

---

*Definiciones*

---

1. **Planificación de la base de datos:** Actividades de gestión que permiten llevar a cabo las etapas del ciclo de vida del desarrollo de sistemas de bases de datos de la forma más eficiente y efectiva posible.
2. **Vista de usuario:** Define qué es lo que se requiere de un sistema de base de datos desde la perspectiva de un determinado rol de la organización o de un área de aplicación (del entorno en el cual se va a implantar la base de datos).
3. **Recopilación y análisis de requisitos:** Proceso de recopilar y analizar la información acerca de la organización a la que el sistema de base de datos va a dar soporte y utilizar esta información para identificar las necesidades del nuevo sistema de base de datos.
4. **Diseño conceptual de la base de datos:** Proceso de construcción de un modelo de los datos utilizados en una organización, de forma independiente de cualquier consideración física.
5. **Transacción:** Acción o serie de acciones que acceden o modifican el contenido de una Base de Datos.
6. **Pruebas:** Proceso de operar una Base de Datos con la intención de localizar posibles errores.
7. **Tipo de entidad:** Grupo de objetos con las mismas propiedades, con existencia independiente física o conceptual.
8. **Entidad:** Elemento identificable de forma unívoca que incluye una entidad de cada uno de los tipos de entidad participantes.
9. **Tipo de Relación:** Conjunto de asociaciones entre tipos de entidad.
10. **Relación:** Asociación identificable de forma unívoca que incluye una entidad de cada uno de los tipos de entidad participantes.
11. **Atributo:** Propiedad de un tipo de entidad o de un tipo de relación.
12. **Clave Candidata:** Conjunto mínimo de atributos que identifican de forma unívoca cada identidad de un tipo de entidad.
13. **Clave Primaria:** La clave candidata que se selecciona para identificar de forma unívoca cada instancia de un tipo de entidad.
14. **Multiplicidad:** Número de entidades de un tipo de entidad que se relacionan con una entidad del resto de tipos de entidad de una relación.

15. **Cardinalidad:** Número máximo de relaciones para una entidad que participa en un tipo de relación.
16. **Participación:** Determina si todas las entidades participan en relaciones en un tipo de relación.
17. **Superclase:** Tipo de entidad que incluye uno o más subgrupos diferentes de sus entidades.
18. **Subclase:** Grupo diferenciado de entidades de un tipo de entidad.
19. **Restricción de participación:** Determina si todo miembro de la superclase debe ser miembro de una subclase.
20. **Restricción de disyunción:** Indica si es posible que un miembro de una superclase sea miembro de más de una subclase.
21. **Agregación:** Abstracción que permite tratar los tipos de relación como tipos de entidad de un nivel conceptual superior.
22. **Relación:** Una relación es una tabla con columnas (atributos) y filas (tuplas).
23. **Dominio:** Conjunto de valores permitidos para uno o más atributos.
24. **Superclave:** Atributo o conjunto de atributos que identifica de forma unívoca cada tupla dentro de una relación.
25. **Clave Externa:** Atributo o conjunto dentro de una relación que se corresponden con una clave candidata de alguna relación.
26. **Valor nulo:** Valor para un atributo que es actualmente desconocido y no aplicable.
27. **Integridad de entidad:** Ningún atributo de una clave candidata de una relación puede ser nulo.
28. **Integridad referencial:** Si hay una clave externa en una relación, el valor de dicha clave externa debe corresponderse con un valor de la clave candidata en la relación referenciada o debe ser nulo.
29. **SELECT:** Poder consultar los datos de una base de datos.
30. **INSERT:** Para insertar datos en una tabla.
31. **UPDATE:** Para actualizar los datos de una tabla.
32. **DELETE:** Para borrar los datos de una tabla.

