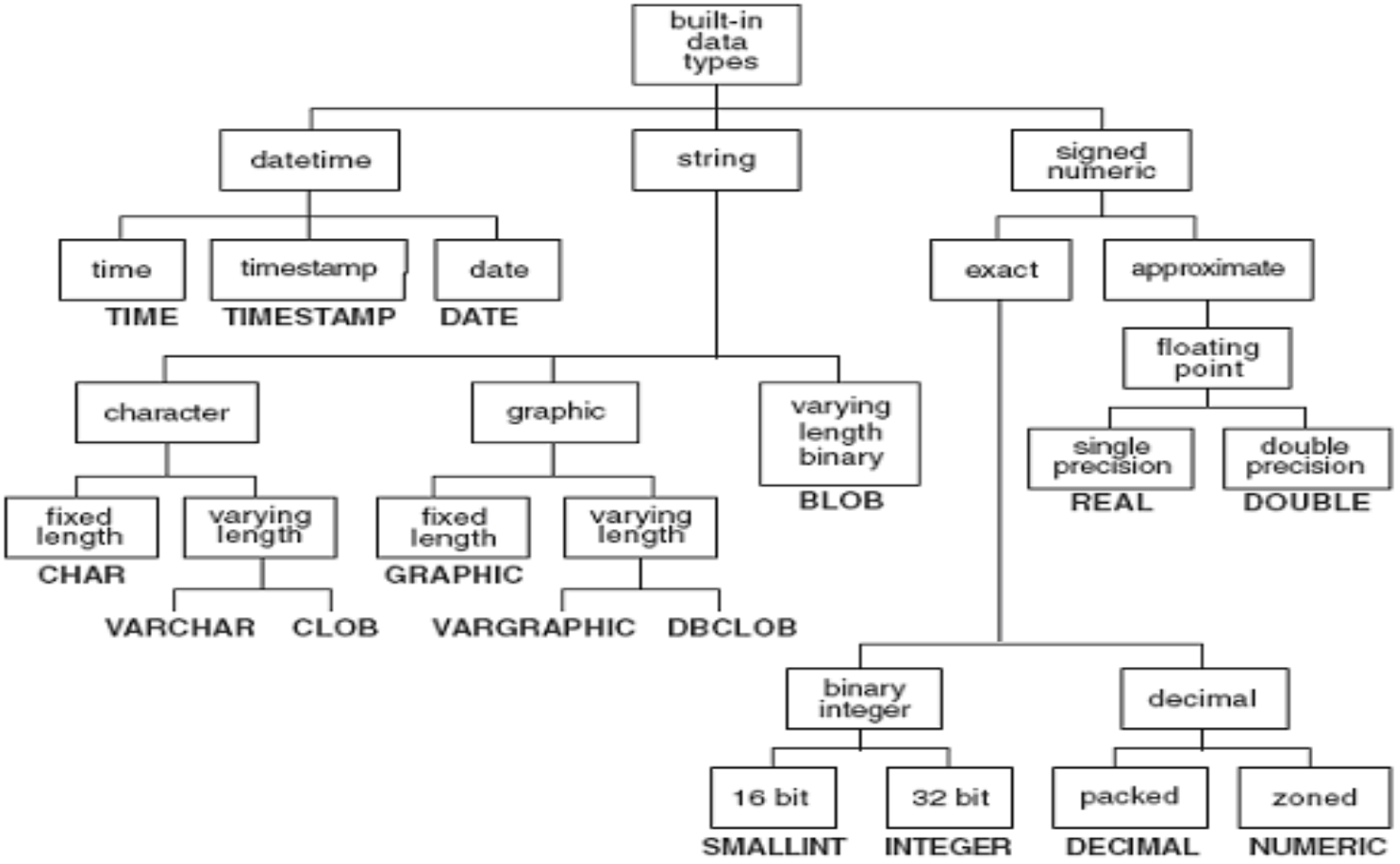
Ejercicios de Laboratorio

Unidad 3
Crear y Administrar Objetos de la Base de Datos

Objetivos del Aprendizaje

- Explicar cómo crear una tabla
- Describir cómo crear una tabla con restricciones
- Listar los cinco tipos importantes de restricciones cuando se crea una tabla
- Definir los tipos de datos
- Describir cómo agregar y modificar columnas en una tabla
- Discutir cómo eliminar tablas
- Describir cómo crear y eliminar vistas
- Explicar alias/sinónimos.

Tipos de Datos de Columnas



Objetos de Base de Datos

- Tablas
- Vistas
- Restricciones de Integridad
  - Reforzar las reglas del negocio
  - Mantener la integridad de la data

Crear una Tabla

- Una tabla es una colección de filas
- Una fila es una colección de columnas

	Columna 1	Columna 2	Columna 3
Fila 1			
Fila 2			
Fila 3			
Fila 4			

Claúsula CREATE

- Una tabla se crea usando el comando CREATE TABLE
- El comando CREATE TABLE es una sentencia para la creación de un objeto de la base de datos
- La claúsula CREATE TABLE es una sentencia del lenguaje de definición de datos - DDL
- Esta sentencia puede ser ejecutada en cualquier herramienta ejecutora de SQL

Claúsula CREATE...2

```
CREATE TABLE nombretabla (
  columna1 TIPODATO,
  columna2 TIPODATO,
  columna3 TIPODATO,
  …
  columnan TIPODATO
)
```

- Cada tabla tiene un nombre
- Cada columna en la tabla tiene nombre

Clausúla CREATE– Un Ejemplo

```
CREATE TABLE empleado (empno INTEGER,
apellido VARCHAR(50),
nombre VARCHAR(50),
salario REAL);
```

- Empleado es el nombre de la tabla

Clausúla CREATE– Un Ejemplo ... 2

empno	apellido	nombre	salario

Restricción de Integridad

- Es una limitación o una restricción impuesta en un objeto de la base de datos
- Asegura que una base de datos preserve la integridad de los datos almacenados en un objeto
- Se especifican al momento de creación del objeto de base de datos o después
- El RDBMS las hace cumplir

Tipos de Restricciones

- **Restricción** NOT NULL
- **Restricción** FOREIGN KEY
- **Restricción** UNIQUE
- **Restricción** PRIMARY KEY
- **Restricción** CHECK

Restricciones NOT NULL(NO NULO)

- Una columna en una tabla puede ser definida para ser NOT NULL
- Las columnas restringidas con NOT NULL no pueden ser NULL

```
CREATE TABLE empleado (
   empno INTEGER NOT NULL,
   apellido VARCHAR(50),
   nombre VARCHAR(50),
   salario REAL,
   telefono INTEGER);
```

Restricción Primary Key (Clave Primaria)

