

Modello relazionale

PROF. DIOMAIUTA CRESCENZO

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA «*LUIGI VANVITELLI*»

Il modello relazionale

- Modello proposto nel 1970 da E.F. Codd, ricercatore IBM.
- Prime apparizioni nel mercato solo nel 1981
- Caratterizzato da un alto livello di astrazione
 - caratterizzato da una elevata indipendenza dei dati
 - ha richiesto l'individuazione di realizzazioni efficienti e di hardware adeguato.

Il modello relazionale

- Relazione matematica: come nella teoria degli insiemi
- Relazione secondo il modello relazionale dei dati
- Relazione (dall'inglese relationship) che rappresenta una classe di fatti, nel modello Entity-Relationship; tradotto anche con associazione o correlazione

dove ricordare una tabella

Lui serve a schematizzare i dati

↓
Come legame fra varie entità
Sono duranti

Il modello relazionale

Dati due insiemi D_1 e D_2 si dice **prodotto cartesiano** di D_1 e D_2 ($D_1 \times D_2$) l'insieme formato dalle coppie ordinate (v_1, v_2) con $v_1 \in D_1$ e $v_2 \in D_2$

Esempio:

$$A = \{1, 2, 3\} \quad B = \{a, b, c\}$$

- $A \times B = \{(1, a), (1, b), (1, c), (2, a), (2, b), (2, c), (3, a), (3, b), (3, c)\}$

Il modello relazionale

Si basa su due concetti:

- Tabella** (concetto primitivo)
 - Relazione**
 - Matematica
 - Secondo la definizione del modello relazionale
 - Con riferimento al modello concettuale entity-relationship
 - La forza del modello è dovuta al suo essere basato su un concetto estremamente intuitivo (**tabella**) e su un concetto estremamente ben formalizzato dal punto di vista matematico (**relazione**).
- Sono stessa cosa: una relazione nella sua base dati corrisponde a una tabella*

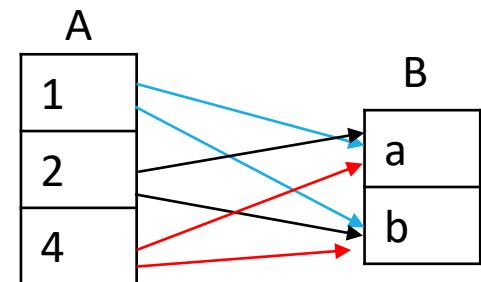
Relazioni e tavole – Prodotto cartesiano

Dati due insiemi D_1 e D_2 si dice **prodotto cartesiano** di D_1 e D_2 ($D_1 \times D_2$) l'insieme formato dalle coppie ordinate (v_1, v_2) con $v_1 \in D_1$ e $v_2 \in D_2$

Esempio:

➤ $A = \{1, 2, 4\}$ $B = \{a, b\}$

➤ $A \times B = \{(1, a), (1, b), (2, a), (2, b), (4, a), (4, b)\}$



$A \times B$

GENERALIZZAZIONE

- 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

- Una relazione matematica sugli insiemi D_1 e D_2 (domini della relazione) è definita come un sottoinsieme di $(D_1 \times D_2)$.
- Es. $\{(1,a),(1,b),(4,b)\}$ relazione su A e B.
- Le relazioni possono essere rappresentati in forma tabellare:

