

IL MODELLO RELAZIONALE DEI DATI

- Un **modello di dati** è un insieme di **meccanismi di astrazione** con associato un insieme predefinito di **operatori** e di **vincoli di integrità**.
- Non tutti i dati sono validi ma ci sono delle precise limitazioni, per esempio: un numero è un vincolo, cioè quel dato deve necessariamente essere presente nel dato.

Un modello di dati è sintesi di **come rappresentare l'informazione**, quali operatori permettono di fare operazioni e **quali sono i vincoli** (come esprimere vincoli su dati, garantire che i dati siano integri).

Il **meccanismo di astrazione** (equivalente alle *strutture dati* presenti in C++), in matematica, è come la definizione di un insieme di numeri e le loro regole (cioè "quali sono i numeri, quello che è rappresentabile e le operazioni che si possono fare con quei numeri").

In questo caso si hanno dati più o meno complessi

Il modello relazionale è stato introdotto da *E.F. Codd* nel '70. Verrà richiamato quando si parlerà di *normalizzazione*.

Prima di *Codd* vi era solo **un'idea** di base di dati relazionale ma non era stata formalizzata e ogni implementazione era diverse dalle altre.

3 accezioni di "relazione"

- La "relazione" si può intendere come **nella teoria degli insiemi**.
- La "relazione" secondo il *modello relazionale dei dati* equivale proprio al vero **concetto di tabella**
- La "relazione" nel modello **Entità-Relazione** cioè a volte si "*mappa*" con l'attributo di una tabella e a volte con un'attributo di una tabella. Questo concetto di relazione è una *connessione fra 2 entità*.

Esempio di relazione matematica

- $D1 = \{a, b\}$
- $D2 = \{x, y, z\}$
- Il prodotto cartesiano $D1 \cdot D2$ è l'insieme delle coppie ordinate, quindi:
- $D1 \cdot D2 = \{(a, x), (a, y), (a, z), (b, x), (b, y), (b, z)\}$

Una **relazione** è un sottoinsieme del prodotto cartesiano, quindi:

$$r \subseteq D1 \cdot D2$$

Formalmente...

- D_1, \dots, D_n (n insiemi anche non distinti)
- **prodotto cartesiano** $D_1 \times \dots \times D_n$:
 - l'insieme di tutte le n -uple (d_1, \dots, d_n) tali che $d_1 \in D_1, \dots, d_n \in D_n$
- **relazione matematica** su D_1, \dots, D_n :
 - un sottoinsieme di $D_1 \times \dots \times D_n$.
- D_1, \dots, D_n sono i **domini** della relazione

Considerando le coppie ordinate possiamo considerare:

- in una n -upla non ci sono ripetizioni;
- ogni n -upla (quindi ogni coppia è ordinata) è ordinato al suo interno (ed è quello che interessa a noi e non ci interessa l'ordine esterno);
- se ad ogni dominio si dà un significato, si rappresentano dei dati.

Esempio:

$Partite \subseteq \text{stringa} \times \text{stringa} \times \text{intero} \times \text{intero}$

Juve	Lazio	3	1
Lazio	Milan	2	0
Juve	Roma	0	2
Roma	Milan	0	1

In questo *esempio*, ogni dato è distinguibile in base alla posizione quindi se si invertono 2 domini, cambia il significato della tabella visto che la posizione è diversa. Pertanto *non si possono fare scambi*.

Vedendo la prima colonna si sa a prescindere che è la squadra di casa. A seguire si ha la squadra ospite, i gol della squadra di casa e i gol della squadra ospite.

La **posizione**, quindi, definisce il significato del dato e quindi si parla di **struttura posizionale**.

