

Basi Dati
Masciari Elio
a.a. 2023-2024

Author
Alessio Romano

March 31, 2025

Contents

1 Sistemi Informatici	3
1.1 Processo Aziendale	3
1.2 Sistema Informativo	3
1.2.1 Componenti del Sistema Informativo	3
1.3 Produzione dell'Informazione	3
1.3.1 Tipi di Informazioni	3
2 Base di Dati	3
2.1 Architettura ANSI-SPARC	4
2.2 Database Management System (DBMS)	4
3 Modello Relazionale	4
3.1 Vincoli di Integrità	4
4 SQL (Structured Query Language)	5
4.1 Creazione di Tabelle	5
4.2 Gestione Utenti e Privilegi	5
5 Modello ER	5
5.1 Progettazione concettuale	6
5.1.1 Entità	6
5.1.2 Attributi	6
5.1.3 Associazione	7
5.1.4 Cardinalità di un'associazione	7
5.1.5 Identificatore di un'entità	7
5.2 Progettazione Logica	8

1 Sistemi Informatici

1.1 Processo Aziendale

Un processo aziendale è una sequenza di attività aziendali volte alla realizzazione di un servizio o prodotto.

1.2 Sistema Informativo

Un sistema informativo è un insieme di informazioni gestite dai processi aziendali.

1.2.1 Componenti del Sistema Informativo

- Patrimonio di dati (database)
- Insieme di procedure (funzioni nel codice)
- Insieme di risorse umane (ruoli nel dominio)
- Insieme di mezzi e strumenti (tool del database)

1.3 Produzione dell'Informazione

La produzione dell'informazione si basa sui seguenti step:

1. Acquisizione del dato
2. Immagazzinamento del dato
3. Elaborazione del dato
4. Immagazzinamento del dato elaborato
5. Trasmissione verso l'utente finale

1.3.1 Tipi di Informazioni

- **Informazioni Elementari:** si ricavano interpretando i dati di un'organizzazione (senza relazioni con altre informazioni).
- **Informazioni Complesse:** insieme di informazioni correlate logicamente associando un'opportuna semantica ai legami.

2 Base di Dati

Una base di dati è l'insieme di informazioni associate a collezioni di dati:

- Dati correlati tra loro
- Dotati di un'opportuna descrizione

Una base di dati contiene un insieme di dati detti **metadati**, ovvero informazioni riguardanti la base di dati stessa, ad esempio il modo in cui vengono immagazzinati i dati.

2.1 Architettura ANSI-SPARC

L'architettura ANSI-SPARC è un'architettura a tre livelli:

- **Livello esterno (vista logica)**: interazione con le applicazioni, personalizzando l'accesso alle informazioni per singoli utenti o classi di utenti.
- **Livello concettuale (logico)**: organizzazione e collegamento logico dei dati.
- **Livello interno (fisico)**: gestione della memorizzazione su disco fisico.

L'architettura garantisce **indipendenza logica** (mediante viste) e **indipendenza fisica**.

2.2 Database Management System (DBMS)

Un **DBMS** è un insieme di programmi che permette di:

- Definire (tipi, strutture, vincoli sui dati)
- Manipolare (inserire, cancellare, aggiornare, recuperare dati)
- Controllare (accesso ai dati, protezione da guasti e accessi indesiderati)

Le transazioni in un DBMS eseguono operazioni come:

- **SELECT**: interrogazione del database.
- **INSERT**: inserimento di dati nel database.
- **UPDATE**: modifica di dati esistenti.
- **DELETE**: rimozione di dati.

3 Modello Relazionale

Una relazione r sui domini D_i è un sottoinsieme del prodotto cartesiano:

$$r \subseteq D_1 \times D_2 \times \dots \times D_n \quad (1)$$

Le relazioni vengono rappresentate tramite tabelle, con righe che rappresentano tuple e colonne che rappresentano attributi.

3.1 Vincoli di Integrità

Un vincolo di integrità è una regola che ogni istanza di uno schema di relazione deve rispettare.

- **Vincoli Intrarelazionali:**
 - Vincoli di dominio (valori di un attributo)
 - Vincoli di tupla (su più attributi della tupla)
 - Vincoli di chiave (identificazione univoca di una tupla)
- **Vincoli Inter-relazionali:**
 - Associano dati tra diverse relazioni

4 SQL (Structured Query Language)

SQL comprende istruzioni per la definizione e manipolazione dei dati.

