

Introduzione sui Database

In informatica un **database** (o base di dati) è un contenitore di grandi quantità di dati.

A livello **fisico (hardware)** i dati sono presenti su dispositivi di archiviazione (dischi, storage, server).

A livello **software**, i dati vengono creati, consultati, modificati e cancellati tramite programmi specifici detti **DBMS (Database Management Systems)**.

Il **DBMS** permette all'utente di accedere al database e offre funzionalità come:

- Gestione di grandi moli di dati
- Accesso simultaneo da parte di più utenti
- Persistenza e sicurezza dei dati

I database sono l'evoluzione digitale dei vecchi archivi fisici.

Fasi della Progettazione di un Database

1. Modello Concettuale

Rappresenta **cosa** vogliamo memorizzare nella realtà.

Si realizza tramite il **modello Entità-Relazione (ER)**, che rappresenta la realtà con:

- **Entità** (oggetti della realtà, es. Automobile)
- **Relazioni** (legami tra entità, es. Proprietario–Automobile)

Esempio: Entità Automobile

- Attributi: marca, modello, targa
- Ogni riga che rappresenta un'automobile è un'**istanza**.
- La **targa** è un attributo che identifica univocamente l'automobile → **chiave primaria**.

Caratteristiche di un attributo:

- **Dominio**: insieme dei valori possibili (es. “Fiat”, “Ford”)
- **Formato**: tipo di dato (numero, stringa, data...)
- **Dimensione**: lunghezza massima del valore
- **Obbligatorietà**: se deve sempre avere un valore (non può essere *null*)

2. Le Relazioni

Una **relazione** associa due o più entità (es. *Donna – Figlio*).

Ogni relazione ha:

- **Grado:** numero di entità coinvolte (es. binaria → 2 entità)
- **Cardinalità:**
 - 1:1 → un solo elemento per parte
 - 1:N → un'entità può essere collegata a più elementi dell'altra
 - N:M → più elementi per entrambe le entità
- **Direzione:** entità padre → entità figlio
- **Esistenza:** se un'entità può esistere senza l'altra

Le **entità forti** esistono indipendentemente (es. *Donna*), le **entità deboli** no (es. *Figlio*).

3. Le Chiavi Esterne

Per collegare le entità si usa la **chiave esterna (Foreign Key – FK)**:

un attributo che riprende il valore della **chiave primaria (Primary Key – PK)** di un'altra entità.

Esempio:

- Tabella **Pazienti (PK = ID_Paziente)**
- Tabella **Esami (FK = ID_Paziente)** → ogni esame è collegato a un paziente

Questo garantisce la **coerenza dei dati** (integrità referenziale).

4. Modello Logico

Dal modello ER si passa al **modello logico relazionale**, basato sul concetto di **tabella**.

Ogni tabella rappresenta un'entità, e:

- Ogni **riga** = un'istanza
- Ogni **colonna** = un attributo
- La **cardinalità** = numero di righe
- Il **grado** = numero di colonne

Regole fondamentali:

- Ogni colonna ha lo stesso tipo di dato
- Ogni riga deve essere unica
- I valori devono essere **atomici** (es. separare “*Mario Rossi*” in nome e cognome)
- Nessuna ridondanza (→ **normalizzazione**)

5. Operazioni tra Tabelle

Principali operazioni del modello relazionale:

Operazione	Descrizione
Unione	Combina due tabelle eliminando i duplicati
Differenza	Mostra le righe della prima tabella non presenti nella seconda
Intersezione	Mostra solo le righe comuni
Prodotto cartesiano	Combina ogni riga della prima con ogni riga della seconda
Proiezione	Seleziona alcune colonne
Selezione	Filtra le righe in base a una condizione
Join	Combina tabelle in base a chiavi comuni

Il DBMS e il Linguaggio SQL

Un **DBMS relazionale (RDBMS)** consente di accedere al database tramite **SQL (Structured Query Language)**.

Tipi di comandi SQL:

Tipo	Nome	Funzione
DDL	Data Definition Language	Creazione e modifica di tabelle e oggetti
DML	Data Manipulation Language	Inserimento, modifica, eliminazione dei dati
DCL	Data Control Language	Gestione dei permessi e accessi
QL	Query Language	Interrogazione dei dati (es. SELECT)

Esempio di comandi SQL:

```
CREATE TABLE Pazienti (
    ID_Paziente INT PRIMARY KEY,
    Nome VARCHAR(50),
    Cognome VARCHAR(50)
);

INSERT INTO Pazienti VALUES (1, 'Mario', 'Rossi');

SELECT * FROM Pazienti;
```

Principali DBMS sul Mercato

- **Oracle Database**
- **Microsoft SQL Server**
- **MySQL** (open source)
- **PostgreSQL** (open source)
- **IBM Db2**

Conclusione

La progettazione di un database passa da tre livelli:

1. **Concettuale (ER)**
2. **Logico (Relazionale)**
3. **Fisico (implementazione su hardware e software DBMS)**

Comprendere la logica di **entità, relazioni, chiavi** e **SQL** è la base per ogni lavoro nel mondo dei dati.