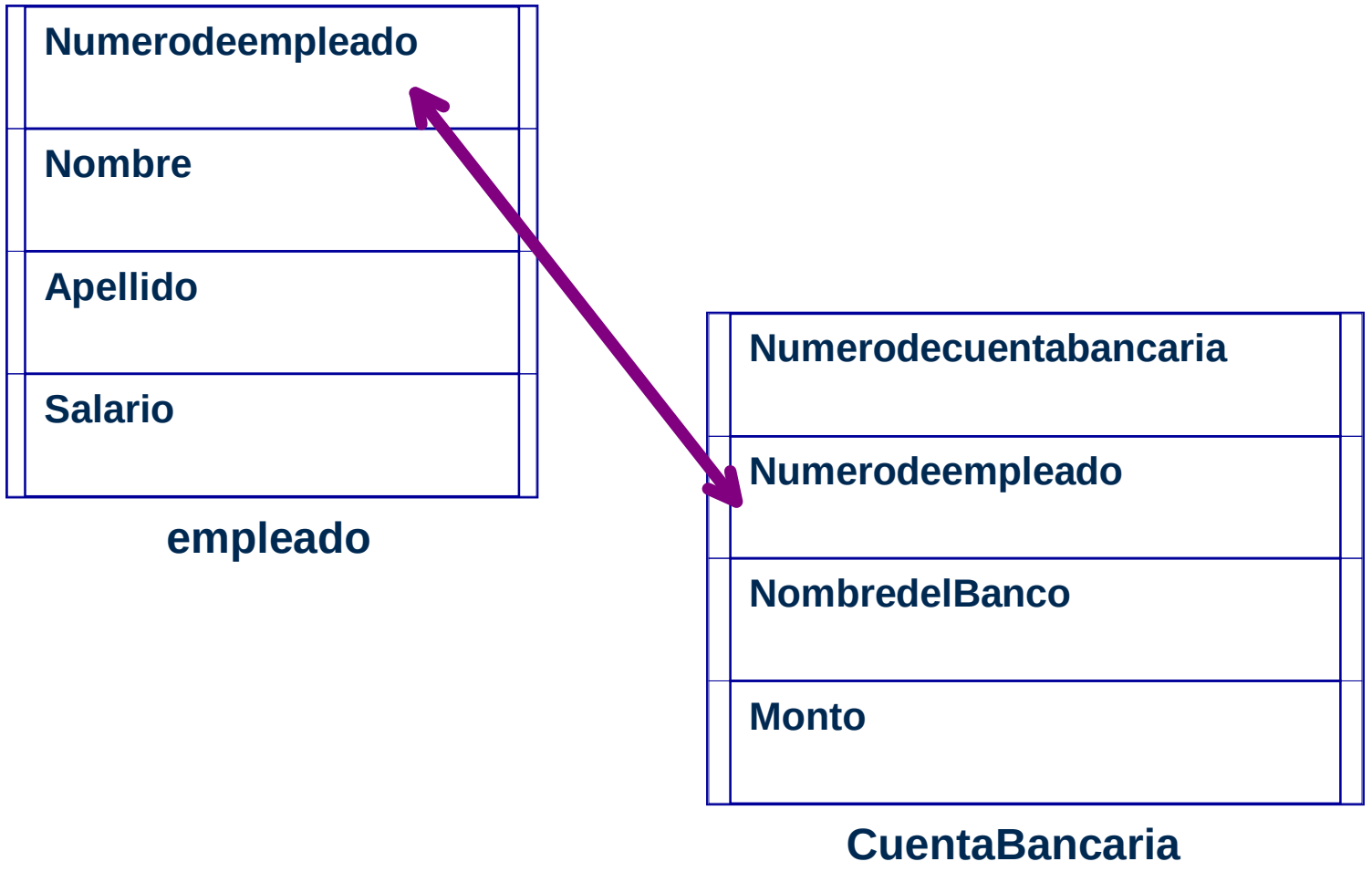
- La restricción primary key se define para una o más columnas en una tabla
- Tales columnas no pueden ser NULL
- La restricción “primary key” asegura que no habrán valores duplicados en esas columnas
- La restricción “primary key” asegura valores únicos para aquellas columnas en la tabla

Restricción Primary Key ... 2

```
CREATE TABLE empleado (
empno INTEGER NOT NULL,
apellido VARCHAR(50),
nombre VARCHAR(50),
salario REAL,
telefono INTEGER,
CONSTRAINT pk_employee_01 PRIMARY KEY (empno));
```

empno es la clave primaria

Relacionar Dos Tablas



Relacionar Dos Tablas... 2

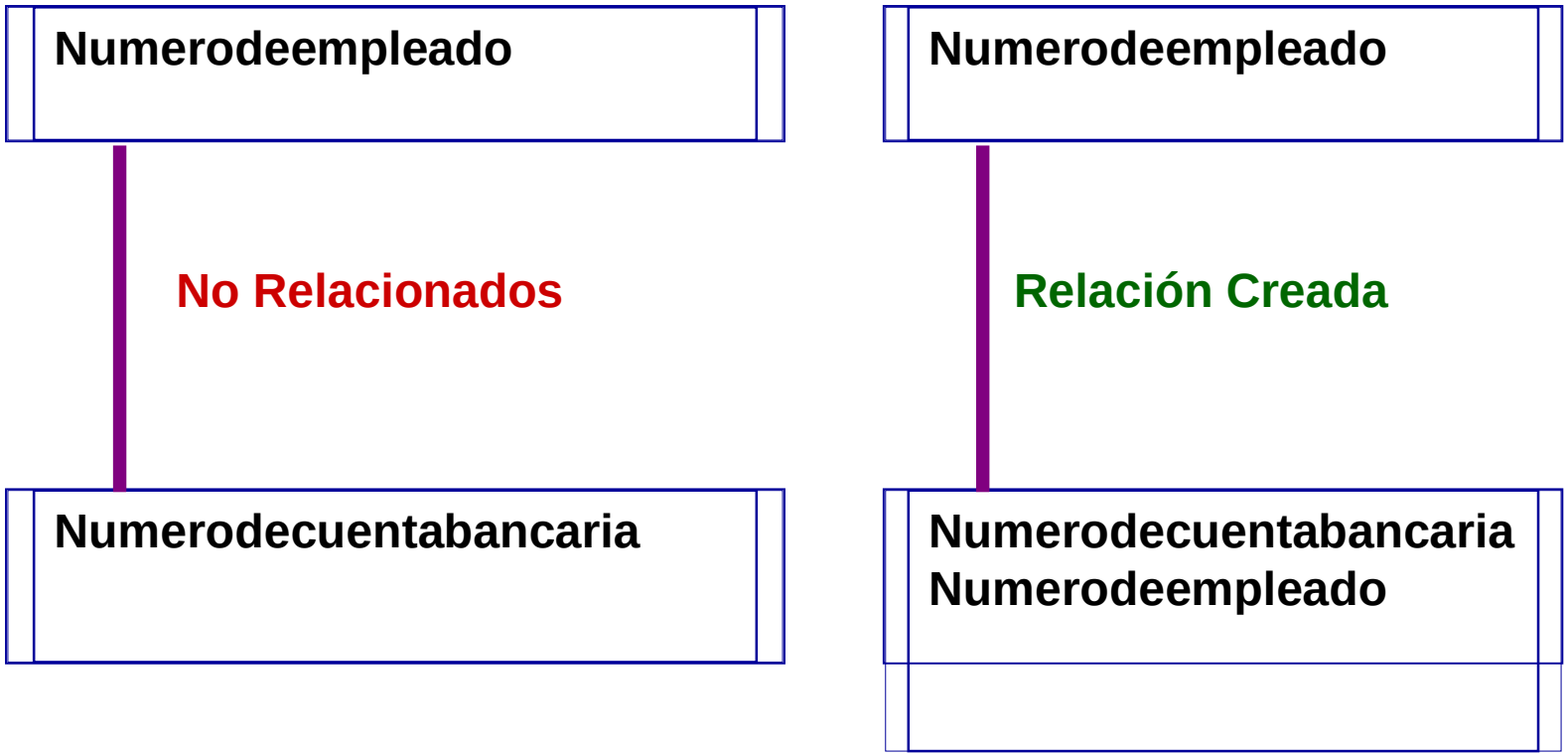
Restricción Foreign Key (Clave Foránea)