In informatica si passa ad una **struttura non posizionale**

A ciascun dominio si associa un nome unico nella tabella (attributo), che ne descrive il "*ruolo*"

Casa	Fuori	RetiCasa	RetiFuori
Juve	Lazio	3	1
Lazio	Milan	2	0
Juve	Roma	0	2
Roma	Milan	0	1

In questo caso **posso scambiare le colonne** e ogni colonna avrà comunque lo stesso significato. La posizione non è più legata al significato. Ogni colonna "*dice il proprio nome*", cioè i dati che contiene.

Con la struttura non posizionale si ottiene una tabella del modello relazionale, con le colonne "nominate".

Definizione:

Uno schema di relazione $R : \{T\}$ è una coppia formata dal **nome di una relazione R** e da un **tipo relazione T** (R nome relazione, T sono gli **attributi** della tabella)

I **tipi primitivi** sono: interi, reali, booleani e stringhe.

- Siano T_1, T_2, \dots, T_n tipi primitivi e A_1, A_2, \dots, A_n etichette, dette attributi, allora $(A_1:T_1, A_2:T_2, \dots, A_n:T_n)$ è un **tipo n-upla di grado n**
- Se T è un tipo n-upla allora $\{T\}$ è un **tipo relazione (tipo insieme di n-uple)**

Sostanzialmente vuol dire che la colonna A_1 avrà attributi T_1 ecc..

Un database (**schema relazionale**) è formato da un **insieme di schemi di relazioni**, cioè un insieme di tabelle. Inoltre ha un insieme di proposizioni (*vincoli di integrità*) che permettono di vedere se i dati inseriti sono validi o no.

Esempio: nel database voti studenti nel database devono rispettare la proprietà: il voto può essere compreso fra 18 e 30)

Esistono *due tipi diversi di aspetti* del database:

- **Aspetto intensionale**, cioè stiamo definendo la **struttura** del database, cioè è la "forma" che si *intende* ottenere;
- **Aspetto estensionale**, cioè un esempio di dato che stiamo andando a rappresentare.

Si definisce ulteriormente:

- *schema di relazione* = struttura di una relazione (com'è formata la tabella);
- *relazione* = una sua particolare istanza;
- *n-upla* = riga di una tabella
- *una relazione R* è un insieme di righe di una colonna;
- *cardinalità di un insieme* = numero di righe, cioè numero di n-uple.

Esempio:

Nome	Matricola	Indirizzo	Telefono
Mario Rossi	123456	Via Etnea 18	777777
Maria Bianchi	234567	Via Roma 2	888888
Giovanni Verdi	345678	Via Etnea 18	999999
Enzo Gialli	456789	Via Catania 3	444444

E' una relazione di tipo:

$\{(Nome : stringa, Matricola : intero, Indirizzo : stringa, Telefono : intero)\}$

Un database è un **insieme di tabelle** con un certo *schema*.

Esempio di 3 tabelle:

```
{
  Studenti : { (Nome : stringa, Matricola : intero(6),
               Indirizzo : stringa, Telefono : intero(6)) }

  Esami : { (Corso : stringa, Matricola : intero(6), Voto: {18,19,...,30}) }

  Corsi : { (Corso: stringa, Professore: stringa) }
}
```

..dove ogni colonna corrisponde ad un attributo.

Questa definizione, solitamente, si può abbreviare nel seguente modo:

- Studenti (Nome, Matricola, Indirizzo, Telefono);
- Esami (Corso, Matricola, Voto);
- Corsi (Corso, Professore);

Il dominio può anche essere **finito e/o infinito** ed essi sono:

- $\text{dominio}(A_i)$ è l'insieme dei valori possibili dell'attributo A_i
- $\text{dominio}(\text{Voto}) = \{18..30\}$

Vincoli di integrità

I **vincoli di integrità** servono a migliorare la qualità delle informazioni contenute nel database.

Un vincolo è un predicato che deve essere soddisfatto da ogni n-upla del database.

Un'**istanza valida** di uno schema di relazione è una relazione dello schema che soddisfa tutti i vincoli di integrità.(un'implementazione dei dati nel database).