$A \times B$

1	a
1	b
2	a
2	b
4	a
4	b

1	a
1	b
4	b

$$r \subseteq A \times B$$

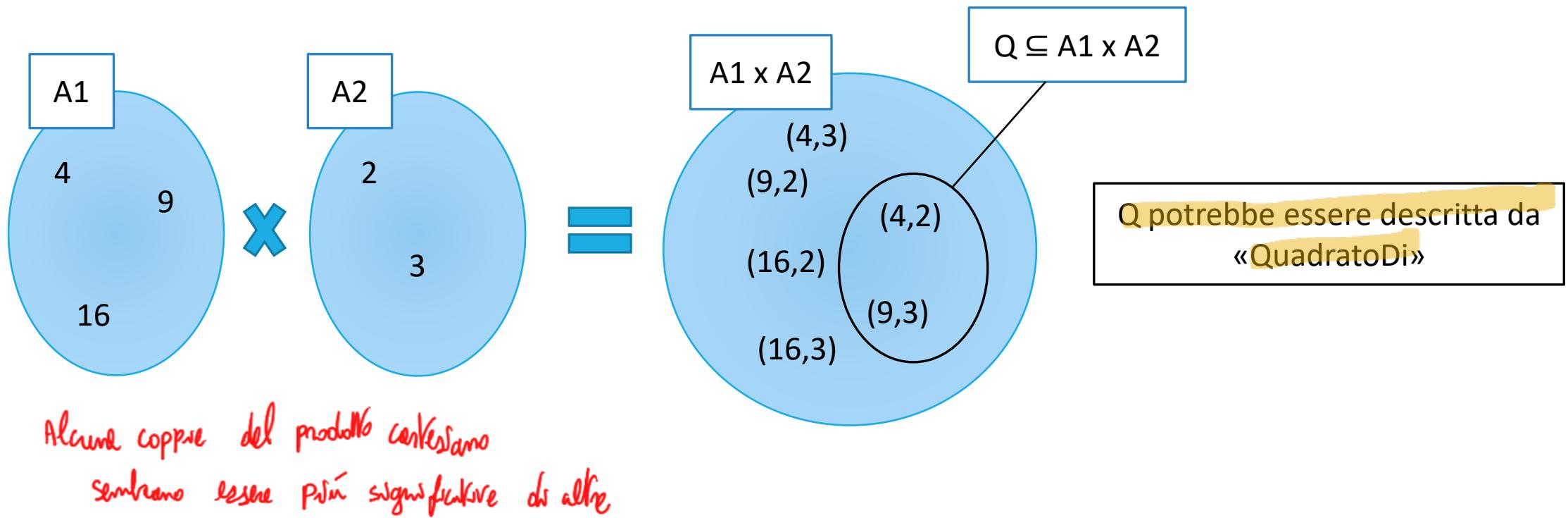


RELAZIONE

GENERALIZZAZIONE

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

Esempio



Esempio

Riferendoci al mondo dell'automobile, consideriamo gli insiemi Modello e Costruttore, così definiti:
Modello = { Panda, Cinquecento, C3, C4 }, Costruttore = { Citroen, Fiat }

Modello x Costruttore

Modello	Costruttore
Panda	Citroen
Cinquecento	Citroen
C3	Citroen
C4	Citroen
Panda	Fiat
Cinquecento	Fiat
C3	Fiat
C4	Fiat

ProdottoDa

Modello	Costruttore
C3	Citroen
C4	Citroen
Panda	Fiat
Cinquecento	Fiat

Relazione matematica

- Le definizioni di prodotto cartesiano e quindi di relazione sono state date con riferimento a coppie di insiemi.
- Esse sono immediatamente generalizzabili rispetto al numero di insiemi coinvolti. *→ numero di colonne*
- Il numero n delle componenti del prodotto cartesiano è detto grado del prodotto cartesiano e della relazione.
- Alle coppie si sostituiscono le n-uple.
- Il numero di elementi (n-uple) della relazione viene detto cardinalità della relazione. *↓ numero di righe*

Dupla = record = tupla/n-upla per una riga generica

Relazioni con Attributi

Una **relazione** è un **insieme**, per cui:

- In una relazione non è definito alcun ordinamento tra le n-uple che la compongono. (*ordine righe*)
- Due tabelle con le stesse righe, ma in ordine diverso, rappresentano la stessa relazione.
- Le n-uple di una relazione sono **distinte** una dall'altra, in quanto tra gli elementi di un insieme non possono essere presenti due elementi uguali.
- ciascuna n-upla è **ordinata**: l' i-esimo valore proviene dall' i-esimo dominio

Relazioni con Attributi

- Ogni n-upla, al suo interno, è ordinata.

Juventus	Lazio	0	1
Milan	Inter	2	0
Roma	Napoli	0	2

Partite \subseteq string \times string \times int \times int

La struttura è posizionale!

- Se scambiamo la colonna 3 con la 4 le informazioni vengono alterate.
- Questa caratteristica è insoddisfacente poiché, in informatica, si tende a privilegiare notazioni non posizionali. *Come la rendo non posizionale? Aggigo nome a colonna*
- La soluzione si ottiene intuitivamente assegnando ad ogni colonna un nome.

