

```
... object to mirror  
mirror_mod.mirror_object
```

```
operation == "MIRROR_X":  
    mirror_mod.use_x = True  
    mirror_mod.use_y = False  
    mirror_mod.use_z = False  
operation == "MIRROR_Y":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = True  
    mirror_mod.use_z = False  
operation == "MIRROR_Z":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = False  
    mirror_mod.use_z = True
```

```
selection at the end -add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_ob))  
mirror_ob.select = 0  
= bpy.context.selected_objects  
data.objects[one.name].select  
print("please select exactly
```

```
-- OPERATOR CLASSES --
```

```
types.Operator):  
    X mirror to the selected  
    object.mirror_mirror_x"  
    mirror X"
```

# SISTEMAS GESTORES DE BASES DE DATOS

COLEGIO CALASANZ. 1º DAW.

# CONTENIDOS

- BBDD
  - Concepto
  - Características
  - Objetivos
  - Ventajas
  - Inconvenientes
  - Tipos de BBDD
- Sistemas gestores de BBDD
  - Conceptos
  - Componentes
  - Servicios proporcionados
  - Niveles de abstracción
  - Lenguajes utilizados

## BBDD (I). CONCEPTO

- Colección de datos relacionados entre sí.
- Colección de información organizada que pertenece a un mismo contexto.
- Datos almacenados formalmente. Estos datos son controlados de forma central para intentar servir a múltiples y diferentes aplicaciones.
- Ej: usuarios de un gimnasio, datos bancarios de una entidad/clientes, comentarios de una página Web, etc.

## BBDD (II). CARACTERÍSTICAS

- Colección de datos.
- Los datos están relacionados según un modelo que refleja las restricciones y relaciones del mundo real.
- Datos compartidos por varios usuarios y/o aplicaciones.
  - Los datos deben ser independientes de esas aplicaciones.
- La definición y descripción han de ser únicas (los datos solo pueden tener una definición).
  - Se deben almacenar junto con los datos.
- Los datos siempre tendrán que conservar la integridad y seguridad.

## BBDD (III). OBJETIVO

- Las aplicaciones puedan acceder a los datos sin necesidad de conocer cómo están almacenados los mismos.
- Se consigue utilizando un **esquema conceptual**.
  - Cualquier desarrollador puede acceder a los datos si conoce este esquema.

## BBDD (IV). VENTAJAS

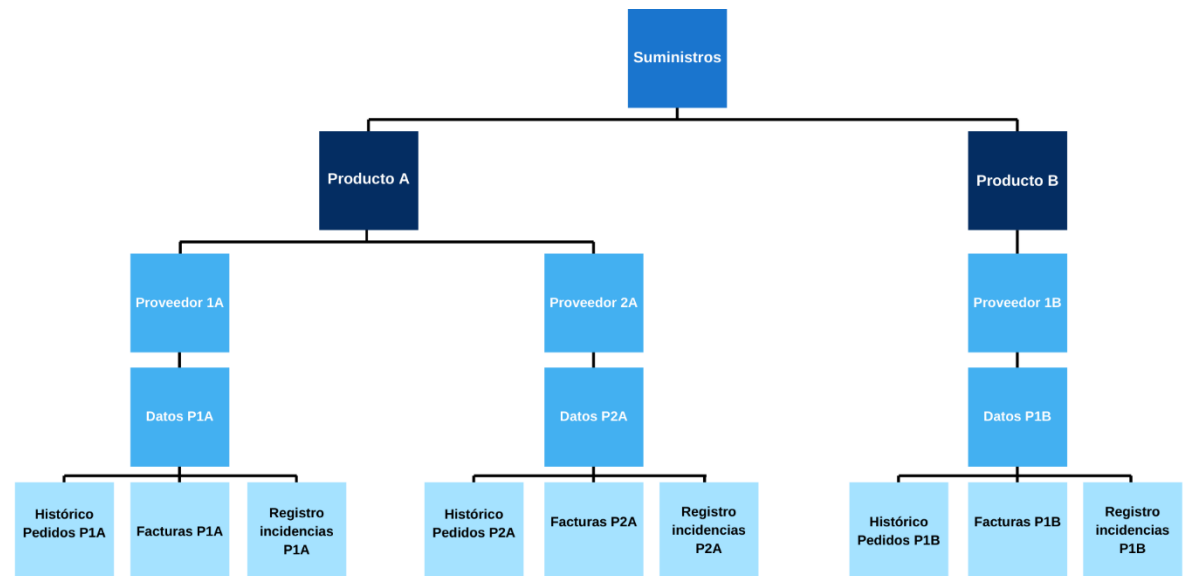
- Menor redundancia de datos.
- Más información sobre los datos.
- Control sobre la consistencia de los datos.
- Compartición de datos (conurrencia).
- Mejora en la integridad de los datos.
- Mejora en la seguridad.
- Mejora en la accesibilidad a los datos.
- Mejora en los servicios de copias de seguridad.
- Menor coste económico con el paso del tiempo.