4.1 Creazione di Tabelle

```
CREATE TABLE nomeTabella (
    nomeAttributo Dominio [Default] [Vincoli],
    {, nomeAttributo Dominio [Default] [Vincoli]}, 
    [, altriVincoli]
);
```

è possibile applicare controlli sui singoli attributi attraverso la direttiva di controllo CHECK()

```
CREATE TABLE squadre
(
    NOME varchar(50)
    CHECK(NOME = "ASCOLI" OR ... OR NOME = "UDINESE"),
    ANNODIFONDAZIONE integer
    CHECK(ANNODIFONDAZIONE > 1900 AND ANNODIFONDAZIONE < 2025),
    STADIO varchar(100)
    primarykey NOME
)
```

4.2 Gestione Utenti e Privilegi

```
CREATE USER username IDENTIFIED BY password;
GRANT privilegio1, ..., privilegioN ON NomeRisorsa TO username;
```

5 Modello ER

La progettazione di una Base di Dati si divide in tre parti

- **Analisi** delle specifiche
- **Progettazione statica:**
 - Progettazione concettuale
 - Progettazione logica
 - Progettazione fisica
- **Progettazione dinamica:**
 - Realizzazione delle procedure per l'effettuazione di transazioni sulla base di dati

La progettazione statica, passa nel pratico attraverso 2 schemi principali

- **Schema Entità-Relazione**
- **Schema Logico**

5.1 Progettazione concettuale

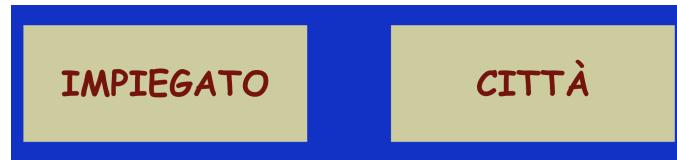
Il **Modello ER** è il principale metodo di progettazione concettuale, esso fornisce una serie di *costrutti* che descrivono la realtà attraverso strutture. Tali strutture consentono di definire gli schemi che descrivono l'organizzazione e la struttura delle istanze di dati. Ci sono 5 costrutti principali

5.1.1 Entità

Un'entità è una classe di oggetti (fatti, persone, cose) dell'applicazione di interesse con proprietà comuni e con esistenza autonoma

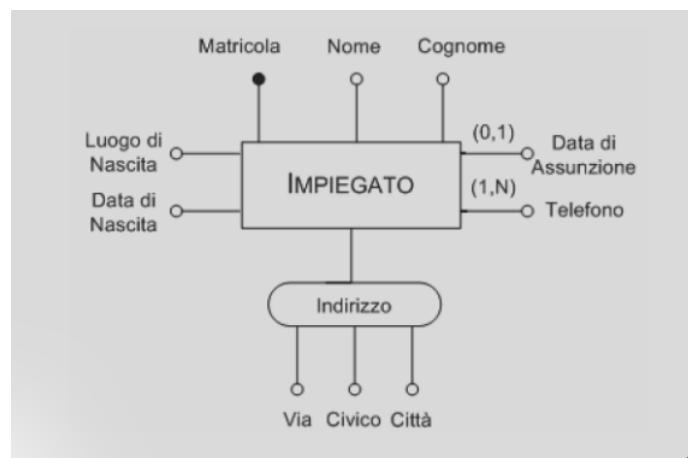
- Ne sono esempi impiegato, città, dipendente...
- Ogni entità ha un nome, di solito singolare che la identifica univocamente nello schema

Nello schema concettuale, si rappresentano le entità, non le singole occorrenze, graficamente si rappresenta nel seguente modo



5.1.2 Attributi

Si intende per attributo di un'entità una sua proprietà descrittiva relativamente al dominio di interesse. Gli attributi sono denotati da una cardinalità che ne definisce il numero (min,max) di occorrenze. Sono rappresentati nel seguente modo



Gli attributi possono essere di diversi tipi

- **Attributi semplici:** sono descritti per mezzo di linee terminate da cerchi e nomi, e sono le proprietà semplici di un'entità o una relazione (es. nome)
- **Identifieri delle entità:** sono usati come strumento per l'identificazione univoca delle occorrenze dell'entità, vengono indicati con cerchietto pieno;
- **Attributi composti:** sono identificati tramite i loro attributi componenti e si rappresentano come un ellisse avente il nome dell'attributo composto, e come attributi i suoi componenti