Relazioni con Attributi

SquadraDiCasa	SquadraOspitata	RetiCasa	RetiOspitata
Juventus	Lazio	0	1
Milan	Inter	2	0
Roma	Napoli	0	2

Non è più possibile

In maniera formale:

- Una relazione è assimilabile ad un insieme di record omogenei.
- Nel caso dei record ad ogni campo è associato un nome, detto attributo. *Indica significato del valore associato a quell'ingresso*
- Si associa a ciascuna occorrenza di dominio nella relazione un attributo che ne descrive il 'ruolo'.
- Nella rappresentazione tabellare gli attributi sono utilizzati come intestazioni per le colonne.

Tabelle e Relazioni

- In una tabella che rappresenta una relazione
 - l'ordinamento tra le righe è irrilevante
 - l'ordinamento tra le colonne è irrilevante
- Una tabella rappresenta una relazione se
 - le righe sono diverse fra loro: *il DBMS ci aiuterà in diversi modi*
 - le intestazioni delle colonne sono diverse tra loro
 - i valori di ogni colonna sono fra loro omogenei

↓ Se da una parte ho un int., de

Relazioni con Attributi

Utilizzando la nozione di attributo, l'ordine delle colonne non è più significativo.

Relazione con Attributi

- Stabiliamo la corrispondenza fra attributi e domini tramite una funzione ***dom*** che associa ad ogni attributo A un dominio ***dom(A)***.
- Si definisce una tupla su un insieme di attributi X la funzione ***t*** che associa ad ogni attributo A un valore del dominio ***dom(A)***.

Relazioni con Attributi

Relazione con attributi \Leftrightarrow insieme di tuple

Ad esempio:

	RELAZIONE			
	SquadraDiCasa	SquadraOspitata	RetiCasa	RetiOspitata
1	Juventus	Lazio	0	1
2	Milan	Inter	2	0
3	Roma	Napoli	0	2

- Con riferimento alla seconda tupla:
 - $t[\text{SquadraDiCasa}] = t.$ SquadraDiCasa = Milan
 - $t[2]$. **$t[X]$ è un record della tabella**

Tupla accoppia valore con il tipo stringa

Relazioni con Attributi

Relazione con attributi \Leftrightarrow insieme di tuple

Ad esempio:

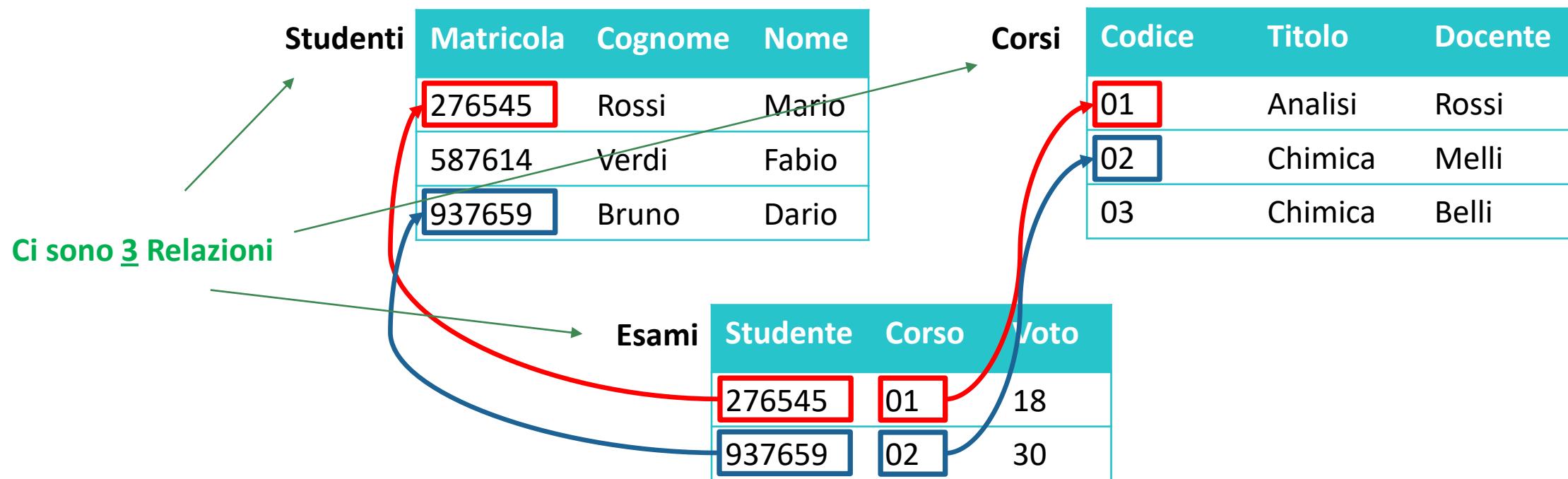
	SquadraDiCasa	SquadraOspitata	RetiCasa	RetiOspitata
1	Juventus	Lazio	0	1
2	Milan	Inter	2	0
3	Roma	Napoli	0	2

