



ARQUITECTURA DE LAS BASES DE DATOS

Bases de Datos I
Sistemas - ETITC
2024



**Escuela Tecnológica
Instituto Técnico Central**
Establecimiento Público de Educación Superior



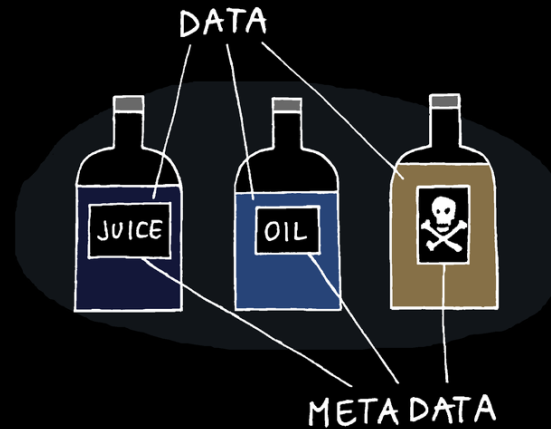
CONTENIDO

- Datos vs Datasets vs BD vs Motores de BD vs Sistema de BD vs SGBD
- Arquitectura de las BD: Una sola, dos, o tres capas (ANSI/SPARC)
 - Nivel externo, interno, y conceptual
- Modelos de datos
- Tipos de diseños
 - Conceptual
 - Relacional
 - Físico

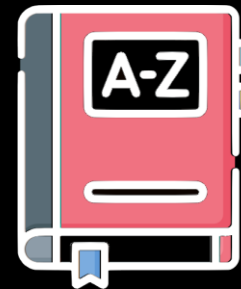
DATOS \neq META DATOS



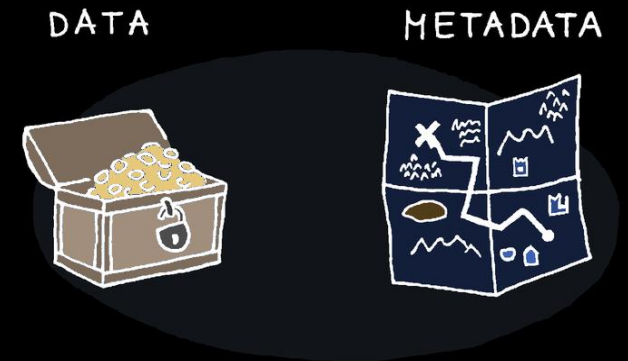
SO YOU'RE TELLING ME THIS WILL TURN INTO VALUE?



- Descriptiva
- Informativa
- Administrativa



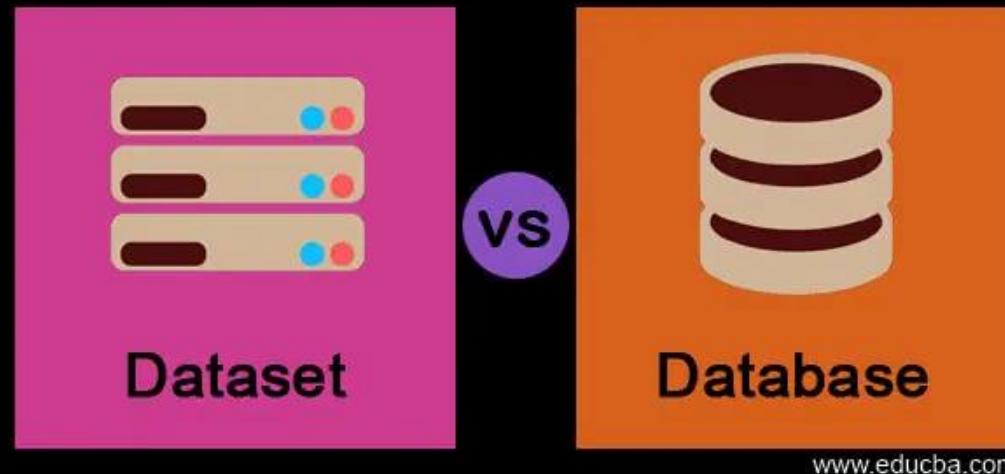
DICCIONARIO



DATA VS METADATA - DO YOU KNOW THE DIFFERENCE?

DATASETS \neq DATABASE

Colección de datos organizados que suelen no cambiar con el tiempo



DATASET VS DATABASE

DATASET VS DATABASE (KEY DIFFERENCES)

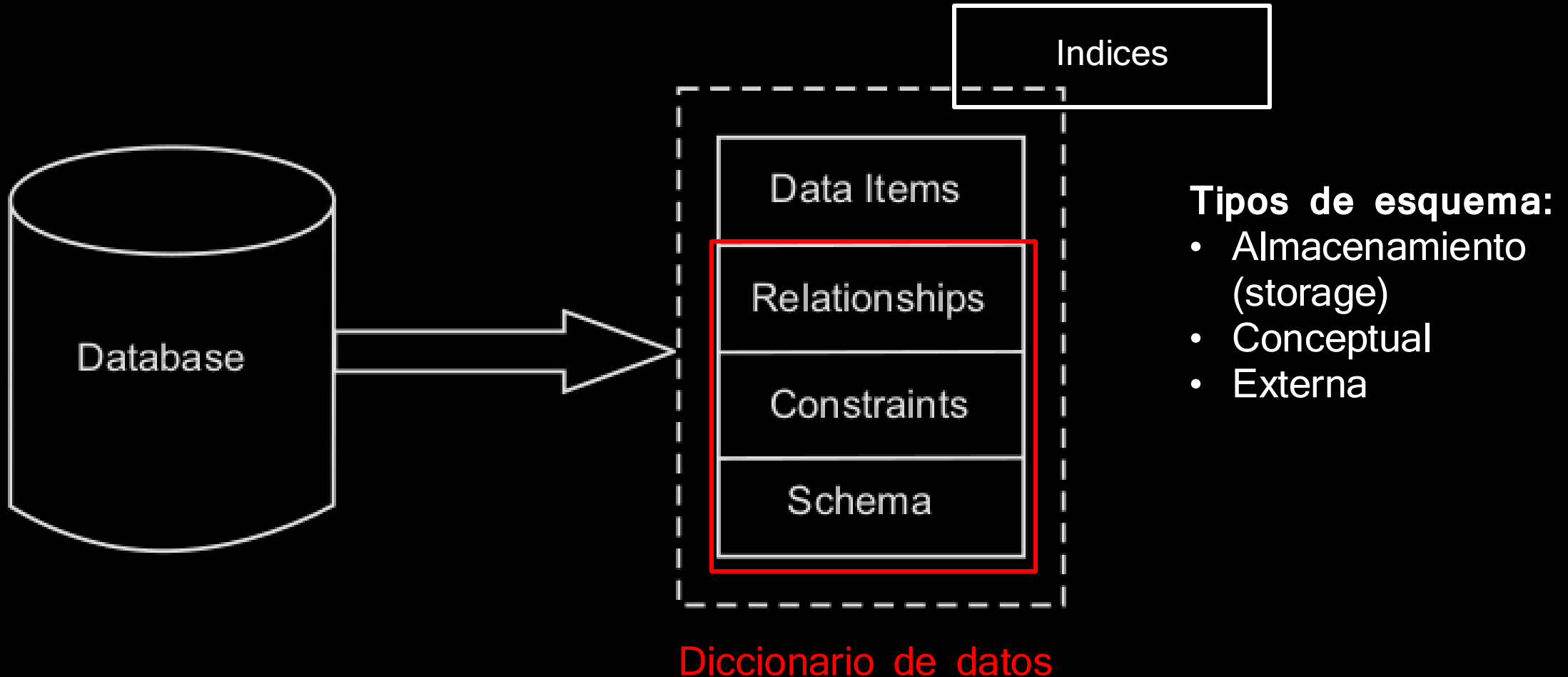
Las BDs son organizadas por:

- Campos (fields)
- Registros (records)
- Archivos (files)

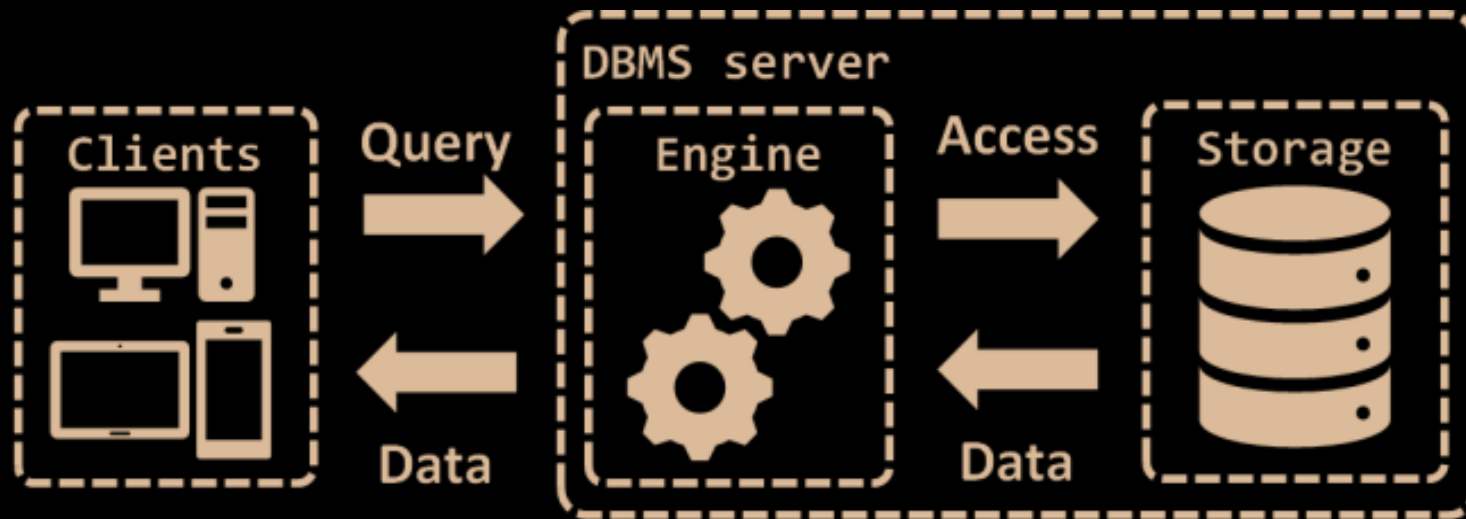


Colección de conjuntos de datos almacenados y administrados electrónicamente

COMPONENTES DE LAS BD



MOTORES DE BD \neq SGBD



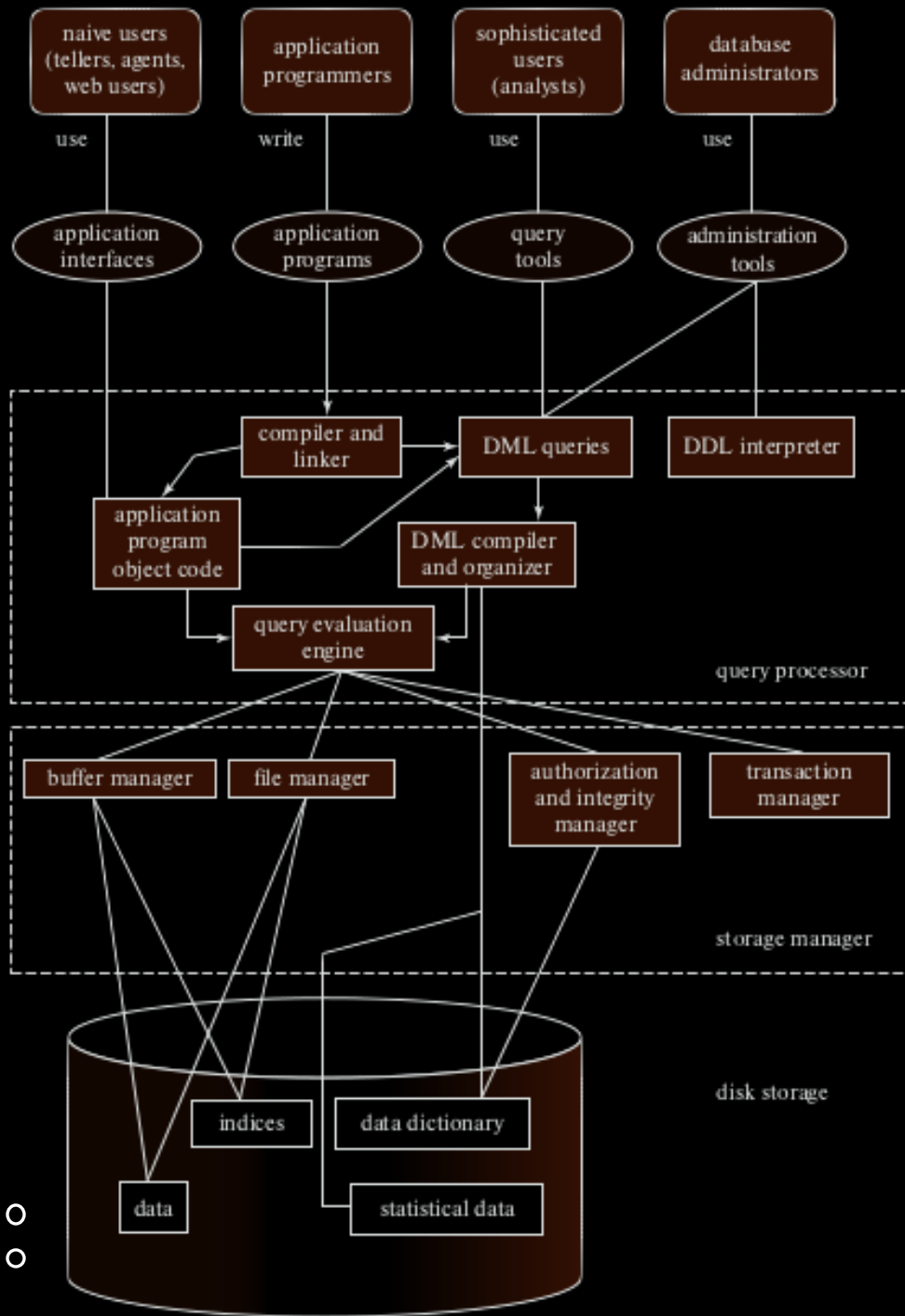
Gerente de almacenamiento

- Transacciones
- Autorización
- Manejo de archivos
- Cargar datos (buffer) a la memoria