## BBDD (V). INCONVENIENTES

- Gran cantidad de espacio en disco y de memoria RAM.
- Vulnerable a los fallos porque es un sistema centralizado.
- Coste económico por el gestor como por el equipamiento adicional.
- Coste de conversión por la migración de datos.

## BBDD (VI). TIPOS DE BBDD (I): JERÁRQUICO

- Es el más antiguo
- Las relaciones entre los registros forman una estructura en árbol invertido
- Solo un nodo padre
- Ej: Registro de Windows





## BBDD (VI). TIPOS DE BBDD (I): JERÁRQUICO - VENTAJAS

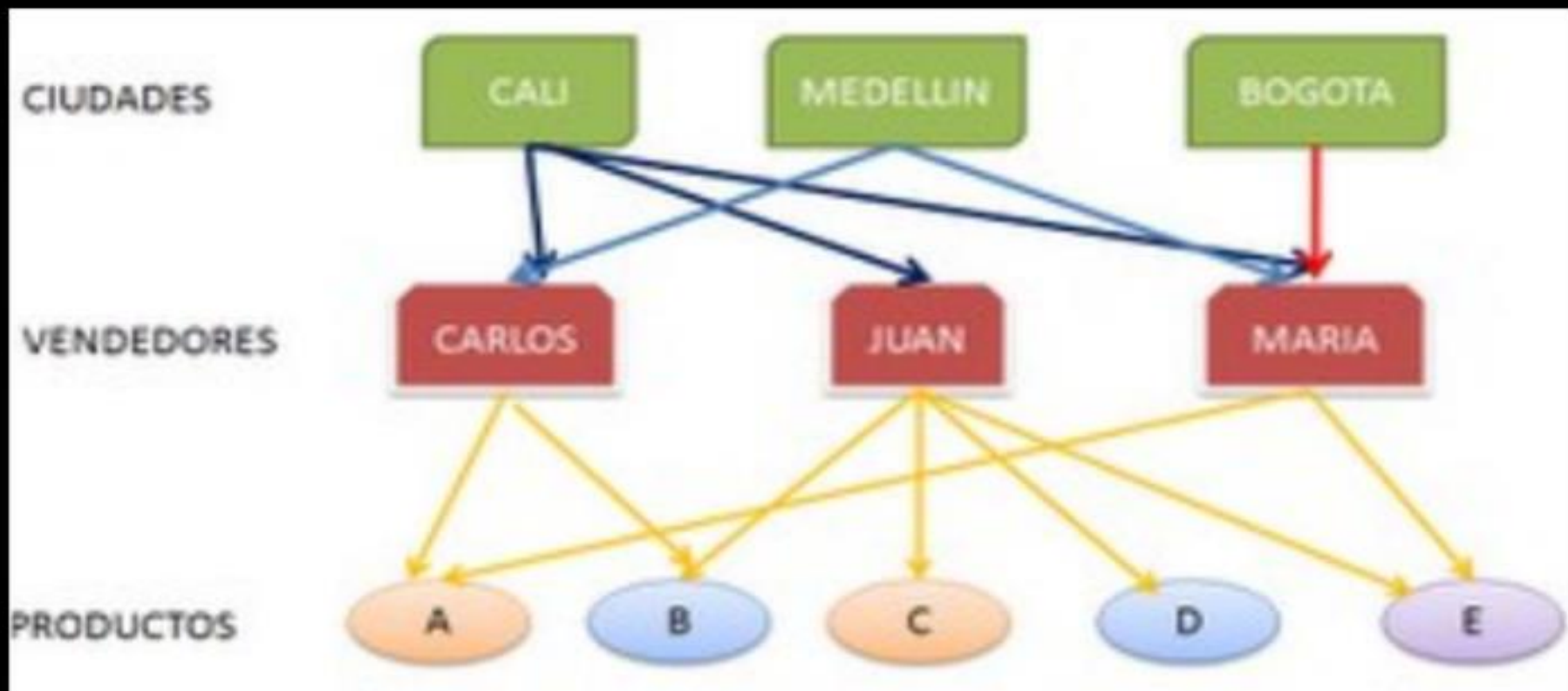
- La navegación dentro del árbol es bastante rápida
- Es una estructura fácil de comprender
- Permite compartir información de forma globalizada dentro de una empresa
- Mantiene la independencia de datos