33. **SQL estático:** La instrucción SQL se conoce en tiempo de compilación del programa.
34. **SQL dinámico:** La instrucción SQL se conoce, aunque sea parcialmente, en tiempo de ejecución del programa.
35. **Dependencia funcional:** Describe la relación existente entre atributos de una relación. Dados A y B, atributos o conjuntos de atributos de la relación, B dependerá funcionalmente de A si cada valor de A está asociado exactamente con un valor de B.
36. **Determinante:** Atributo o grupo de atributos que inician una dependencia funcional.
37. **Dependencia funcional completa:** Si A y B son atributos o conjuntos de atributos de una relación, B depende funcionalmente de manera completa de A si B depende funcionalmente de A, pero no de ninguno de sus subconjuntos.
38. **Dependencia funcional transitiva:** Si A, B y C son atributos o conjuntos de una relación tales que B depende funcionalmente de A y C depende funcionalmente de B, entonces C depende transitivamente de A a través de B.
39. **NOT NULL:** Datos requeridos.
40. **CHECK:** Restricción de dominio.
41. **PRIMARY KEY:** Integridad de entidad.
42. **FOREIGN KEY:** Integridad referencial.
43. **ON DELETE / ON UPDATE:** Políticas de integridad referencial.
44. **COUNT:** Devuelve el número de valores en una columna especificada.
45. **SUM:** Devuelve la suma de los valores contenidos en una columna especificada.
46. **AVG:** Devuelve la media de los valores contenidos en una columna especificada.
47. **MIN:** Devuelve el mínimo de los valores contenidos en una columna especificada.
48. **MAX:** Devuelve el máximo de los valores contenidos en una columna especificada.
49. **Restricción (RESTRICT):** En este caso no se puede actualizar/eliminar un valor en la tabla referenciada sin actualizar/eliminar previamente las apariciones en la relación referenciante.
50. **Cascada (CASCADE):** En este caso se actualizarán/eliminarán las apariciones en la relación referenciante cuando se actualicen/eliminen los datos en la tabla referenciada.

51. **Fijar (SET DEFAULT y SET NULL):** En este caso, al actualizar/eliminar un valor en la tabla referenciada se pondrán a un valor, determinado o nulo, las apariciones en la tabla referenciante.

---

*Otros conceptos para tener en cuenta*

---

- A. **Entidad fuerte:** Tiene existencia independiente de cualquier tipo de entidad del Modelo Entidad-Relación. Como consecuencia de esta existencia independiente, tiene atributos que le permiten formar, al menos, una clave candidata (y, por tanto, tiene clave primaria).
- B. **Entidad débil:** No tiene existencia independiente y necesita de, al menos, un tipo de entidad fuerte para completar su definición. Como consecuencia de esta necesidad, no tiene atributos que le permitan formar una clave candidata (y, por tanto, necesita atributos de un tipo de entidad fuerte para formar su clave primaria). Para formar una clave candidata de un tipo de entidad débil se necesita el apoyo de una clave candidata de, al menos, un tipo de entidad fuerte de la cual presenta lo que se denomina una dependencia de existencia.
- C. **Relación ternaria vs Relación binaria:** En ambos casos se relacionan entre si los tres tipos de entidad. La diferencia radica en que en un tipo de relación ternaria los tres tipos de entidad establecen la relación de forma simultánea pudiendo tener valores independientes cada una de ellas. Mientras que, en un tipo de relación binaria con un tercer tipo de entidad agregada, la relación independiente se forma entre los tipos de entidad de la relación y el tercer tipo de entidad sólo se puede relacionar con relaciones de los dos tipos de entidad definidas previamente.
- D. **Especialización vs Generalización:** Su diferencia es meramente conceptual. Si se quiere primar la idea de que un conjunto inicial se separa en diferentes conjuntos se hablaría de especialización. Sin embargo, si se quiere transmitir la idea de que un conjunto de elementos tiene algo en común se hablaría de generalización. En la representación del Modelo Entidad-Relación no se puede expresar el concepto que produjo la decisión sino sólo su resultado. Esto es debido a que en la representación sólo existe un tipo de generalización-especialización.
- E. **Lenguaje de Definición de Datos (DDL):** Conjunto de instrucciones de SQL que sirven para definir la estructura de la base de datos y controlar el acceso a los datos.
- F. **Lenguaje de Manipulación de Datos (DML):** Conjunto de instrucciones de SQL que sirven para extraer y actualizar los datos.
- G. **Reunión sin pérdidas:** garantiza que cualquier instancia de la relación original pueda ser identificada a partir de las instancias correspondientes de las relaciones más pequeñas.

