I DATABASE

UN DATABASE E' UN MODELLO (SU COMPUTER) DELLA REALTA'.

Gli archivi costituiscono una memoria di lavoro indispensabile per gestire quantità ingenti di informazioni, per ordinare gli elementi utili, metterli in collegamento e selezionare i dati che devono essere utilizzati nelle varie circostanze.

La creazione di software specifici per la gestione di banche dati, chiamati **DBMS (Data Base Management System)**, ha permesso di unificare in un unico programma applicativo le funzionalità di archiviazione e gestione dei dati.

DBMS

Un Data Base Management System è una raccolta di strumenti che permettono di strutturare, manipolare e gestire i dati di un database che deve possedere almeno le seguenti caratteristiche:

- **Strumenti per la definizione dei dati**
- **Strumenti per l'immissione e la modifica dei dati**
- **Strumenti per l'interrogazione l'estrazione dei dati**
- **Strumenti per la generazione di report**
- **Strumenti di programmazione**
- **Strumenti di amministrazione**

MODELLI DI DATABASE

Esistono vari modelli organizzativi di database, ossia basi teoriche a cui si conformano i progettisti dei DBMS.

- Il modello gerarchico. Trattazione omessa.
- Il modello reticolare. Trattazione omessa.

Il modello **RELAZIONALE** è quello che trattiamo in questo corso

MODELLO RELAZIONALE

Il nome RELAZIONALE deriva dalle **TABELLE**, che sono il modo, facile ed intuitivo, scelto per rappresentare gli "insiemi", le "collezioni" di dati.

Proprio la teoria degli insiemi, dove il termine "relazione" è sinonimo di "tabella" sta alla base di questo modello.

MODELLO RELAZIONALE

Origini del modello relazionale
Il modello relazionale è stato proposto dal matematico **Edgar Codd**, un ricercatore dell'IBM, alla fine degli anni 60. La vera forza del modello relazionale è che (almeno finora) esso è l'unico modello definito per mezzo di rigorosi principi matematici astratti. La grande innovazione di Edgar Codd, infatti, era l'aver intuito che **il modello organizzativo dei dati poteva essere basata sulla teoria matematica degli insiemi** e che si potevano avere risultati rigorosamente prevedibili tramite le normali operazioni insiemistiche applicate ai dati di un database.
In altri termini, il modello relazionale può essere dedotto a partire da principi primi o assiomi matematici.

MODELLO RELAZIONALE

Progettazione (disegno o design)
di un database relazionale.
La teoria relazionale di Codd ha il grande pregio della formalizzazione, e quindi rigorosità, matematica. Però metterla in pratica nella sua interezza si è dimostrato in concreto impossibile. Ecco quindi che nel 1976 è stata elaborata da Peter Pin-Shan Chen una diversa proposta, che sarà quella da noi adottata:

Il modello Entità – Relazioni (E/R)

"**The Entity – Relationsip Model** – Toward A Unified View Of Data" (Marzo 1976).

E/R – ENTITA'

Entità

Il primo passo della metodologia E/R consiste nell'individuare le Entità, cioè quelle cose o quegli oggetti del mondo reale (ricordiamo la definizione iniziale di database!) dei quali abbiamo necessità di conoscere delle informazioni.

Questo, all'interno dell'E/R è il punto più critico in quanto il concetto di entità risulta scarsamente formalizzato. Si tratta di un concetto non definito in termini rigorosamente matematici, bensì fortemente intuitivo....

E/R – ENTITA'

… Questa caratteristica fa sì che l'intera metodologia posta dall'E/R risulti scarsamente scientifica e che quindi nonostante l'insieme dei passi da seguire per pervenire allo schema E/R sia molto ben definito, l'effetto finale è che non esiste uno schema E/R "giusto", nel senso che l'analisi di persone diverse potrebbe portare allo sviluppo di schemi differenti, in quanto già i punti di partenza potrebbero essere diversi (potrebbero essere cioè diverse le entità individuate in questa prima fase).
Esempi di entità : le Persone, i Prodotti, le Ditte, i Clienti, i Soci del club, ecc.(si usa scrivere con la maiuscola)

E/R – ATTRIBUTI

Attributo

Il secondo passo della metodologia E/R consiste nell'identificare e descrivere l'insieme delle caratteristiche degli elementi, ossia degli oggetti, appartenenti a ciascuna entità. Tali caratteristiche si dicono attributi.