- Nel caso di più attributi abbiamo una tupla su più attributi:
- $t[\text{SquadraOspitata}, \text{RetiOspitata}] = \text{milan}, 0$

Relazioni e basi di dati

Nel modello relazionale, una Base di Dati è costituita da più relazioni collegate, dove necessario e/o opportuno, attraverso valori comuni sulle tuple.

Esempio:



Relazioni e basi di dati

- L'esempio mostra una delle caratteristiche fondamentali del modello relazionale:
 - Il modello relazionale è **basato su valori**
 - I riferimenti fra dati in relazioni diverse sono cioè rappresentati attraverso valori uguali dei domini che compaiono nelle tuple. I modelli gerarchico e reticolare sono basati su “record e puntatori”.

Relazioni e basi di dati

Vantaggi del modello basato su valori:

- Rappresenta solo ciò che è rilevante per l'utente. *È significativa anche a chi non capisce molto*
- Tutta l'informazione è contenuta nei valori => è facile trasferire informazioni tra vari contesti (**DB migration**).
- Non vi è alcun legame tra rappresentazione logica e fisica (indipendenza fisica dei dati).
↳ Come vengono memorizzati

- X {
- A livello fisico anche un modello relazionale può essere (e, ovviamente, è) basato su puntatori.
 - Il punto è che essi non sono né visibili né necessari a livello logico.

Definizioni nel Modello Relazionale

- Uno **schema di relazione** è costituito da una simbolo ***R***, detto **nome della relazione** e da un insieme di attributi. Ad ogni attributo è associato un dominio, es:
 - **Studente (Matricola, Cognome, Nome)**
Nome seguito dai suoi campi
- Uno **schema di base di dati** è un insieme di schemi di relazione con nomi diversi (prima definizione)
 - $R = \{ \text{Studente}(\text{Matricola}, \text{Cognome}, \text{Nome}),$
Corsi(Codice, Titolo, Docente),
Esami(Studente, Corso, Voto) }
- Una **istanza di relazione** (o relazione) è un insieme di tuple.
- Una **istanza di base di dati** sullo schema $R\{R_1, \dots, R_N\}$ è un insieme di relazioni $r\{r_1, \dots, r_n\}$ dove ogni r_i è una istanza di relazione sullo schema R_i .

Valori nulli

- Il modello relazionale è molto semplice e potente, ma è rigido (dati omogenei).

Esempio

Persone (Nome, Cognome, Indirizzo, Telefono)

- Il valore dell'attributo Telefono potrebbe non essere disponibile per tutte le tuple
- Non è concettualmente corretto usare un valore del dominio per rappresentare l'assenza di informazione
- Potrei usare ad esempio il valore «0» per indicare le persone che non possiedono il telefono.
(SOLUZIONE CORRETTA SOLO IN ALCUNI CASI)

Tipi di valore nullo

(Almeno) **tre casi differenti**

- valore **sconosciuto**
- valore **inesistente**
- valore **senza informazione**

I **DBMS** non distinguono i tipi di valore nullo

Valori nulli

NOTA: NULL non riguarda colonna o tabella. A differenza di slide dopo

- Il problema è risolto estendendo il concetto di relazione, si prevede che una tupla possa assumere o un valore del dominio o un valore speciale, detto valore nullo (**NULL**).
- Questa soluzione è semplice ed efficace ma, ovviamente
 - non risolve tutti i problemi
 - ne introduce di nuovi.
- Il valore «**NULL**» deve essere usato con estrema attenzione sia per quegli attributi usati per stabilire relazioni fra tabelle sia perché si possono generare dubbi sulla identità delle tuple.

→ Ci sono campi che non potranno avere valore NULL.

