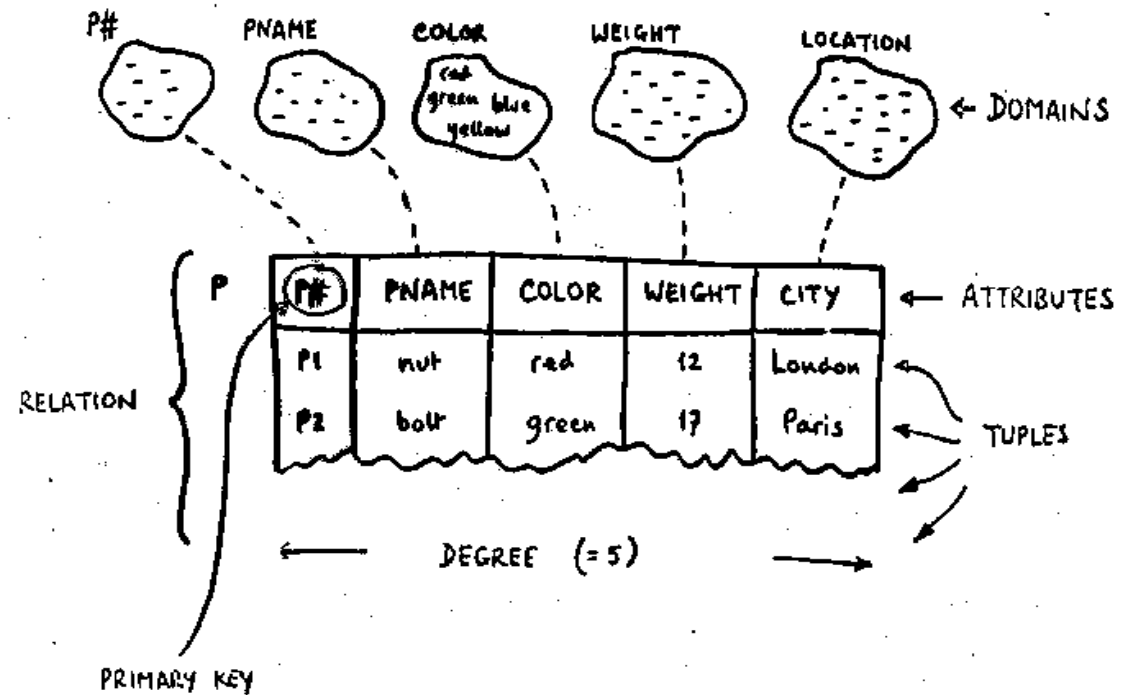


TERMINOLOGY



Il modello relazionale

Joseph Giovanelli, Annalisa Franco, Dario Maio
Università di Bologna

Introduzione al modello relazionale

- Il modello relazionale fu introdotto nel 1970 da EF. Codd (presso i laboratori IBM di San Jose, CA) allo scopo di favorire l'indipendenza dei dati.
- I modelli preesistenti, gerarchico (hierarchical model) e reticolare (network model) erano fortemente influenzati da considerazioni di natura fisica, che enfatizzavano maggiormente gli aspetti di efficienza piuttosto che la semplicità d'uso.
- Il sistema di gestione IMS, basato sul modello gerarchico, fu sviluppato originariamente nel 1966 da IBM insieme a Rockwell e Caterpillar per il programma Apollo. È tutt' ora operativo per ambienti OLTP industriali.
- Il modello reticolare è stato introdotto da Charles Bechman e standardizzato dal consorzio CODASYL nel 1969.
- Rispetto ai modelli gerarchico e reticolare il modello relazionale si caratterizza per:
 - ▮ la totale assenza di legami costruiti con puntatori; nel modello relazionale, infatti, si fa uso solo di valori;
 - ▮ la presenza di una teoria utile per la progettazione di DB, per la definizione di linguaggi d'interrogazione e per l'ottimizzazione di query.