## BBDD (VI). TIPOS DE BBDD (I): JERÁRQUICO - INCONVENIENTES

- Escasa independencia entre los registros (nodos). Se debe navegar por los registros padres hasta llegar a un nodo.
- Mucha redundancia al tener que duplicar un nodo si tiene varias relaciones
- Esto genera problemas de integridad y coherencia de datos
- Es difícil de diseñar y de modificar por su rigidez. Se debe tener un conocimiento muy amplio de la información y las relaciones entre sí.

## BBDD (VII). TIPOS DE BBDD (II): EN RED

- Mejoras con respecto al modelo jerárquico.
- Mayor independencia y flexibilidad de los datos.
- Existen varios caminos entre los datos.

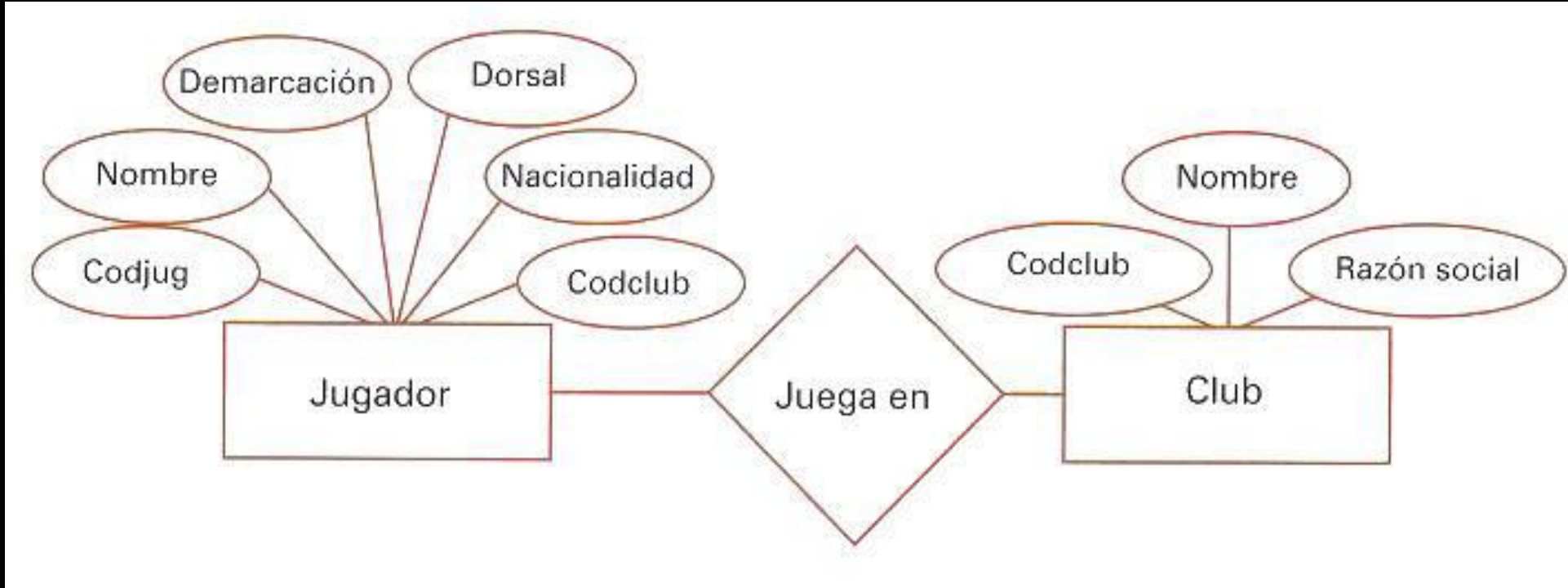


## BBDD (VIII). TIPOS DE BBDD (III): RELACIONAL (I)

- El modelo más utilizado en la actualidad.
- Información representada en forma de entidades y relaciones entre ella.
- Ej: MySQL, MariaDB, Oracle.

## BBDD (IX). TIPOS DE BBDD (III): RELACIONAL (II)

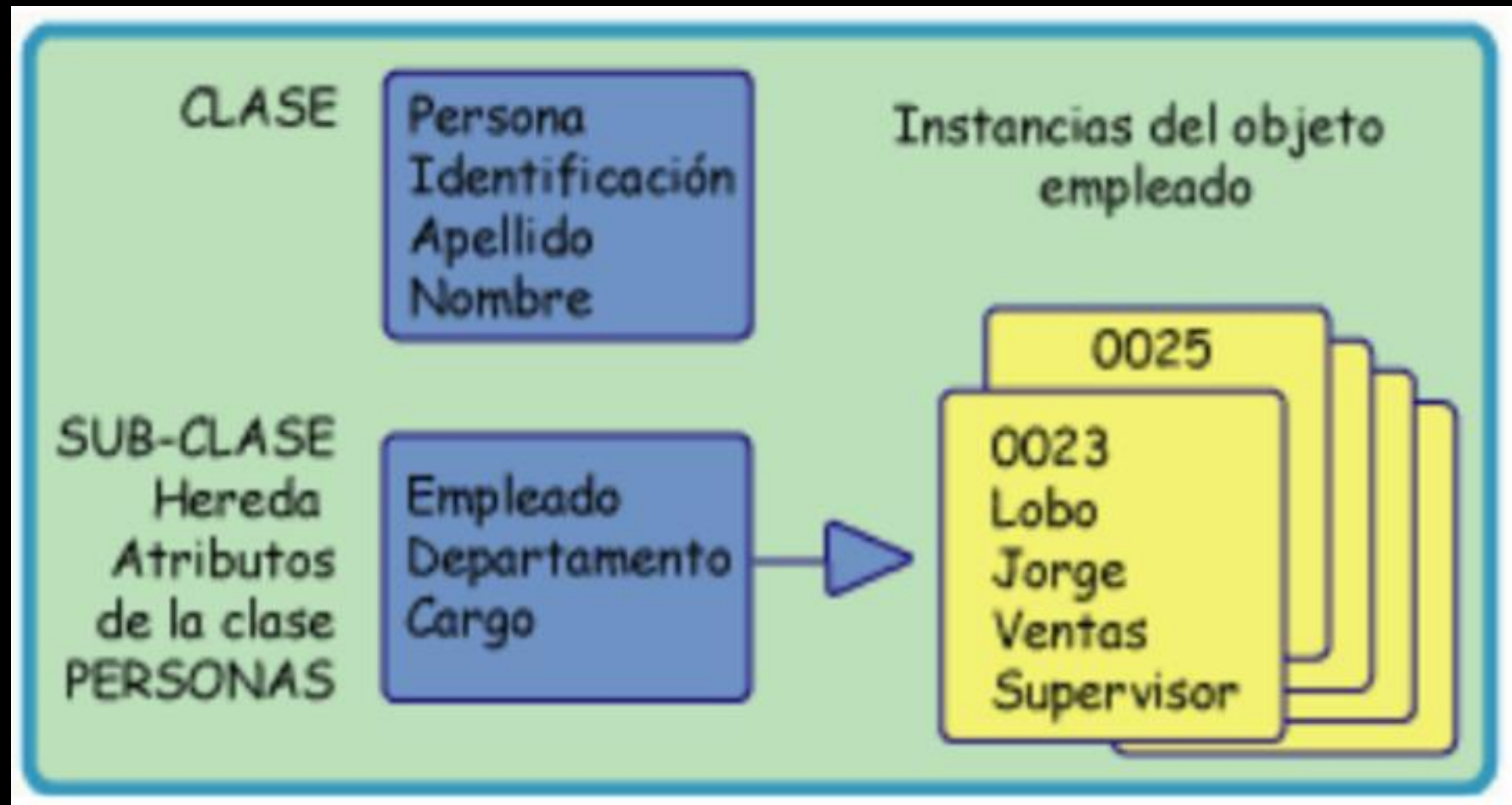
- Entidad: representa un objeto o un concepto del mundo real.
  - Ej: un jugador de un club deportivo
- Atributo: representa alguna propiedad de interés de una entidad.
  - Ej: nombre del empleado
- Relación: describe la interacción entre dos o más entidades.