Studenti	Matricola	Cognome	Nome
	276545	Rossi	Mario
	NULL	Verdi	Fabio
	NULL	Bruno	Dario

Esami	Studente	Voto	Corso
	276545	18	01
	NULL	30	NULL
	200768	24	NULL

Corsi	Codice	Titolo	Docente
	01	Analisi	Rossi
	03	Chimica	NULL
	NULL	Chimica	Belli

Vincoli di integrità : *Anomie avvolte dai vincoli*

□ Ci sono situazioni che non dovrebbero verificarsi. Chi controlla questi inserimenti?

1. Chi inserisce dati (utenti tramite applicazioni o direttamente)
2. Direttamente al DBMS => vincolo di integrità

Studenti

Matricola	Cognome	Nome
200768	Rossi	Mario
937653	Verdi	Fabio
937653	Bruno	Dario

Esami

Studente	Voto	Lode	Corso
200768	35		01
937653	28	lode	02
200765	30	lode	02

Corsi

Codice	Titolo	Docente
01	Analisi	Rossi
02	Chimica	NULL
03	Chimica	Belli

Vincoli di integrità

: operazione su elenco non fatta se vincolo non è soddisfatto

- È stato introdotto il concetto di *vincolo di integrità*, inteso come proprietà che deve essere soddisfatta dalle istanze (informazioni corrette).
- Un *vincolo* è un predicato che associa alla generica istanza il valore *vero* o *falso*.
 - Se il *predicato* è *vero*, l'istanza soddisfa il *vincolo*.
- Ad uno *schema* di base di dati sono assegnati un certo numero di *vincoli*.
- Una *istanza* è *leccita* (o corretta, o consistente) se soddisfa tutti i *vincoli* di una *base di dati*.

Vincoli di integrità, nota

- alcuni tipi di vincoli (ma non tutti) sono "supportati" dai DBMS:
 - possiamo quindi specificare vincoli di tali tipi nella nostra base di dati e il DBMS ne impedisce la violazione
- per i vincoli "non supportati", la responsabilità della verifica è dell'utente o del programmatore

Vincoli di integrità

riguarda 2 o più relazioni:

legame tra 2 tabelle: matricola lega studente e esame
Voto invece no.

- Vincolo intrarelazionale** \Leftrightarrow il suo soddisfacimento è definito relativamente alla singola relazione.
- Vincolo di tupla** \Leftrightarrow il suo soddisfacimento è definito a livello di tupla.
- Vincolo su valori o vincolo di dominio** \Leftrightarrow il suo soddisfacimento riguarda singoli valori.
- Vincolo interrelazionale** \Leftrightarrow coinvolge due o più relazioni.

Vincoli intrarelazionali

- Vincoli di Tupla**
 - Esprimono condizioni sui valori della singola tupla.
 - $(\text{Voto} \geq 18) \text{ AND } (\text{Voto} \leq 30)$
 - $(\text{Voto} = 30) \text{ OR NOT } (\text{Lode} = \text{'e lode'})$

- Vincoli di Chiave**
 - Sono i più importanti vincoli intrarelazionali e sono la base stessa del modello.
 - Si consideri il campo Matricola: non esistono due studenti con lo stesso numero di matricola

Vincoli di tupla: Esempio

Stipendi	Impiegato	Lordo	Ritenute	Netto
	Rossi	55.000	12.500	42.500
	Neri	45.000	10.000	35.000
	Bruni	47.000	11.000	36.000

Lordo = (Ritenute + Netto)

Vincoli di tupla: Violazione

Stipendi	Impiegato	Lordo	Ritenute	Netto
	Rossi	55.000	12.500	42.500
	Neri	45.000	10.000	35.000
	Bruni	50.000	11.000	36.000

$$\text{Lordo} = (\text{Ritenute} + \text{Netto})$$

Chiave

Per il concetto di relazione non esistono 2 tuple identiche. Quindi, già tutte gli attributi sono una chiave.

Una chiave è un insieme di attributi che identificano univocamente una tupla di una relazione.

- Un insieme di attributi K si dice **superchiave** per la relazione r se r non contiene due tuple distinte t_1 e t_2 tali che $t_1[K]=t_2[K]$.
- Un insieme di attributi si dice **chiave** se è una **superchiave minimale** di r , cioè se non esiste una superchiave K_1 contenuta propriamente in K .