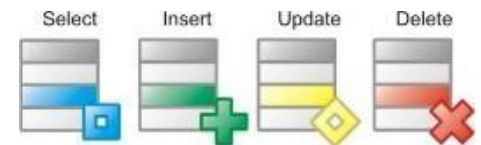
Un po' di storia...

- Lo sviluppo di efficienti DBMS basati sul modello relazionale è stato piuttosto lento, a causa dell'elevato livello d'astrazione offerto rispetto ai modelli precedenti. Le prime soluzioni commerciali risalgono alla prima metà degli anni 80.

Anni 70: definizione del modello, prima versione del linguaggio **SQL** (Structured Query Language, inizialmente denominato **SEQUEL**), studi fondamentali sulla tecnologia relazionale (ottimizzazione, transazioni, recovery, ...) e primi prototipi di DBMS relazionali (RDBMS): System R (IBM, laboratorio di ricerca di San Jose, CA, USA), Ingres (Università di Berkeley, CA, USA)

Anni 80: prima standardizzazione di SQL, primi prototipi commerciali: SQL/DS (derivato da System R), Oracle, IBM DB2

Anni 90: standard ISO-ANSI SQL-2 (anche noto come SQL-92).



- Esiste oggi lo standard ISO-ANSI SQL-3 (o SQL:1999) e sono state definite ulteriori estensioni: SQL 2003, SQL 2008, ecc.
- Nonostante il nome, SQL non è un semplice linguaggio di interrogazione perché alcune istruzioni sono dedicate alla creazione, alla gestione e all'amministrazione del database.

Modello logico relazionale: note

- Il modello relazionale è un **modello logico** nel senso che risponde al requisito di indipendenza dalla particolare rappresentazione dei dati adottata a livello fisico.
- Nel contesto di un DB relazionale, gli utenti che accedono ai dati e i programmatori che sviluppano applicazioni fanno riferimento **solo al livello logico**, senza specificare i percorsi di accesso per eseguire le operazioni.
- I modelli logici **gerarchico** (rappresentazione con **strutture ad albero**) e **reticolare** (rappresentazione con **strutture a grafo**), ricordano più da vicino tecniche di organizzazione dei dati a livello fisico e richiedono al programmatore di esplicitare i cammini per accedere ai dati.
- Nel modello relazionale l'unica astrazione è il concetto di **relazione**; non vi sono costrutti concettuali di alto livello in grado di descrivere entità, associazioni, generalizzazioni, specializzazioni, aggregazioni.
- Tuttavia, una relazione può rappresentare opportunamente ed efficientemente i suddetti concetti astratti. Ciò motiva la necessità di ricorrere, nella fase di progettazione logica, a una traduzione da schemi concettuali E/R a schemi logici relazionali.

Relazione nel modello relazionale

- Nella **rappresentazione tabellare**:
 - gli attributi sono le **intestazioni delle colonne**,
 - le istanze sono le **righe**.

TeamCasa	TeamOspite	PuntiCasa	PuntiOspite
Enel Brindisi	Sidigas Avellino	92	88
MIA Cantù	Virtus Bologna	94	87
Fiat Torino	Vanoli Cremona	88	80
The Flex Pistoia	Consultinvest Pesaro	86	83

- **L'ordine di attributi (colonne) e istanze (righe) non ha rilevanza.**

Relazione: una definizione formale

- Si consideri un insieme di attributi $X = \{A_1, A_2, \dots, A_n\}$.
- Si indichi con $\text{dom}(A)$ il dominio dell'attributo A .
- Una **tupla** t su X è una funzione che associa a ogni $A_i \in X$ un valore di $\text{dom}(A_i)$.
- Uno **schema di relazione** su X è definito da un **nome** R (della relazione) e dall'insieme di attributi X , e si indica con $R(X)$.

attributi

PARTITE

	TeamCasa	TeamOspite	PuntiCasa	PuntiOspite
	Enel Brindisi	Sidigas Avellino	92	88
	MIA Cantù	Virtus Bologna	94	87
t	Fiat Torino	Vanoli Cremona	88	80
	The Flex Pistoia	Consultinvest Pesaro	86	83

Livelli intensionale ed estensionale

- Uno schema $R(X)$ definisce a livello intensionale una relazione.
Esempio:

PARTITE(TeamCasa, TeamOspite, PuntiCasa, PuntiOspite)

- A **livello estensionale** si usa la notazione r o $r(X)$ per indicare l'estensione come insieme di tuple.



