appunti_verifica_informatica_1

1. Gli archivi

Un **archivio** è un insieme formato dalle **informazioni**.
Le **informazioni** sono un insieme di **dati**, hanno *senso compiuto*.
Ha alcune caratteristiche fondamentali:

- esiste un **nesso logico** tra di esse (sono dello stesso argomento);
- Sono rappresentate secondo un **formato**, che ne rende possibile l'interpretazione;
- Sono registrate con un **supporto** su cui si possono scrivere e rileggere le informazioni, anche a distanza di tempo; (es. archivio cartaceo, database digitale)
- Sono **organizzate** in modo da essere facilmente consultabili;

In un archivio queste informazioni vengono raggruppate secondo un'**unità logica** (es. elenco telefonico, i dati relativi a un singolo abbonato). Formano un unico soggetto chiamato **record**.
Le singole informazioni che lo compongono si chiamano **campi**.
L'elenco dei campi che lo compongono viene chiamato **tracciato del record**.

La **creazione di un archivio** richiede vari parametri preliminari:

- **Nome dell'archivio**, che lo identifica.
- **Tracciato record**, cioè le informazioni che comporranno il record
- Il tipo di **supporto** (carta, nastri magnetici...)
- La **dimensione massima** dell'archivio
- La **struttura/organizzazione** dei dati

In un archivio si possono eseguire **2 operazioni**:

- **Manipolazione**, cioè l'*aggiunta* di nuovi dati, o *modifiche* a dati già inseriti;
- **Interrogazione**, quando si *reperiscono informazioni* dall'archivio;

2. I file e le memorie di massa

Questi archivi di record vengono memorizzati in un **file**. I file possono contenere **qualsiasi tipo di informazione**, come testuale, ma anche multimediale.
Nelle procedure gestionali, gli *archivi* sono costituiti da un **insieme di record omogenei**, che vengono chiamati *record logici*.

I supporti usati per memorizzare tutti questi dati vengono chiamati **memorie di massa**.
*Anche chiamate **memorie ausiliari**, perché estendono la memoria centrale del computer, e consentono la permanenza dei file nel tempo*.
Su queste memorie vengono eseguite le operazioni di *lettura* (**Input**) e *scrittura* (**Output**),

vengono quindi anche chiamate *unità di I/O*.
Hanno *4 caratteristiche principali*:

- **Tipo di accesso** ai dati, *diretto* (o random) come nei dischi o *sequenziale* come nei nastri;
- **Capacità**, cioè la quantità di dati che possono contenere. Misurata in *GB/TB*;
- **Tempo medio di accesso**, misurato in *ms*, costituisce il tempo medio necessario per trovare i dati e trasferirli nell'unità centrale;
- **Velocità di trasferimento**, dalla memoria di massa all'unità centrale. Misurata in *MB/s*.
  Le unità periferiche che contengono i supporti di memorizzazione, sono gestite dal **drive** (hardware), collegate al computer tramite *schede di controllo*, che consentono la *lettura e scrittura dei dati*.
  Il *driver* è invece il software di gestione della periferica.

Nei computer moderni i dati sono per lo più codificati in **ASCII esteso**, codifica *8 bit*. Non basta, però, per rappresentare i tanti simboli dei diversi alfabeti, simboli di interpunzione e simboli matematici.
Viene quindi usata la codifica **Unicode**, a *16 bit*.
Che include la stessa codifica per i caratteri *ASCII standard*.

Vengono anche inclusi i *bit di controllo*, anche chiamati **bit di parità**.
Durante la registrazione, viene aggiunto un 1 o un 0, per rendere *pari* o *dispari* il numero di bit con valore 1.
Nella *fase di lettura*, questo bit sarà ricalcolato, e se non è corretto, sarà segnalato un errore di parità.
È in grado di riconoscere un solo errore e viene chiamato **controllo di parità**.