Procesador de consultas

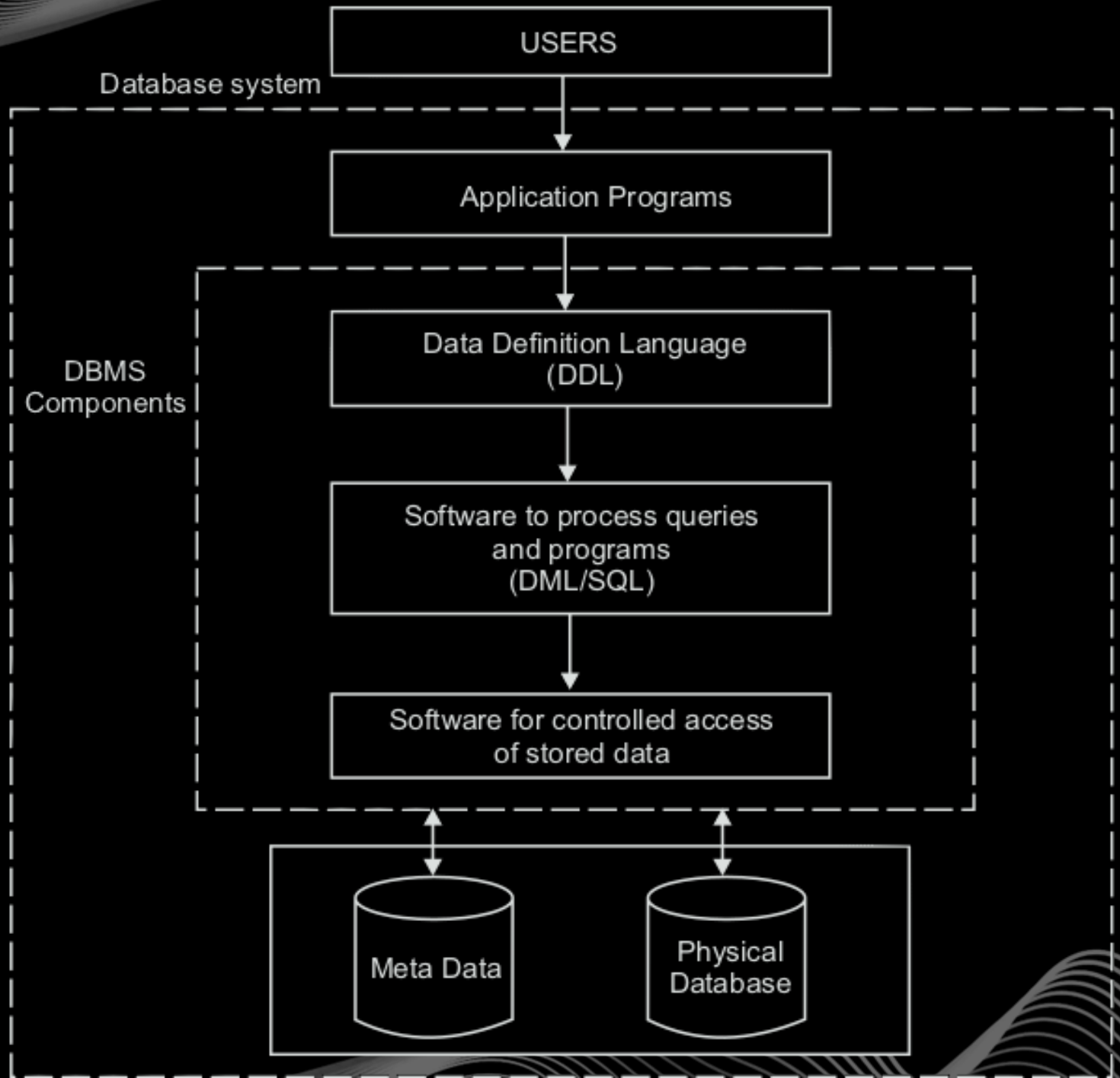
- Interprete deñ Data Definition Language (DDL)
- Compilador del Data Manipulation Language (DML)
- Evaluador del lenguaje de consulta (SQL - mySQL)

What is a Database System (DBMS)? [Dev Concepts #36]

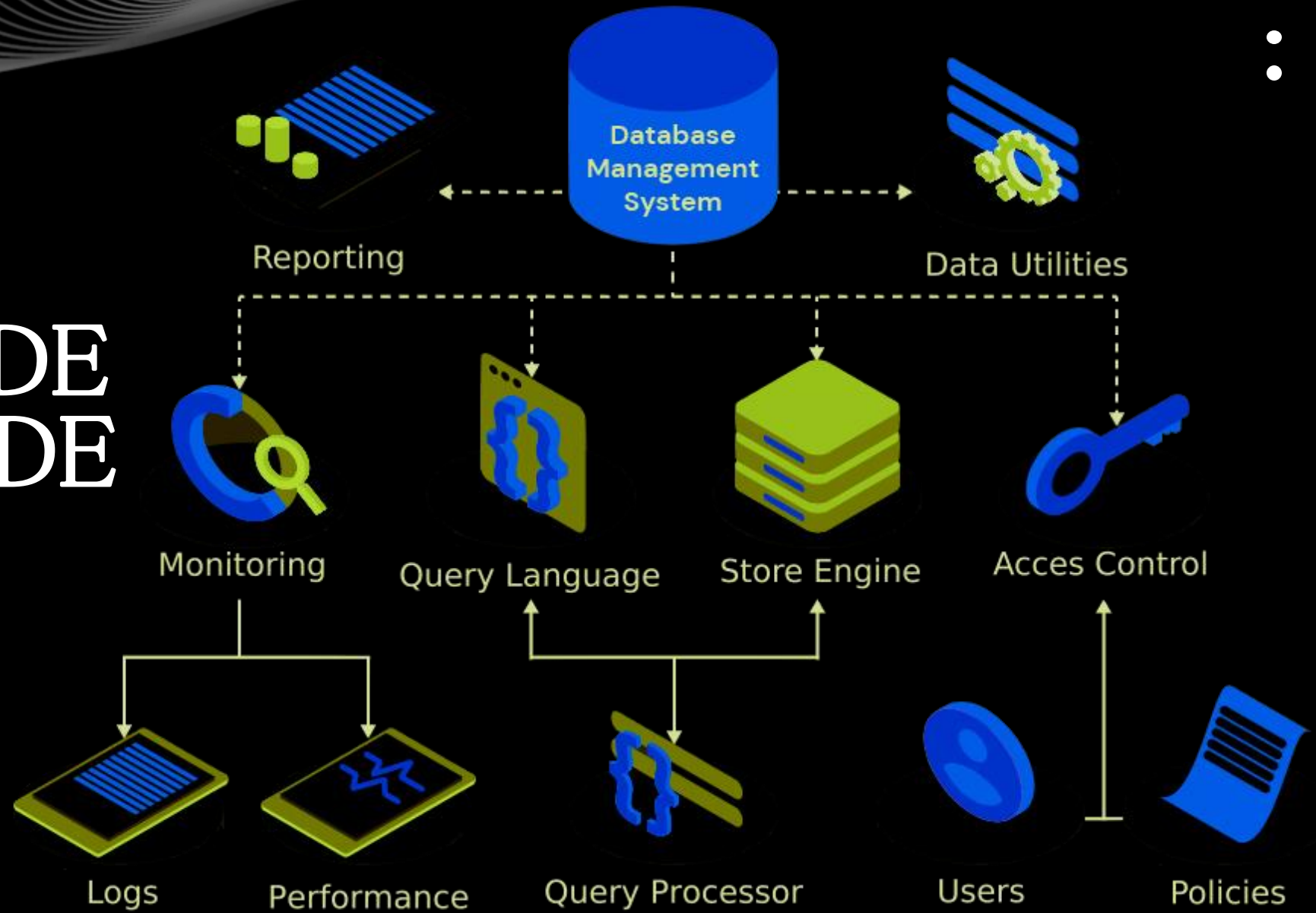


MOTORES DE BD
≠
SISTEMA DE
GESTIÓN DE BD
(SGBD)

SISTEMA DE GESTIÓN DE BD (SGBD)



SISTEMA DE GESTIÓN DE BD (SGBD)

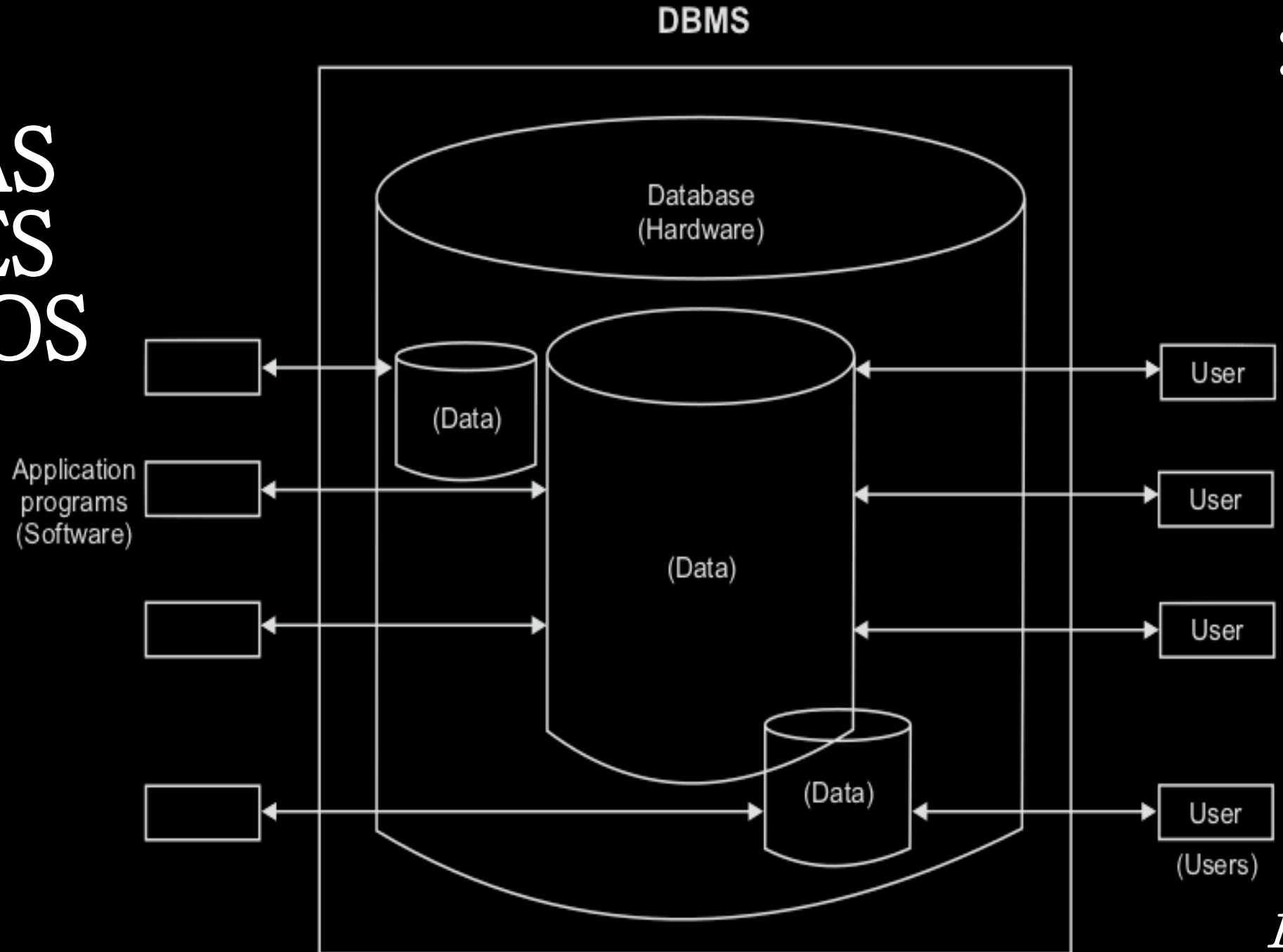


What Is a DBMS: Understanding Database Management Systems

SISTEMAS DE BASES DE DATOS

Un sistema de base de datos consiste en 4 componentes:

1. Datos
2. Hardware
3. Software
4. Usuarios



EVOLUCIÓN DE LAS BD

Data Models

GENERATION	TIME	MODEL	EXAMPLES	COMMENTS
First	1960s–1970s	File system	VMS/VSAM	Used mainly on IBM mainframe systems Managed records, not relationships
Second	1970s	Hierarchical and network	IMS ADABAS IDS-II	Early database systems Navigational access
Third	Mid-1970s to present	Relational	DB2 Oracle MS SQL-Server	Conceptual simplicity Entity relationship (ER) modeling support for relational data modeling
Fourth	Mid-1980s to present	Object-oriented Extended Relational	Versant VFS/FastObjects Objectivity/DB DB/2 UDB Oracle 10g	Support complex data Extended relational products support objects and data warehousing Web databases become common
Next Generation	Present to future	XML	dbXML Tamino DB2 UDB Oracle 10g MS SQL Server	Organization and management of unstructured data Relational and object models add support for XML documents

ARQUITECTURA DE BD



La arquitectura del DBMS ayuda a los usuarios a realizar sus peticiones mientras se conectan a la base de datos. Elegimos la arquitectura de la base de datos en función de varios factores, como el tamaño de la base de datos, el número de usuarios y las relaciones entre los usuarios.

ARQUITECTURA 1-TIER

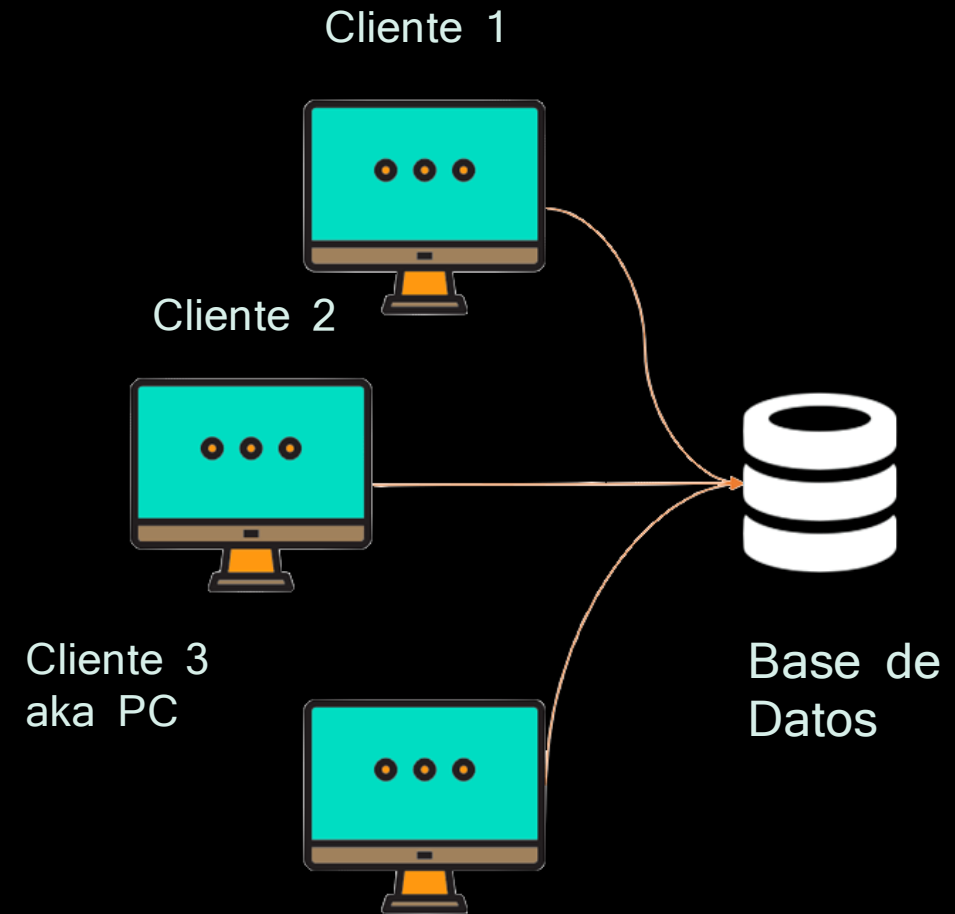
- La arquitectura más directa y rápida
- La BD es accesada por el usuario final
- El SGBD se conoce como el sistema local de base de datos
- Los cambios realizados se reflejan directamente en la BD y todo pasa en un solo servidor
- Utilizada para aprender SQL



Database Architecture in DBMS: 1-Tier, 2-Tier and 3-Tier

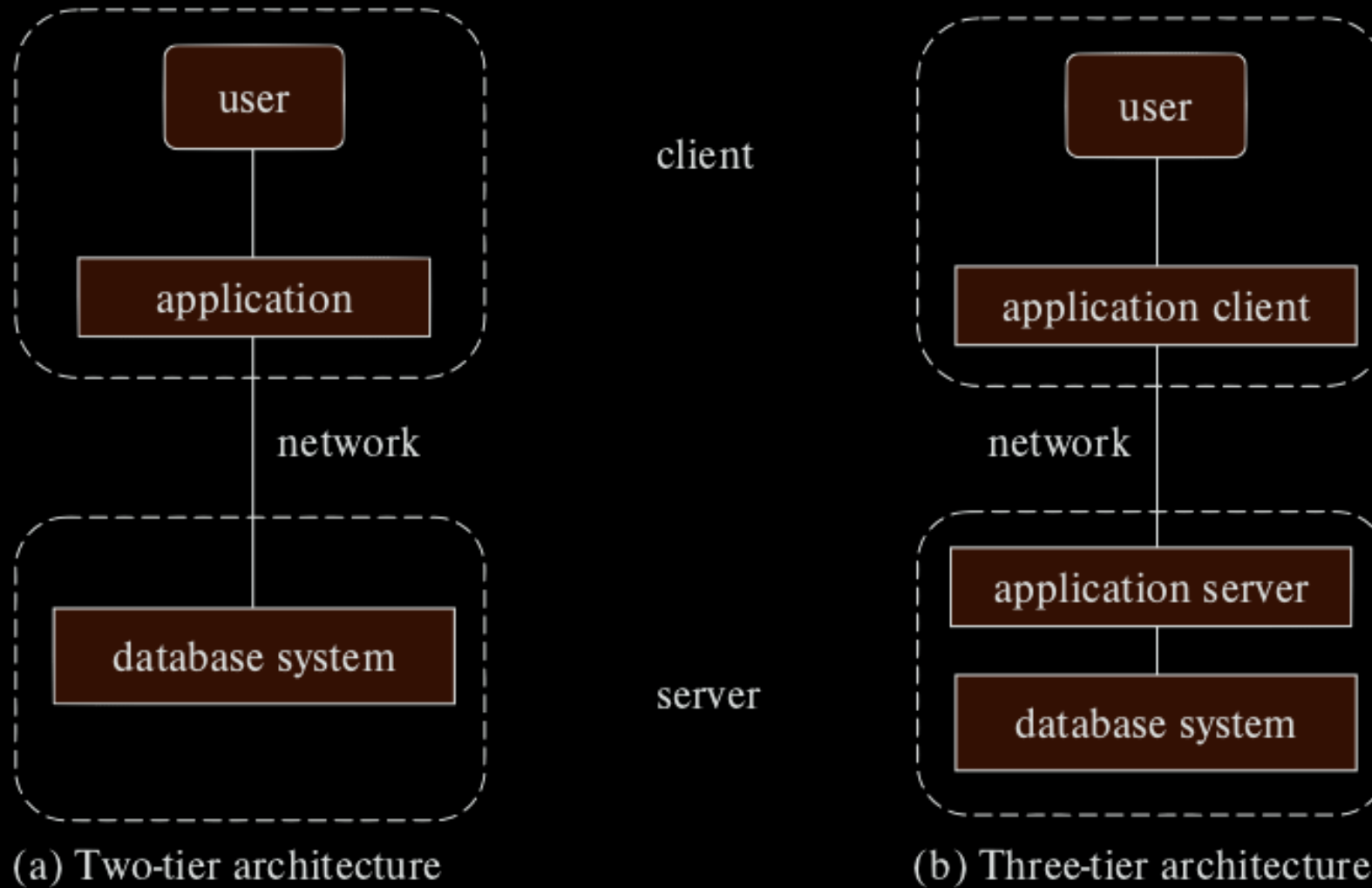
ARQUITECTURA 2-TIER

Es una arquitectura similar al modelo básico de servidor-cliente. Se caracteriza porque la capa de visualización esta en el cliente y los datos son almacenados en el servidor quien se encarga de brindar el proceso de consultas y las funcionalidades del manejo de las transacciones