- Las claves foráneas son la base para relacionar tablas
- Una columna se llama clave foránea solamente si hace referencia a la columna de clave primaria de otra tabla
- Una restricción definida entre una columna de clave foránea y una columna de clave primaria se llama *restricción de integridad referencial*

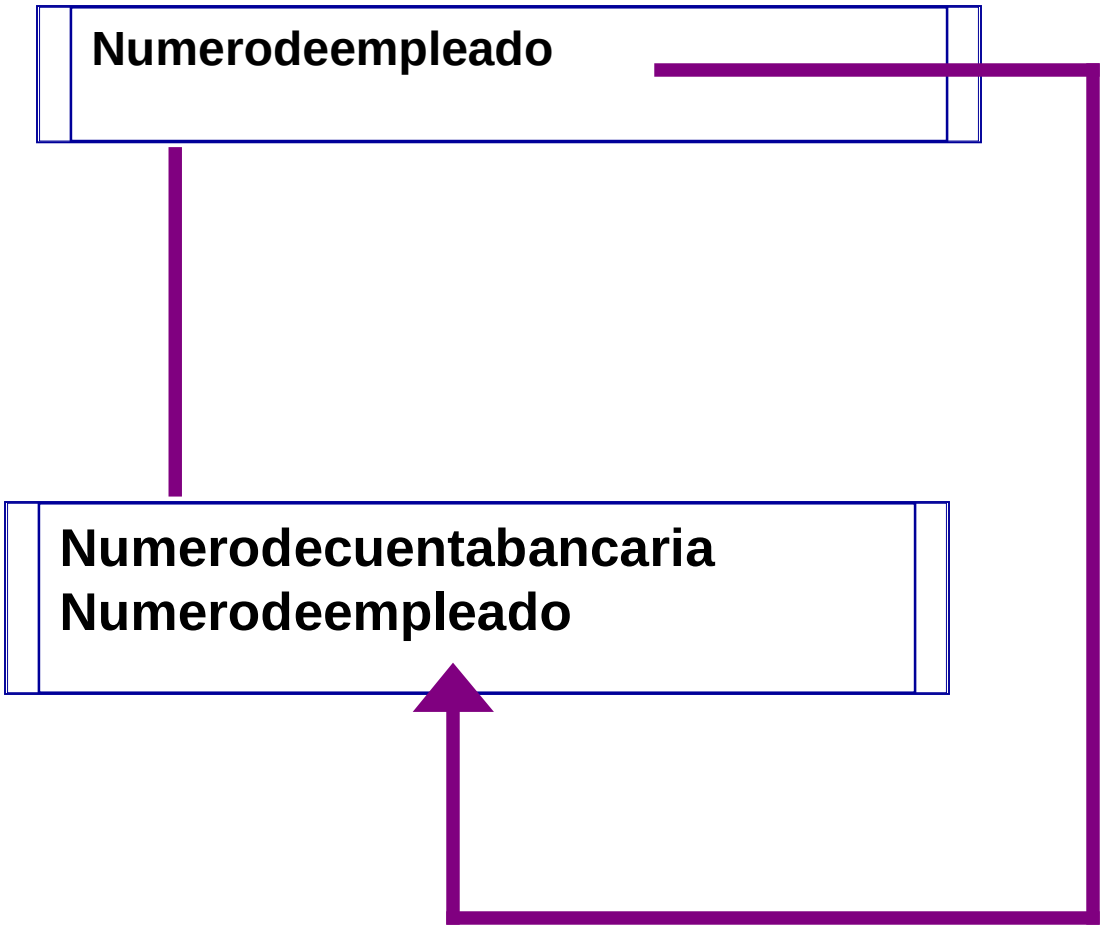
Restricción Foreign Key (Clave Foránea) … 2

- La tabla que contiene la columna de clave primaria, la cual es referenciada por la clave foránea, es llamada la tabla *padre*
- La tabla que contiene la clave foránea se llama la tabla *hija*
- Los valores permitidos en una columna de clave foránea son los valores de columna referenciados de la tabla padre o NULL

Restricción Foreign Key ... Un Ejemplo



Inclusion de una Restricción Foreign Key



Se establece como una Clave Foránea en la Tabla cuentabancaria

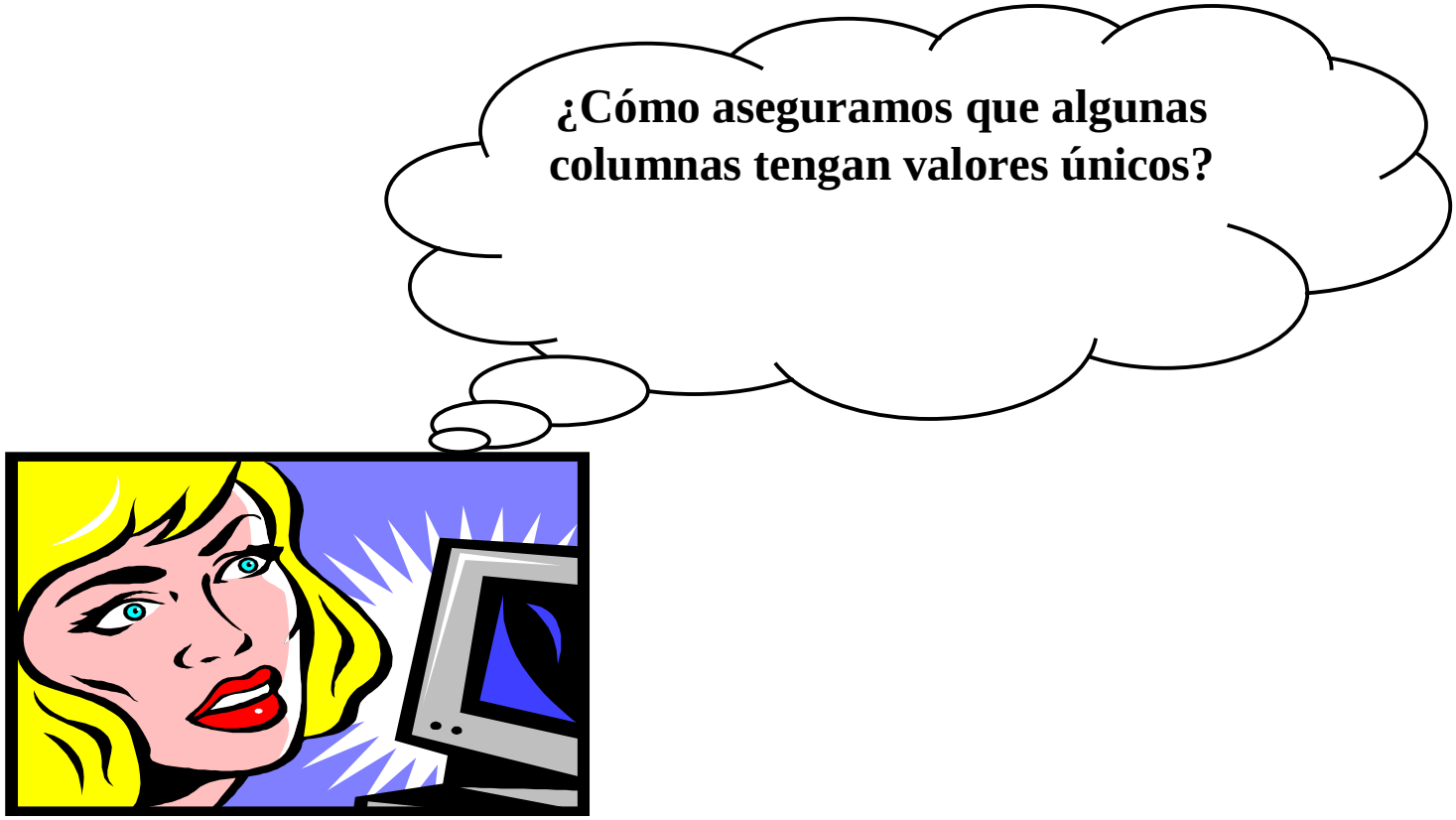
Especificar una Clave Foránea en una Tabla

```
CREATE TABLE Cuentabancaria (
Nocuentabancaria VARCHAR(50) NOT NULL,
Empno INTEGER,
Nombrebanco VARCHAR(50),
CONSTRAINT pk_bank_acct_no1 PRIMARY KEY
  (nocuentabancaria),
CONSTRAINT fk_bank_acct_01 FOREIGN KEY (empno)
  REFERENCES empleado(empno));
```