Chiavi

A volte non è immediato trovare una chiave

Matricola	Cognome	Nome	Nascita	Corso
001	Rossi	Luigi	29/10/2000	Ing. Informatica
002	Rossi	Dario	29/10/2000	Ing. Informatica
003	Rossi	Luca	18/11/2001	Ing. Civile
004	Neri	Luca	18/11/2001	Ing. Meccanica
005	Neri	Luca	07/04/2000	Ing. Meccanica

- L'insieme $K = \{\text{Matricola}\}$ è una **superchiave**, è anche una **superchiave minimale** (ha un solo attributo), è quindi una **chiave**;
- L'insieme $\{\text{Cognome, Nome, Nascita}\}$ è superchiave, è minimale ed è anche una chiave
 - Ci sono delle tuple uguali se prendiamo insiemi a due elementi, es: $\{\text{Cognome, Nascita}\}$

↳ Potrebbe non essere in futuro: un altro Luca nero.

Chiavi

Matricola	Cognome	Nome	Nascita	Corso
001	Rossi	Luigi	29/10/2000	Ing. Informatica
002	Rossi	Dario	29/10/2000	Ing. Informatica
003	Rossi	Luca	18/11/2001	Ing. Civile
004	Neri	Luca	18/11/2001	Ing. Meccanica
005	Neri	Luca	07/04/2000	Ing. Meccanica

- L'insieme {Matricola, Corso} è una superchiave, non è superchiave minimale poiché ha come sottoinsieme proprio {Matricola}, quindi non è chiave;
- L'insieme {Nome, Corso} non è superchiave

Chiavi

Matricola	Cognome	Nome	Nascita	Corso
001	Rossi	Dario	29/10/2000	Ing. Informatica
002	Rossi	Luca	18/11/2001	Ing. Civile
003	Neri	Luca	18/11/2001	Ing. Meccanica
004	Neri	Luca	07/04/2000	Ing. Meccanica

- {Corso, Cognome} è una superchiave minimale e quindi è una chiave.
- Che questa relazione sia chiave non è detto sia sempre vera.
- Questo significa che {Cognome, Corso} è «casualmente» una chiave per la relazione mostrata sopra.

Chiavi

- Ogni relazione e ogni schema di relazione **ha sempre una chiave.**
- Infatti:
 - Una relazione è, per definizione, un insieme di elementi tra loro differenti
 - Non possono esistere due tuple uguali
 - L' insieme di tutti gli attributi di una relazione è una superchiave
 - esso è chiave
 - esso non è chiave poiché esiste un' altra superchiave in esso contenuta.
 - Si può procedere ricorsivamente, poiché il numero di attributi è finito, il processo termina in numero finito di passi con una superchiave minimale.

Chiavi e valori nulli

Matricola	Cognome	Nome	Nascita	Corso
NULL	Rossi	Dario	NULL	Ing. Informatica
002	Rossi	Luca	18/11/2001	Ing. Civile
003	Neri	Luca	NULL	NULL
NULL	Neri	Luca	07/04/2000	Ing. Meccanica

- Ipotesi: esistono 2 chiavi
 - {Matricola}
 - {Cognome, Nome, Nascita}
- Le tuple relative a “Neri Luca” si riferiscono a studenti diversi o lo stesso studente è rappresentato 2 volte?

Chiave primaria

- Esiste allora la necessità di porre dei limiti alla presenza di valori nulli nelle chiavi delle relazioni.
→ Su cui non posso inserire null
- In pratica, su una delle chiavi (detta chiave primaria), si vieta la presenza di valori nulli

<u>Matricola</u>	Cognome	Nome	Nascita	Corso
3976	Rossi	Mario	NULL	Ing. Informatica
4766	Rossi	Luca	18/11/2001	Ing. Civile
4856	Neri	Luca	07/05/2000	NULL
5591	Neri	Luca	07/04/2000	Ing. Civile

*Posso avere diverse chiavi,
ma di chiave primaria 1*

- La chiave primaria si rappresenta sottolineando l'attributo relativo (o gli attributi relativi)
- In ogni relazione “deve” esistere una e una sola chiave primaria.