TeamCasa	TeamOspite	PuntiCasa	PuntiOspite
Enel Brindisi	Sidigas Avellino	92	88
MIA Cantù	Virtus Bologna	94	87
Fiat Torino	Vanoli Cremona	88	80
The Flex Pistoia	Consultinvest Pesaro	86	83

Notazione di base

- Se t è una tupla su X e $A \in X$, allora $t[A]$ o $t.A$ è il valore di t su A .

t {

TeamCasa	TeamOspite	PuntiCasa	PuntiOspite
Enel Brindisi	Sidigas Avellino	92	88
MIA Cantù	Virtus Bologna	94	87
Fiat Torino	Vanoli Cremona	88	80
The Flex Pistoia	Consultinvest Pesaro	86	83

$t[\text{TeamOspite}] = t.\text{TeamOspite} = \text{'Vanoli Cremona'}$

- La stessa notazione si usa per insiemi di attributi, e denota una tupla
 - $t[\text{TeamOspite}, \text{PuntiOspite}]$ è una tupla su $\{\text{TeamOspite}, \text{PuntiOspite}\}$

Data Base relazionale

- Lo schema di un DB relazionale è un insieme di schemi di relazioni con nomi distinti:

$$R = \{R_1(X_1), R_2(X_2), \dots, R_m(X_m)\} \quad (R_i \neq R_j \quad \forall i \neq j)$$

- Una **estensione** di un DB con schema

$$R = \{R_1(X_1), R_2(X_2), \dots, R_m(X_m)\}$$

è un insieme di estensioni

$$r = \{r_1, r_2, \dots, r_m\}$$

- Esempio: **AZIENDA** =

{
 IMPIEGATI(Matricola, Cognome, Nome, Livello, Stipendio),
 FILIALI(CodiceFiliale, Nome, Indirizzo, Direttore),
 FORNITORI(RagioneSociale, Indirizzo, PartitaIVA) }
}

N.B. In realtà la definizione di uno schema di relazione e di uno schema di DB comprende anche l'indicazione di un insieme di vincoli d'integrità.

Una estensione di un semplice DB relazionale

STUDENTI

Matricola	Cognome	Nome	DataNascita
29323	Bianchi	Giorgio	21/06/1978
35467	Rossi	Anna	13/04/1978
39654	Verdi	Marco	20/09/1979
42132	Neri	Lucia	15/02/1978

CORSI

CodCorso	Titolo	CodDocente	Anno
483	Analisi	0201	1
729	Analisi	0021	1
913	Sistemi Informativi	0123	2

ESAMI

Matricola	CodCorso	Voto	Lode
29323	483	28	no
39654	729	30	sì
29323	913	26	no
35467	913	30	sì

DOCENTI

CodDocente	Cognome	Nome	DataNascita
0021	Biondi	Carlo	21/06/1958
... 22

Tabelle vs Relazioni

- In realtà i termini “**tabella**” e “**relazione**” non sono affatto sinonimi; una relazione del modello relazionale può essere vista come un particolare tipo di tabella (che Codd chiama R-table).
- Una **tabella** rappresenta una relazione se:
 - ▮ i valori di ciascuna colonna sono tra loro omogenei (definiti sullo stesso dominio);
 - ▮ le righe sono tra loro diverse;
 - ▮ le intestazioni delle colonne sono diverse tra loro.
- In **una tabella che rappresenta una relazione**:
 - ▮ l'ordinamento delle righe è irrilevante;
 - ▮ l'ordinamento delle colonne è irrilevante.
- Il linguaggio SQL nei DBMS commerciali consente di gestire tabelle che non sono relazioni, e che ammettono righe duplicate.