### NIVELES DE ABSTRACCIÓN

Existen tres niveles de abstracción, es decir, tres niveles diferentes mediante los cuales pueden describirse los elementos de datos. Estos niveles forman una arquitectura en tres niveles que comprende un nivel **externo (físico)**, otro **conceptual (lógico)** y otro **interno**. La forma en que los usuarios perciben los datos se denomina nivel externo. La forma en que el SGBD y el sistema operativo perciben los datos es el nivel interno, y en este nivel es donde se almacenan realmente los datos utilizando las estructuras de datos y organizaciones de archivos. El nivel conceptual proporciona tanto la correspondencia como la necesaria independencia entre los niveles externo e interno.

### INDEPENDENCIA DE DATOS

1. **Independencia lógica de los datos:** El concepto de independencia lógica de los datos hace referencia a la inmunidad de los esquemas externos a las modificaciones que se efectúen en el esquema conceptual.
2. **Independencia física de los datos:** El concepto de independencia física de los datos hace referencia a la inmunidad del esquema conceptual a los cambios que se efectúen en el esquema interno.

### CREACIÓN DE TABLAS – SQL (CASO\_09)

```
CREATE TABLE DEPARTAMENTOS (
    nombre VARCHAR(20),
    titulacion VARCHAR(20),
    PRIMARY KEY nombre,
);
```

```
DROP TABLE DEPARTAMENTOS;
```

```
INSERT INTO DEPARTAMENTOS VALUES ('Matemáticas', 'Grado en Matemáticas')
```

```
UPDATE DEPARTAMENTOS SET 'Titulación' = 'XXX'
WHERE nombre = 'Matemáticas'
```

```
CREATE TABLE AREAS (
    nombre VARCHAR(20) PRIMARY KEY,
    investigacion VARCHAR(30),
    FK nombre REFERENCES DEPARTAMENTOS(nombre),
);
```

```
DROP TABLE areas CASCADE;
```

```
INSERT INTO areas VALUES ('DepartamentoInvestigación', 'Álgebra', 'Matemáticas')
```

**NORMALIZACIÓN**

Cuando una tabla está en 1FN, decimos que es una tabla relacional correcta. Normalmente van a estar en 3FN (al igual que todas sus relaciones).

La 1FN es la única obligatoria del modelo relacional. Se prohíben los atributos compuestos y los multivaluados, tienen que ser todos simples.

En la 2FN, todo atributo que no está en la clave principal depende funcionalmente de manera completa de la clave principal.

**AXIOMAS DE ARMSTRONG**

- (1) **Reflexividad:** Si B es un subconjunto de A, entonces  $A \rightarrow B$
- (2) **Aumentación:** Si  $A \rightarrow B$ , entonces  $A, C \rightarrow B, C$
- (3) **Transitividad:** Si  $A \rightarrow B$  y  $B \rightarrow C$ , entonces  $A \rightarrow C$
- (4) **Autodeterminación:**  $A \rightarrow A$
- (5) **Descomposición:** Si  $A \rightarrow B, C$ , entonces  $A \rightarrow B$  y  $A \rightarrow C$
- (6) **Unión:** Si  $A \rightarrow B$  y  $A \rightarrow C$ , entonces  $A \rightarrow B, C$
- (7) **Composición:** Si  $A \rightarrow B$  y  $C \rightarrow D$ , entonces  $A, C \rightarrow B, D$

6

**SQL**

El orden de ejecución es FROM, WHERE, GROUPBY, HAVING, SELECT, ORDER BY.

FROM T1 INNER JOIN T2 ON B1=B2  
 FROM T1 NATURAL JOIN T2 –Elimina las columnas repetidas

FROM T1 LEFT/RIGHT OUTER JOIN T2  
 FULL OUTER JOIN (todas las filas)  
 CROSS JOIN (join sin condición)

WHERE ... atributo <= ALL (subclase aquí) //filas de una tabla o valores de una columna  
 (NOT) IN (subclase aquí) //si coincide cn valor de una subconsulta o lista

Redondear (ROUND), valor absoluto (ABS), truncar (TRUNC)