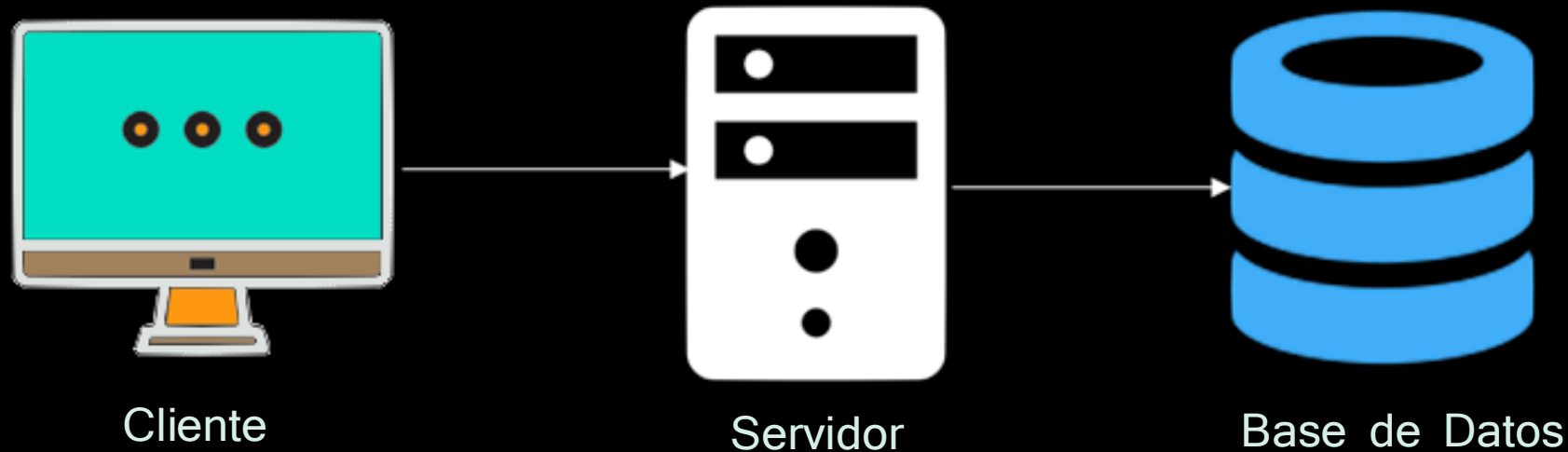


Database Architecture in DBMS: 1-Tier, 2-Tier and 3-Tier

ARQUITECTURA 2-TIER



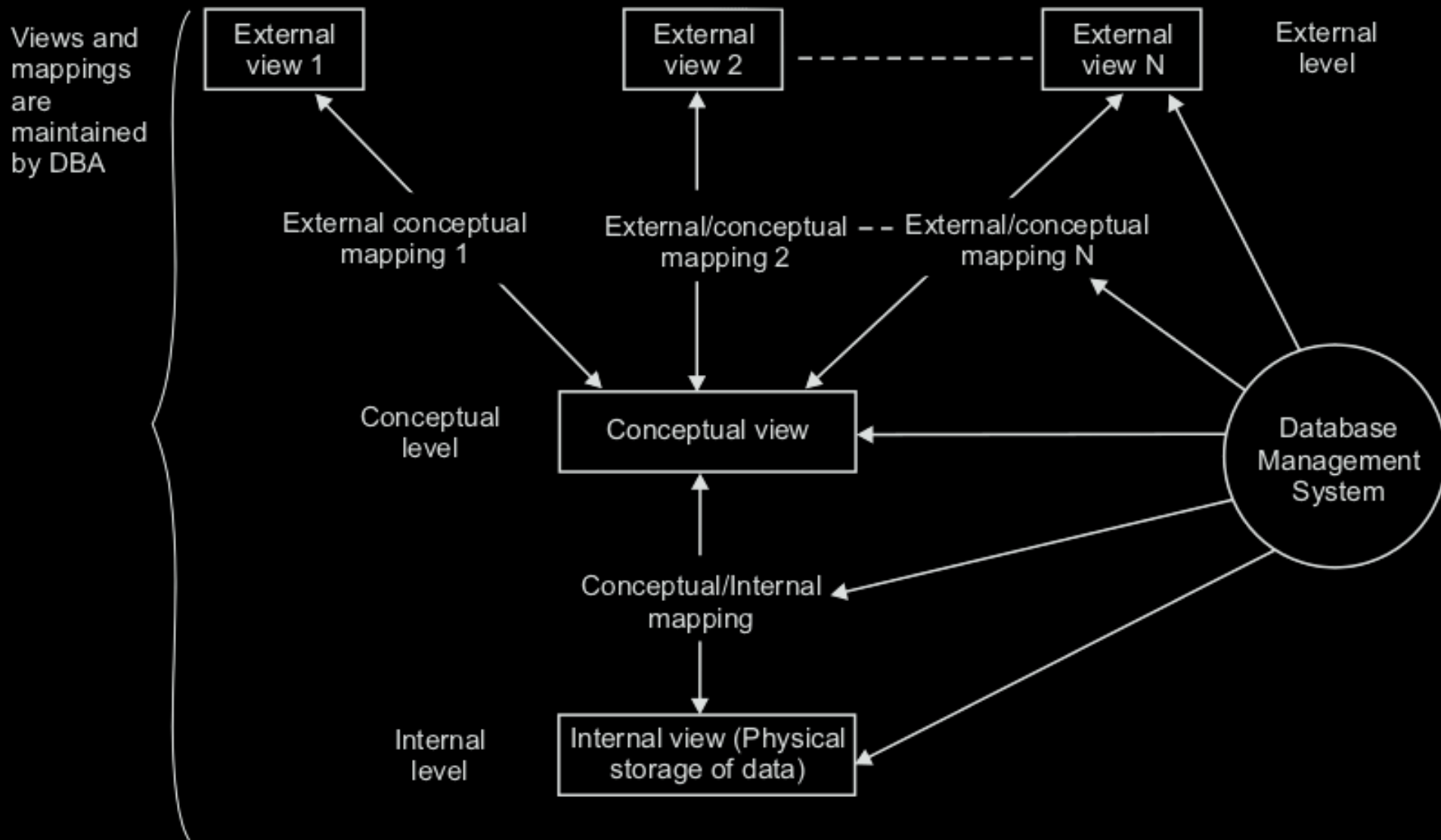
ARQUITECTURA 3-TIER



Propuesta por el American National Standards Institute/ Standards Planning And Requirements Committee (ANSI/SPARC) en 1978 con el fin de separar las aplicaciones del usuario de la base de datos física. Se caracteriza porque el esquema se divide en tres partes: externo, conceptual, e interno.

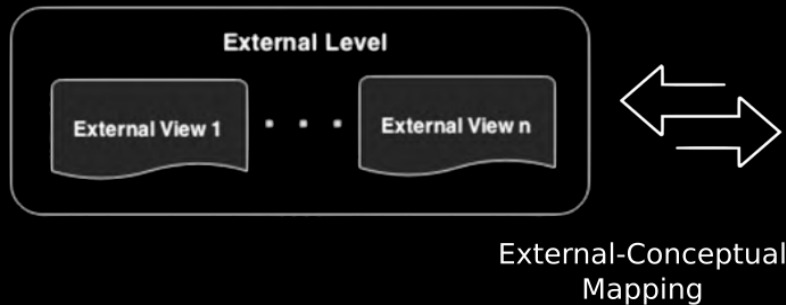
[Database Architecture in DBMS: 1-Tier, 2-Tier and 3-Tier](#)

ARQUITECTURA 3-TIER



ARQUITECTURA 3-TIER

Tipicamente, el DSL consiste del *data definition language* (DDL), un *data manipulation language* (DML), y el *data control language* (DCL).

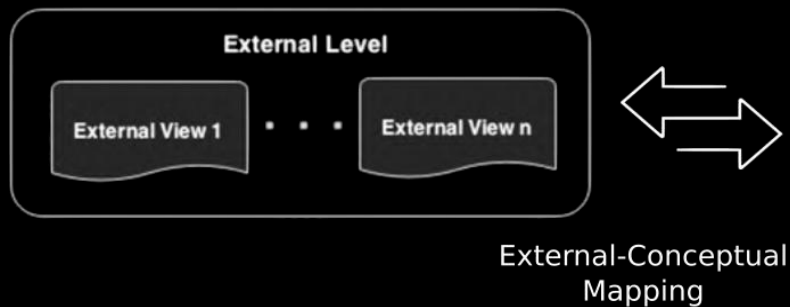


EXTERNO

Encargado de los espectadores. Tipicamente la BD es accesada a través del esquema externo. El programador de la aplicación usa el host language (high-level languages (HLL); examples include C++, Java, C#, Python, PHP, and PL/SQL) y el data sublanguage (DSL) (SQL, KQML, o QUEL)

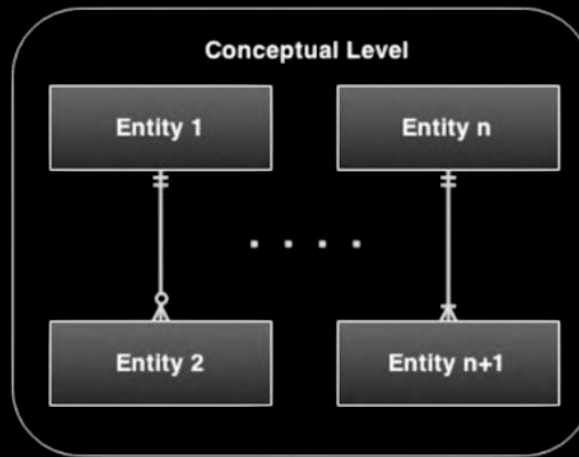
ARQUITECTURA 3-TIER

Tipicamente, el DSL consiste del *data definition language* (DDL), un *data manipulation language* (DML), y el *data control language* (DCL).



EXTERNO

Encargado de los espectadores. Tipicamente la BD es accesada a través del esquema externo. El programador de la aplicación usa el host language (high-level languages (HLL); examples include C++, Java, C#, Python, PHP, and PL/SQL) y el data sublanguage (DSL) (SQL, KQML, o QUEL)



CONCEPTUAL

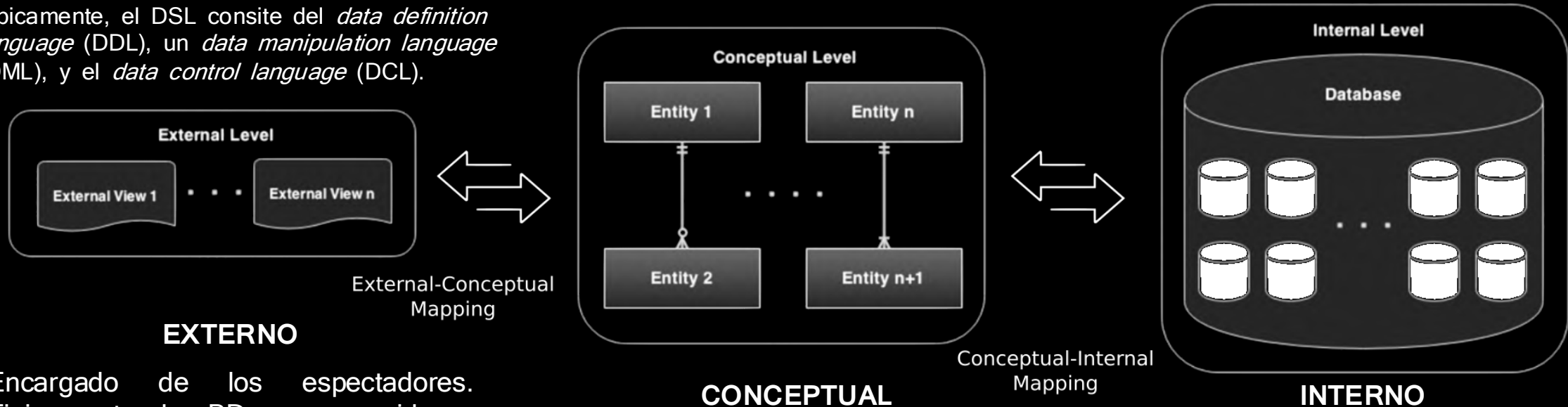
Es una representación abstracta de toda la información contenida en la BD. Es definido utilizando el esquema conceptual, también conocido como esquema lógico, que incluye la definición de cada objeto persistente (almacenados permanentemente) en la BD.

Este esquema incluye la estructura de la BD, restricciones de seguridad y operacionales, y verificación de integridad

- o
- o

ARQUITECTURA 3-TIER

Tipicamente, el DSL consiste del *data definition language* (DDL), un *data manipulation language* (DML), y el *data control language* (DCL).



Encargado de los espectadores. Tipicamente la BD es accesada a través del esquema externo. El programador de la aplicación usa el host language (high-level languages (HLL); examples include C++, Java, C#, Python, PHP, and PL/SQL) y el data sublanguage (DSL) (SQL, KQML, o QUEL)

Es una representación abstracta de toda la información contenida en la BD. Es definido utilizando el esquema conceptual, también conocido como esquema lógico, que incluye la definición de cada objeto persistente (almacenados permanentemente) en la BD.

Este esquema incluye la estructura de la BD, restricciones de seguridad y operacionales, y verificación de integridad

También conocido como la *vista del almacenamiento* es la representación de bajo nivel de la base de datos. Este nivel es definido por el esquema interno, que se encarga de los dispositivos de almacenamiento, índices, campos de representación, secuencia de almacenamiento físico de los registros, el acceso a los datos y escritura en el DDL

ARQUITECTURA 3-TIER

VENTAJAS

1. Cada usuario puede acceder a los mismos datos pero tener una vista personalizada diferente de los datos según sus necesidades.
2. Los cambios en la organización física del almacenamiento no afectan a la estructura interna de la base de datos. por ejemplo, mover la base de datos a un nuevo dispositivo de almacenamiento.
3. Para utilizar la base de datos, el usuario no necesita preocuparse por los detalles del almacenamiento físico de los datos.
4. La estructura conceptual de la base de datos puede ser modificada por el DBA sin afectar a ningún usuario.
5. La estructura de almacenamiento de la base de datos puede ser modificada por el DBA sin afectar a la vista del usuario.

DEPENDENCIA DE LOS DATOS

Es la característica de un sistema de BD a no ver afectado sus esquemas cuando se hace una modificación a un esquema de otro nivel. También puede definirse como la inmunidad de los programas de aplicación a los cambios en la representación física y las técnicas de acceso de la base de datos. Los dos tipos de independencia que hay son:

1. **Física:** Se pueden cambiar los dispositivos de almacenamiento y estructuras sin ningún problema.
2. **Lógica:** Se puede cambiar el esquema conceptual sin cambiar el esquema externo. Evita que se dañe la integridad de los datos.

ARQUITECTURA N-TIER

Presentation tier

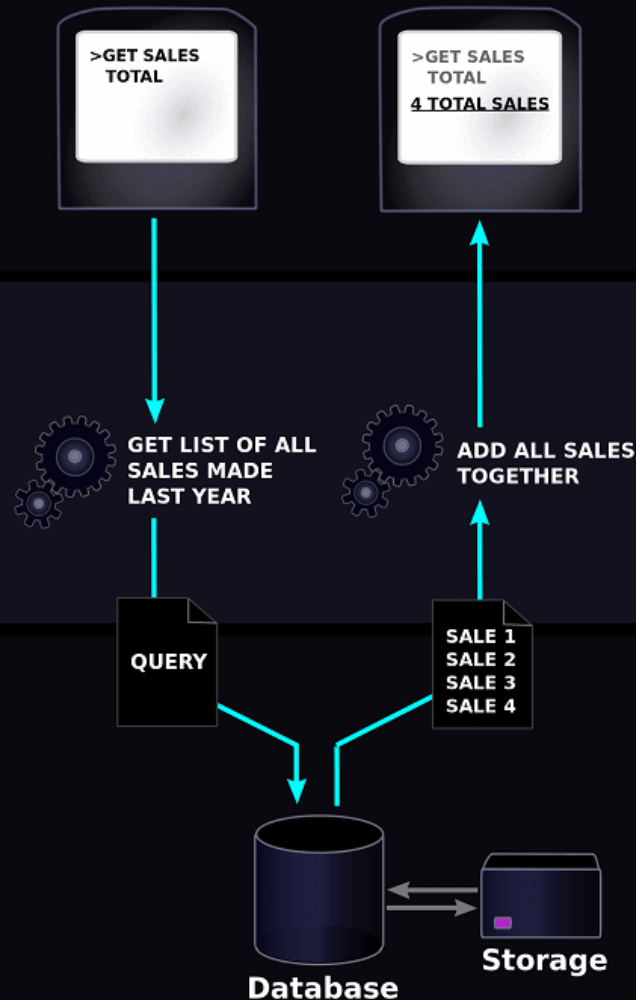
The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.