Eliminación en Cascada

```
CREATE TABLE Cuentabancaria (
nocuentabancaria VARCHAR(50) NOT NULL,
Empno INTEGER,
Nombrebanco VARCHAR(50),
CONSTRAINT pk_bank_acct_no1
  PRIMARY KEY (nocuentabancaria),
CONSTRAINT fk_bank_acct_01 FOREIGN KEY (empno)
  REFERENCES empleado (empno)
ON DELETE CASCADE);
```

Valores Únicos en Columnas



Restricción UNIQUE (Única)

```
CREATE TABLE empleado (
empno INTEGER NOT NULL,
apellido VARCHAR(50),
nombre VARCHAR(50),
salario REAL, telefono INTEGER NOT NULL,
CONSTRAINT pk_employee_01 PRIMARY KEY (empno),
CONSTRAINT uk_employee_01 UNIQUE (telefono));
```

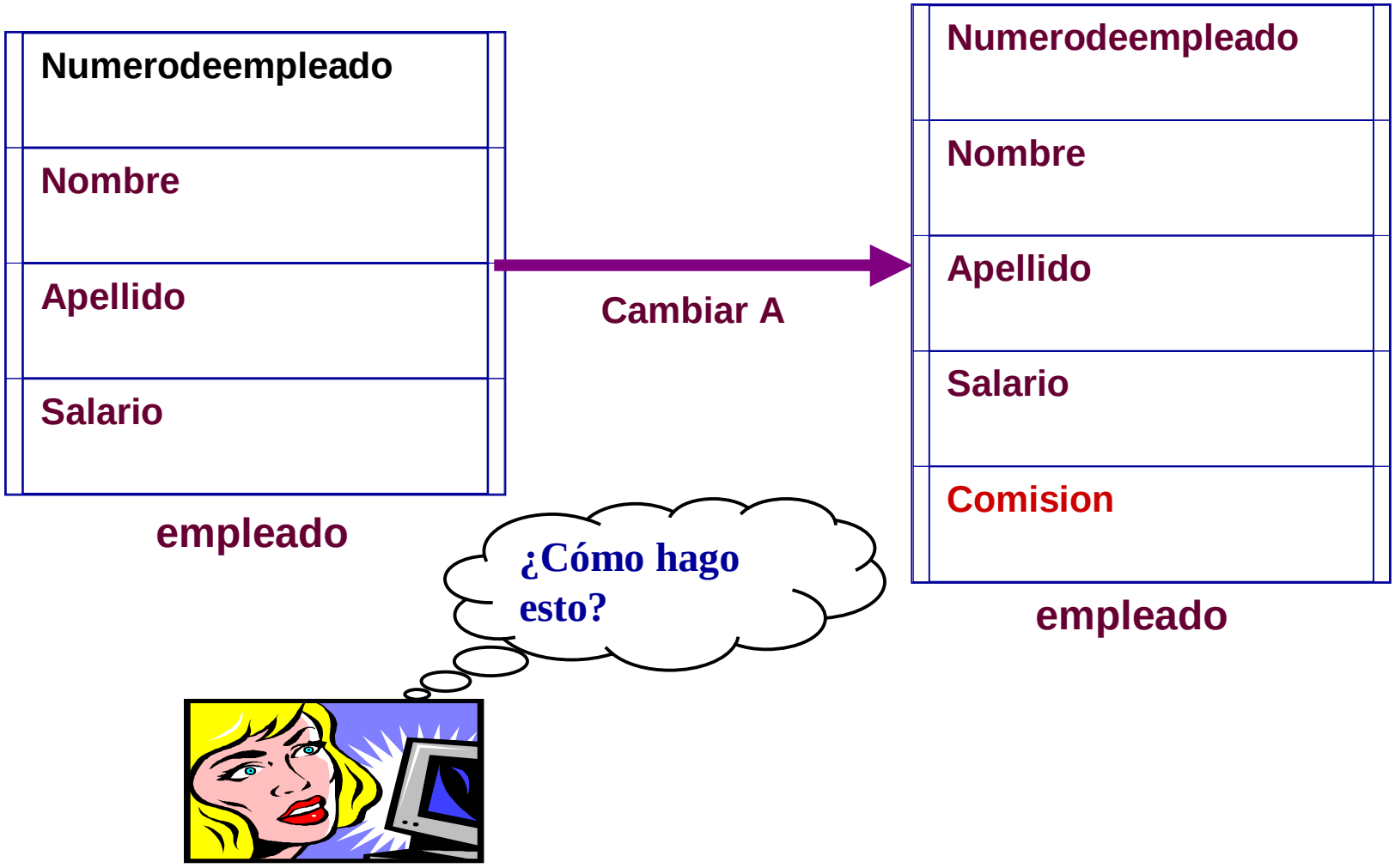
Restricción CHECK (CHEQUEO)

- Una restricción CHECK verifica la validez de los datos ingresados a una tabla contra un conjunto de restricciones
- Mientras se usa una restricción CHECK, el RDBMS retornará un error cuando el usuario intente ingresar un valor que no está permitido
- Una restricción CHECK puede hacer referencia solamente a un conjunto específico de valores constantes, u operaciones sobre dichos valores
- Una restricción CHECK hacer referencia a una columna o a una fila en no puede una tabla, incluyendo aquella donde se define la restricción
- Una restricción CHECK no puede hacer referencia a cualquier palabra clave especial específica del RDBMS que pueda tener valores en ella, como por ejemplo la fecha del sistema.