Esempi : nome-cognome-data di nascita delle Persone; codice-costo-descrizione dei Prodotti; ecc.

E/R – RELAZIONI

Relazioni (legami)

Il terzo passo della metodologia E/R consiste nell'identificare e descrivere i legami, i collegamenti, le correlazioni fra le entità, che per uniformità con la terminologia di Access/Base chiameremo RELAZIONI.

Esempi : fra le Ditte e i Prodotti ci potrebbe essere il legame (la relazione) "quale ditta produce un certo prodotto", oppure fra Clienti e Fatture "a quale cliente è stata emessa una fattura", ecc.

E/R – RELAZIONI

I legami possono essere di diversa natura:

- **uno a uno**
- **uno a molti (molti a uno)**
- **molti a molti**
- (ricorsività che non trattiamo)

Questo concetto è basilare e compreso a fondo per capir database !!

E/R – RELAZIONI

Un esempio di legami **uno a molti** è quello che intercorre tra un cliente e un ordine che ha fatto (es. il sig Pinco e il suo ordine di bibite). Infatti un ordine fa capo ad un unico cliente, ma quel cliente può fare molti ordini.

Invece il legame tra potrebbe intercorrere fra due ipotetiche entità, le Persone e i Telefoni, sarebbe un legame **molti a molti** in quanto una persona può avere più telefoni e ad uno stesso telefono, possono corrispondere più persone.

Un tipico legame **uno a uno** è quello tra una persona e una carta di identità: una persona può avere una carta di identità, una carta di identità deve appartenere ad una sola persona.

E/R – RELAZIONI

Nell'esempio “un Cliente fa MOLTI Ordini, mentre un Ordine proviene da UN SOLO Cliente” abbiamo detto che c'è una relazione **1 a M.**

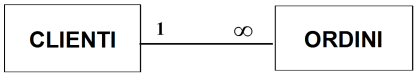
Però **ATTENZIONE** ! Nel giudicare il tipo delle relazioni si deve pensare alla POSSIBILITA' che l'evento reale accada. Per es un Cliente PUO' fare Ordini, non è detto che tutti i clienti li facciano. Cioè la condizione è sufficiente, ma non necessaria. In compenso un Ordine NON PUO' appartenere a più Clienti, quindi NON c'è un legame 1 a M fra Ordine e Cliente.

Altro esempio : 1 Ordine (contiene) M prodotti, ma anche 1 Prodotto (sta in) M Ordini. Si tratta di un legame **Molti a Molti.**

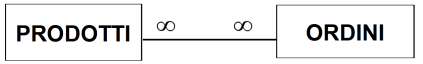
RELAZIONI RAPPRESENTAZIONE GRAFICA

Legami : rappresentazioni grafiche

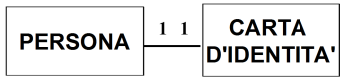
Uno a Molti → 1 - M



Molti a Molti → M – M



Uno a Uno → 1 - 1



CHIAVE PRIMARIA PRIMARY KEY (PK)

Gli oggetti appartenenti ad una entità devono essere identificabili in maniera univoca, in modo ciascuno di loro risulti separato e identificabile da tutti gli altri della stessa entità. P.es. nell'entità Persone ogni persona si deve distinguere dalle altre. A tale scopo è necessario individuare all'interno di ciascuna entità un attributo tale per cui ogni oggetto ha un valore diverso da tutti gli altri (p.es. il codice fiscale delle persone). Tale attributo viene chiamato

CHIAVE PRIMARIA
(o Primary Key, abbreviato PK)

CHIAVE PRIMARIA PRIMARY KEY (PK)

Talvolta l'attributo opportuno non esiste: in tal caso si sceglie una combinazione di attributi (p.es. nome+cognome+data di nascita), oppure si “inventa” un nuovo attributo atto allo scopo (p.es. un numero di matricola o un semplice contatore progressivo).

Per quanto sia possibile creare entità prive di PK (p.es. Tabelle di decodifica), si sconsiglia vivamente di farlo (e noi non lo faremo MAI), ma specialmente, come vedremo,
la PRESENZA di PK è l'elemento INDISPENSABILE, all'atto pratico, per definire le RELAZIONI fra entità.

