

DBMS e Modello Relazionale

Daniele Riboni

Università degli Studi di Cagliari

Dipartimento di Matematica e Informatica

Argomenti delle prossime lezioni

DataBase Management System (DBMS), modelli dei dati,
linguaggi e utenti

Relazioni e tabelle

Vincoli di integrità

Libro di testo: Capitoli 1 e 2

Parte 1

DataBase Management System (DBMS)

Modelli dei dati

Linguaggi e utenti

Informazioni e dati

- **Informazione**: notizia, nozione, elemento che consente di avere conoscenza più o meno esatta di fatti, situazioni, modi di essere
- **Dato**: ciò che è immediatamente presente alla conoscenza, prima di ogni elaborazione;
(in informatica) elementi di informazione costituiti da simboli che devono essere elaborati

Gestione delle informazioni

- Nelle attività umane, le **informazioni** vengono gestite in forme diverse:
 - idee informali
 - linguaggio naturale (scritto o parlato)
 - disegni, grafici, schemi
 - sistemi informatici complessi
- e su vari supporti
 - mente umana, carta, dispositivi elettronici



Gestione dei dati

- I DBMS relazionali sono specializzati nel trattare dati, non informazioni
- I dati sono spesso il risultato di forme di organizzazione schematica delle informazioni
- Ad esempio, nei servizi anagrafici:
 - Nome e cognome
 - Codice fiscale



Base di dati

- Due accezioni:
- **Insieme organizzato di dati** utilizzati per il supporto allo svolgimento di attività (azienda, ufficio, persona...)
- Insieme di dati **gestito da un Database Management System** (DBMS) – cioè, un software specializzato nella memorizzazione e interrogazione di dati

Uno dei datacenter di Google



Server racks in un datacenter di Google



Database Management System

Sistema che gestisce collezioni di dati:

- Grandi – Persistenti – Condivise

Garantendo:

- Privatezza – Affidabilità – Efficienza

Esempi:

- IBM DB2, Oracle, Microsoft SQLServer, MySQL, PostgreSQL, svariati DBMS di nuova generazione...

Le basi di dati sono ... grandi

- Dimensioni **molto maggiori della memoria centrale** dei sistemi di calcolo utilizzati
- Esempi di dimensioni molto grandi
 - 500 Gigabyte (dati transazionali)
 - 10 Terabyte (dati decisionali)
 - 500 Terabyte (dati scientifici)
 - 100 miliardi di record (post pubblicati negli ultimi 6 mesi)

Le basi di dati sono ... persistenti

- Hanno un tempo di vita indipendente dalle singole esecuzioni dei programmi che le utilizzano
- Devono **persistere su memoria di massa** (non possono stare solo in memoria primaria!)

Le basi di dati sono... condivise

- Una base di dati è una risorsa integrata, condivisa fra applicazioni
- Conseguenze
 - Attività diverse su dati condivisi:
 - meccanismi di **autorizzazione**
 - Accessi di più utenti ai dati condivisi:
 - controllo della **concorrenza**

I DBMS garantiscono ... privacy

- Si possono definire meccanismi di **autorizzazione**
 - l'utente A è autorizzato a leggere tutti i dati e a modificare X
 - l'utente B è autorizzato a leggere i dati X e a modificare i dati Y

I DBMS garantiscono... affidabilità

- Affidabilità (per le basi di dati):
 - **Resistenza a malfunzionamenti** hardware e software
- Una base di dati è una risorsa pregiata e quindi deve essere conservata a lungo termine

I DBMS debbono essere...efficienti

- Cercano di utilizzare al meglio le **risorse** di spazio di **memoria** (principale e secondaria) e **tempo** (di esecuzione e di risposta)
- I DBMS, con tante funzioni, rischiano l'inefficienza e per questo ci sono grandi investimenti e competizione

I DBMS debbono essere...efficaci

- Cercano di rendere **produttive** le attività dei loro utilizzatori, offrendo funzionalità articolate, potenti e flessibili