Restricción CHECK– Un Ejemplo

```
CREATE TABLE empleado (
empno INTEGER NOT NULL,
apellido VARCHAR(50),
nombre VARCHAR(50),
salario REAL,
telefono INTEGER NOT NULL,
CONSTRAINT pk_employee_01 PRIMARY KEY (empno),
CONSTRAINT uk_employee_01 UNIQUE (telefono),
CONSTRAINT ck_employee_01 CHECK (salario < 1000000));
```

Modificar una Tabla



Modificar una Tabla ... 2

- Las columnas pueden ser agregadas o modificadas en la base de datos usando la sentencia **`ALTER TABLE.`**
- Considere el siguiente ejemplo:

```
ALTER TABLE empleado
ADD COLUMN Comision REAL;
```

Modificar una Columna

```
ALTER TABLE empleado
ALTER COLUMN Apellido
   SET DATA TYPE VARCHAR(100);
```

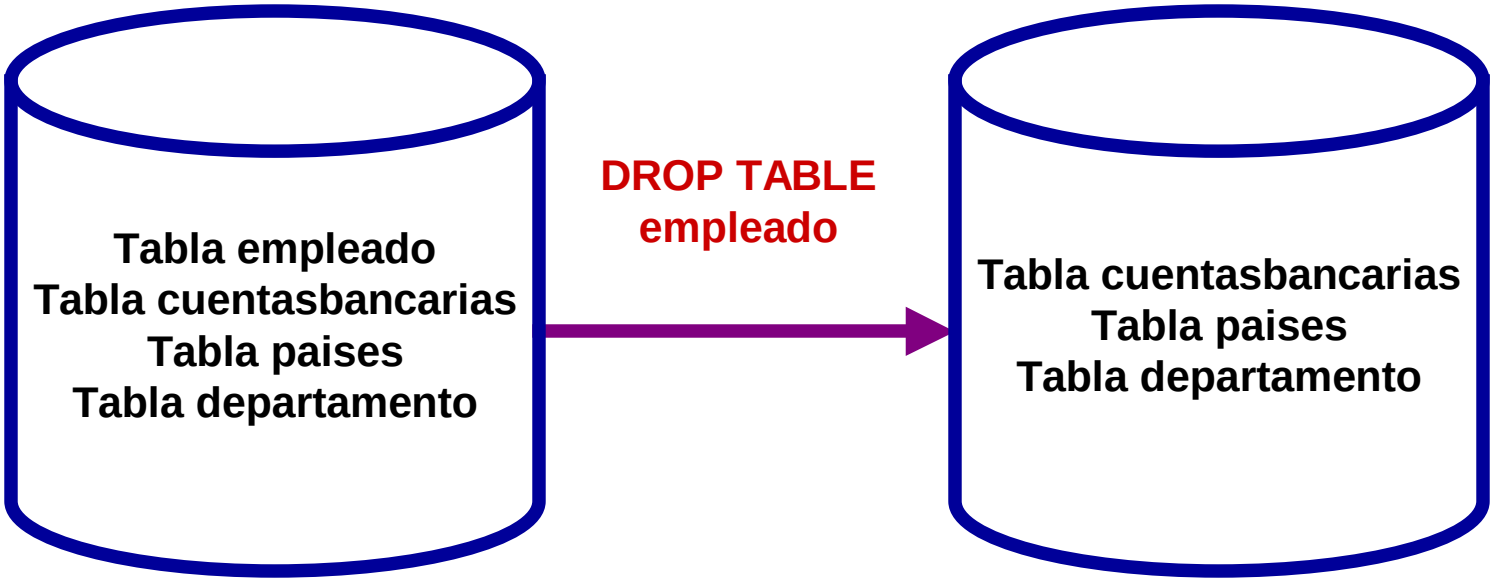
La sentencia anterior modifica la columna existente `lastname`, cambiando el ancho de la columna de 50-caracteres a 100-caracteres

```
ALTER TABLE empleado
ADD CONSTRAINT Checksalary
                  CHECK(Salario>0);
```

Eliminar una Tabla

DROP TABLE empleado;

Elimina la tabla `empleado` de la base de datos



Crear Vistas

- Una vista es una tabla virtual
- Una vista no almacena los datos
- Una vista sólo puede actuar como una ventana para una o más tablas
- Las vistas son creadas para dar una instantánea de los datos disponibles en la base de datos
- Las vistas permiten la visión restringida de los datos

CREATE VIEW

Un vista creada con dos columnas, `empno` y `lastname`

```
CREATE VIEW employee_view AS (
 SELECT empno, apellido
 FROM empleado
    );

SELECT * FROM employee_view;
```

El resultado

empno	apellido
1	JONES
2	JAMES
3	ROGER

Crear Índices

- Un índice es un objeto de la base de datos DB2 usado para localizar filas en una tabla que cumplen cierto criterio
- Creado en columnas específicas de una tabla
- Las restricciones PRIMARY KEY y UNIQUE crean sus propios índices
- Estos índices son creados automáticamente por la base de datos

Crear Índices – Un Ejemplo

- Crear un índice en una sola columna

```
CREATE INDEX empindex ON empleado(telefono);
```

- Construir un índice en más de una columna

```
CREATE INDEX empindex
             ON empleado(nombre, apellido)
```

Esquemas

- Un esquema (Schema) es un conjunto de objetos con nombre.
- Los esquemas proporcionan una clasificación lógica de los objetos de la base de datos y puede contener tablas, vistas, apodos, activadores, funciones y otros objetos
- Un nombre de esquema se utiliza como la parte más a la izquierda de las dos partes del nombre de objeto

Sinónimos

- Un sinónimo es un objeto de la base de datos que permite hacer referencia a otro objeto por un nombre diferente
- Los sinónimos pueden ser usados en la base de datos para transparencia de esquema
- Los sinónimos proporcionan un método alternativo de hacer referencia a una tabla existente
- Los sinónimos permiten a los usuarios acceder a un objeto de la base de datos ya sea con un nombre diferente o sin hacer referencia al dueño del objeto

Sinónimos ... 2

```
CREATE SYNONYM dept FOR Departamento
CREATE ALIAS dept FOR Departamento
```

Se puede hacer referencia a Departamento como dept

Resumen

- Se estudió cómo crear una tabla
- Se explicó cómo crear una tabla con restricciones
- Se presentaron los cinco tipos importantes de restricciones cuando se crean las tablas
- Se definieron los tipos de datos
- Se explicó cómo agregar y modificar columnas en una tabla
- Se explicó cómo eliminar tablas
- Se explicó cómo crear y eliminar vistas
- Se estudiaron alias/sinónimos

Unidad 4

Laboratorio de Creación y Administración de Objetos de la Base de Datos

Ejercicios de Laboratorio

Unidad 5:

Diccionario de Datos, Acceso y Seguridad de Base de Datos

Objetivos del Aprendizaje

- Explicar las vistas del catálogo del sistema
- Describir algunas de las vistas del catálogo usadas frecuentemente
- Explicar cómo consultar las vistas del catálogo del sistema
- Identificar algunas autorizaciones y privilegios básicos sobre la base de datos
- Explicar cómo administrar y controlar el acceso a la base de datos

Algunos Términos Preliminares

- Una tabla es un objeto de la base de datos que almacena datos
- Una vista es un objeto de la base de datos que captura datos de la tabla subyacente
- Las restricciones son algunos de los objetos creados en las tablas
- Los datos sobre datos se denomina diccionario de datos
- El diccionario de datos de una base de datos es almacenado en tablas del sistema
- Las tablas del sistema son tablas que almacenan la información de la metadata acerca de la base de datos, objetos de la base de datos, usuarios, privilegios, archivos de la base de datos, etc.