## BBDD (X). TIPOS DE BBDD (IV): ORIENTADO A OBJETOS

- Se creó para dar respuesta a la necesidad de almacenamiento de: imágenes, documentos, vídeo, etc.
- Aplica el concepto de la programación orientada a objetos.
- Ej: Objectivity





---

## BBDD (XI). TIPOS DE BBDD (V): OBJETO-RELACIONAL

- Híbrido de los dos modelos.
- Concepto relacional que se adapta a las nuevas tendencias.
- Ej: Oracle, Microsoft SQL Server, IBM DB2, IBM Informix, PostgreSQL.

## BBDD (XII). TIPOS DE BBDD (VI): OTROS MODELOS

- Otros modelos utilizados:
  - **Documental**: almacenamiento e indexación de grandes documentos.
  - **Orientado al documento**: gestión datos de documentos estructurados (XML, JSON). Ej: MongoDB.
  - **Multidimensional**: algoritmos de inteligencia artificial para tratar la información.
  - **Deductivo**: almacena reglas de inferencia para generar deducciones a partir de unos datos.
  - **Grafos**. Esta especializada en grafos y permite relaciones que las bbdd relacionales no permiten.

## SISTEMA GESTOR DE BBDD (I). CONCEPTO

- Un sistema gestor de BBDD (SGBD) es el conjunto de herramientas que facilitan, en la BBDD:
  - Las consultas
  - El uso
  - La actualización
- Un SGBD es una aplicación que permite a los usuarios:
  - Definir, crear y mantener una BBDD
  - Proporcionar acceso controlado a la BBDD

## SISTEMA GESTOR DE BBDD (II). COMPONENTES (I)

- Hardware

- Máquinas donde se almacena la información.
- Volúmenes de almacenamiento secundario donde reside la BBDD: discos, cintas, etc.
- Dispositivos asociados: unidad de control, tarjetas, memoria, etc.

- Software

- Sistema gestor de BBDD, que permite el manejo de la BBDD.
- Entorno gráfico, estadísticas, backups, tareas programadas, etc.

## SISTEMA GESTOR DE BBDD (III). COMPONENTES (II)

- Usuarios
  - Personas que manipulan los datos.
  - Tipos:
    - **Administrador**: diseño físico, implementación, mantenimiento y gestión de usuarios/aplicaciones de la BBDD.
    - **Diseñador**: diseño lógico de la BBDD. Identifican los datos, relaciones de datos y las restricciones..
    - **Programador**: implementa los programas que acceden a los datos.
    - **Usuario final**: clientes de la BBDD.

## SISTEMA GESTOR DE BBDD (IV). COMPONENTES (III)

- Datos
  - Datos que se necesitan almacenar junto con los metadatos.
  - Metadatos: describen lo que se almacena en la BBDD.

## SISTEMA GESTOR DE BBDD (V). SERVICIOS PROPORCIONADOS (I)

- Definir la BBDD mediante el "lenguaje de definición de datos (*DDL – Data Description Language*)".
  - DDL permite especificar la estructura, el tipo de los datos y sus restricciones.
- Permitir inserción, actualización, eliminación y consulta de datos mediante el "lenguaje de manejo de datos (*DML – Data Manipulation Language*)".