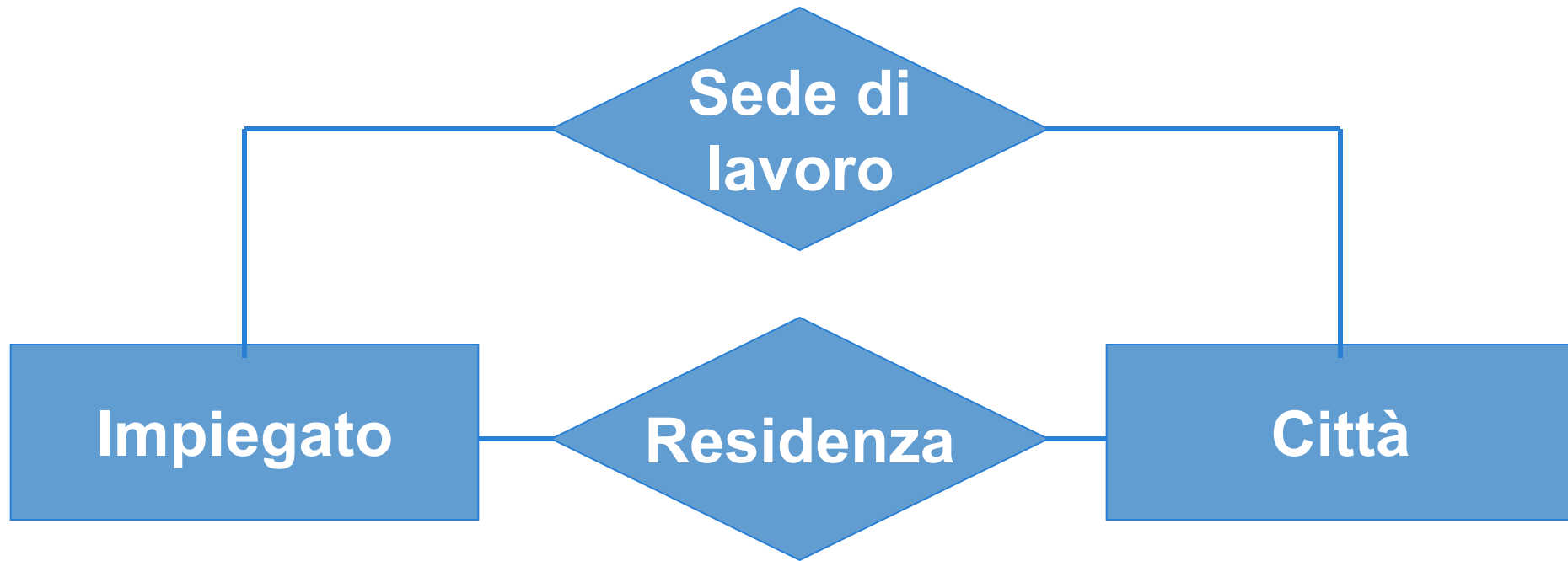
Due tipi (principali) di modelli

- Modelli concettuali
- Modelli logici

Modelli concettuali

- Permettono di rappresentare i dati in modo **indipendente da ogni sistema**
 - cercano di descrivere i concetti del mondo reale
 - sono utilizzati nelle fasi preliminari di progettazione
- Il più diffuso è il modello **Entity-Relationship (ER)**

Esempio di modello concettuale ER



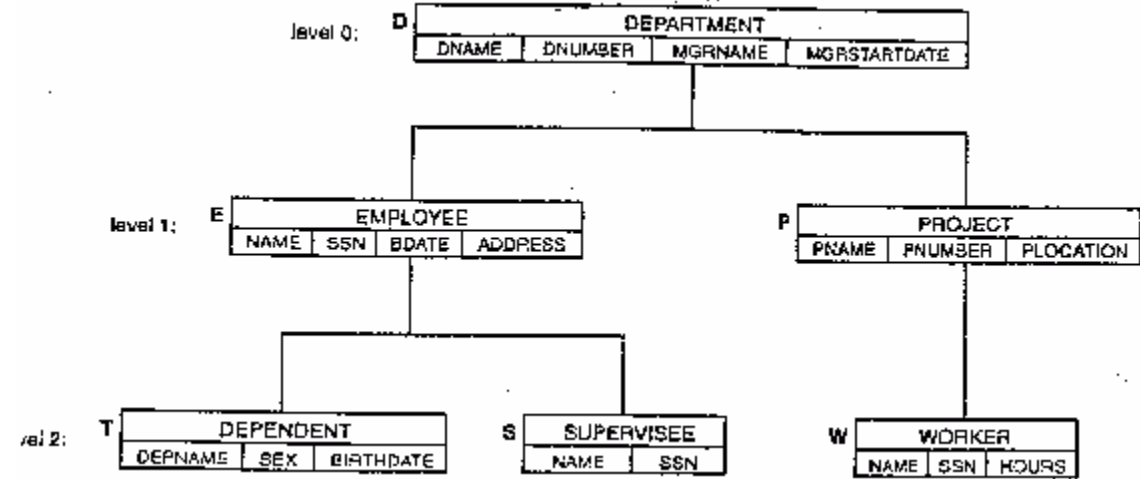
Lo stesso impiegato può risiedere in una città e avere sede di lavoro diversa

Modello logico dei dati

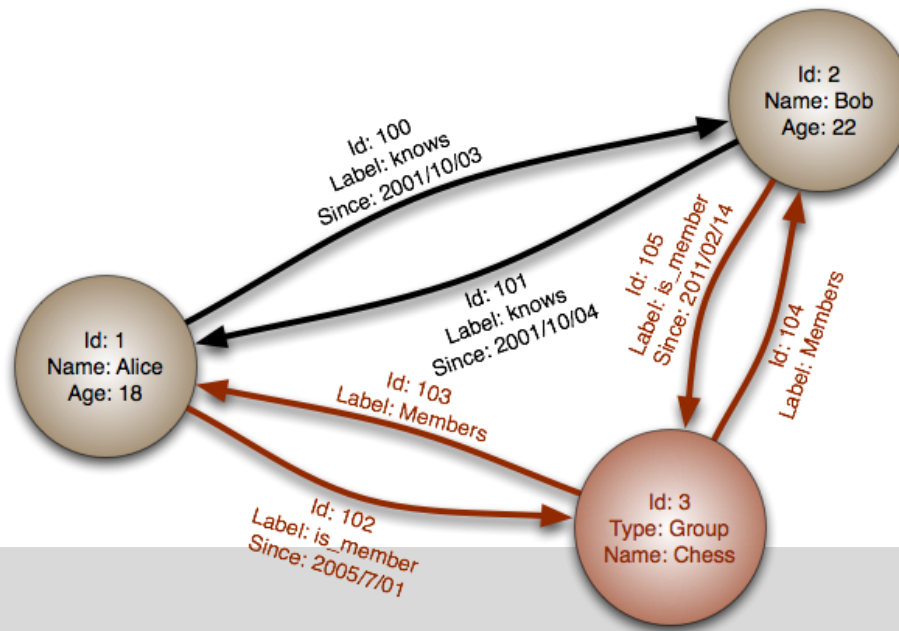
- Insieme di costrutti utilizzati per **organizzare** i dati di interesse
- Componente fondamentale: meccanismi di **strutturazione**
- Esempio: il **modello relazionale** prevede il costruttore **relazione**, che permette di definire insiemi di **record** omogenei