Il problema dei duplicati


- Esempio di derivazione in SQL di una tabella da una relazione.

STUDENTI

Matricola	Cognome	Nome	DataNascita
2106103423	Bianchi	Giorgio	21/06/1978
2106111021	Rossi	Anna	13/04/1978
1602042312	Rossi	Anna	11/03/1978

- `SELECT Cognome, Nome`


`FROM STUDENTI ;`


una tabella che
non è una relazione

Cognome	Nome
Bianchi	Giorgio
Rossi	Anna
Rossi	Anna

- `SELECT DISTINCT Cognome, Nome`

`FROM STUDENTI ;`


una tabella che
è una relazione

Cognome	Nome
Bianchi	Giorgio
Rossi	Anna

Nome dello schema di una relazione

- Vi sono diverse “scuole di pensiero” a riguardo della convenzione da adottare per il nome di uno schema di relazione. Molto spesso si ricorre all’uso di un sostantivo e alcuni preferiscono indicare il nome al singolare, altri al plurale: **PERSONA** o **PERSONE**, **CORSO** o **CORSI**?
- Naturalmente sono possibili, se opportuno, anche altre categorie grammaticali. Esistono inoltre casi dove il problema “singolare o plurale” non si pone, laddove una parola di per sé indica una pluralità di elementi (es. **STAFF**, **PERSONALE**,...).
- La motivazione principale per il **singolare** risiede nel fatto che il nome dello schema esprime un’asserzione che denota il tipo di ogni tupla della relazione in qualunque stato si trovi.
- Chi propende per il **plurale** pensa invece che il nome debba indicare un insieme ovvero una pluralità di elementi, e in questo modo concentra l’attenzione sull’aspetto estensionale della relazione e sulle operazioni che si effettuano. Entrambe le opinioni hanno pro e contro.

Nome di una SQL table

- Qual è la conseguenza del nome ai fini pratici in SQL?

CodImpiegato	Nome	Cognome	Ruolo
E001	Carlo	Rossi	Analista
E003	Mario	Bianchi	Programmatore
E006	Giorgio	Grigi	Sistemista
E007	Carlo	Verdi	Programmatore

IMPIEGATI.Nome

oppure

IMPIEGATO.NOME ?

Select Cognome, Nome
From IMPIEGATI

oppure

Select Cognome, Nome
From IMPIEGATO ?

- In SQL lo statement **CREATE TABLE** definisce al contempo gli attributi di ogni tupla e alloca un certo spazio per ospitare i record che saranno inseriti. Si tratta dunque di una vera e propria **collezione di record**.

1NF, ovvero solo domini semplici

- Il modello relazionale non permette di usare domini arbitrari per la definizione delle relazioni; in particolare **non è in generale possibile usare domini strutturati** (array, set, liste, ...).
 - ▮ Vi sono **eccezioni** notevoli (esempi: **date** e **stringhe**).
- Concisamente, una relazione in cui ogni dominio è “**atomico**” (non ulteriormente decomponibile) si dice che è in
Prima Forma Normale, o **1NF** (First Normal Form)
- In molti casi è pertanto richiesta preliminarmente un’attività di **normalizzazione dei dati** che dia luogo a relazioni in 1NF e che preservi l’informazione originale.

Strutture nidificate: normalizzazione in 1NF

Ricevuta n. 231 del 12/02/2002		
Coperti	2	3,00
Antipasti	1	5,80
Primi	2	11,45
Secondi	2	22,30
Caffè	2	2,20
Vino	1	8,00
Totale (Euro)		52,75

Ricevuta n. 352 del 13/02/2002		
Coperti	1	1,50

RICEVUTE



Numero	Data	Totale
231	12/02/2002	52,75
352	13/02/2002	...
...