REALIZZAZIONE DI UN DATABASE IN UN DBMS

Come si poteva intuire già dalle slides introduttive, le ENTITA' vengono realizzate, all'atto pratico nei DBMS, tramite delle **TABELLE**, in cui gli **ATTRIBUTI** sono le **COLONNE** (in perfetta analogia con i fogli di calcolo), mentre le **RIGHE** sono gli oggetti appartenenti all'entità.

In terminologia database :

- **Le colonne sono chiamate CAMPI**
- **Le righe sono chiamate RECORD**
- **Uno o più campi costituiscono la PK**

REALIZZAZIONE DELLE RELAZIONI NEI DBMS

Le relazioni 1 → M sono molto frequenti nei database, e vengono definite in Access/Base mediante un metodo grafico, però abbisognano di una operazione preliminare a livello di progettazione.

Per realizzare una 1 → M si deve aggiungere alla tabella lato Molti un NUOVO CAMPO, (cioè colonna) corrispondente alla PK del lato Uno, che viene chiamato

CHIAVE ESTERNA (Foreign Key o FK).

P.es.il Codice (fiscale) del cliente va aggiunto – come nuovo campo - alla tabella Ordini

REALIZZAZIONE DELLE RELAZIONI NEI DBMS

Non è necessario che il campo FK abbia lo stesso nome del campo PK, la corrispondenza NON avviene in base al nome. Consiglio, anzi, di usare per la FK un nome che richiami la tabella della PK (specie quando il nome della PK è generico, tipo "Codice"). Quindi ATTENZIONE al significato delle chiavi e non al loro nome. Come detto, in Base/Access la realizzazione pratica delle relazioni si fa con un metodo grafico che prescinde dai nomi dati ai campi.

REALIZZAZIONE DELLE RELAZIONI NEI DBMS

Le relazioni M ←→ M non sono hanno una rappresentazione diretta nel DBMS: anche qui è necessaria una **soluzione preliminare a livello di progettazione**.

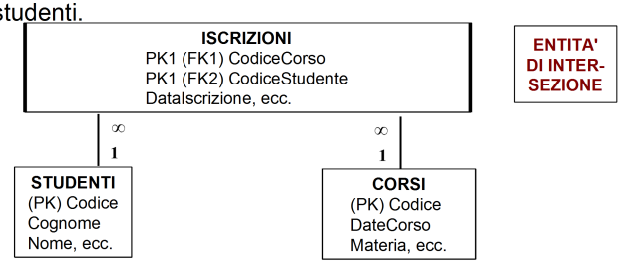
Bisogna individuare una terza tabella, da inserire in mezzo alle due, la quale avrà una relazione 1 → M con ciascuna delle due tabelle di partenza....(segue)

RELAZIONI M - M

Per realizzare una M←→M si deve AGGIUNGERE UNA NUOVA tabella ("di intersezione") contenente, come FK, le 2 PK delle tabelle di partenza. Tabella che si troverà dal lato Molti di 2 relazioni 1 → M .
La PK della tabella di intersezione sarà costituita dalla combinazione delle 2 FK.

RELAZIONI M - M

Esempio di M – M. Le iscrizioni degli studenti ai corsi.
UNO studente si iscrive a MOLTI corsi, ad UN corso si iscrivono MOLTI studenti.



RELAZIONI M - M

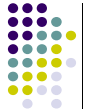
Non è obbligatorio inserire nella tabella di intersezione altri campi oltre alle due FK, però facendolo si chiarisce meglio la natura della relazione M-M, e in numerosi casi è NECESSARIO inserire in questa tabella campi che altrimenti non troverebbero collocazione nelle altre 2 tabelle. Si capirà meglio quando vedremo gli esempi pratici.

RELAZIONI 1 - 1

Le relazioni 1 - 1 non sono altro che relazioni 1 – M in cui il lato M viene limitato a 1.

P.es. si potrebbe imporre la regola che ogni Cliente può fare al massimo UN Ordine. Quindi il caso del legame 1-1 si risolve come quello 1 – M, ma è possibile scegliere da quale parte mettere la FK.

Si obietta spesso che in realtà le relazioni 1 – 1 non dovrebbero esistere, in quanto si potrebbe inserire tutti gli attributi (delle 2 tabelle) in un'unica tabella. I motivi che giustificano la loro esistenza stanno nella situazione REALE che il database deve rappresentare.



I DATABASE

Fine della parte teorica.

Passeremo ora a delle realizzazioni pratiche utilizzando il software Base della suite LibreOffice oppure MS Access.