Modello logico dei dati - Esempi

- Modello **gerarchico**

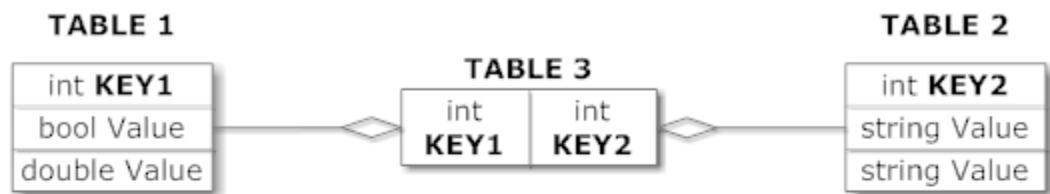


- Modello **a grafo**



Modello dei dati - Esempi

Relational Model



Document Model

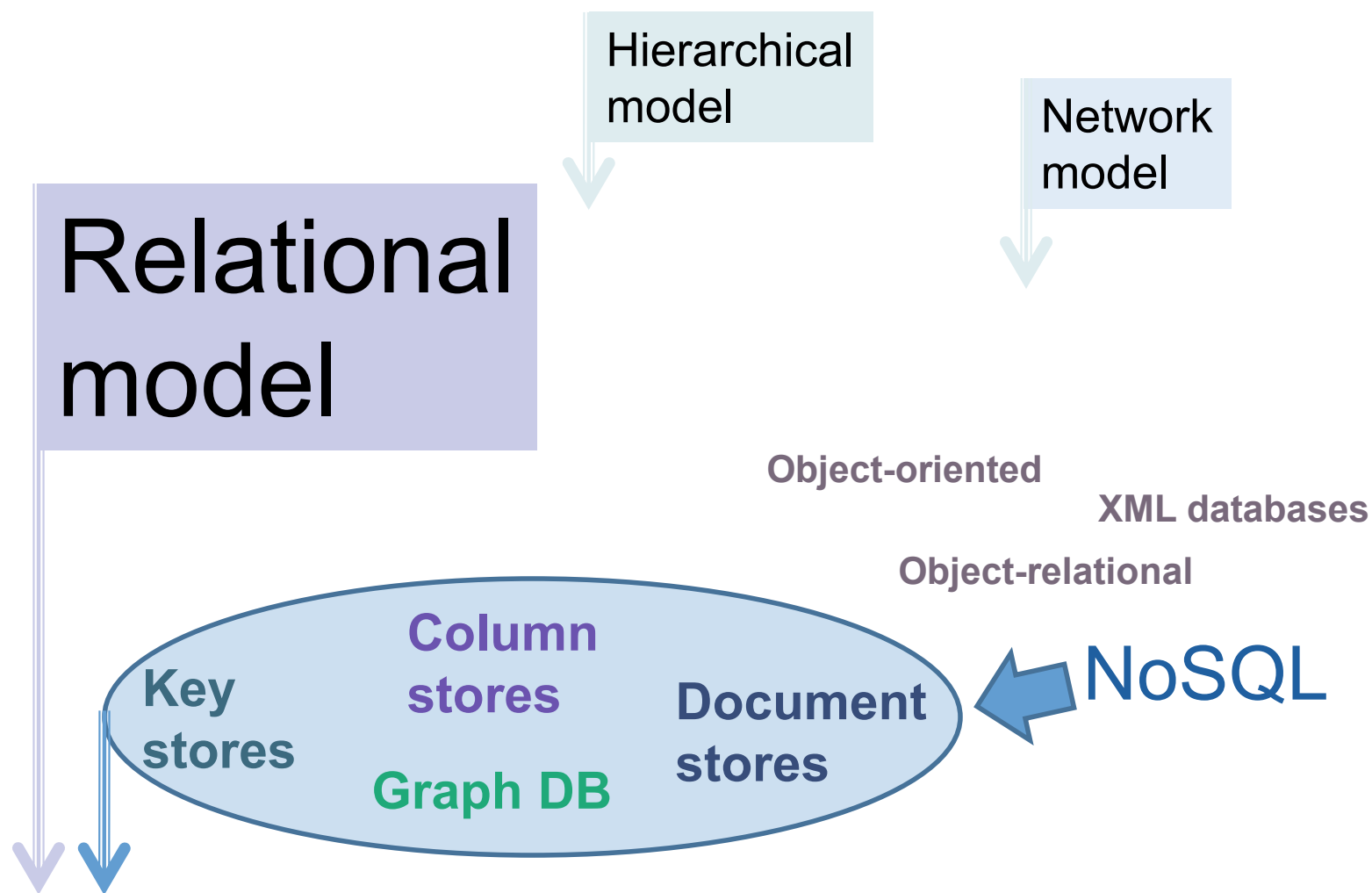
Collection ("Things")



- Noi studieremo il **Modello Relazionale**
















Storia dei modelli logici per DBMS

- 1960s
- 1970s
- 1980s
- 1990s
- 2000s
- 2010s
- 2020s



DBMS popularity – September 2025

424 systems in ranking, September 2025

Rank			DBMS	Database Model	Score		
Sep 2025	Aug 2025	Sep 2024			Sep 2025	Aug 2025	Sep 2024
1.	1.	1.	Oracle	Relational, Multi-model 	1170.62	-50.08	-115.97
2.	2.	2.	MySQL	Relational, Multi-model 	891.77	-23.69	-137.72
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model 	717.32	-36.84	-90.45
4.	4.	4.	PostgreSQL	Relational, Multi-model 	657.17	-14.08	+12.81
5.	5.	5.	MongoDB 	Document, Multi-model 	380.50	-15.08	-29.74
6.	6.	 7.	Snowflake	Relational	190.19	+11.29	+56.47
7.	7.	 6.	Redis	Key-value, Multi-model 	145.17	-2.02	-4.25
8.	8.	 9.	IBM Db2	Relational, Multi-model 	124.19	-3.12	+1.14
9.	9.	 14.	Databricks	Multi-model 	124.06	+8.25	+39.82
10.	10.	 8.	Elasticsearch	Multi-model 	118.26	+3.99	-10.53

From <http://db-engines.com/en/ranking>

Basi di dati: schema e istanza

Tabella “Corsi” ← **Schema** della base di dati

Insegnamento	Docente	Aula	Ora
Analisi matem. I	Luigi Neri	N1	8:00
Basi di dati	Piero Rossi	N2	9:45
Chimica	Nicola Mori	N1	9:45
Fisica I	Mario Bruni	N1	11:45
Fisica II	Mario Bruni	N3	9:45
Sistemi inform.	Piero Rossi	N3	8:00

← **Istanza** della base di dati

Schema e istanza

In ogni base di dati esistono:

- Lo **schema**, che ne descrive la struttura (statica)
(es.: l'intestazione di una tabella)
- L'**istanza** (i valori attuali, che possono cambiare)
(es.: il “corpo” della tabella)

Strutturazione dei dati e interrogazioni

Corsi

Corso	Docente	Aula
Basi di dati	Rossi	DS3
Sistemi	Neri	N3
Reti	Bruni	N3
Controlli	Bruni	G

Aule

Nome	Edificio	Piano
DS1	OMI	Terra
N3	OMI	Terra
G	Pincherle	Primo

**Risultato di una
interrogazione:
“Dimmi edificio
e piano delle aule
dei corsi”**

Corso	Aula	Edificio	Piano
Sistemi	N3	OMI	Terra
Reti	N3	OMI	Terra
Controlli	G	Pincherle	Primo

Esercizio

Schema LOGICO (relazionale)

- Quali informazioni descrive questo schema?
- Qual è il modello concettuale dello schema?

PAZIENTI

Cod	Cognome	Nome
A102	Necchi	Luca
B372	Rossini	Piero
B543	Missoni	Nadia
B444	Missoni	Luigi
S555	Rossetti	Gino

REPARTI

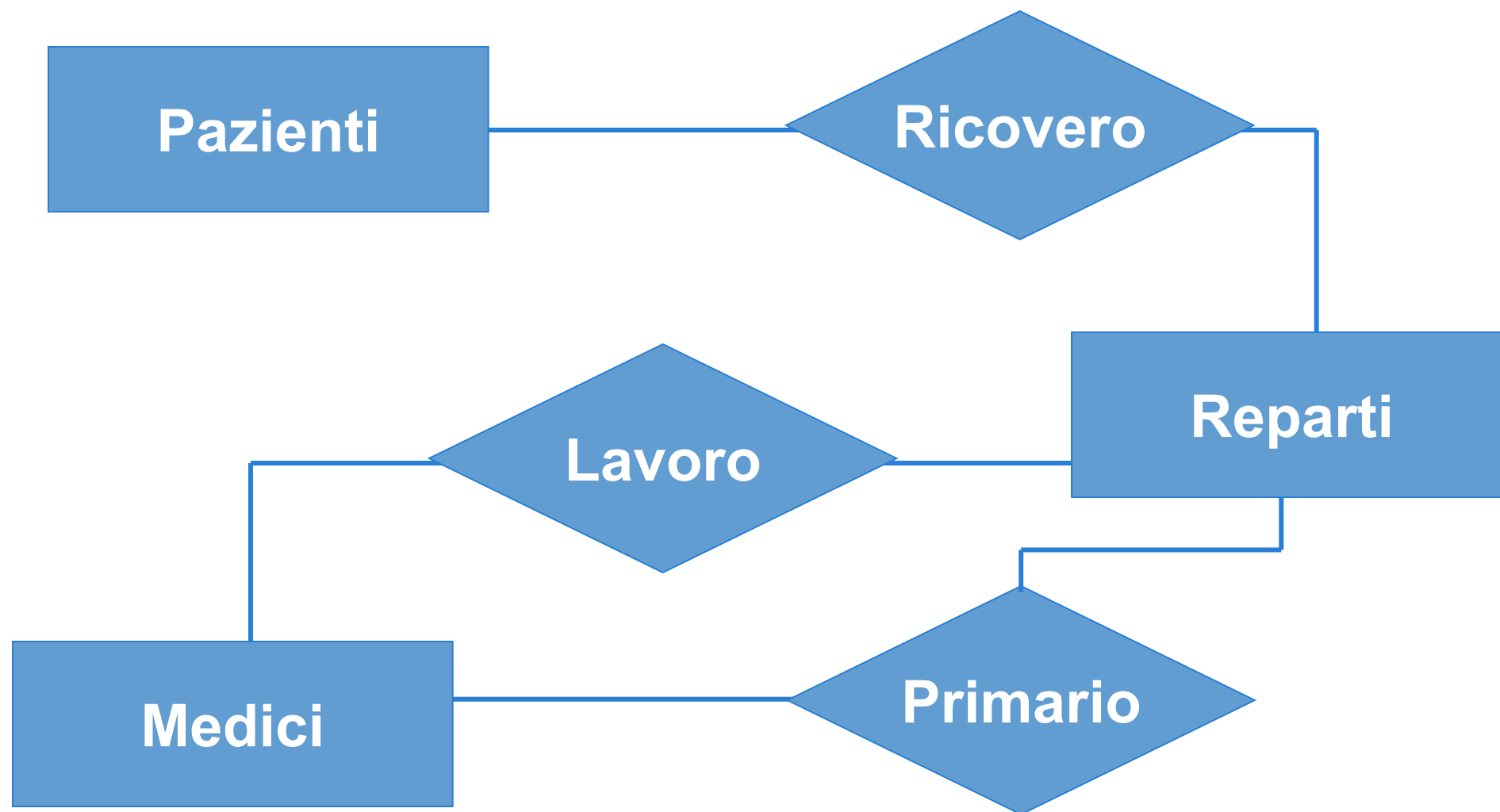
Cod	Nome	Primario
A	Chirurgia	203
B	Medicina	574
C	Pediatria	530