Chiaramente si possono usare SOLO le istanze valide, quindi quelle non valide vengono scartate.

Il database software memorizza solo istanze valide

Esempio di vincoli d'integrità

- Il voto deve essere compreso tra 18 e 30;
- La lode può apparire solo se il voto = 30;
- Ogni studente deve avere un numero di matricola;
- Il numero di matricola di uno studente deve essere univoco;
- Esami dati devono fare riferimento solo a corsi offerti.

Tipi di vincoli

I 3 tipi di vincoli più importanti specificano:

- Vincoli **intrarelazionali** fanno riferimento solo ad una relazione:
 - Quali attributi non possono assumere il valore **NULL**
 - Quali attributi sono **chiave**
- Vincoli **interrelazionale** fanno riferimento a più relazioni:
 - Quali attributi sono **chiavi esterne**

Nome	Matricola	Indirizzo	Telefono
Mario Rossi	123456	Via Etnea 18	777777
Maria Bianchi	234567	Via Roma 2	888888
Giovanni Verdi	345678	NULL	999999
Enzo Gialli	456789	Via Catania 3	NULL

Gli attributi obbligatori sono quegli attributi che non possono assumere valore NULL e vuol dire che in quella particolare cella ci deve necessariamente essere un dato.

Ogni colonna non ammette mai NULL a meno che non viene esplicitamente indicato che quella colonna può usare NULL.

Chiave e superchiave

La **chiave** di una relazione è il **sottoinsieme di attributi** che mi permette di identificare **univocamente un record**, ovvero una riga.

Se cerco specificatamente una chiave, trovo una riga. Praticamente mi serve saper rispondere alla domanda: "come identifico in maniera unica un dato all'interno del database? Qual è la chiave?"

- Una superchiave, un insieme di attributi X si dice superchiave se non esistono, per ogni possibile istanze valide, due record che sono diversi ma che nella superchiave hanno lo stesso valore. Il **valore della superchiave è univoco per tutti i record**. Pertanto la posso usare per fare *riferimento ad uno specifico record*. Non ci possono essere 2 record diversi che presentano la stessa superchiave.
- La **chiave** è la *superchiave minimale*, la superchiave che contiene il *minor numero possibili di attributi* che permette di identificare i record in maniera univoca.

In sostanza...

chiave = *numero minimo di relazioni* che permettono di identificare un record univocamente

La chiave *non è unica* ma posso avere tanti attributi che hanno lo scopo di chiave (valori univoci) ma ne devo scegliere sempre uno solo che diventa la chiave della relazione.

In una relazione **c'è sempre** la chiave per definizione di relazione. Ogni relazione ha SEMPRE come superchiave l'insieme degli attributi su cui è definita (cioè tutti gli attributi) e quindi ha (almeno) una chiave

L'esistenza delle chiavi *garantisce l'accessibilità a ciascun dato del database*.

Per *collegare 2 tabelle* vengono usate proprio le chiavi. Se dovessi collegare la tabella A alla tabella B, allora:

- su B ho attributi che provengono da A e su A gli attributi devono essere **chiavi** (permettono di identificare univocamente un record)

Chiave primaria e chiave esterna

La **Chiave Primaria** è una delle chiavi scelta per un dato sistema, quindi è la chiave della tabella.

Essa *non* ammette valori **NULL** e la notazione è la **sottolineatura**.

Esempio:

Nome	<u>Matricola</u>	Indirizzo	Telefono
Mario Rossi	123456	Via Etnea 18	777777
Maria Bianchi	234567	Via Roma 2	888888
Giovanni Verdi	345678	Via Etnea 18	999999
Enzo Gialli	456789	Via Catania 3	444444

La **Chiave Esterna** è un **insieme di attributi** ed è esterna dalla tabella R verso un'altra tabella S .
 In pratica, per ogni record della tabella R , se prendo un attributo R_i trovo nella tabella S lo stesso valore S_i , cioè nella tabella dove è presente la chiave esterna.