Vincoli di integrità referenziale

- La più importante classe di vincoli interrelazionali.
- Una tupla in una relazione che fa riferimento ad un'altra relazione, deve far riferimento ad un tupla esistente.
Se ho l'elenco degli studenti con quelle matricole e ho l'elenco degli esami che svolgono gli studenti, non posso in esami assegnare un numero di studenti che non compare nella prima relazione

Tabella esterna

	Matricola	Cognome	Nome
	200768	Rossi	Mario
	937653	Verdi	Fabio
	937654	Bruno	Dario

Studenti

Studente	Voto	Lode	Corso
200768	35		01
937653	28	Iode	02
937654	30	Iode	02

Esami

Tabella interna

Con una direzione, comincio che le due relazioni sono collegate con attivato.

Vincoli di integrità referenziale

- Un vincolo di integrità referenziale (**foreign key** o **referential integrity constraint**) fra un insieme di attributi X di R_1 e uno Y di R_2 è vero se i valori su X di ogni tupla dell' istanza di R_1 compaiono come valori di Y della chiave primaria di R_2

- L'integrità referenziale richiede che, per ogni valore non nullo della chiave esterna, esista un valore corrispondente della chiave primaria nella tabella associata

CHIAVE ESTERNA: è un legame che ho via chiave primaria e una serie di attributi che ho definito nella tabella esterna. Deve sempre avere relazione 1 a 1 (dalla relaz. di avvio posso sempre tornare a quella chiave)

STESSA COSA DI VINCITO DI INTEGRITÀ REFERENZIALE

Vincoli di integrità referenziale: Esempi

- Caso 1: X costituito da un solo attributo.

□ Spesso chiave di R_2 B (singolo attributo);

□ per ogni tupla t_1 in R_1 esiste una tupla t_2 in R_2 tale che: $t_1[A]=t_2[B]$.

R_2 è la relazione dei Vigili

R_1 è quella delle Infrazioni

→ Tupla su attrib. matr. col.

Tupla su attrib. vigile

L'uni. al legame al vigile può
che alla semplice matr. col.

Vigili

	Matricola	Cognome	Nome
	3987	Rossi	Luca
	3295	Neri	Piero
	9345	Neri	Mario
	7543	Mori	Gino

Infrazioni

	Codice	Data	Vigile	Prov	Numero
	34321	1/2/15	3987	MI	39548K
	53524	4/3/15	3295	TO	E39548
	64521	5/4/16	3295	PR	839548
	73321	5/2/18	9345	PR	839548

Come capisco quali sono le chiavi esterne? Vedremo

In auto, Prov e Numero è chiave primaria. Questa diventa chiave esterna nella relazione di auto

Vincoli di integrità referenziale: Esempi

- Caso 2: X composto da più attributi.

- In questo caso, è necessario far corrispondere ad ogni attributo di X un attributo della chiave K di R₂.

- Bisogna stabilire un ordinamento sugli attributi di X=A₁,A₂,..A_p e di K=B₁,B₂,..B_p

- Il vincolo è vero se per ogni tupla t₁ di R₁ esiste una tupla t₂ in R₂ tale che : t₁[A_i] = t₂[B_i] per 1 <= i <= p

Infrazioni

<u>Codice</u>	Data	Vigile	Prov	Numero
34321	1/2/15	3987	MI	39548K
53524	4/3/15	3295	TO	E39548
64521	5/4/16	3295	PR	839548
73321	5/2/18	9345	PR	839548

Auto	<u>Prov</u>	<u>Numero</u>	Cognome	Nome
	MI	39548K	Rossi	Mario
	TO	E39548	Rossi	Mario
	PR	839548	Neri	Luca

↳ Attributo Vigile non appartiene alla relazione Vigile; ho 2 vincoli di integrità referenziale

Importante mantenere orologiamenTo

Vincoli di integrità referenziale: Esempi

- Consideriamo una base di dati sullo schema

IMPIEGATI (Matricola, Cognome, Nome, Dipartimento)

DIPARTIMENTI (Codice, Nome, Sede)

→ Chiavi esterne qui c'è vincolo
di integrità referenziale verso
il codice della relazione dipartimenti

Dipartimenti

Impiegati

<u>Matricola</u>	<u>Cognome</u>	<u>Nome</u>	<u>CapoImpiegato</u>	<u>Dipartimento</u>
200768	Rossi	Mario	Si	001
937653	Verdi	Fabio	No	001
937654	Bruno	Dario	No	002

<u>Codice</u>	<u>Nome</u>	<u>Sede</u>
001	Logistica	Napoli
002	Tecnico	Roma
003	Amministrativo	Roma

- Esistono anche altri vincoli che possiamo immaginare che esistano, ad es:

- Lo stipendio di un impiegato deve essere \leq dello stipendio del proprio capo
-

(Sarebbe da gestire tutto programm.)

