

appunti_informatica_3

I database e i sistemi informativi

un **sistema informativo** è l'insieme di tutte le *risorse umane, strumenti e procedure*, con il compito di *raccogliere, memorizzare ed elaborare i dati*, per produrre informazioni per soggetti interni ed esterni all'attività di un'azienda.

Invece, il **sistema informatico** rappresenta l'insieme delle risorse hardware e software.

Il *sistema informatico* viene integrato nel *sistema informativo*, creando il **sistema informativo automatizzato**.

Durante la fase iniziale della transizione verso questi sistemi, era comune il fenomeno delle **isole informatiche**, dove molte informazioni erano isolate su singole postazioni dell'azienda.

I **database**, anche chiamati *archivi*, sono degli insiemi di **informazioni**.

Sono degli insiemi logicamente organizzati riferiti a un unico soggetto. (detto **record**)

Le singole informazioni sono dette **campi**.

In passato gli *archivi* erano soprattutto in formato *cartaceo*, che hanno problemi di **ridondanza** (*duplicazione inutile dei dati*) e **disponibilità** (*accesso ai dati*).

Verranno introdotte le **basi di dati**, o anche *database*, che sono invece archivi in formato digitale.

Hanno molti vantaggi come:

- **disponibilità maggiore**, perché più persone possono accedere ai dati contemporaneamente e da luoghi diversi
 - **Integrità**, perché i dati sono più controllati e aggiornati
 - **Sicurezza**, grazie a una miglior gestione dei permessi
- **ridondanza ridotta**, perché si evita che gli stessi dati siano memorizzati più volte

I *database* sono gestiti tramite i **DBMS**.

*colui che gestisce il database viene chiamato **DBA*** (*DataBase Administrator*)

Questi software utilizzano diversi linguaggi, come:

- **DDL**, *Data Definition Language*, per definire la struttura dell'archivio e i suoi vincoli
- **DML**, *Data Manipulation Language*, per la manipolazione dei dati nel database (*inserimento, aggiornamento e rimozione*)

- **QL**, *Query Language*, per estrarre informazioni dal database

Sono stati successivamente introdotti *linguaggi non procedurali* ma dichiarativi.

Il più popolare è il linguaggio **SQL**, *Structured Query Language*, detto anche linguaggio per database completo perché si possono usare funzionalità di tutti i tipi. (**DDL**, **DML** e **QL**)

(Utilizzato da DBMS popolari come MariaDB e PostgreSQL)

La realizzazione di un archivio avviene in 4 fasi:

1. Analisi della realtà di riferimento

2. Modello concettuale

Tramite strutture come il *modello E/R*.

Il **Modello E/R** è composto da **Entità** e **Relazioni**.

- Le *entità* sono quindi i soggetti degli attributi, che ne definiscono le caratteristiche.
- Le *relazioni* sono il legame logico tra due o più entità.
 - La *cardinalità* specifica il numero minimo e massimo di occorrenze che un'entità può essere associata ad un'occorrenza di un'altra entità

3. Progettazione logica

Una volta realizzato il *modello concettuale*, esso si *converte* in un *modello logico*, più vicino alla struttura finale del database nel *DBMS*.

4. Progettazione fisica

Vincoli

Sono delle regole che possono essere applicati ad uno o più campi

Intra-Relazionali

Vincoli che interagiscono su una sola entità.

Vincoli espressi con condizioni logiche, che devono essere soddisfatti all'interno di una singola relazione.

Ci sono 2 tipi principali:

- **Di dominio (o di chiave)**
Sono dei vincoli che riguardano i singoli campi della tupla, come la **chiave primaria**
La chiave primaria è composta dai vincoli *unique* e *not null*.
- **Di tupla**
Vincoli tra i valori di più campi della stessa tupla.
es. si può avere la lode solo se si prende 30

Altri 2 vincoli che abbiamo sono *UNIQUE* e *NOT NULL*.

- *UNIQUE* non permette valori duplicati ad un'attributo
- *NOT NULL* non permette valori nulli ad un'attributo

Inter-Relazionali

Abbiamo il *vincolo di integrità referenziale*, usato per collegare 2 entità diverse tra loro.