DETTAGLI



Numero	Quantità	Descrizione	Prezzo
231	2	Coperti	3,00
231	1	Antipasti	5,80
231	2	Primi	11,45
231	2	Secondi	22,30
231	2	Caffè	2,20
231	1	Vino	8,00
352	1	Coperti	1,50

Considerazioni sulla normalizzazione in 1NF

- Il fatto che una rappresentazione normalizzata sia adeguata o meno dipende (molto) dal contesto.
 - Ad esempio: l'ordine delle righe nella ricevuta è rilevante o meno?
- Analogamente per eventuali ridondanze che si possono osservare.
 - Ad esempio: il coperto e il caffè hanno un prezzo che non varia da ricevuta a ricevuta?
- In generale è bene ricordare che ogni caso presenta una sua specificità e pertanto non deve essere trattato “automaticamente”.
- Normalizzare in 1NF è, a tutti gli effetti, un'attività di progettazione (logica), e in quanto tale può essere solo oggetto di “regole guida” che però non hanno validità assoluta.

Informazione incompleta

- Le informazioni che si vogliono rappresentare mediante relazioni non sempre corrispondono pienamente allo schema prescelto; in particolare, per alcune tuple e per alcuni attributi potrebbe non essere possibile specificare, per diversi motivi, un valore del dominio.

PERSONE

Codice	Cognome	Nome	DataMorte
A001	Rossi	Mario	20/02/1954
A002	Verdi	Paolo	
A003	Bianchi	Bruno	
A004	Grigi	Carlo	

- Paolo Verdi è ancora vivo (**valore non applicabile**);
- Bruno Bianchi è deceduto, ma non conosciamo la data di morte (**valore applicabile ma ignoto**);
- Carlo Grigi è scomparso misteriosamente, non sappiamo se è vivo o se è deceduto (**ignota l'applicabilità**).

Quale soluzione?

- In diversi casi, in mancanza di informazione, si tende a usare un “valore speciale” del dominio (0, “”, “-1”, “9999”, ecc.) che non si utilizza per altri scopi.
- Questa pratica è **fortemente sconsigliata**, in quanto, anche dove possibile:
 - ▮ valori inutilizzati potrebbero successivamente diventare significativi;
 - ▮ le applicazioni dovrebbero sapere “che cosa significa in realtà” il valore usato allo scopo.
- Esempio (reale!): nel 1998, analizzando i clienti di un’assicurazione, si scoprì una strana concentrazione di ultra-novantenni... tutte le date di nascita ignote erano state codificate con “01/01/1900”!
- Nel modello relazionale si opera in maniera pragmatica: si adotta il concetto di **valore nullo (NULL)**, che denota assenza di un valore nel dominio (e **non è un valore del dominio**);
- ... pertanto $t[A] \in \text{dom}(A) \cup \{\text{NULL}\}$.

Valori nulli: restrizioni

- La presenza di valori nulli non può essere sempre tollerata, ovvero è necessario imporre delle **restrizioni** al loro uso; si consideri ad esempio il caso della registrazione di esami:

ESAMI

Matricola	CodCorso	Voto	Lode
29323	483	28	no
NULL	729	30	sì
29323	913	NULL	no
35467	913	30	no

- ▮ Un valore nullo per Matricola non permetterebbe di sapere quale studente ha sostenuto l'esame.
- ▮ Un valore nullo per Voto non è proprio ammissibile nel contesto considerato.



Istanze di questo tipo non sono accettabili!

Vincoli di integrità

- La “**correttezza sintattica**” di una estensione di una relazione non è condizione sufficiente affinché i dati rappresentino un’informazione possibile nel contesto reale considerato.