I trasferimenti I/O tra le periferiche e la memoria centrale non è di singoli caratteri, ma di **blocchi**. la loro dimensione varia tra i *kilobyte*.
Nella fase di lettura i dati che vengono trasferiti nella memoria centrale finiscono nel **buffer di I/O** dove, i dati aggiunti o modificati, nel buffer, saranno ricopiati sul disco in operazioni di scrittura.
Il blocco è l'*unità fisica* della memorizzazione dei dati.
Possono anche contenere più record logici, che permette quindi di ridurre il numero di accessi alla periferica.
Si definisce **fattore di blocco** di un file il numero di record logici contenuti in un blocco.
I file con record possono essere a **lunghezza fissa**, o a **lunghezza variabile**, dove bisogna, specificare la fine di ogni record.
Può essere indicata in 2 modi:

- Viene indicato all'interno di ogni record il *numero di caratteri*.
- Viene usata una *sequenza speciale* (Come i 2 caratteri ASCII, *Carriage Return* e *Line Feed*. CR e LF).

Il sistema operativo gestisce i file tramite il modulo **file system**.
Gestisce *l'organizzazione, l'assegnamento, la protezione e il ritrovamento di insiemi di dati*.
In particolare svolge queste funzioni:

- Tiene traccia dei file, della *posizione*, dello *stato*, usando le strutture dati *file directory*;
- Decide secondo le richieste, le *protezioni e i diritti di accesso*, a quali programmi assegnare un file o una sua parte.
- Gestisce l'assegnazione dei file ai programmi, togliendoli in caso siano necessari ad altri programmi.

Il file system consente di riferirsi ai file tramite le *directory* o *tabelle dei descrittori*, che includono per ognuno un **descrittore del file**, formato da:

- Nome
- Tipo (testo, sottodirectory, eseguibile, di sistema...)
- Tipo di protezione e accessi consentiti (lettura, scrittura, esecuzione, nascosto, di sistema...)
- Nome dell'autore e del modificatore
- Dimensione del file in byte
- Informazioni necessarie per trovarlo su disco (indirizzo del primo blocco di memoria del file, o una struttura dati con le informazioni sulla collocazione nel disco...)
- Data di creazione, ultima modifica e ultimo accesso
  I file possono essere bloccati con password.

5. Basi dati

I **database** sono insiemi digitali di dati, progettati con tecniche di modellazione dei dati e gestiti con appositi software.
Negli anni sono stati creati software che hanno migliorato l'**efficienza** e la **produttività** dell'organizzazione degli archivi.
Quando si parla di efficienza e produttività, si intende:

- cercare le informazioni desiderate, anche con diversi criteri di ricerca e scopi diversi.
- Gestire i dati con una buona velocità di elaborazione.
  - Garantire la sicurezza dei dati e l'**integrità** delle registrazioni.
  - **Integrità** significa garantire che le operazioni effettuate sul database non provochino una perdita di consistenza ai dati.
  - Garantire **consistenza** degli archivi significa assicurare che siano significativi e effettivamente utilizzabili nelle applicazioni.

I **DBMS** sono un software usato per la gestione dei database, permettendo all'utente di concentrarsi sulla gestione/progettazione di esso, senza preoccuparsi dell'organizzazione fisica dei dati.
L'**utente finale** è colui che usa le informazioni contenute all'interno del database.

Colui che li gestisce è chiamato **amministratore del database**, che si occupa della loro progettazione e manutenzione.
Il **database**, invece, è solo un'*insieme di dati*.

è anche comune l'uso di **database distribuiti**, dove più nodi archivi possono essere anche lontani tra di loro.
Possono avere basi dati su più memorie di massa, o una sola.
Permettono da dovunque l'esecuzione di lavori di manutenzione e controllo sulla sicurezza, sempre garantendo la disponibilità immediata dei dati a tutti gli utenti, qualunque sia la loro posizione.

7. Organizzazione degli archivi, mediante basi di dati

Le funzioni del **DBMS** si dividono in vari linguaggi:

- **DDL** (*Data Definition Language*), che viene usato per creare nuovi archivi, gestire l'accesso e impostare vincoli.
  - **DCL** (*Data Control Language*)
- **DML** (*Data Manipulation Language*), usato per l'inserimento, la cancellazione e variazione dei dati nel database.
- **QL** (*Query Language*), usato per interrogare/estrarre dati dal database.