- **Attributo multivalore:** sono costituiti da un insieme di valori ed indicati tramite due numeri che esprimono la cardinalità dell'attributo
- **Attributi opzionali:** attributi con cardinalità (0,1) e sono usati quando non è necessario specificare un valore dell'attributo in quanto la sua assenza non compromette la significatività del concetto

5.1.3 Associazione

un'associazione è un legame concettuale tra due o più rilevante ai fini dell'applicazione. Ne sono esempi:

- Afferenza(fra impiegato e dipartimento)
- Fornitura(fra fornitore, prodotto e dipartimento)

Ogni associazione ha un nome, che la identifica univocamente nello schema. Viene rappresentata come



Una associazione a sua volta può avere degli attributi, ad esempio studente - esame - corso, l'associazione può prevedere attributi quali DataEsame, Voto ecc

5.1.4 Cardinalità di un'associazione

Si tratta di una coppia di valori relativa a ciascun entità legata ad un'associazione che sintetizza la partecipazione di ciascun'entità alla relazione stessa.



Tali coppie specificano il massimo e il minimo di occorrenze dell'associazione cui ciascun'occorrenza di un'entità può partecipare. Le associazioni si distinguono in

- **Associazione Binaria:** Se si tratta di un'associazione che coinvolge esattamente due entità
- **Associazione N-Aria:** coinvolge più di due entità
- **Associazione Ricorsiva:** coinvolge un'unica entità, se stessa (es: collega - collabora - collega)

5.1.5 Identificatore di un'entità

L'identificatore di un'identità è lo strumento che permette di identificare univocamente le occorrenze di un'entità. Ogni entità ne deve possedere almeno un identificatore. È costituito da

- **attributi dell'entità:** ossia l'identificatore interno (primary key)
- **attributi esterni:** entità esterne collegate mediante relazioni (foreign key)

Se per identificare un'entità si deve far ricorso ad attributi di una (o più) entità ad essa riferite attraverso un'associazione si parla di identificatore esterno. Un identificatore esterno di un'Entità E può esistere solo se la cardinalità della associazione tra E e le altre Entità coinvolte è (1,1).

5.2 Progettazione Logica

La progettazione logica, che segue lo sviluppo del diagramma ER, consiste nel tradurre il modello concettuale in uno schema logico, tipicamente basato sul modello relazionale. Questa fase è fondamentale per definire la struttura dati e le regole di integrità che il DBMS dovrà rispettare. In particolare, si prevede quanto segue:

- **Mappatura del modello ER:**
 - **Entità:** ogni entità viene trasformata in una tabella. Gli attributi diventano le colonne e viene individuata la chiave primaria.
 - **Relazioni:**
 - * **Relazioni uno a uno e uno a molti:** implementate tramite l'inserimento di chiavi esterne nelle tabelle appropriate.
 - * **Relazioni molti a molti:** gestite mediante la creazione di tabelle intermedie (o di giunzione) che contengono le chiavi primarie delle tabelle correlate.
- **Gestione degli attributi particolari:**
 - **Attributi multivалore:** solitamente vengono modellati con tabelle separate per mantenere la relazione con l'entità principale tramite chiavi esterne.
 - **Attributi derivati:** si decide se memorizzarli oppure calcolarli in maniera dinamica.
 - **Entità deboli:** vengono rappresentate con tabelle che, oltre alla chiave parziale, includono la chiave primaria dell'entità "padre" per garantire l'unicità.
- **Definizione dei vincoli di integrità:**
 - Stabilire le chiavi primarie ed esterne per garantire le relazioni tra le tabelle.
 - Definire vincoli di dominio e regole di integrità referenziale per assicurare la coerenza dei dati.
- **Normalizzazione:**
 - Eliminare ridondanze e anomalie (inserimento, aggiornamento e cancellazione) attraverso il rispetto delle forme normali (prima, seconda, terza e, se necessario, forme superiori).
- **Documentazione e validazione:**
 - Redigere una documentazione dettagliata dello schema logico, descrivendo tabelle, attributi, chiavi e vincoli.
 - Validare lo schema per verificare che risponda correttamente alle esigenze del dominio applicativo.
- **Iterazione e ottimizzazione:**

- Revisionare e perfezionare lo schema logico in base ai feedback degli stakeholder e alle nuove esigenze, anche considerando possibili denormalizzazioni controllate per migliorare le prestazioni.

Questo processo garantisce che lo schema logico sia solido, coerente e pronto per essere implementato nel DBMS, assicurando al contempo l'integrità e l'efficienza nella gestione dei dati.