STUDENTI	Matricola	Cognome	Nome	DataNascita
	35467	Bianchi	Giorgio	21/06/1978
	35467	Rossi	Anna	13/04/1978
	39654	Rossi	Anna	13/04/1978

- La prima e la seconda tupla hanno la stessa Matricola?
- La seconda e la terza tupla hanno gli stessi valori per Nome, Cognome e DataNascita, ma questo in linea di principio è possibile!
- Un **vincolo di integrità** è una proprietà che deve essere soddisfatta da ogni possibile stato osservabile di una relazione; ogni vincolo può quindi essere descritto da una funzione booleana che associa a ogni stato il valore VERO o FALSO.

Vincoli di dominio

- Un vincolo che si riferisce ai valori ammissibili per un singolo attributo è detto **vincolo di dominio** (o sui **valori**).

ESAMI	Matricola	CodCorso	Voto	Lode
	29323	483	28	no
	39654	729	30	sì
	29323	913	31	no
	35467	913	30	forse

- ▮ Il Voto deve essere compreso tra 18 e 30 :
(Voto \geq 18) AND (Voto \leq 30)
- ▮ La Lode può solo assumere i valori 'sì' o 'no' :
(Lode = 'sì') OR (Lode = 'no')

Vincoli di tupla

- I vincoli di dominio sono un caso particolare dei **vincoli di tupla**, ovvero vincoli che esprimono **condizioni su ciascuna tupla**, indipendentemente dalle altre.

ESAMI	Matricola	CodCorso	Voto	Lode
	29323	483	28	no
	39654	729	30	sì
	29323	913	26	sì
	35467	913	30	no

- La **Lode** si può assegnare solo se il **Voto** è 30:
 $(\text{Voto} = 30) \text{ OR NOT}(\text{Lode} = \text{'sì'})$
- Nello schema **PAGAMENTI**(Data, ImportoLordo, Ritenute, Netto) si ha:

$$\text{ImportoLordo} = \text{Netto} + \text{Ritenute}$$

Vincoli di chiave: intuizione

- I vincoli di chiave, che giocano un ruolo molto importante, vietano la presenza di tuple distinte che hanno lo stesso valore su uno o più attributi.

STUDENTI

Matricola	CodiceFiscale	Cognome	Nome	DataNascita
210629323	BNCGRG78L21A944Z	Bianchi	Giorgio	21/07/1978
216635467	RSSNNA78A53A944N	Rossi	Anna	13/01/1978
160239654	VRDMRC79H20F839U	Verdi	Marco	20/06/1979
214842132	VRDMRC79H20G125T	Verdi	Marco	20/06/1979

- Un valore di Matricola **identifica univocamente** uno studente;
- analogamente per il CodiceFiscale
- e ogni insieme di attributi che includa Matricola o CodiceFiscale
 - {Matricola, Cognome}, {CodiceFiscale, Nome}, ... ;
- ma possono esistere due tuple uguali su {Cognome, Nome, DataNascita}.

Chiavi e superchiavi

- Dato uno schema $R(X)$, un insieme di attributi $K \subseteq X$ è:
 - ▮ una **superchiave** se e solo se
in ogni stato ammissibile r di $R(X)$ non esistono due tuple distinte t_1 e t_2 tali che $t_1[K] = t_2[K]$;
 - ▮ una **chiave** se e solo se
è una superchiave minimale, ovvero non esiste $K' \subset K$ con K' superchiave.
- Una chiave è pertanto un **identificatore minimale** per ogni r su $R(X)$.
- Nella relazione **STUDENTI**:
 - $\{\text{Cognome}, \text{Nome}, \text{DataNascita}\}$ **non è superchiave**;
 - $\{\text{Matricola}, \text{Cognome}\}$ e $\{\text{CodiceFiscale}, \text{Nome}\}$ sono **superchiavi**;
 - $\{\text{Matricola}\}$ e $\{\text{CodiceFiscale}\}$ sono **chiavi**.