Logic tier

This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.

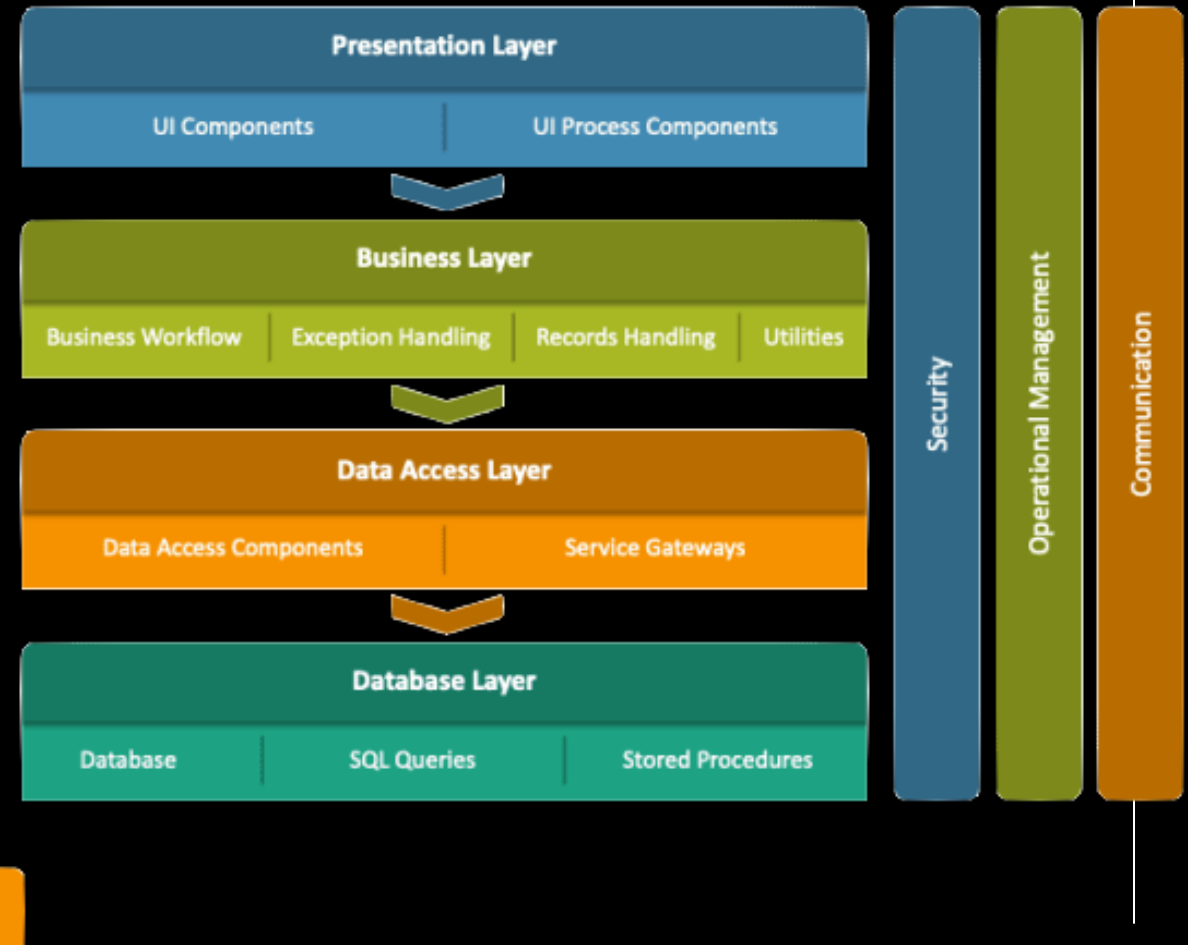
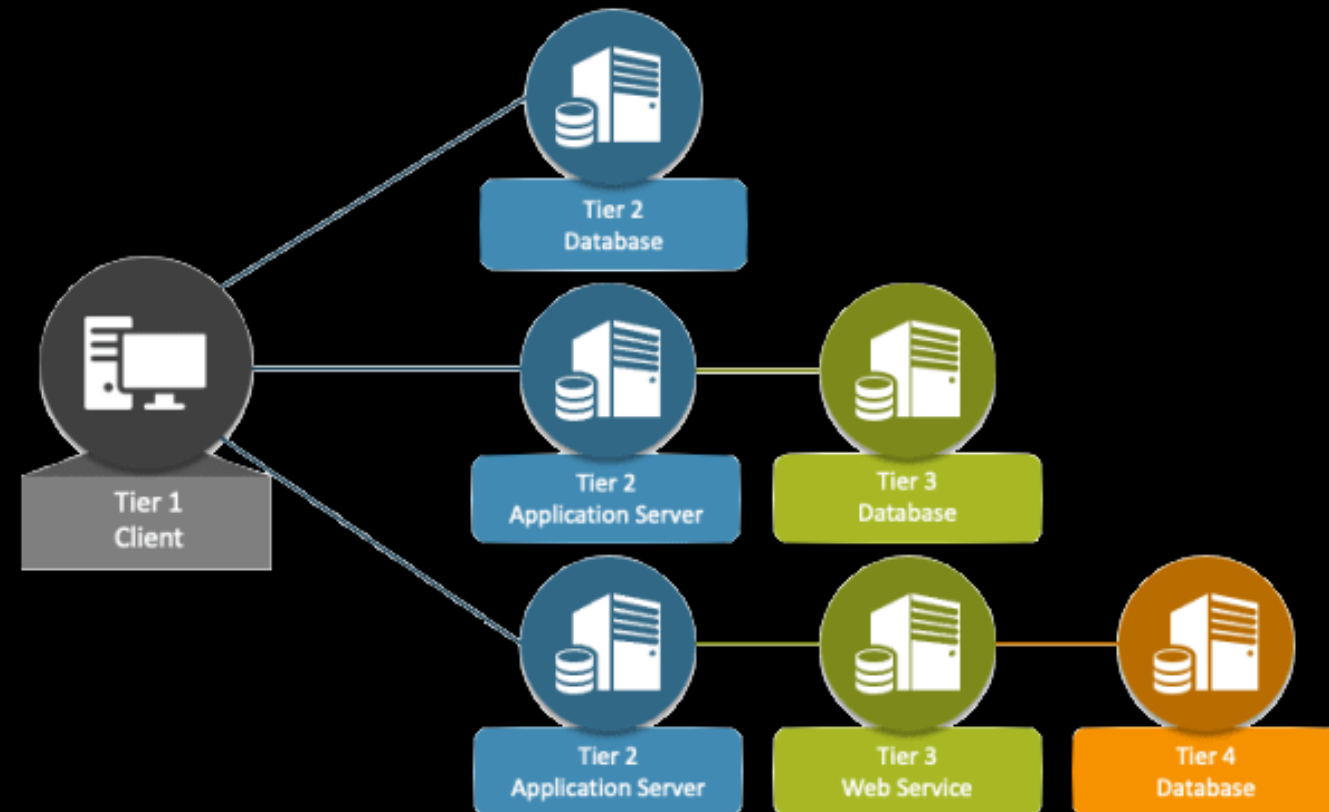
Data tier

Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.