RICOVERI

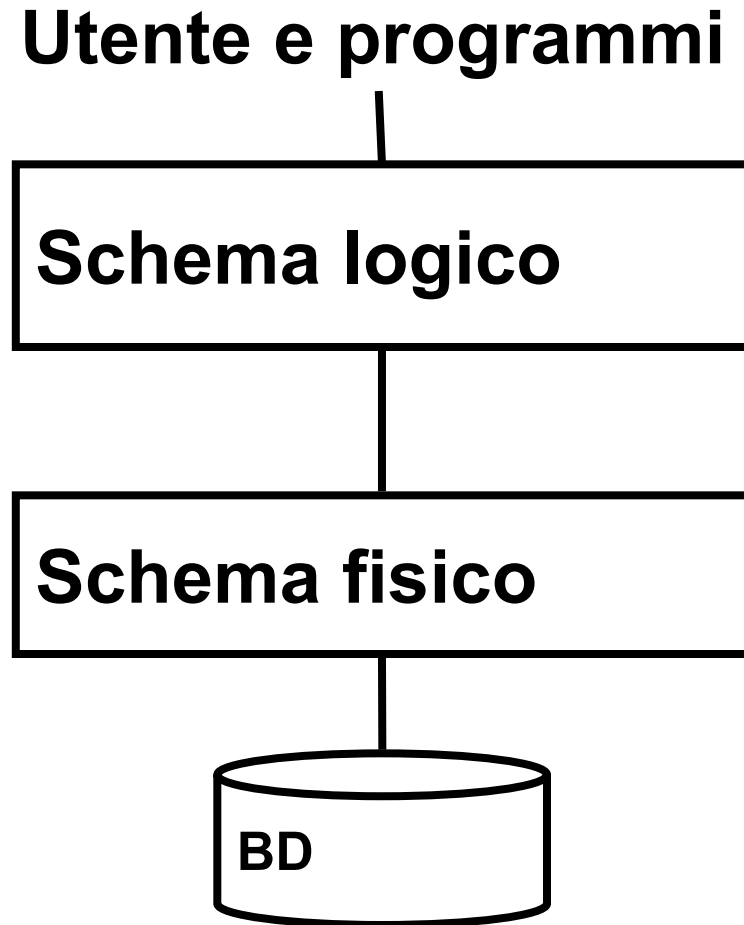
Paziente	Inizio	Fine	Reparto
A102	2/05/94	9/05/94	A
A102	2/12/94	2/01/95	A
S555	5/10/94	3/12/94	B
B444	1/12/94	2/01/95	B
S555	5/10/94	1/11/94	A

MEDICI

Matr	Cognome	Nome	Reparto
203	Neri	Piero	A
574	Bisi	Mario	B
431	Bargio	Sergio	B
530	Belli	Nicola	C
405	Mizzi	Nicola	A
201	Monti	Mario	A

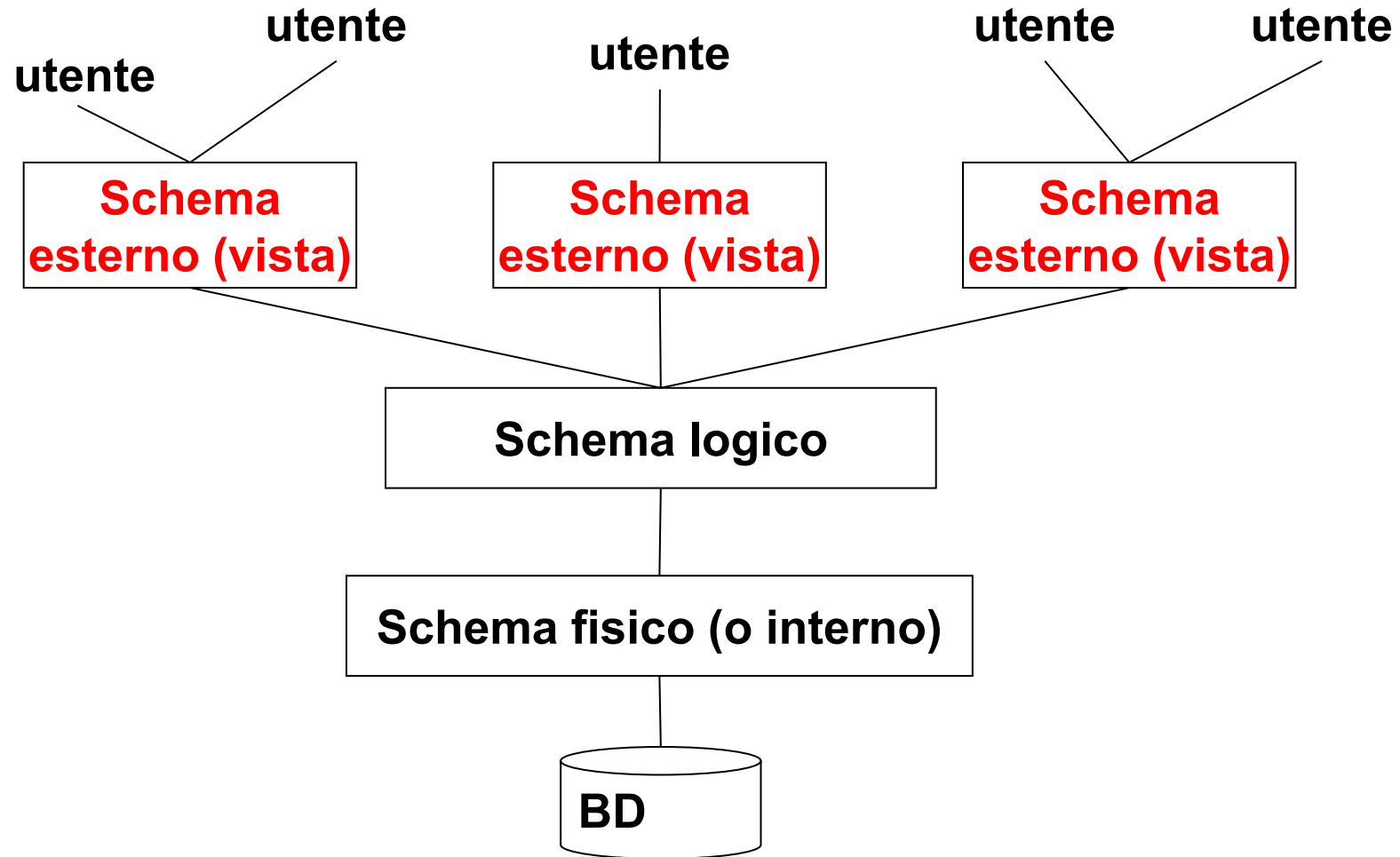


Architettura di un DBMS



Il progettista definisce lo **schema concettuale**, che viene tradotto nello **schema logico** (tabelle), implementate nel DBMS relazionale tramite strutture dati e algoritmi opportuni (**livello fisico**). L'utente e i programmi interagiscono solo con lo schema logico.