## SISTEMA GESTOR DE BBDD (VI). SERVICIOS PROPORCIONADOS (II)

- Proporcionar acceso controlado a la BBDD mediante:
  - Un **sistema de seguridad**, con el "lenguaje de control de datos (*DCL – Data Control Language*)".
  - Un **sistema de integridad**, que mantiene la integridad y la consistencia de los datos.
  - Un **sistema de control de concurrencia**, que permite el acceso compartido a la BBDD.
  - Un **diccionario de datos** accesible por el usuario que contiene la descripción de los datos de la BBDD.
  - Un mecanismo de vistas. Las vistas se comportan como tablas virtuales.

## SGBD (VII). ARQUITECTURA DEL SGBD. NIVELES DE ABSTRACCIÓN(I)

- Los sistemas de bases de datos tienen una arquitectura de tres niveles.
  - Objetivo: separar la BBDD física de los programas de aplicación y de los programas que manejan los usuarios.
- Niveles:
  - Nivel físico:
    - Esquema interno que describe la estructura física de la BBDD.
    - El esquema describe los detalles para el almacenamiento y los métodos de acceso.

## SGBD (VIII). ARQUITECTURA DEL SGBD. NIVELES DE ABSTRACCIÓN(II)

- Nivel conceptual:
  - Esquema conceptual que describe la estructura de la BBDD.
  - El esquema oculta los detalles de las estructuras físicas.
  - Se definen las entidades, los atributos, relaciones, las operaciones de los usuarios y las restricciones.
- Nivel externo:
  - Varios esquemas externos o vistas de usuario.
  - Cada esquema describe la parte de la BBDD que interesa a un grupo de usuarios determinado, ocultando el resto de la BBDD.

## SGBD (IX). ARQUITECTURA DEL SGBD. NIVELES DE ABSTRACCIÓN(III)

- Objetivo a conseguir: independencia de datos.
  - Capacidad para modificar el esquema de un nivel sin tener que modificar al nivel superior.
  - Tipos de independencia de datos:
    - Independencia lógica: capacidad para modificar el esquema conceptual sin tener que modificar los esquemas externos ni los programas de aplicación.
      - Ej: se reduce la BBDD eliminando una entidad, los esquemas externos que no se refieran a esa entidad no se verán afectados.
    - Independencia física: capacidad para modificar el esquema físico sin tener que modificar el esquema conceptual ni los esquemas externos.
      - Ej: se reorganizan algunos ficheros físicos para mejorar su rendimiento sin afectar al esquema conceptual.
      - Es más fácil conseguir la independencia física que la independencia lógica.

## SGBD (X). LENGUAJES UTILIZADOS POR EL SGBD

- Lenguaje de definición de datos (DDL)
  - Define el esquema conceptual y el esquema interno de la BBDD
  - Modifica el esquema de la BBDD. Ej: CREATE, ALTER Y DROP.
- Lenguaje de manejo de datos (DML)
  - Permite manipular los datos, realizar consultas, insertar, etc.
  - Modifica el estado de la BBDD. Ej: INSERT, UPDATE, DELETE Y SELECT.
- Lenguaje de control de datos (DCL)
  - Permite definir la seguridad, las restricciones, etc.
  - Controla el acceso a los datos de la BBDD. Ej: GRANT Y REVOKE.

```
... object to mirror  
mirror_mod.mirror_object
```

```
operation == "MIRROR_X":  
    mirror_mod.use_x = True  
    mirror_mod.use_y = False  
    mirror_mod.use_z = False  
operation == "MIRROR_Y":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = True  
    mirror_mod.use_z = False  
operation == "MIRROR_Z":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = False  
    mirror_mod.use_z = True
```

```
selection at the end -add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_ob))  
mirror_ob.select = 0  
= bpy.context.selected_objects  
data.objects[one.name].select  
print("please select exactly
```

```
-- OPERATOR CLASSES -----
```

```
types.Operator):  
    X mirror to the selected  
    object.mirror_mirror_x"  
    mirror X"
```

# SISTEMAS GESTORES DE BASES DE DATOS

COLEGIO CALASANZ. 1º DAW.