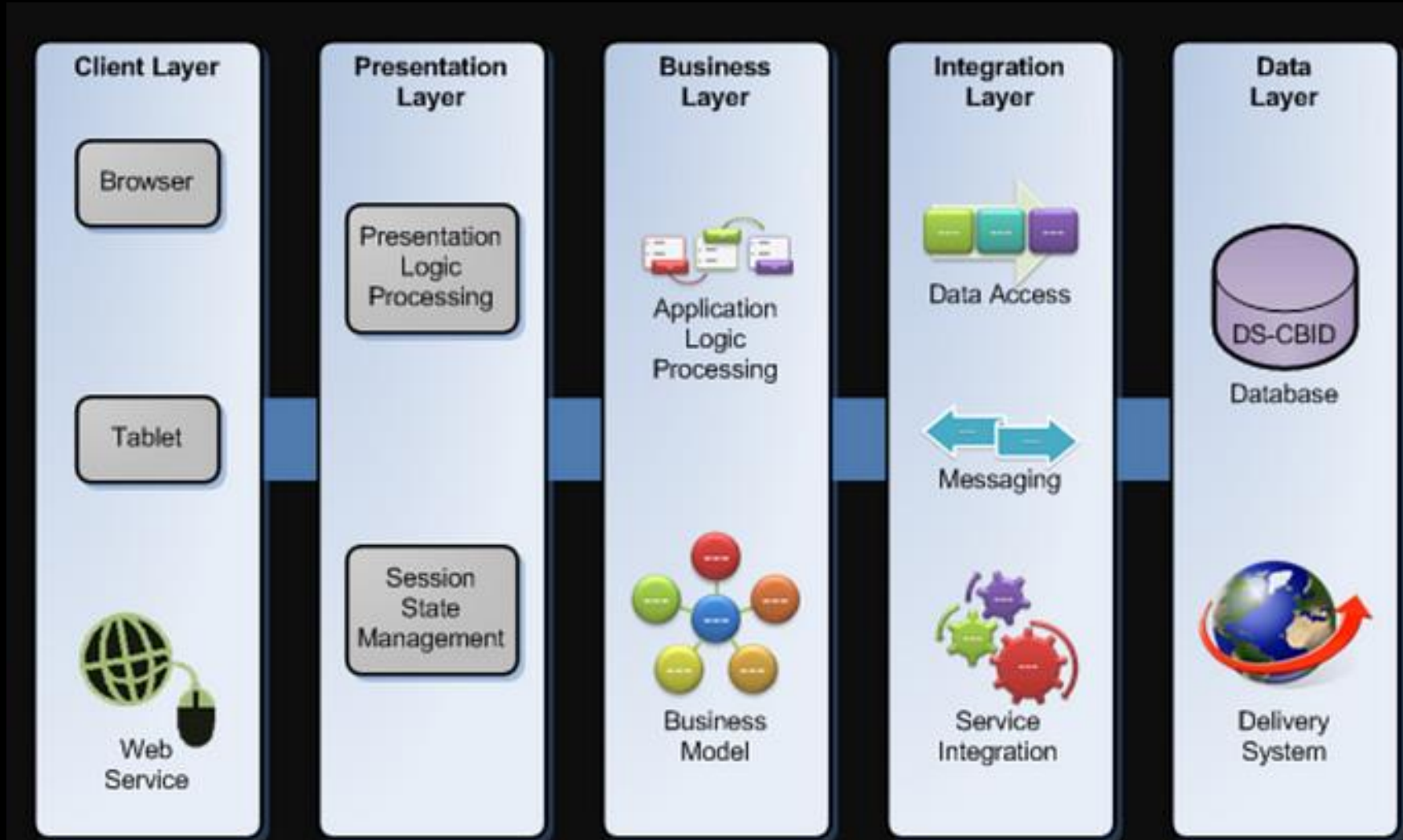


What is N-Tier Architecture? How It Works, Examples, Tutorials, and More

ARQUITECTURA N-TIER



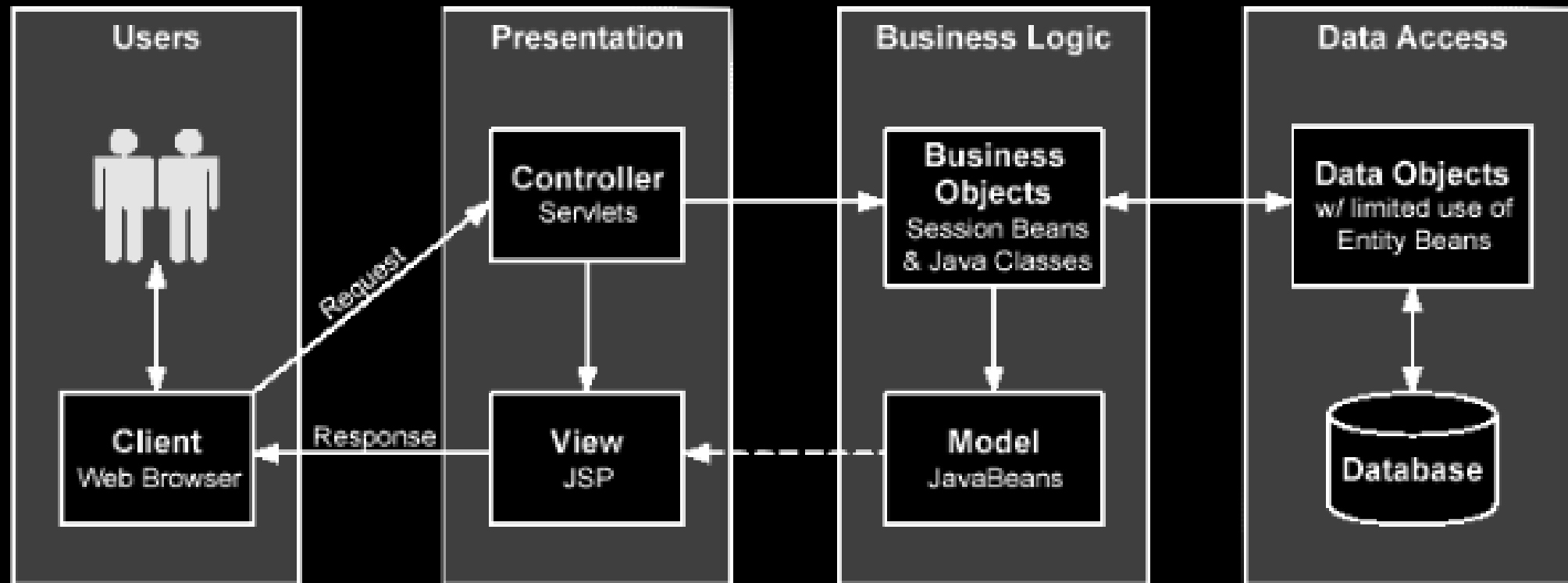
ARQUITECTURA N-TIER



Drawbacks of N-tier application development

ARQUITECTURA N-TIER

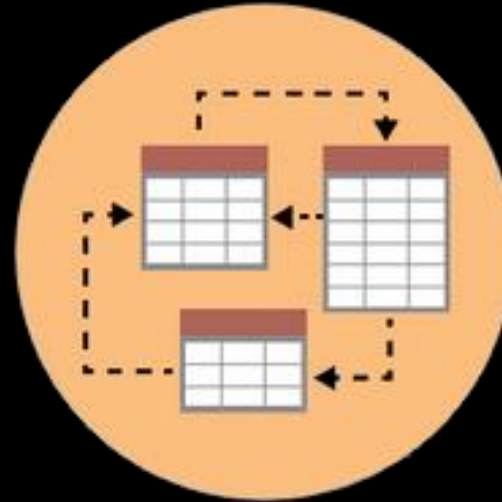
APLICACIÓN WEB



MODELOS DE DATOS

Colección de conceptos que son usados para la estructura de las BD incluido los tipos de datos, relaciones, y las restricciones que se le pueden aplicar a los datos. Los modelos de datos ayudan a entender el significado de los datos para así asegurarnos que entendemos:

- Los requerimientos de cada usuario
- El uso de los datos a través de varias aplicaciones
- La naturaleza de la independencia de los datos de su representación física



MODELOS DE DATOS

**What is a
Data
Model?**

intricity



MODELOS DE DATOS

Un modelo de datos debe poseer las siguientes características para que pueda obtenerse la mejor representación posible de los datos.

- I. Representación diagramática del modelo de datos.
- II. Simplicidad en el diseño, es decir, que los datos y sus relaciones puedan expresarse y distinguirse fácilmente.
- III. Independiente de la aplicación, de modo que distintas aplicaciones puedan compartirlo.
- IV. La representación de los datos debe ser sin duplicaciones.
- V. Debe seguirse un enfoque ascendente.
- VI. Debe mantenerse la coherencia y la validación de la estructura.

TIPOS DE MODELOS DE DATOS

Basados en Registros (records based)

Estos modelos representan los datos mediante estructuras de registros. Estos modelos se sitúan entre los modelos de datos basados en objetos y los modelos de datos físicos.

Basados en Objetos (objects based)

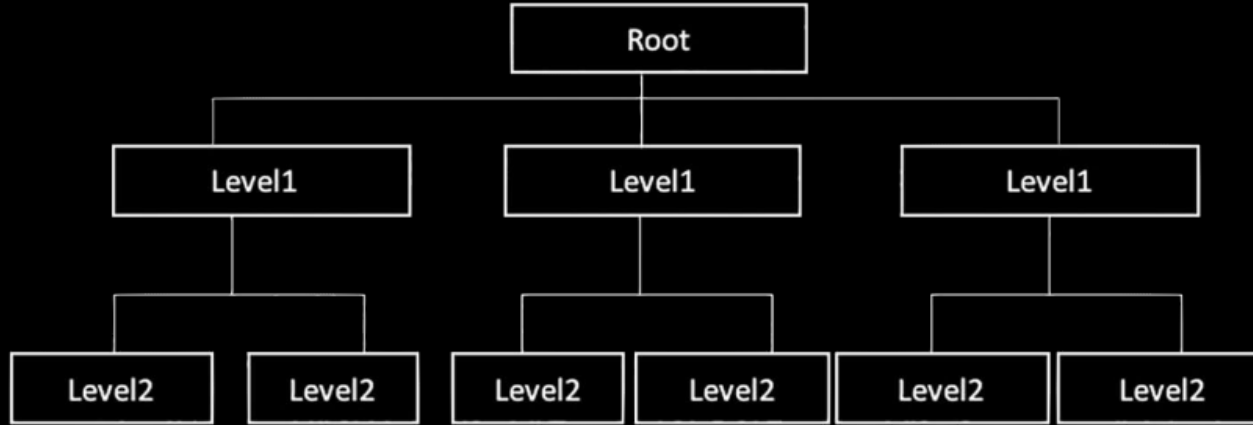
Estos modelos se utilizan para describir los datos a nivel lógico y de usuario. Estos modelos permiten a los usuarios especificar implícitamente las restricciones de los datos.

Físicos (physical)

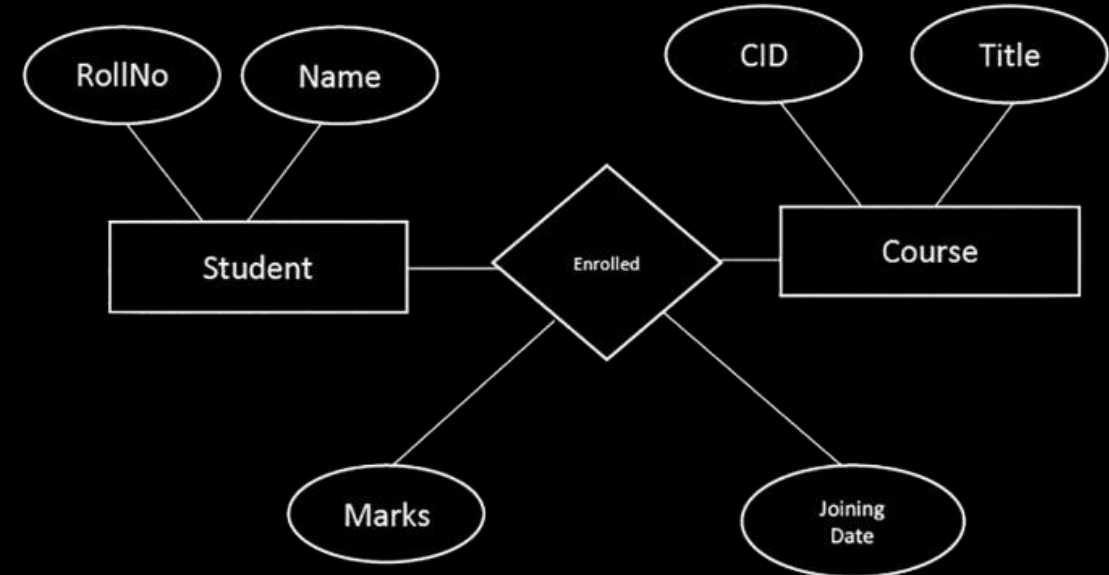
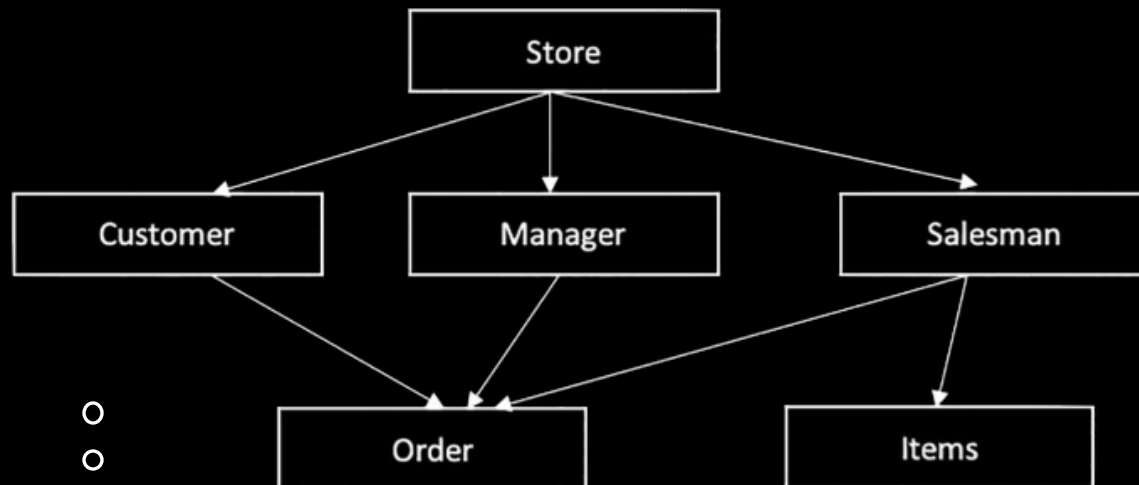
Estos modelos proporcionan los conceptos que describen los detalles de cómo se almacenan los datos en el ordenador junto con sus estructuras de registro, rutas de acceso y ordenación. Sólo los usuarios especializados o profesionales pueden utilizar estos modelos.

TIPOS DE MODELOS DE DATOS

BASADOS EN REGISTROS



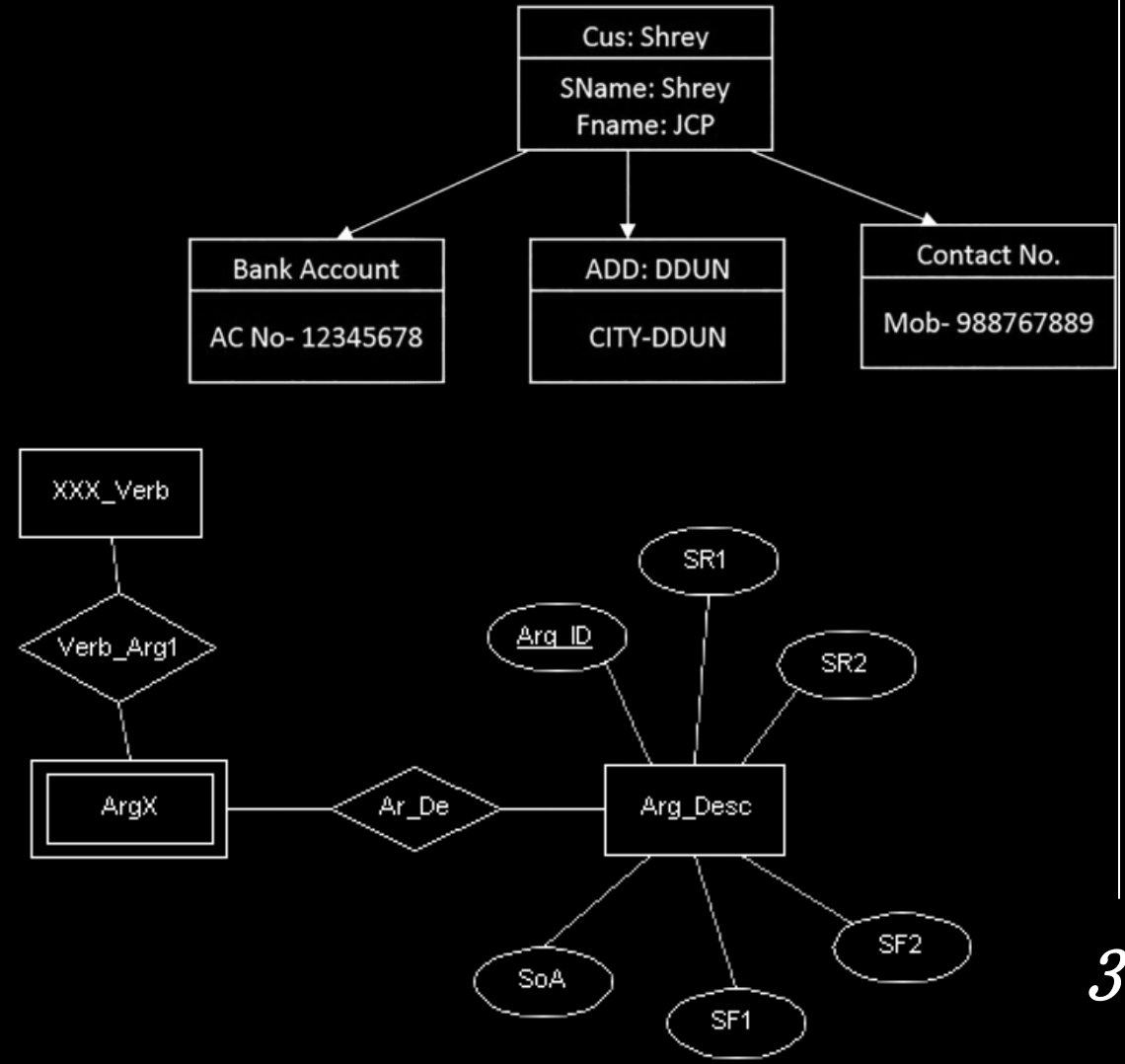
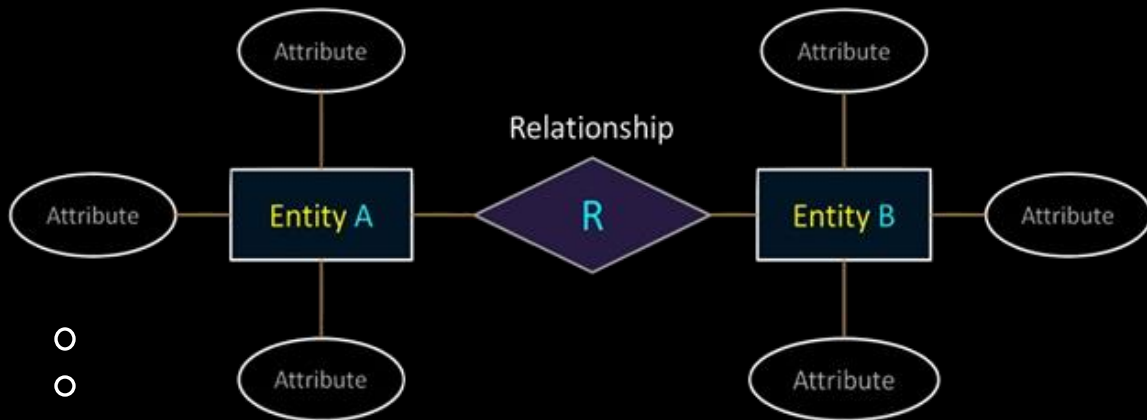
- Modelo de datos jerárquicos
- Modelo de datos de red
- Modelo de datos relacionales.