Architettura standard a tre livelli per DBMS



Architettura standard per DBMS

- **Schema esterno**: “vista” parziale o derivata di una parte della base di dati in un modello logico
- **Schema logico**: descrizione dell'intera base di dati nel modello logico “principale” del DBMS
- **Schema fisico**: rappresentazione dello schema logico per mezzo di strutture fisiche di memorizzazione

Indipendenza dei dati

- Conseguenza della articolazione in livelli
- L'accesso avviene solo tramite il livello esterno (che può coincidere con il livello logico)
- Due forme:
 - indipendenza **fisica**
 - indipendenza **logica**

Indipendenza fisica

- Il livello logico e quello esterno sono indipendenti da quello fisico
- Una relazione è utilizzata nello stesso modo qualunque sia la sua realizzazione fisica
- La realizzazione fisica può cambiare senza che debbano essere modificati i programmi

Indipendenza logica

- Il livello esterno è indipendente da quello logico
- Aggiunte o modifiche alle viste non richiedono modifiche al livello logico
- Modifiche allo schema logico che lascino inalterato lo schema esterno sono trasparenti

Linguaggi per basi di dati

- Disponibilità di vari **linguaggi e interfacce**
 - ⇒ linguaggi testuali interattivi (SQL)
 - ⇒ comandi (SQL) immersi in un linguaggio ospite (Python, Java, C ...)
 - ⇒ con interfacce amichevoli (senza linguaggio testuale)

DDL e DML

Data definition language (DDL)

- per la definizione e la modifica di **schemi** (logici, esterni, fisici) e altre operazioni generali

Data manipulation language (DML)

- per l'interrogazione e l'aggiornamento (inserimento, cancellazione, modifica) di **istanze** nella base di dati

Structured Query Language (SQL)

- "Trovare i corsi tenuti in aule al piano terra"

Corsi

Corso	Docente	Aula
Basi di dati	Rossi	DS3
Sistemi	Neri	N3
Reti	Bruni	N3
Controlli	Bruni	G

Aule

Nome	Edificio	Piano
DS1	OMI	Terra
N3	OMI	Terra
G	Pincherle	Primo

SQL come linguaggio interattivo

```
SELECT Corso, Aula, Piano  
FROM Aule, Corsi  
WHERE Nome = Aula  
      AND Piano = 'Terra'
```

Corso	Aula	Piano
Sistemi	N3	Terra
Reti	N3	Terra

Un'operazione DDL (sullo schema)

Codice SQL per **creare una tabella**:

```
CREATE TABLE orario (  
    insegnamento    CHAR(20) ,  
    docente          CHAR(20) ,  
    aula             CHAR(4)  ,  
    ora              CHAR(5)  )
```

Vero o falso?

1. L'indipendenza dei dati permette di scrivere programmi senza conoscere le strutture fisiche dei dati
2. L'indipendenza dei dati permette di modificare le strutture fisiche dei dati senza dover modificare i programmi che accedono alla base di dati
3. L'indipendenza dei dati permette di scrivere programmi conoscendo solo lo schema concettuale della base di dati

Vero o falso?

1. La distinzione fra DDL e DML corrisponde alla distinzione fra schema e istanza
2. Le istruzioni DML permettono di modificare la base di dati ma non di interrogarla
3. Le istruzioni DDL permettono di specificare la struttura della base di dati ma non di modificarla
4. Non esistono linguaggi che includono sia istruzioni DDL sia istruzioni DML

Ruoli nei DBMS

Progettisti e sviluppatori di DBMS

Progettisti e amministratori della base di dati (DBA)

Analisti, progettisti e sviluppatori di applicazioni

Utenti

***a quali livelli lavorano
queste figure?***

Database administrator (DBA)

- Persona o gruppo di persone responsabile del controllo centralizzato e della **gestione** del sistema, delle **prestazioni**, dell'**affidabilità**, delle **autorizzazioni**
- Le funzioni del DBA includono quelle di **progettazione**, anche se in progetti complessi ci possono essere distinzioni

Riepilogo: concetti chiave

- DBMS, dato e informazione, base di dati, modello dei dati
- Ridondanza, incongruenza, DBMS vs file system
- Schema e istanza, modello concettuale e logico, schema fisico, indipendenza logica e fisica dei dati
- DDL e DML
- Ruoli nei DBMS, DBA

Parte 2

Il modello relazionale

Relazioni e tabelle

Chiavi e vincoli di integrità

Il modello relazionale

- È il modello logico attualmente più **diffuso**
- Proposto nel 1970
- Disponibile in DBMS commerciali dal 1981
- Si basa sul concetto matematico di **relazione**
- Le relazioni hanno naturale rappresentazione per mezzo di **tabelle**

Relazione: tre accezioni

- Relazione **matematica**: come nella teoria degli insiemi
- Relazione secondo il **modello relazionale** dei dati
- Relazione (dall'inglese relationship) tra entità astratte nel **modello Entity-Relationship**
(noi le chiameremo **associazioni**)

Relazione matematica, esempio

- $D_1 = \{a, b\}$
- $D_2 = \{x, y, z\}$
- prodotto cartesiano $D_1 \times D_2 = \{(a, x), (a, y), (a, z), (b, x), (b, y), (b, z)\}$

a	x
a	y
a	z
b	x
b	y
b	z

- una **relazione** $r \subseteq D_1 \times D_2$

a	x
a	z
b	z

Relazione matematica

- 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

Relazione matematica, proprietà

- Una relazione matematica è un insieme di n-uple i cui elementi (d_1, d_2, \dots) sono ordinati:
 - (d_1, \dots, d_n) tali che $d_1 \in D_1, \dots, d_n \in D_n$
- **Siccome la relazione matematica è un insieme:**
 - non c'è ordinamento fra le n-uple
 - le n-uple (“**tuple**”) sono distinte: un insieme non può contenere elementi ripetuti!
 - ciascuna tupla è ordinata: l' i-esimo valore proviene dall' i-esimo dominio