I DBMS devono anche risolvere i problemi tipici dell'approccio **file-based**.
Di conseguenza, sono caratterizzati da varie proprietà:

- **Eliminazione della ridondanza e inconsistenza**, il database non può presentare campi uguali con valori diversi in archivi diversi.
  (Essendo costituito da archivi integrati di dati, nel database gli stessi dati non compaiono più volte in archivi diversi)
- **Facilitare l'accesso ai dati**, il ritrovamento dei dati è facile e trasparente al programmatore, e svolto con grande velocità, anche nel caso di database molto grandi, con richieste contemporanee da più utenti.
- **Interrogazioni non predefinite**, dev'essere possibile interrogare i dati con un linguaggio semplice, anche mediante interfacce grafiche intuitive.
- **Integrità dei dati**, i *vincoli di integrità* vengono memorizzati nel database e controllati dal DBMS, in modo evitare anomalie ai dati causate da programmi e le applicazioni dell'utente.
- **Indipendenza dalla struttura logica e fisica dei dati**, dev'essere possibile apportare modifiche alla definizione delle strutture della base di dati senza modificare il software applicativo. E viceversa, dev'essere possibile modificare i supporti su cui sono registrati i dati e le modalità di accesso alla memoria di massa, senza dover apportare cambiamenti alle applicazioni.
- **Utilizzo da più utenti e controllo della concorrenza**, i dati devono poter essere usati da più utenti, consentendo anche una visione solo parziale del database ai singoli utenti.

(Il DBMS deve garantire che le operazioni svolte in concorrenza non interferiscano tra di loro)
- **Sicurezza dei dati**, prevede procedure di controllo per impedire accessi non autorizzati.

I *database* memorizzano anche un **dizionario dei dati**, formato dai **metadati**, cioè dati che descrivono i dati.
Questo garantisce l'indipendenza logica dei programmi dai dati.
Il DBMS tramite il dizionario è in grado di:

- Realizzare controlli per garantire l'integrità
- Autorizzare gli utenti in base alla politiche definite per l'accesso
- Fornire servizi per il controllo della consistenza

I DBMS offrono anche **interfacce** per facilitare l'interazione tra l'utente e il software di gestione.
Spesso includono ambienti software che semplificano la creazione di **applicazioni** che interrogano e manipolano il database.

11. I linguaggi dei database

Lo sviluppo delle tecniche di gestione ha portato alla creazione di DBMS per basi di dati relazionali. (**RDBMS**)
Non si usa un **linguaggio procedurale**, ma **dichiarativo**. Rimuove la necessità dell'utente di sapere come sono stati registrati i dati e dove trovarli.
Vengono chiamati **linguaggio per basi di dati**.
Le funzioni dei *DDL*, *DML* e *QL* vengono unificate in un unico linguaggio.
I tipici linguaggi usati sono **SQL** e **QBE** (*Query By Example*)

- *QBE* permette di eseguire interrogazioni dichiarando, in un'apposita maschera, come si deve presentare la risposta alla query. Nell'output si possono effettuare calcoli partendo dai valori dei campi, o esprimere condizioni che dovranno essere verificate sui valori.
- *SQL* è, invece, un *linguaggio per database completo*. Si possono, infatti, sia *definire*, *manipolare* e *interrogare* dati.

ESEMPIO

SQL usato come **DDL** (linguaggio di definizione dei dati) per creare lo schema di una tabella di nome *Dipartimenti*:

```
CREATE TABLE Dipartimenti (
CodDip            CHAR(10)   PRIMARY KEY,
NomeDipartimento  CHAR(30),
Indirizzo         CHAR(50),
);
```

ESEMPIO

SQL usato come **DML** (linguaggio di manipolazione dei dati) per inserire i dati del dipartimento di Medicina nella tabella *Dipartimenti*:

```
INSERT INTO Dipartimenti(CodDip, NomeDipartimento, Indirizzo)
VALUES ('Med', 'Medicina', 'Piazza Mentana');
```