Vincoli espressi a livello di schema

- I vincoli di chiave si esprimono a livello di schema, sulla base di un'analisi della realtà che si vuole modellare mediante relazioni, e **limitano l'insieme di estensioni legali** (o “ammissibili”, “corrette”, “valide”, ecc.).

ESAMI	Matricola	CodCorso	Voto	Lode
	29323	483	28	no
	39654	729	30	sì
	29323	913	26	no
	35467	913	30	sì

- ▮ La (unica) **chiave** è {Matricola, CodiceCorso}.
- ▮ Questa particolare estensione soddisfa anche altri vincoli, ad esempio {Matricola, Voto} è un identificatore, **ma ciò è puramente casuale**.

Importanza delle chiavi

- L'esistenza delle chiavi garantisce l'accessibilità a ciascun dato del DB, in quanto ogni singolo valore è univocamente individuato da:



- Le chiavi sono lo strumento principale attraverso il quale vengono correlati i dati in relazioni diverse (“il modello relazionale è basato su valori”).

Legami basati sui valori

LINEE_AEREE

<u>CodLine</u>	NomeLinea
L001	TWA
L002	ALITALIA
...	...

DESTINAZIONI

<u>CodDest</u>	NomeDest	Nazione
FCO	ROMA	ITALIA
JFK	NEW YORK	USA
...

VOLI

<u>NumVolo</u>	CodLinea	CodDest	Giorno	Ora	Durata	Attivo
TW056	L001	JFK	LUN	9:00	2	SI
AZ854	L002	FCO	MER	22:30	8	SI
...			

Chiavi e valori nulli

- In presenza di valori nulli entrambe le funzioni svolte dalle chiavi (identificazione e correlazione) possono venire a mancare.

STUDENTI

Matricola	CodiceFiscale	Cognome	Nome	DataNascita
NULL	NULL	Bianchi	Giorgio	21/07/1978
216635467	RSSNNA78A53A944N	Rossi	Anna	13/01/1978
NULL	VRDMRC79H20F839U	Verdi	Marco	20/06/1979
214842132	NULL	Verdi	Marco	20/06/1979


- La prima tupla non è identificabile in alcun modo, pertanto:
 - ▮ è necessario specificare il valore di almeno una chiave!
- La terza e quarta tupla potrebbero riferirsi allo stesso studente, pertanto:
 - ▮ non è sufficiente specificare il valore di una chiave!

Chiave primaria

- Per evitare i problemi visti è necessario scegliere una chiave, detta **chiave primaria (primary key)**, su cui non si ammettono valori nulli.
- Convenzionalmente si sottolineano gli attributi che costituiscono la chiave primaria.

STUDENTI

<u>Matricola</u>	CodiceFiscale	Cognome	Nome	DataNascita
210629323	NULL	Bianchi	Giorgio	21/07/1978
216635467	RSSNNA78A53A944N	Rossi	Anna	13/01/1978
160239654	VRDMRC79H20F839U	Verdi	Marco	20/06/1979
214842132	NULL	Verdi	Marco	20/06/1979



Nei casi in cui per nessuna chiave si possa garantire la disponibilità di valori, è necessario introdurre un nuovo attributo (un “**codice**”) che svolga le funzioni di chiave primaria; si pensi ad esempio al caso in cui non si riesca a identificare un paziente all’arrivo a un pronto soccorso ospedaliero.

Vincoli di integrità referenziale

- I vincoli sinora visti sono tutti di tipo **intra-relazionale**, in quanto interessano una relazione alla volta.
- Viceversa, i **vincoli di integrità referenziale** sono importanti tipi di vincoli **inter-relazionali** che enfatizzano come le correlazioni tra le tuple siano spesso ottenute usando i valori delle chiavi.
- Si considerino due relazioni $R_1(X_1)$ e $R_2(X_2)$ di un DB R , e sia Y un insieme di attributi in X_2 .
- Un vincolo di integrità referenziale su Y impone che in ogni stato $r = \{r_1, r_2, \dots\}$ del DB l'insieme dei valori di Y in r_2 sia un sottoinsieme dell'insieme dei valori della chiave primaria di $R_1(X_1)$ presenti nello stato r_1 .
- L'insieme Y viene detto una **foreign key** (o “**chiave importata**”).