Esempi di violazione

Studenti

Matricola	Cognome	Nome	DataNascita	Dipartimento
545	Rossi	Mario	NULL	125
653	Verdi	Fabio	20/09/1998	125
768	Bruni	Dario	30/10/1999	NULL
653	Neri	Franco	32/10/1998	180
314	Rossi	Maria	27/10/1999	185

La data di nascita non è nota. Non è una violazione se il dato non è obbligatorio.

Non è nota la scuola frequentata da Bruni. Non è una violazione al vincolo di integrità referenziale.

Scuole

Codice	Nome
125	ITC Manzoni
180	Liceo Dante
190	Liceo Fermi
NULL	IT Carli

Violazione del vincolo di chiave: non ci possono essere chiavi duplicate.

Data di nascita assurda: si tratta di una violazione di un vincolo di dominio.

CodScuola con valore scorretto: non c'è nessuna scuola con Codice =185. Violazione a un vincolo di integrità referenziale.

Violazione del vincolo di chiave: la chiave non può avere valore nullo. Non c'è modo di collegare la scuola con chi la frequenta.

Azioni compensative

Esempio:

- Viene eliminata una tupla causando una violazione

Comportamento “standard”:

- Rifiuto dell'operazione

Azioni compensative:

- Eliminazione in cascata
- Introduzione di valori nulli

Elimina anche le tuple associate al progetto eliminato.
Non conviene quasi mai.

Impiegati

1. *Insere valore NULL*

Progetti

Matricola	Cognome	Progetto
34321	Rossi	IDEA
53524	Neri	XYZ
64521	Verdi	NULL
73032	Bianchi	IDEA

Codice	Inizio	Durata	Costo
IDEA	01/2017	36	200
XYZ	07/2016	24	120
BOH	09/2016	24	150

Impiegati

Matricola	Cognome	Progetto
34321	Rossi	IDEA
53524	Neri	NULL
64521	Verdi	NULL
73032	Bianchi	IDEA

Progetti

Codice	Inizio	Durata	Costo
IDEA	01/2017	36	200
BOH	09/2016	24	150

Esercizio 1

Considerare le informazioni per la gestione dei prestiti di una biblioteca personale. Il proprietario presta libri ai suoi amici, che indica semplicemente attraverso i rispettivi nomi o soprannomi (così da evitare omonimie) e fa riferimento ai libri attraverso i titoli (non possiede 2 libri con lo stesso titolo). Quando presta un libro, prende nota della data prevista di restituzione. Definire uno schema di relazione per rappresentare queste informazioni, individuando opportuni domini per i vari attributi e mostrare un'istanza in forma tabellare. Indicare la chiave (o le chiavi) della relazione.

Esercizio 1

PRESTITO (Titolo, Nome, DataRestituzione)

<u>Titolo</u>	Nome	DataRestituzione
Il signore degli anelli	Marco	01/03/2022
Il piccolo principe	Mario	18/02/2022
Harry Potter e la pietra filosofale	Danilo	10/12/2021
La caduta di Gondolin	Vittorio	20/12/2021

Esercizio 2

Definire uno schema di base di dati per organizzare le informazioni di un'azienda che ha impiegati (ognuno con codice fiscale, cognome, nome e data di nascita) e filiali (con codice, sede e direttore, che è un impiegato). Ogni impiegato 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 soddisfi i vincoli.

Esercizio 2

Impiegati (CF, Cognome, Nome, DataNascita, Filiale)

Filiali (Codice, Sede, Direttore)

Impiegati

<u>CF</u>	Cognome	Nome	DataNascita	Filiale
RSSMRA76E27H501Z	Rossi	Mario	27/05/1976	GT09
BRNGNN90D03F205E	Bruni	Giovanni	03/04/1990	AB04
GLLBRN64E04F839H	Gialli	Bruno	04/05/1964	GT09
NREGNI64L01G273Y	Neri	Gino	01/07/1964	PT67

Filiali

<u>Codice</u>	Sede	Direttore
AB04	Roma	BRNGNN90D03F205E
GT09	Livorno	GLLBRN64E04F839H
PT67	Napoli	NREGNI64L01G273Y