Relazione matematica, esempio

- Ciascuno dei domini ha due ruoli diversi, distinguibili attraverso la posizione:
 - La struttura della relazione matematica è posizionale

$$Partite \subseteq String \times String \times Int \times Int$$

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

Relazione nel modello relazionale

- Negli RDBMS* la struttura è 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

* RDBMS = Relational DBMS

Proprietà di tabelle e relazioni

Nel modello relazionale:

- In una tabella che rappresenta una relazione
 - l'ordinamento tra le righe è irrilevante
 - l'ordinamento tra le colonne è irrilevante

Il modello relazionale è basato su valori

- Gerarchico e reticolare
 - utilizzano riferimenti espliciti (**puntatori**) fra record
- Relazionale "è basato su valori"
 - anche i riferimenti fra dati in strutture (relazioni) diverse sono rappresentati per mezzo dei **valori** stessi

Matricola	Cognome	Nome	Data di nascita	studenti
6554	Rossi	Mario	05/12/1978	
8765	Neri	Paolo	03/11/1976	
9283	Verdi	Luisa	12/11/1979	
3456	Rossi	Maria	01/02/1978	

Studente	Voto	Corso	esami
3456	30	04	
3456	24	02	
9283	28	01	
6554	26	01	

Codice	Titolo	Docente	corsi
01	Analisi	Mario	
02	Chimica	Bruni	
04	Chimica	Verdi	

Il modello relazionale
 è basato sui valori:
 i riferimenti fra diverse
 tuple sono determinati
 dai valori che
 compaiono nelle tuple

Schema di relazione e di database

- **Schema di relazione:** $R (A_1, A_2, \dots, A_n)$

Un nome R (nome della relazione, che diventerà una tabella del database) con un insieme di attributi A_1, \dots, A_n (le colonne della tabella)

Ogni attributo A_i ha un dominio (tipo di dato)

- **Schema della base di dati:**

Insieme di schemi di relazione (le varie tabelle che comporranno il database)

Esercizio

- Considerare le informazioni per la gestione delle presenze di pesci nelle vasche di un acquario civico
- Ogni tipo di pesce ha un nome univoco (es. Piranha), e se ne conosce la famiglia (ciclidi, caracidi, ecc)
- Ogni vasca ha un nome univoco, e se ne conosce la sala, il piano e la capienza in litri
- Si vuole tenere traccia delle presenze e della numerosità dei pesci nelle vasche
- Definire uno schema logico per rappresentare queste informazioni, individuando opportuni domini per i vari attributi e mostrarne un'istanza in forma tabellare

Schema logico (relazionale)

- Pesce (NomeP, Famiglia)
- Vasca (NomeV, Sala, Piano, Capienza)
- Presenze (NomeP, NomeV, Quantità)

PESCE

NomeP	Famiglia
Scalare	Cicli di
Cardinale	Caracidi

VASCA

NomeV	Sala	Piano	Capienza
SudAmerica	S3	1	2000
Messico	S3	2	800

PRESENZE

NomeP	NomeV	Quantità
Scalare	SudAmerica	12
Scalare	Messico	4
Cardinale	Messico	70

Informazione incompleta: che fare?

NomeP	NomeV	Quantità
Scalare	Sudamerica	12
Scalare	Messico	← ???
Cardinale	Messico	70

Informazione incompleta: soluzioni?

- **Non conviene usare valori del dominio** (0, stringa vuota, “???”, ...):
 - potrebbero non esistere valori “non utilizzati”
 - valori “non utilizzati” potrebbero diventare significativi
 - in fase di utilizzo (nei programmi) sarebbe necessario ogni volta tener conto del “significato” di questi valori

Informazione incompleta nel relazionale

- Tecnica rudimentale ma efficace:
 - **NULL**: denota l'assenza di un valore del dominio
- **Non è un valore vero e proprio**, ma rappresenta la mancanza di un valore per quell'attributo
- Rappresenta un valore **inesistente oppure sconosciuto**

Identificazione delle tuple

- Identificare univocamente le tuple consente di **evitare ambiguità**
- È quindi necessario garantire che ogni tupla sia identificabile univocamente, **indipendentemente dalle modifiche** che verranno apportate ai dati
- L'identificazione della tupla permette di **correlare** i dati in relazioni diverse
 - (il modello relazionale è basato su valori)

Superchiave e chiave

- **Superchiave**: insieme di attributi che identificano univocamente le tuple di una relazione
- Formalmente:
 - un insieme **K** di attributi è **superchiave** per la relazione R se R non contiene due tuple distinte t_1 e t_2 con gli stessi valori per gli attributi in K
 - K è chiave per R se è una **superchiave minimale** per R (cioè non contiene un'altra superchiave)

Identificazione delle tuple: la Chiave

- es:
 - Se l'attributo NomeV è **chiave** di Vasca, significa che non posso avere due vasche con lo stesso nome
 - Se la **chiave** di Vasca è la coppia di attributi $\langle \text{NomeV}, \text{Piano} \rangle$, significa che non posso avere due vasche con lo stesso nome sullo stesso piano (ma potrei averle su piani diversi!)

Una chiave

- Matricola è una **chiave**:
 - è **superchiave**
 - contiene un solo attributo e quindi è **minimale**

Matricola	Cognome	Nome	Corso	Nascita
27655	Rossi	Mario	Ing Inf	5/12/78
78763	Rossi	Mario	Ing Inf	3/11/76
65432	Neri	Piero	Ing Mecc	10/7/79
87654	Neri	Mario	Ing Inf	3/11/76
67653	Rossi	Piero	Ing Mecc	5/12/78

Un'altra chiave

- <Cognome, Nome, Nascita> è un'altra chiave:
 - È **superchiave**
 - È **minimale**

Matricola	Cognome	Nome	Corso	Nascita
27655	Rossi	Mario	Ing Inf	5/12/78
78763	Rossi	Mario	Ing Inf	3/11/76
65432	Neri	Piero	Ing Mecc	10/7/79
87654	Neri	Mario	Ing Inf	3/11/76
67653	Rossi	Piero	Ing Mecc	5/12/78