ESEMPIO

SQL usato come **QL** (linguaggio di interrogazione) per estrarre l'elenco degli studenti del dipartimento di Giurisprudenza; l'elenco contiene cognome, nome e numero di matricola:

```
SELECT Cognome, Nome, Matricola
FROM Studenti
WHERE CodDip = 'Giu';
```

Il comando SELECT cerca nella tabella *Studenti* le righe per le quali vale la condizione *CodDip* = *'Giu'* e, di queste righe, mostra i valori dei campi *Cognome*, *Nome* e *Matricola*.

Questi linguaggi si basano sulla visione tabellare dei dati.
I comandi del linguaggio operano su gruppi di righe o l'intera tabella, invece che una riga per volta.
Sono semplificate le operazioni per mettere in connessione tra loro tabelle diverse e per presentare o stampare risultati di interrogazioni, ben impaginati, per facilitarne la lettura e la comprensione.

I software DBMS, soprattutto le versioni per personal computer, offrono interfacce utente, con le quali si può interagire usando la lingua nazionale, offrono menu per selezionare i comandi, sotto menu, con le modalità caratteristiche di un'interfaccia grafica.

MODELLO ER

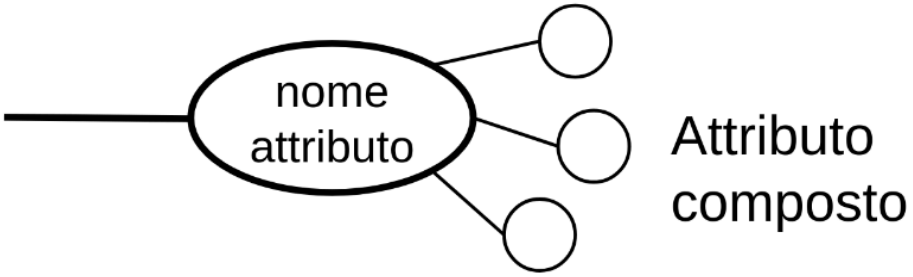
Modello concettuale per progettare il database, basato sulle entità e relazioni.
L'entità è il soggetto, il record, il soggetto a cui vengono attribuite proprietà.
Le relazioni stabiliscono invece i legami tra le entità.

- **Schema portante**, la struttura delle entità e delle relazioni.

Nel Modello ER le entità possono avere **6 tipi di attributi**:

- **Attributo semplice**

  Proprietà/caratteristica dell'entità.
- **Attributo composto**



  Proprietà/caratteristica dell'entità *formata da più informazioni*.
  Si aggiunge un simbolo di attributo semplice per ogni informazione che compone l'attributo composto.
- **Chiave primaria**

  Un attributo *univoco*. *Non può essere nullo*.
- **Chiave esterna**
  Rappresentata da una linea che interseca l'attributo e la linea di relazione all'entità con la chiave primaria.
- **Chiave opzionale**
  Chiave con *cardinalità (0,1)* ,cioè che può avere anche un valore **nullo**.
  (Viene definita al momento di creazione del database)
- **Chiave multivalore**
  Chiave con *cardinalità (1,N)*. È un attributo che può assumere più di un valore.

Anche le relazioni possono avere attributi.
Infatti, la cardinalità può anche essere intesa per relazioni con lo stesso valore dell'attributo.

Le **Cardinalità** vengono usate per definire un *vincolo* sull'entità, per definire quante relazioni può avere con altre entità.

MODELLO ER SEMPLIFICATO

È una versione semplificata, compatibile con il **modello logico**.

Tanti attributi subiscono trasformazioni:

- Gli elementi degli **Attributi composti** diventano tanti **attributi semplici**.
- La **Chiave multivalore** viene trasformata in una nuova **entità**.
  (La nuova entità ha cardinalità (1,1), mentre l'entità padre ha cardinalità (1, N))
- L'**Attributo opzionale** rimane, perché viene definito al momento di creazione del database che può assumere valori nulli