Gerarchie di Specializzazione/Generalizzazione

Strategie/strumento utilizzato quando abbiamo più entità che hanno attributi in comune. Sarà definita un'**entità padre**, composta da tutti gli *attributi comuni*, che avrà **entità figlie**, composte dagli *attributi non-comuni*. Queste gerarchie hanno **2 vincoli**.

Vincoli

Copertura

Ci sono 2 tipi di **vincoli di copertura**:

- **Totale**
Una gerarchia di tipo *totale* si ha quando le entità figlie coprono tutti i casi. (es. genitore 1 e 2 in una famiglia)
- **Parziale**
Una gerarchia di tipo *parziale* si ha quando le entità figlie *non* coprono tutti i casi. (es. staff scolastico diviso solo in professori e personale ATA)

Disgiunzione

Ci sono 2 tipi di **vincoli di disgiunzione**:

- **Disgiunta**
Si dice *disgiunta* quando l'entità può essere di un solo tipo. (es. un pagamento può essere fatto solo in contanti o con carta)
- **Sovrapposta**
Si dice *sovrapposta* quando un l'entità può essere di più tipi. (es. una persona può essere sia studente che lavoratore)

Con il modello ER Semplificato

Le gerarchie possono essere convertite nel modello ER semplificato con 3 possibili soluzioni.

Accorpamento delle sottoclassi nella superclasse

Gli attributi delle entità figlie sono aggiunti nell'entità padre come attributi opzionali, assieme a un nuovo attributo *tipo*. (che indica il tipo di entità figlia della tupla)

Accorpamento della superclasse nelle sottoclassi

Tutti gli attributi dell'entità padre sono aggiunti alle entità figlie. (Soprattutto ottimale per il caso disgiunta, perché le informazioni dell'entità padre vengono perse)

Sostituzione della generalizzazione con tante relazioni

La gerarchia viene riscritta sotto forma di tante relazioni tra le entità.

Normalizzazione

La normalizzazione delle tabelle consiste nella creazione di un database ottimizzato per ridurre le **ridondanze** e le **anomalie**.

Ci sono 3 tipi di *anomalie*:

- anomalie di **cancellazione**, cioè, l'errata cancellazione di altri dati o della tupla.
- anomalie di **aggiornamento**, cioè, che l'aggiornamento di un dato sia fatto correttamente anche nelle sue relazioni.
- anomalie di **inserimento**, cioè, che il dato sia inserito correttamente, rispettando i propri vincoli.

Forme normali

Le forme normali sono dei metodi per ridurre la possibilità di anomalie.

1FN

Prevede che **tutti gli attributi siano di dominio semplice (atomico)**.

Cioè, che devono contenere una *sola informazione*. (non dobbiamo avere attributi composti, come *indirizzo*)

2FN

Dipende dalla 1FN.

Una relazione si dice in 2FN se tutti gli attributi non-chiave dipendono dalla chiave completa, e non solo da parte di essa.

Dipendenza funzionale

Si considera una relazione R in cui ci sono gli attributi X e Y .

Se per ogni coppia di tuple di R con gli stessi valori su X , $T1$ e $T2$ hanno gli stessi valori anche su Y . Si dice che X determina funzionalmente Y , o è determinato funzionalmente da Y .

Quando alcune colonne hanno gli stessi valori, cioè che gli attributi hanno gli stessi valori in più tuple, vuol dire che uno degli attributi ha un valore univoco che determina il valore di un altro attributo. (Cioè una dipendenza funzionale)

3FN

Dipende dalla 1FN e la 2FN.

Nella 3FN vengono eliminate le dipendenze funzionali transitive.

Dipendenza funzionale transitiva

Abbiamo 3 attributi: **A**, **B** e **C**.

- **A** è una *chiave primaria*
- **B** e **C** sono *attributi non-chiave*

Se **C** è determinato da **B**, e **B** è determinato da **A**, vuol dire **C** dipende transitivamente da **A**.