

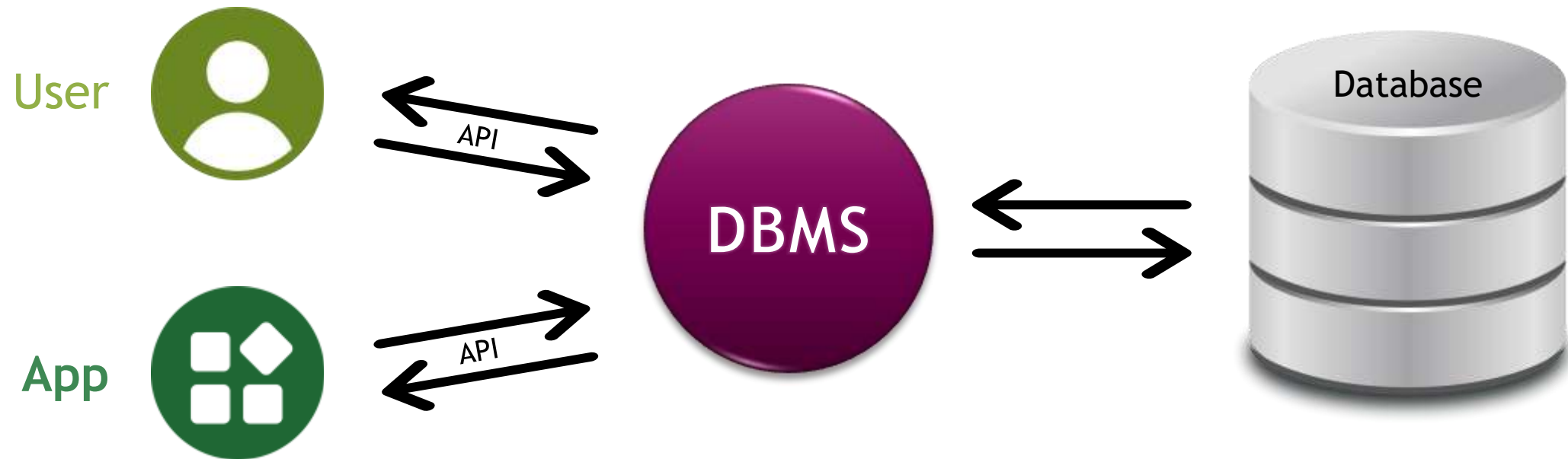
Introduzione ai database

Database

- Il **database** è una collezione di dati logicamente correlati e condivisi, che ha lo scopo di soddisfare i fabbisogni informativi di una specifica organizzazione.



- I dati, congiuntamente con la loro descrizione, sono gestiti da un unico sistema, chiamato **DBMS** (*DataBase Management System*), che li amministra e ne controlla gli accessi.



Alcuni DBMS...

ORACLE®
DATABASE



Lo standard di riferimento per l'enterprise, offre massima potenza, sicurezza e affidabilità per applicazioni mission-critical.



Soluzione robusta e integrata nell'ecosistema Microsoft, ideale per aziende che utilizzano stack tecnologici Windows e .NET.



PostgreSQL

Database open source avanzato e ricco di funzionalità, apprezzato per il rigore sugli standard SQL e la gestione di dati complessi.

Il database open source più utilizzato al mondo, celebre per la sua semplicità, affidabilità e vasta adozione nel web.



Il principale database NoSQL, progettato per archiviare dati non strutturati in documenti JSON, garantendo massima flessibilità e