TIPOS DE MODELOS DE DATOS

BASADOS EN OBJETOS

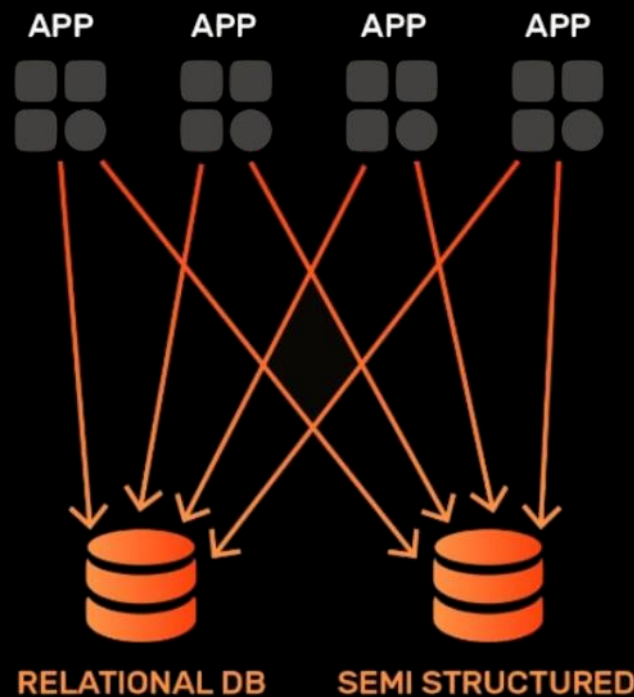
- Modelo de relación entre entidades (ER-Model)
- Modelo orientado a objetos
- Modelo de datos semánticos
- Modelo de datos funcional



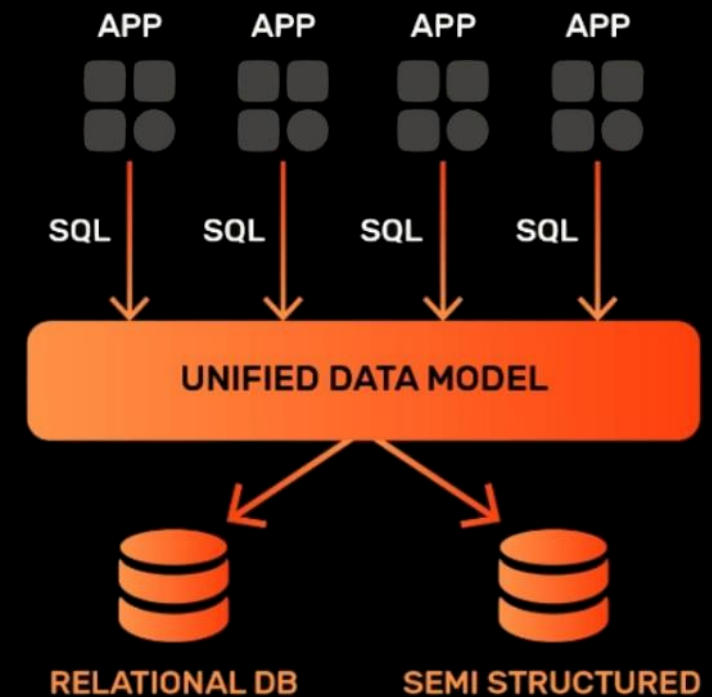
TIPOS DE MODELOS DE DATOS FÍSICOS

- Modelo unificador
- Modelo de memoria marco.

WITHOUT A UNIFIED DATA MODEL



UNIFIED DATA MODEL



TIPOS DE MODELOS DE DATOS

COMPARACIÓN

Property	Hierarchical	Network	Relational	E-R Diagram	Object-oriented
1. Data element organization	Files, records	Files, records	Tables/tuples	Objects, entity sets	Objects
2. Identity	Record based	Record based	Value based	Value based	Record based
3. Data Independence	Yes	Yes	Yes	Yes	Yes
4. Relationship Organization	Logical proximity in a linearised tree.	Intersecting Networks	Identifiers of rows in one table are embedded as attribute values in another table.	Relational extenders that support specialized applications.	Logical containment
5. Access Language	Procedural	Procedural	Non-procedural	Non-procedural	Procedural
6. Structural Independence	No	No	Yes	Yes	Yes

LOS 3 NIVELES DE DISEÑO DE BD

Es el tipo de modelo de datos más abstracto y de alto nivel. Se centra en las principales entidades, atributos y relaciones de los datos, sin especificar ningún detalle técnico, como tipos de datos, restricciones o claves.

Añade más información y precisión a los datos, como los atributos, dominios y claves específicos de cada entidad, y la cardinalidad y opcionalidad de cada relación. Un modelo de datos lógico es independiente de cualquier SGBD, pero sigue las reglas y normas de un tipo de modelo de datos, como el relacional, el jerárquico o el de red.

Es el nivel más detallado y técnico del proceso de diseño de bases de datos. En esta fase, los diseñadores toman decisiones sobre cómo se implementará el diseño lógico en un SGBD específico. Las consideraciones incluyen la indexación, el almacenamiento, la optimización del rendimiento y las medidas de seguridad.