Vistas del Diccionario de Datos

- Para asegurar la seguridad e integridad, se niega el acceso a un usuario regular al diccionario de datos.
- La información almacenada en las tablas del diccionario de datos le puede ser útil a un usuario regular, por lo cual se crean vistas.
- Estas vistas permiten a los usuarios consultar la información almacenada en el diccionario de datos, no es posible modificar la información presente en ellas.
- Toda la información almacenada en las tablas del diccionario de datos esta en MAYUSCULAS.

Vistas del Catálogo del Sistema más Usadas

Sno	Vistas del Catálogo del Sistema
1	SYSCAT.TABLES
2	SYSCAT.COLUMNS
3	SYSCAT.KEYCOLUSE
4	SYSCAT.REFERENCES
5	SYSCAT.CHECKS
6	SYSCAT.INDEXES
7	SYSCAT.VIEWS

Consultar Vistas del Catálogo del Sistema

- Algunos de los requerimientos comunes de un desarrollador que busca cierta información del diccionario de datos son:
  - Lista de tablas disponibles en la base de datos
  - Lista de columnas disponibles en una tabla
  - Lista de restricciones en las columnas de una tabla
  - Lista de índices a los cuales tiene acceso el usuario
  - Lista de vistas que le interesan al usuario
  - Información general acerca de los usuarios

Lista de Tablas Disponibles en la Base de Datos

- Ejecutar el comando **LIST TABLES**
- Este comando devuelve la lista de tablas disponibles en el esquema actual del usuario
- Una consulta sobre una vista llamada SYSCAT.TABLES

```
SELECT tabname
FROM SYSCAT.TABLES
WHERE tabschema='USER NAME' AND type='T'
```

producirá un resultado similar

Columnas en syscat.tables

Nombre de Columna	Descripción
TABNAME	Nombre de Objeto
TABSCHEMA	Dueño del objeto
TYPE	Tipo del Objeto T- Tabla V- Vista A- Alias

Lista de Columnas Disponibles en una Tabla

- Ejecutar el siguiente comando:

```
DESCRIBE SELECT * FROM <nombretabla>
```

- Ejecutar una consulta sobre una vista llamada `syscat.columns`

```
SELECT tabname, colname, typename
FROM syscat.columns
WHERE  tabname ='nombre tabla' AND
tabschema='Nombre Esquema';
```

Columnas en syscat.columns

Nombre de Columna	Descripción
TABNAME	Nombre de la Tabla
TABSCHEMA	Dueño de la tabla
TYPENAME	Tipo de Dato Por ejemplo : VARCHAR,INTEGER,DATE y DECIMAL

Lista de Restricciones en las Columnas de una Tabla

- Para obtener información sobre: Claves primarias, Claves foráneas y restricciones unique(único), ejecute una consulta sobre una vista llamada `syscat.keycoluse`

```
SELECT tabschema, constname, tabname,
colname, colseq
FROM syscat.keycoluse
```

- Esta consulta listará el nombre de la tabla, el nombre del dueño de la tabla, las columnas que tengan restricciones, y el nombre de las restricciones

SYSCAT.REFERENCES y SYSCAT.CHECKS

Vista de Catálogo	Descripción
SYSCAT.REFERENCES	Esta vista lista las restricciones FOREIGN KEY (Clave Foránea). Nombre de tabla padre, nombre de tabla hija, nombre(s) de columna padre, nombre(s) de columna hija están disponibles en esta vista
SYSCAT.CHECKS	Esta vista lista las restricciones CHECK (Chequeo) definidas en la base de datos. Nombre de restricción, nombre de tabla, y la condición de verificación son definidas en esta vista
SYSCAT.COLCHECKS	Esta vista lista las restricciones CHECK (Chequeo) definidas en una tabla, junto con el nombre de columna en la que se define la condición

Lista de Índices Disponibles en la Base de Datos

La información acerca de índices está disponible en SYSCAT.INDEXES

```
SELECT indname, tabschema, tabname
FROM syscat.indexes WHERE tabschema ='SCOTT'; ;
```

INDNAME	TABSCHEMA	TABNAME
PK_CONST	SCOTT	DEPT1
PK_CONST1	SCOTT	EMP1
PK_DEPT	SCOTT	DEPT
PK_EMP	SCOTT	EMP
SQL010219174830540	SCOTT	EMPLOYEE

Lista de Vistas Disponibles en la Base de Datos

La información acerca de las vistas está disponible en SYSCAT.VIEWS.

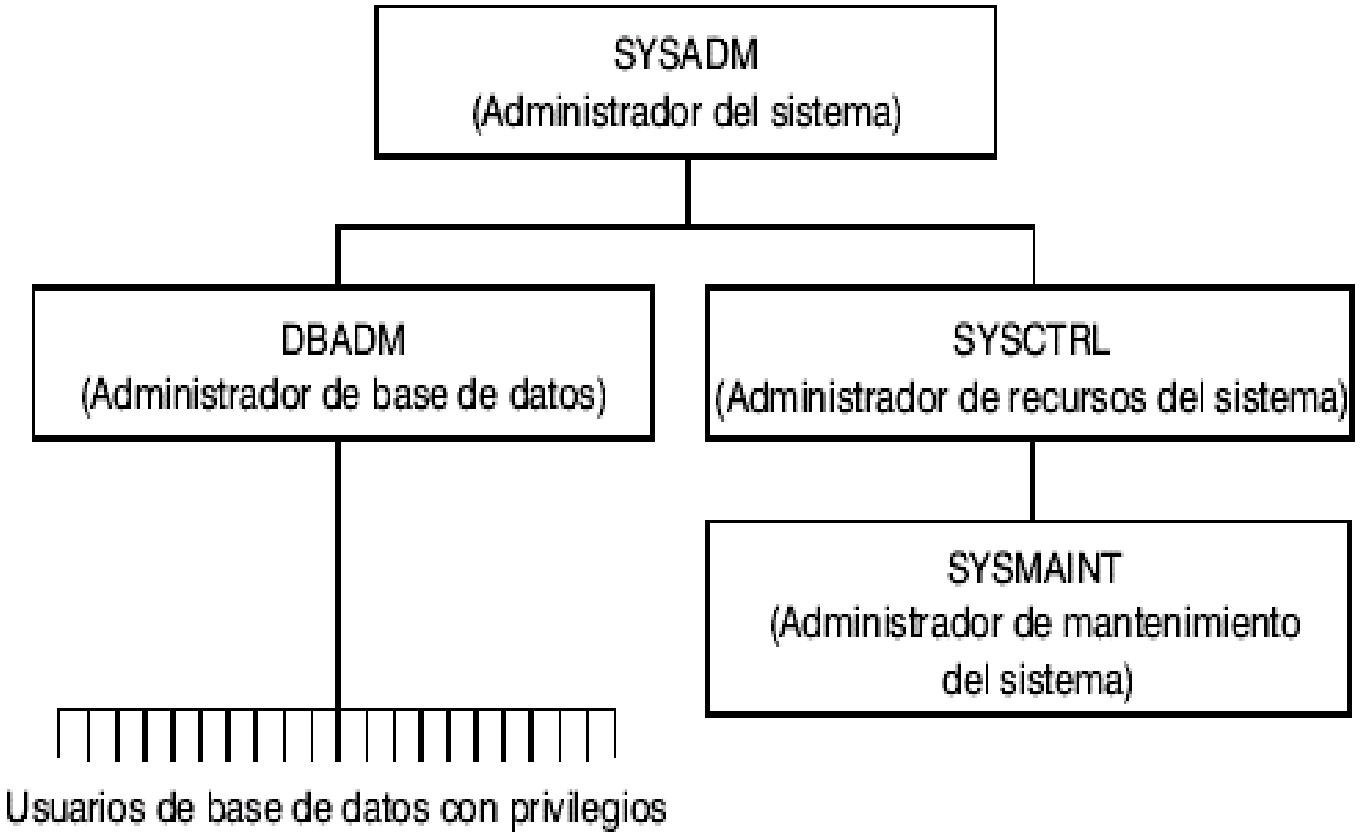
```
SELECT VIEWNAME, VIEWNAME
FROM SYSCAT.VIEWS;

VIEWNAME        TEXT
-----------     ------
V               select emp1.empno,empname,
                basic, da,total from
                emp1, sal1
```