- La notazione della chiave esterna è la **sottolineatura tratteggiata**.
- la chiave esterna collega più tabelle differenti tramite valori di determinati chiavi primarie nella tabella "*collegante*";
- la chiave esterna deve essere sempre chiave primaria nella tabella "*collegante*".

Esempio:

Esami

Notazione chiave esterna:
 Sottolineatura tratteggiata

Corso	<u>Matricola</u>	Voto
Programmazione 1	345678	27
Architettura	123456	30
Matematica discreta	234567	19
Basi di Dati	345678	28

Studenti

Nome	<u>Matricola</u>	Indirizzo	Telefono
Mario Rossi	123456	Via Etnea 18	777777
Maria Bianchi	234567	Via Roma 2	888888
Giovanni Verdi	345678	Via Etnea 18	999999
Enzo Gialli	456789	Via Catania 3	444444

Esami

<u>Corso</u>	<u>Matricola</u>	Voto
Programmazione 1	345678	27
Architettura	123456	30
Matematica discreta	234567	19
Algoritmi	345678	28

Corsi

<u>Corso</u>	Professore
Architettura	Barbanera
Programmaizione 1	Cincotti
Matematica discreta	Milici
Algoritmi	Cantone

Esercizi

1. Definire uno schema relazionale per organizzare le informazioni di un'azienda che ha impiegati (ognuno con codice fiscale, cognome, nome e data di nascita), filiali (con codice, sede e direttore, che è un impiegato). Ogni lavoratore lavora presso una filiale. Indicare le chiavi e i vincoli di integrità referenziale dello schema. Mostrare un'istanza della base di dati e verificare che soddisfa i vincoli.

FILIALI

CODICE	SEDE	DIRETTORE
F02LN	GROSSETO (GR)	RSPRBR78L02F205Q
ZFF31	PISTOIA (PT)	CHNNCL93L23F205R
RTZ5W	CUNEO (CN)	TRPNNT65A41F205E
I36AV	BOLZANO (BZ)	PVRVSS64M53H501W

I 3 VINCOLI PRINCIPALI

CHIAVI

- CODICE FISCALE (*IMPIEGATI*)
- CODICE (*FILIALI*)

CHIAVI ESTERNE

- FILIALE (*IMPIEGATI* : da codice della filiale)
- DIRETTORE (*FILIALE*: da codice fiscale di impiegati)

NULL

- Si può omettere tutto tranne il codice fiscale e la filiale in **IMPIEGATI**
- Si può omettere la sede in **FILIALI**

ALTRI VINCOLI IMPIEGATI

- Ogni dato dell'attributo filiale deve appartenere alla relazione "filiali" in particolare all'attributo codice.
- Il codice fiscale deve essere composto da 16 caratteri e deve essere una stringa alfanumerica
- Nome e cognome non devono avere caratteri speciali (escluso apostrofo e accenti)
- La data di nascita deve essere del tipo GG/MM/AAAA (giorno, mese, anno)
- Tutti i caratteri devono avere il formato maiuscolo
- Il codice fiscale deve essere univoco

ALTRI VINCOLI FILIALI

- Ogni dato dell'attributo direttore deve appartenere alla relazione "impiegati" in particolare all'attributo codice fiscale.
- Le sedi devono essere del formato: CITTA(sigla provincia)
- Il codice deve essere univoco

1. Un albero genealogico rappresenta, in forma grafica, la struttura di una famiglia (o più famiglie, quando è ben articolato). Mostrare come si possa rappresentare, in una base di dati relazionale, un albero genealogico, cominciando eventualmente da una struttura semplificata, in cui si rappresentano solo le discendenze in linea maschile (cioè i figli vengono rappresentati solo per i componenti di sesso maschile) oppure solo quelle in linea femminile.