Esempi di foreign key

STUDENTI

<u>Matricola</u>	Cognome	Nome	DataNascita
29323	Bianchi	Giorgio	21/06/1978
35467	Rossi	Anna	13/04/1978
39654	Verdi	Marco	20/09/1979
42132	Neri	Lucia	15/02/1978

CORSI

<u>CodCorso</u>	Titolo	CodDocente	Anno
483	Analisi	0201	1
729	Analisi	0021	1
913	Sistemi Informativi	0123	2

ESAMI

foreign key

<u>Matricola</u>	<u>CodCorso</u>	Voto	Lode
29323	483	28	no
39654	729	30	sì
29323	913	26	no
35467	913	30	sì

In **CORSI**, {**CodDocente**} è una foreign key.

In **ESAMI**, {**Matricola**} è una foreign key, così come {**CodCorso**}.

Foreign key: precisazioni

- In generale la foreign key Y e la primary key K di $R1(X1)$ possono includere attributi con nomi diversi:

CORSI	<u>Codice</u>	Titolo	CodDocente	Anno
	483	Analisi I	0201	1
	729	Analisi I	0021	1

ESAMI	<u>NumMatricola</u>	<u>CodCorso</u>	Voto	Lode
	29323	483	28	no


- Foreign key e primary key possono far parte della stessa relazione, ovviamente con $Y \neq K$.

PERSONALE	<u>Codice</u>	Cognome	...	CodResponsabile
	123	Rossi	...	325
	134	Verdi	...	325
	325	Neri

Foreign key: valori nulli

- In presenza di **valori nulli**, i vincoli di integrità referenziale si possono parzialmente rilassare:

PERSONALE



<u>Codice</u>	Cognome	...	CodResponsabile
123	Rossi	...	325
134	Verdi	...	325
325	Neri	...	NULL

- Nei DBMS un vincolo di integrità referenziale può anche esprimersi con riferimento a una generica chiave (quindi anche non primaria):

STUDENTI

<u>Matricola</u>	CodiceFiscale	Cognome	Nome	DataNascita
29323	BNCGRG78L21A944Z	Bianchi	Giorgio	21/07/1978
35467	RSSFLV78M53G125O	Rossi	Flavia	13/08/1978

REDDITI

<u>CF</u>	Imponibile
BNCGRG78L21A944Z	45300

Foreign key: notazioni negli schemi

- Si usano, nei vari testi sui DB relazionali, diverse notazioni per indicare nella definizione di uno schema le foreign key. Ad esempio:

AGENZIE(Agenzia, Luogo)

IMPIEGATI(CodImpiegato, Cognome, Nome, CodAgenzia)

FK: CodAgenzia REFERENCES AGENZIE(Agenzia)

o più semplicemente

FK: CodAgenzia REFERENCES AGENZIE

oppure

AGENZIE(Agenzia, Luogo)

IMPIEGATI(CodImpiegato, Cognome, Nome, CodAgenzia: AGENZIE)

Queste notazioni sono da intendersi come semplificazioni rispetto alla sintassi del linguaggio SQL.

Valori nulli e altri vincoli: notazioni negli schemi

- Per quanto riguarda gli attributi e le foreign key che ammettono valori nulli, nella definizione di schemi relazionali a volte si usa il simbolo * proprio per denotare che è ammesso NULL come valore.
- Per denotare che un attributo A ammette solo valori unici, ovvero non ripetuti, a volte si usa scrivere: Unique(A).

AGENZIE(CodAgenzia, Nome, Sede, Direttore: IMPIEGATI)

Unique(Direttore)

IMPIEGATI(Codice, Nome, Cognome, Agenzia*: AGENZIE, DataInizio*)

Domande?