Acceso y Seguridad de Base de Datos

- El acceso a los datos necesita ser controlado
- DB2 UDB proporciona herramientas administrativas para controlar autorizaciones y privilegios sobre objetos de la base de datos
- Las *autorizaciones* de base de datos otorgan acciones sobre una base de datos como un todo.

Autorizaciones de Base de Datos



Privilegios de la Base de Datos

- Involucran acciones sobre objetos específicos dentro de la base de datos
- Le da a un usuario o a un grupo el derecho de acceder a un objeto específico de la base de datos de una manera específica
- Los privilegios y autorizaciones controlan de manera conjunta el acceso a la base de datos
- Los usuarios pueden acceder solamente a aquellos objetos para los cuales tienen la autorización apropiada

Tipos de Privilegios de la Base de Datos

SELECT

Otorga el privilegio de recuperar filas de la tabla, ver o crear vistas sobre la tabla

INSERT

Otorga el privilegio de insertar filas en una tabla

UPDATE

Otorga el privilegio de usar la sentencia UPDATE en una tabla

DELETE

Otorga el privilegio de eliminar filas de una tabla

Tipos de Privilegios de la Base de Datos ... 2

INDEX

Otorga el privilegio de crear un índice en una tabla.

REFERENCES

Otorga el privilegio de crear y eliminar una clave foránea que haga referencia a la tabla padre

ALTER

- Otorga el privilegio de agregar columnas a la definición de una tabla base, o
- Crear o eliminar una restricción PRIMARY KEY o UNIQUE en una tabla base

Tipos de Privilegios de la Base de Datos ... 3

ALL o **ALL PRIVILEGES**

Otorga todos los privilegios apropiados, excepto CONTROL, en la vista de la tabla base

CONTROL

Otorga todos los privilegios apropiados en la lista, es decir, ALTER, CONTROL, DELETE, INSERT, INDEX, REFERENCES, SELECT y UPDATE para tablas base y CONTROL, DELETE, INSERT, SELECT y UPDATE para vistas.

Otorgar Privilegios

- La sentencia GRANT permite a un usuario autorizado otorgar privilegios
  - ü A uno o más usuarios en una sentencia
  - ü A PUBLIC

- Un nombre de autorización puede ser tanto un usuario individual como un grupo

```
GRANT SELECT ON EMPLOYEE TO USER SCOTT;
GRANT SELECT ON EMPLOYEE TO GROUP SCOTT;
```

Vistas del Catálogo del Sistema y Privilegios

Vistas del Catálogo del Sistema	Privilegios
SYSCAT.DBAUTH	Registra las autorizaciones de base de datos que ostentan los usuarios.
SYSCAT.TABAUTH	Contiene los privilegios de tablas y vistas
SYSCAT.COLAUTH	Contiene una o varias filas para cada usuario o grupo a los que se ha otorgado un privilegio a nivel de columna, que indican el tipo de privilegio y si se puede otorgar o no.
SYSCAT.INDEXAUTH	Contiene los privilegios de índices
SYSCAT.SCHEMAAUTH	Contiene los privilegios de esquemas

Revocar Privilegios

- La sentencia REVOKE permite a los usuarios autorizados revocar los privilegios otorgados a otros usuarios
- Para revocar privilegios, el usuario debe tener la autoridad **DBADM**, la autoridad **SYSADM** o el privilegio **CONTROL** sobre dicho objeto

  Por ejemplo:

```
REVOKE SELECT ON EMPLOYEE FROM USER SCOTT;
REVOKE SELECT ON EMPLOYEE FROM GROUP SCOTT;
```

Administrar el Acceso a la Base de Datos

- Limitar los usuarios a sólo algunas columnas de una tabla
- Por ejemplo:

```
CREATE VIEW detalleEmp
AS SELECT EmpNo, Enombre, trabajo,
 fechaingreso
FROM Empleado;
GRANT SELECT ON detalleEmp TO USER SCOTT;
```

Administrar el Acceso a la Base de Datos ... 2

- Limitar el acceso a sólo un subconjunto de filas específico en una tabla

```
CREATE VIEW EmpProduction
AS SELECT EmpNo, Enombre, trabajo, fechaingreso,
Dept
FROM Empleado
WHERE Dept='Production';
GRANT SELECT ON EmpProduction TO GROUP
Production;
```

Resumen

- Explicar las vistas del catálogo del sistema
- Describir algunas de las vistas del catálogo usadas frecuentemente
- Explicar cómo consultar las vistas del catálogo del sistema
- Identificar algunas autorizaciones y privilegios básicos sobre la base de datos
- Explicar cómo administrar y controlar el acceso a la base de datos.