scalabilità orizzontale.
mongoDB®

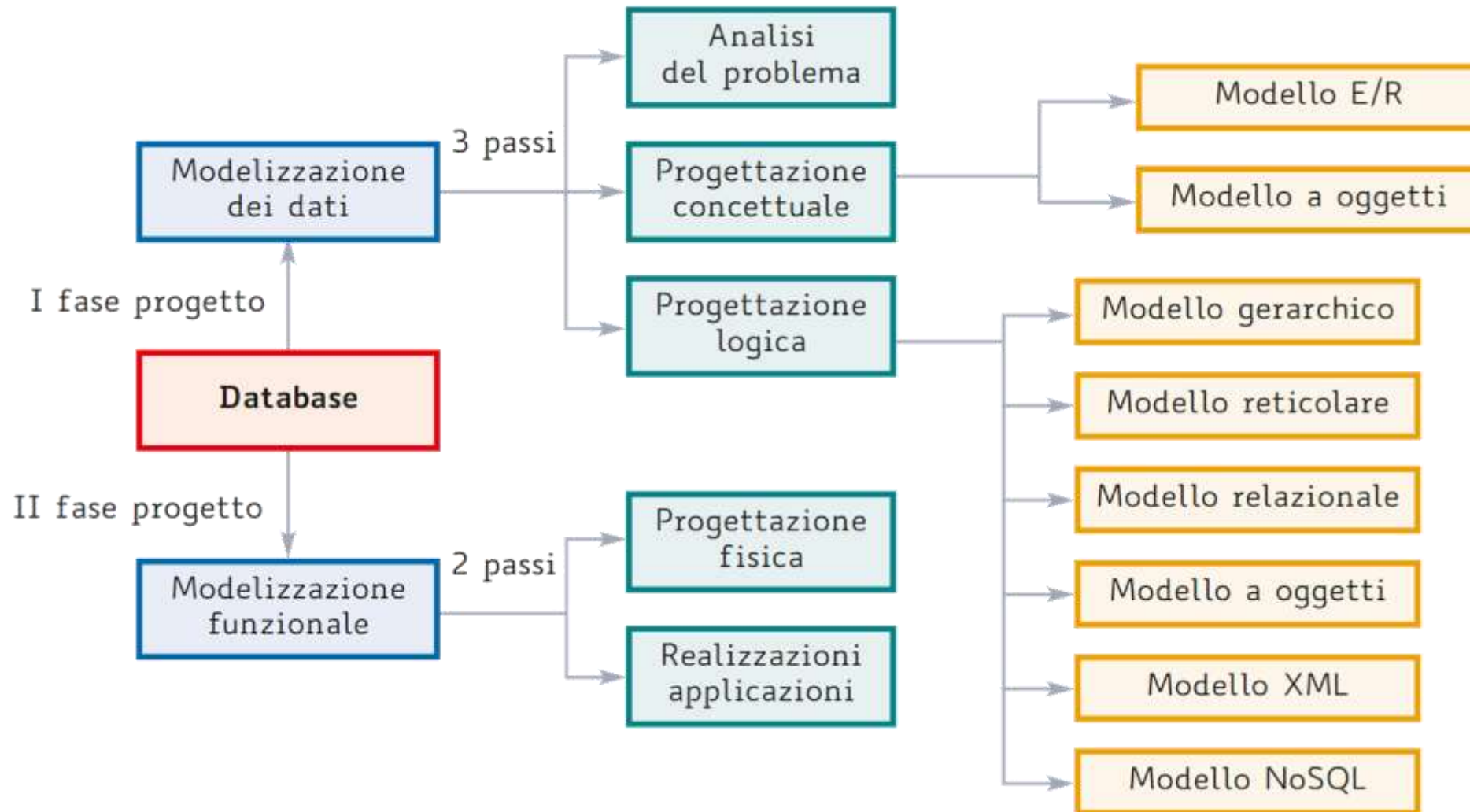
Caratteristiche di un database



- Un database deve essere:
 - **sicuro**: ossia deve impedire che essa possa essere danneggiato da eventi accidentali (come crash di sistema) o da accessi non autorizzati;
 - **consistente**: ossia i dati in esso contenuti devono essere significativi ed effettivamente utilizzabili;
 - **integro**: ossia deve garantire che le operazioni effettuate da utenti autorizzati non possano provocare una perdita di consistenza dei dati;
 - **condivisibile**: ossia applicazioni ed utenti diversi devono poter accedere, secondo opportune modalità, ai dati comuni;
 - **persistente**: ossia deve avere un tempo di vita che non è limitato a quello delle singole esecuzioni dei programmi che lo utilizzano;
 - **efficiente**: l'utilizzo delle risorse (CPU e memoria) deve essere ottimizzato.