1

Conceptual

2

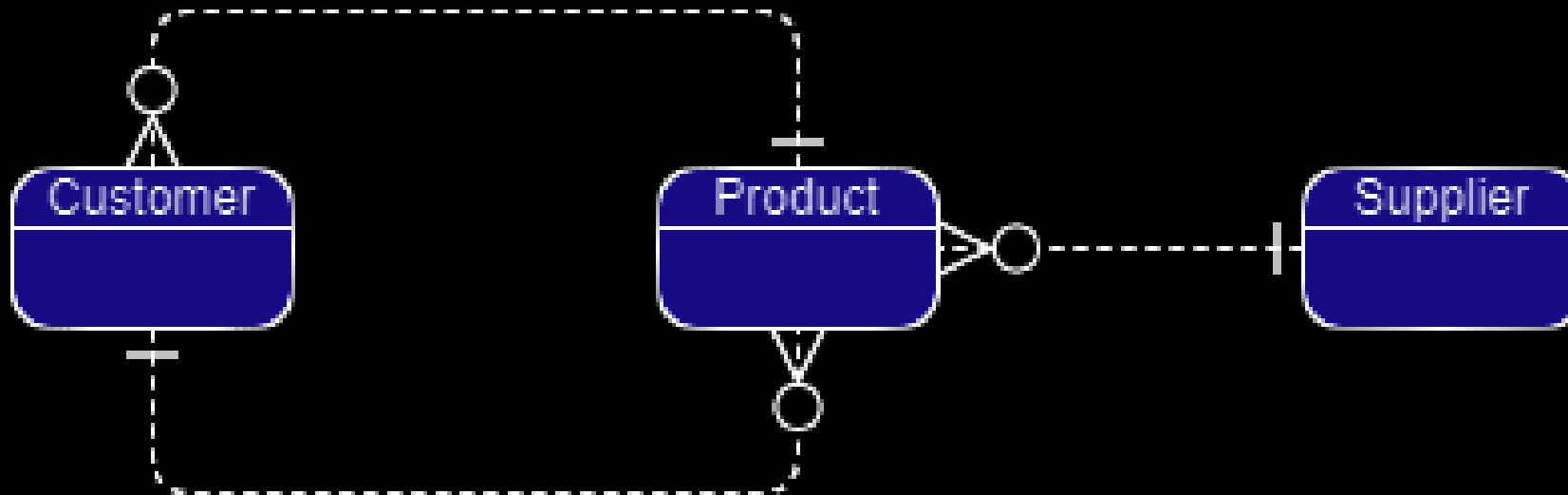
Lógico

3

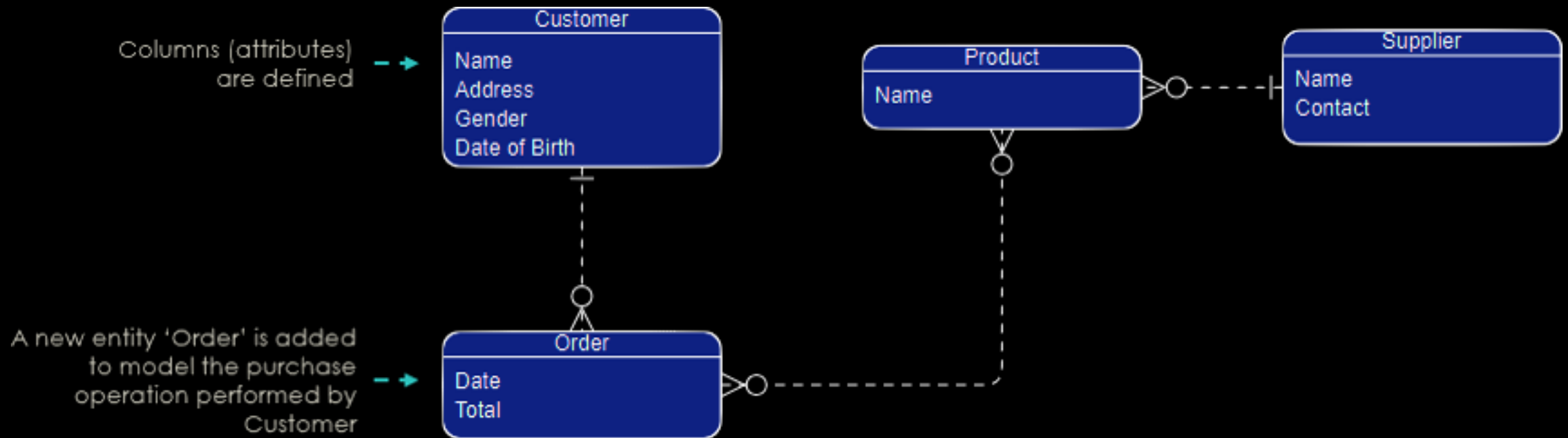
Físico

NIVELES DE DISEÑO

CONCEPTUAL

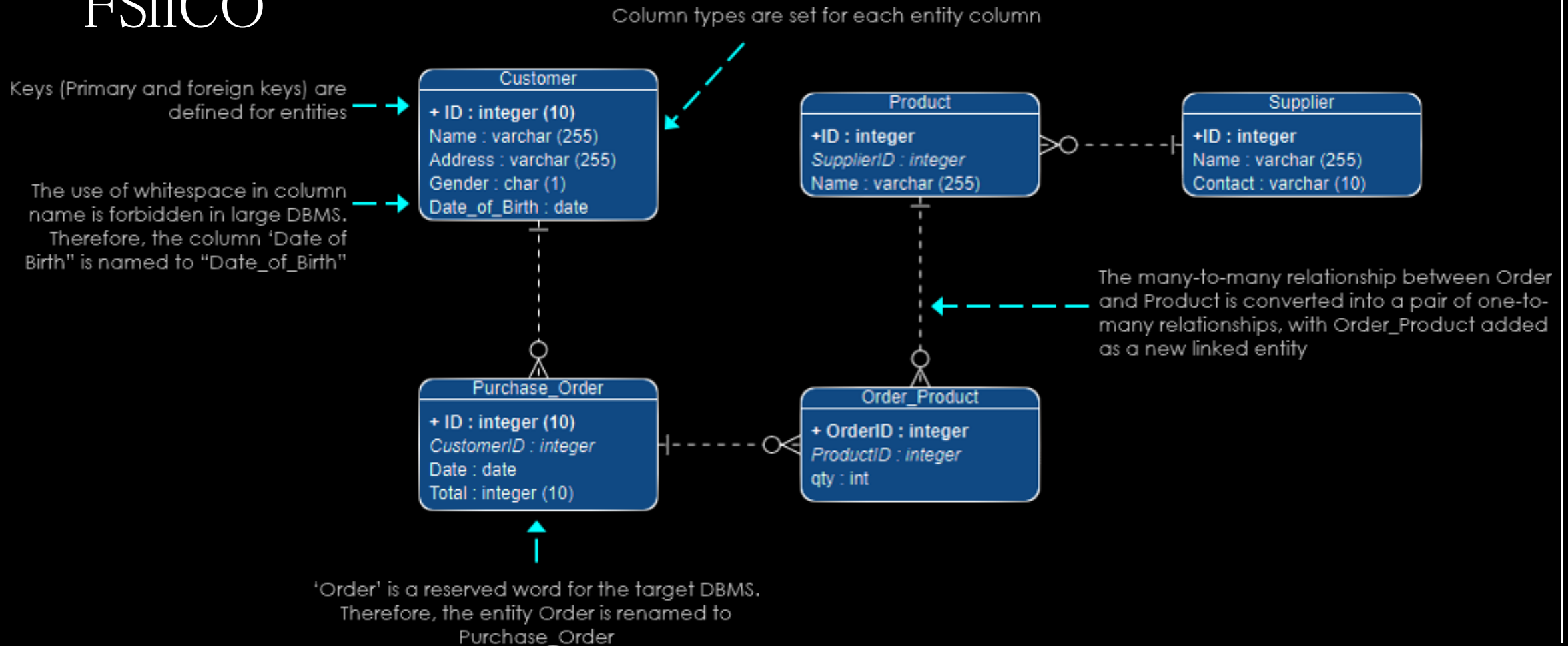


NIVELES DE DISEÑO LÓGICO

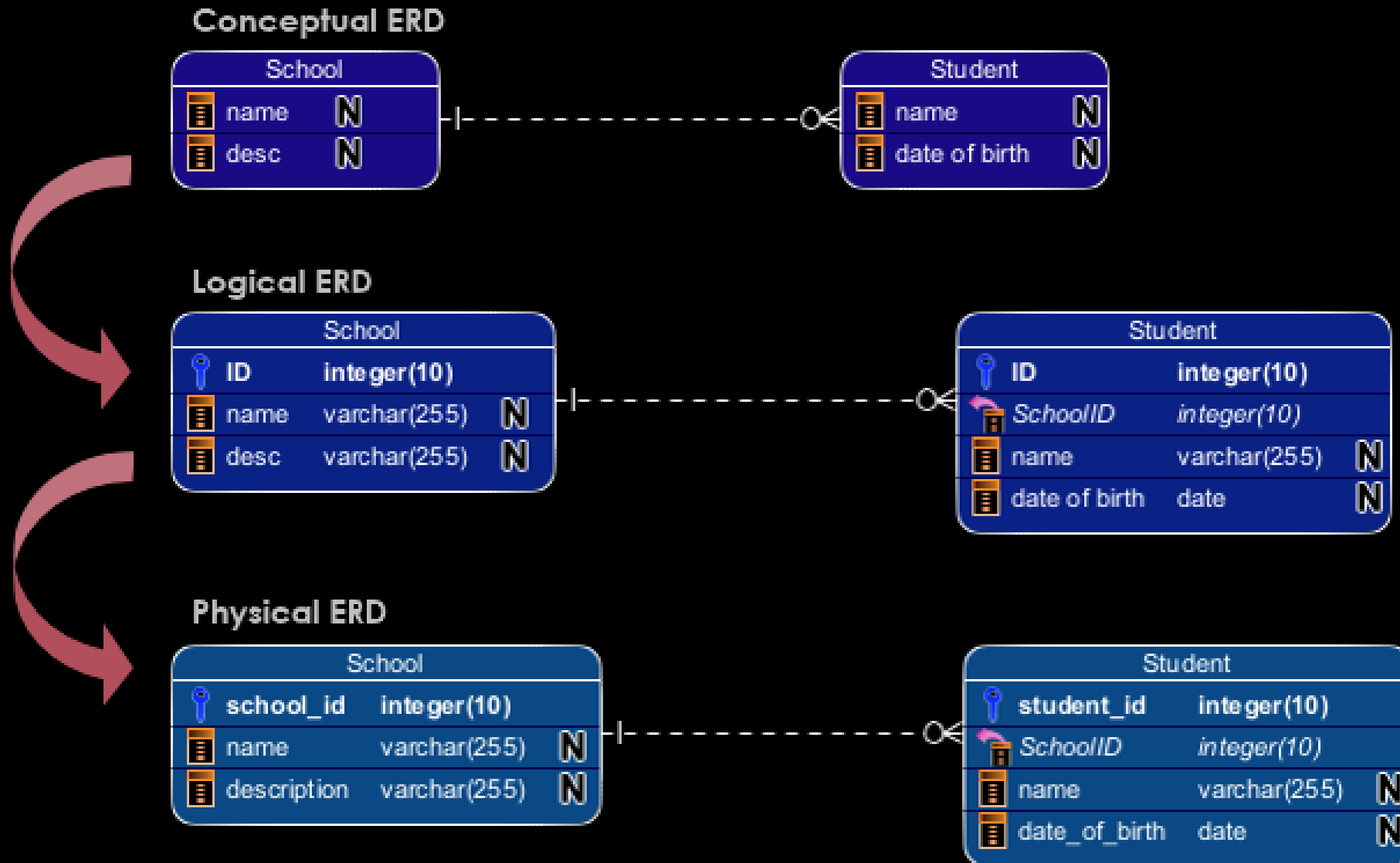


- **Entidades:** Estudiante, Curso, Instructor, Departamento
- **Relaciones:** El alumno se matricula en el curso, el profesor imparte el curso, el departamento gestiona el curso

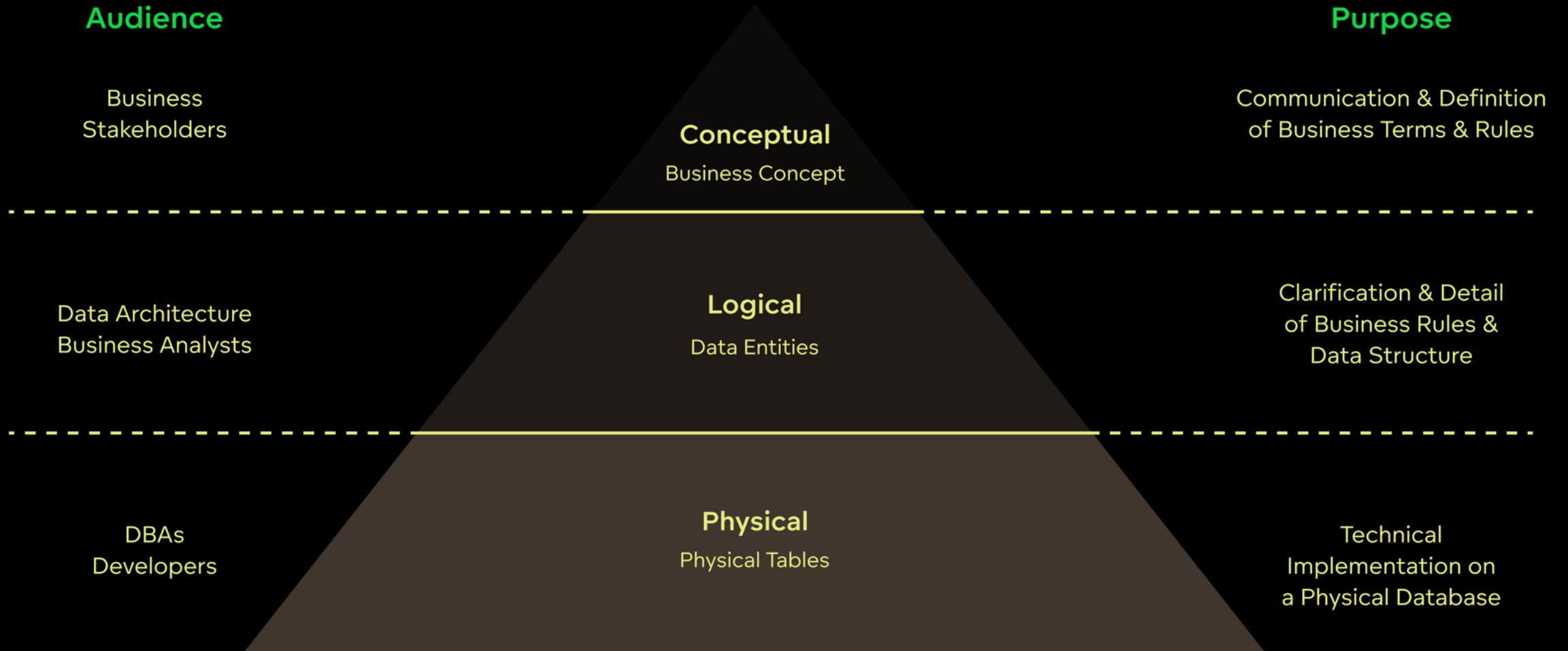
NIVELES DE DISEÑO FÍSICO



LOS 3 NIVELES DE DISEÑO DE BD

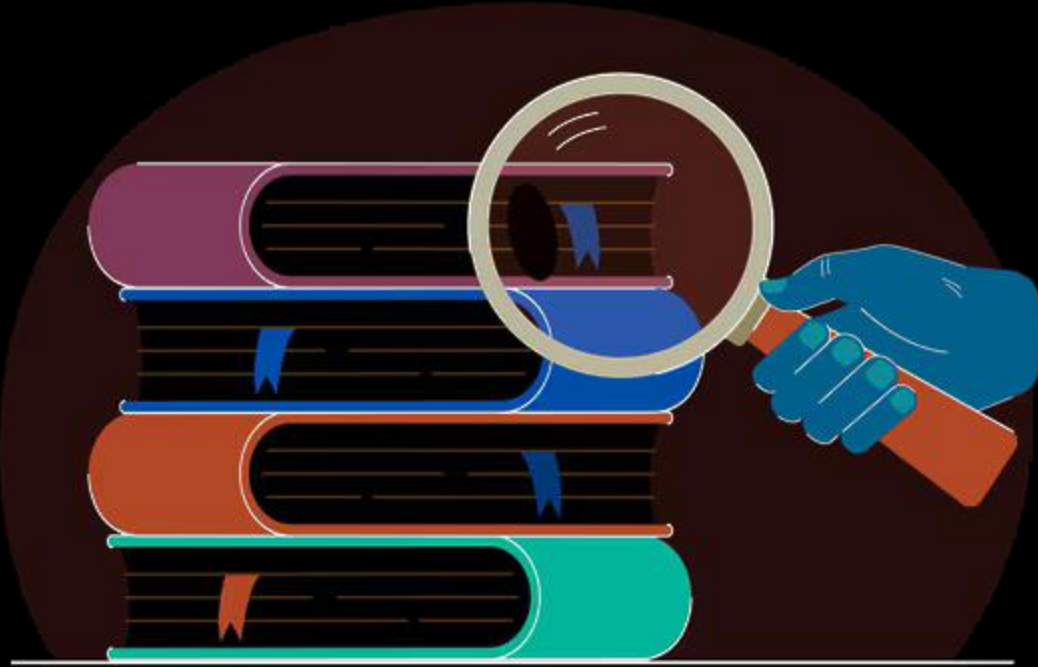


NIVELES DE DISEÑO DE BD



REFERENCIAS

- [Three Tier Architecture](#)
- [What is N-Tier Architecture? How It Works, Examples, Tutorials, and More](#)
- [Understanding Layers, Tiers, and N-Tier Architecture in Application Development](#)
- [7 Database Paradigms](#)



¡Gracias!

Sebastián Aguilera Novoa
saguileran@itc.edu.co