Unidad 6

Laboratorio de Diccionario de Datos

Ejercicios de Laboratorio

Unidad 7

Programación de Estructura, UDT y UDF

Objetivos del Aprendizaje

- Explicar los conceptos de:
  - ü Tipos de Datos Definidos por el Usuario (User Defined Data-Types - UDT)
  - ü Funciones Definidas por el Usuario (User Defined Functions - UDF)
- Discutir cómo implementar UDT

Preliminares

- El DB2 UDB proporciona soporte para el desarrollo de aplicaciones en una variedad de lenguajes tales como C, C++, Java, etc
- El soporte para el desarrollo de aplicaciones viene con el paquete DB2 Application Development Client
- C, C++, Java (Embedded SQL en Java usando SQLJ), COBOL y FORTRAN, etc.
- DB2 proporciona además soporte para Perl y Java (programación de base de datos interpretada vía JDBC)
- Con DB2, los programadores pueden definir su propio conjunto de funciones asociadas con los tipos de dato
- Estas funciones son conocidas como UDT y UDF

Objetos Predefinidos

- Soporta tipos de datos primitivos como datos numéricos y de cadenas
- Soporte para Objetos Largos Binarios (Binary Large Objects - BLOBs)
- Objetos Largos de Caracteres (Character Large Objects - CLOBs)
- Objetos Largos de Caracteres de Dos Bytes (Double-Byte Carácter Long Objects)

Tipos de Datos – Definidos por el Usuario

Objetos Definidos por el Usuario

- Los usuarios pueden definir los objetos usando tipos de datos predefinidos
- Permiten al usuario traducir objetos del sistema en objetos de base de datos, definiendo la semántica y el comportamiento del objeto
- Los usuarios pueden definir nueva semántica que sea específica para los objetos definidos por el usuario

Necesidad del UDT

- Almacenando Objetos basados en el Contexto
  - ü El contexto para los datos pertenecientes a algún campo especial en una aplicación
  - ü UDTs proporciona el contexto para los datos
- Tipado Fuerte (Strong Typing)
  - ü Las aplicaciones son más confiables
- Basado en tipos predefinidos
- Una vez definida, UDT se comporta de manera diferente de los tipos de datos predefinidos
- El tipado fuerte es una ventaja proporcionada por los UDTs

Tipos de UDT

- Tipos Distinct
  - Pueden extender los tipos básicos de dato y tener nueva semántica definida sobre ellos.
  - Estos nuevos Tipos Definidos por el Usuario forman la base de los tipos de datos distinct.
- Tipos Estructurados
  - Una agregación de algunos de los tipos básicos.
  - Similar a las estructuras C.
- Tipos de Referencia
  - Para cada tipo estructurado que crea el usuario, DB2 crea automáticamente un tipo referencia.

Definir Tipos de Dato Distinct Definidos por el Usuario

- Pueden ser creados ya sea usando SQL o usando al asistente proporcionado por DB2
- Usando la sentencia **SQL CREATE DISTINCT TYPE**

```
CREATE DISTINCT TYPE <<Distinct Type Name>>
AS <<Source Basic Data Type>> WITH COMPARISONS
```

Ejemplos de Tipos de Dato Distinct Definidos por el Usuario

```
CREATE DISTINCT TYPE GALON AS DECIMAL (9,2)
  WITH COMPARISONS;

CREATE DISTINCT TYPE ONZAS_LIQUIDAS
  AS DECIMAL (9,2) WITH COMPARISONS;

CREATE DISTINCT TYPE PINT AS DECIMAL (9,2)
  WITH COMPARISONS;

CREATE DISTINCT TYPE LITRO AS DECIMAL (9,2)
  WITH COMPARISONS;
```

Usar UDT en Columnas de Tabla

```
CREATE TABLE FuelSalesData (
    ItemId      INTEGER,
    ItemName    VARCHAR (20),
    SalesVolume     GALON
);

CREATE TABLE BeverageSalesData (
    BeverageId INTEGER,
    BeverageName VARCHAR (20),
    SalesVolume   ONZAS_LIQUIDAS
);
```

Convertir Tipos Básicos UDT

```
CREATE FUNCTION source-basic-data-type
(distinct-type-name)
 RETURNS source-basic- data-type

CREATE FUNCTION distinct-type-name (source-
basic-data-type) RETURNS distinct-type-name
```

Las funciones anteriores permiten al usuario hacer algunas comparaciones como las siguientes:

```
SELECT SalesVolume
FROM FuelSalesData
WHERE SalesVolume=GALLON(1000)
```

Operadores de Comparación para UDT

- Menor que (<)
- Mayor que (>)
- Menor o igual que (<=)
- Mayor o igual que (>=)
- No igual a (<>)
- Igual a (=)

Definir Tipos Estructurados

```
CREATE TYPE Estudiante_t AS (
  IDNO INT,
   NOMBRE VARCHAR (50),
   FECHANACIMIENTO DATE)
 REF USING INT
 INSTANTIABLE
 MODE DB2SQL;
```

Definir Tipos Estructurados… 2

Se pueden crear tipos sub-estructurados a partir del Tipo Estructurado padre

```
CREATE TYPE Asistente_inves_t
UNDER Estudiante_t AS (
 TopicIdInvest INTEGER,
 Topicinvestigacion VARCHAR(100)
 )
INSTANTIABLE
MODE DB2SQL;
```

Definir Tipos Estructurados... 3

```
CREATE TABLE Estudiante OF Estudiante_t
(REF IS Oid USER GENERATED);

CREATE TABLE Estudianteinvestigador
OF Asistente_inves_t
(REF IS Oid USER GENERATED);
```

Operaciones sobre UDT

```
INSERT INTO Estudiante
(Oid, IDNO,Nombre, fechanacimiento)
VALUES(Estudiante_t      (1),1,      'John',
  '01/02/1999');

INSERT INTO Estudianteinvestigador
(Oid,    IDNO,    nombre,    fechanacimiento,
  TopicIdInvest, Topicinvestigacion)
VALUES  (Asistente_inves_t(1),    1,  'John',
  '01/02/1999', 1,'Geografia');
```

Definir Tipos Estructurados… 4

• El tipo de dato estructurado puede ser usado además como un tipo de columna en una definición de tabla. La sintaxis para esto es como sigue:

1,- Creación del tipo estructurado:

```
CREATE TYPE Direccion_t AS
 (calle VARCHAR(30),
 numero CHAR(15),
 ciudad VARCHAR(30),
 estado VARCHAR(10))
 REF USING INT
                    INSTANTIABLE
                    MODE DB2SQL;
```

Definir Tipos Estructurados… 5

2.- Creación de la tabla trabajador:

```
create table trabajador(
idtra int,
Nombre VARCHAR(20),
Edad int,
salario double,
direccion direccion_t);
```

Trabajador

idtra	nombre	edad	salario	direccion			
				Calle	Numero	Ciudad	Estado

Definir Tipos Estructurados… 6

- Operación de inserción:

```
 insert into trabajador
(idtra,nombre,edad,salario,direccion)
VALUES
(1,'Juana
   Peña',22,12345.87,Direccion_t()..calle('Av
   Acacias')..numero('12')..ciudad('Caracas')..estad
   o('Dtto. Cap'));
```

Definir Tipos Estructurados… 7

- Operación de consulta:

Buscar los empleados que viven en la calle "Av Acacias"

```
select idtra,nombre,salario,
direccion..calle,direccion..ciudad,direccion..estado
from trabajador
where direccion..calle ='Av Acacias';
```

Definir Tipos Estructurados… 8

- Operación de actualización:

```
UPDATE trabajador
SET direccion = direccion..numero('15')
WHERE direccion..calle='Av Acacias';
```

UDF

- Funciones definidas por el usuario.
- DB2 proporciona la facilidad para que los desarrolladores de aplicaciones creen sus propias funciones, estas se llaman UDF.
- Cuando desarrollamos una aplicación personalizada, podemos requerir funciones especiales aparte de las funciones estándar. Este requerimiento puede realizarse desarrollando UDF en DB2.

Resumen

- Se explicaron los conceptos de UDT.
- Se aprendió acerca de como implementar UDT.

Unidad 8

Laboratorio de Programación de Estructura, UDT y UDF

Ejercicios de Laboratorio