Progettazione di database



Esempio di progettazione

➤ Fase 1. Analisi dei requisiti

- **Obiettivo:** Creare un database per gestire la collezione di vini di un cliente.
- **Requisiti:**
 - Registrare informazioni specifiche di ogni bottiglia
 - Tenere traccia della posizione e dello stato
 - Monitorare l'invecchiamento



Esempio di progettazione

➤ Fase 2. Progettazione concettuale

- **Obiettivo:** descrivere cosa deve contenere il database senza preoccuparsi di come verrà implementato.
- **Modello Entità-Relazione:**



Bottiglia
<u>codiceBottiglia</u>
nomeVino
annata
gradazione
tipoVino
cantina
dataAcquisto
prezzoAcquisto
posizioneCantina
stato



Esempio di progettazione

► Fase 3. Progettazione logica

► **Obiettivo:** Traduzione del modello concettuale in uno schema logico (tabelle o altro in base al modello di database scelto)

► **Modello logico:**

Bottiglia
<u>codiceBottiglia</u>
nomeVino
annata
gradazione
tipoVino
cantina
dataAcquisto
prezzoAcquisto
posizioneCantina
stato



Campo	Tipo LOGICO	Descrizione	Vincoli	Dominio
CodiceBottiglia	Codice	Identificativo unico	PRIMARY KEY, NOT NULL	Stringa alfanumerica (10 caratteri)
NomeVino	Testo	Nome commerciale	NOT NULL	Stringa (max 50 caratteri)
Annata	Intero	Anno vendemmia	NOT NULL	Intero (1900-anno corrente)
Gradazione	Decimale	Gradazione alcolica		Numero (0.0-20.0)
TipoVino	Categoria	Tipologia		{Rosso, Bianco, Rosato, Spumante}
Cantina	Testo	Produttore		Stringa (max 50 caratteri)
DataAcquisto	Data	Data acquisto		Data valida
PrezzoAcquisto	Monetario	Prezzo pagato		Numero positivo (2 decimali)
PosizioneCantina	Testo	Collocazione		Stringa (max 20 caratteri)
Stato	Categoria	Stato corrente	DEFAULT 'In cantina'	{In cantina, Bevuta, Danneggiata}



Esempio di progettazione



► Fase 4. Progettazione fisica

- Obiettivo: creazione del database
- Codice SQL:

Campo	Tipo LOGICO	Descrizione	Vincoli	Dominio
CodiceBottiglia	Codice	Identificativo unico	PRIMARY KEY, NOT NULL	Stringa alfanumerica (10 caratteri)
NomeVino	Testo	Nome commerciale	NOT NULL	Stringa (max 50 caratteri)
Annata	Intero	Anno vendemmia	NOT NULL	Intero (1900-anno corrente)
Gradazione	Decimale	Gradazione alcolica		Numero (0.0-20.0)
TipoVino	Categoria	Tipologia		{Rosso, Bianco, Rosato, Spumante}
Cantina	Testo	Produttore		Stringa (max 50 caratteri)
DataAcquisto	Data	Data acquisto		Data valida
PrezzoAcquisto	Monetario	Prezzo pagato		Numero positivo (2 decimali)
PosizioneCantina	Testo	Collocazione		Stringa (max 20 caratteri)
Stato	Categoria	Stato corrente	DEFAULT 'In cantina'	{In cantina, Bevuta, Danneggiata}



```
-- Creazione del database
CREATE DATABASE CantinaVini;
USE CantinaVini;
```

```
-- Creazione della tabella BOTTIGLIE
CREATE TABLE Bottiglie (
  CodiceBottiglia VARCHAR(10) PRIMARY KEY,
  NomeVino VARCHAR(50) NOT NULL,
  Annata INTEGER NOT NULL,
  Gradazione DECIMAL(3,1),
  TipoVino VARCHAR(10) CHECK (TipoVino IN ('Rosso', 'Bianco', 'Rosato', 'Spumante')),
  Cantina VARCHAR(50),
  DataAcquisto DATE,
  PrezzoAcquisto DECIMAL(6,2),
  PosizioneCantina VARCHAR(20),
  Stato VARCHAR(15) DEFAULT 'In cantina'
);
```



Esempio di progettazione



➤ Fase 5. Realizzazione applicazioni

➤ Inserimento dati:

```
INSERT INTO Bottiglie  
VALUES ('BOTT007', 'Chardonnay', 2021, 13.0, 'Bianco', 'Cantina Puglia',  
'2023-03-10', 22.50, 'Scaffale B1', 'In cantina');
```

➤ Interrogazioni:

```
-- Vini rossi con più di 5 anni  
SELECT NomeVino, Annata, Cantina  
FROM Bottiglie  
WHERE TipoVino = 'Rosso' AND Annata <= 2018;
```

```
-- Valore totale della collezione  
SELECT SUM(PrezzoAcquisto) as ValoreTotale  
FROM Bottiglie  
WHERE Stato = 'In cantina';
```