Teorema di esistenza delle chiavi

- Ogni relazione ha almeno una chiave

Dimostrazione:

- Una relazione non può contenere tuple distinte ma uguali
- Quindi, ogni relazione ha come superchiave l'insieme S di tutti i suoi attributi
- ...e quindi ha (almeno) una chiave S

Chiave primaria

- Possono esistere diverse chiavi (ad esempio, sia Matricola, che <Nome, Cognome>)
- Devo scegliere una **chiave primaria** che poi verrà usata per la tabella del database
- In generale, preferisco chiavi composte dal minor numero di attributi

Chiave artificiale

- E se la chiave è troppo “complessa”?
es: consideriamo la relazione “Esami medici”

Esami medici	Paziente	Esame	Esito	Medico	Data	<u>ID</u>
	p23	TAC	POS	03	3/1/16	1
	p5	ECG	NEG	01	2/1/16	2
	p5	ECG	NEG	01	1/1/16	3
	p7	TAC	NEG	04	2/1/16	4

- Creo una **chiave artificiale**: aggiungo un attributo (ID) e gli assegno dei valori univoci (ad esempio, un contatore incrementale)

Chiave primaria

- Notazione: SOTTOLINEATURA

<u>Matricola</u>	Cognome	Nome	Corso	Nascita
86765	Bianchi	Mario	Ing Inf	5/12/88
78763	Rossi	Mario	Ing Civile	3/11/89
65432	Neri	Piero	Ing Mecc	10/7/92
87654	Neri	Mario	Ing Inf	10/7/94
43289	Neri	Mario	Mat	10/7/95

Vero o falso?

- 1. ogni attributo appartiene al massimo ad una chiave
- 2. possono esistere attributi che non appartengono a nessuna chiave
- 3. possono esistere attributi che non appartengono a nessuna superchiave
- 4. una chiave può essere sottoinsieme di un'altra chiave

Vero o falso?

- 5. può esistere una chiave che coinvolge tutti gli attributi
- 6. può succedere che esistano più chiavi e che una di esse coinvolga tutti gli attributi
- 7. una relazione può avere due chiavi primarie
- 8. una superchiave non minimale può essere sottinsieme di una chiave primaria

Esercizio

- Cosa scegli come chiave di Esami?
Ricorda che la chiave può essere formata da più di un attributo
Ricorda che non posso avere 2 tuple con gli stessi valori di chiave!

Esami	Studente	Voto	Lode	Corso
	222222	30		01
	111111	30	e lode	02
	333333	27		03
	222222	24		04

Come chiave
scelgo la coppia:
<Studente, Corso>
(lo stesso studente
può dare più esami
ma non per lo
stesso corso)

Esercizio

- Se scelgo come chiave Studente, non posso avere due esami fatti dallo stesso studente! (nemmeno se fatti per corsi diversi)
- Se scelgo <Studente, Voto>, non ammetto che lo studente possa prendere lo stesso voto in 2 esami diversi

Esami	Studente	Voto	Lode	Corso
	222222	32		01
	111111	30	e lode	02
	333333	27	e lode	03
	222222	24		04

Come chiave
scelgo la coppia:
<Studente, Corso>
(lo stesso studente
può dare più esami
ma non per lo
stesso corso)

Chiavi e valori nulli

- In presenza di valori nulli, i valori della chiave non permettono
 - di identificare le tuple
 - di realizzare i riferimenti da altre relazioni
- Gli attributi che compongono la chiave primaria **non possono assumere valori nulli!**

Esercizio: Trova la chiave primaria

- Cliente (codFisc, telefono, città)
 - Prodotto (Id, nome, marca, prezzo)
 - Acquisto (CF, IdProd, quantità, data)
-
- Cliente (codFisc, telefono, città)
 - Prodotto (Id, nome, marca, prezzo)
 - Acquisto (CF, IdProd, quantità, data)
-
- E se compro più volte lo stesso prodotto?

Esercizio: Trova la chiave primaria

- Cliente (codFisc, telefono, città)
 - Prodotto (Id, nome, marca, prezzo)
 - Acquisto (CF, IdProd, quantità, data)
-
- Cliente (codFisc, telefono, città)
 - Prodotto (Id, nome, marca, prezzo)
 - Acquisto (CF, IdProd, quantità, data)
-
- E se compro più volte lo stesso prodotto lo stesso giorno?

Esercizio: Trova la chiave primaria

- Cliente (codFisc, telefono, città)
 - Prodotto (Id, nome, marca, prezzo)
 - Acquisto (CF, IdProd, quantità, data)
-
- Cliente (codFisc, telefono, città)
 - Prodotto (Id, nome, marca, prezzo)
 - Acquisto (codice, CF, IdProd, quantità, data)
-
- La scelta migliore dipende dalle specifiche richieste...

Una base di dati "scorretta"

- Esistono istanze di basi di dati che non rappresentano informazioni possibili per l'applicazione di interesse

Esami	<u>Studente</u>	Voto	Lode	<u>Corso</u>
	111111	32		01
	111111	30	e lode	02
	111111	18	e lode	03
	333333	24		04

Studenti	<u>Matricola</u>	Cognome	Nome
	111111	Rossi	Mario
	787643	Neri	Piero
	787643	Bianchi	Luca

Vincolo di integrità

- **Proprietà** che deve essere soddisfatta dalle **istanze** che rappresentano informazioni corrette per l'applicazione
- Un vincolo è una **funzione booleana** (un predicato): associa ad ogni istanza il valore vero o falso

Tipi di vincoli

- Vincoli **intra-relazionali**
 - vincoli su valori (o di dominio)
es: $\text{Stipendio} > 0$
 - vincoli di tupla
es: $\text{DataFine} \geq \text{DataInizio}$
- Vincoli **inter-relazionali**
 - tra relazioni diverse

Vincoli di integrità, nota

- Alcuni tipi di vincoli (ma non tutti) sono supportati dai DBMS:
 - Possiamo specificarli nella nostra base di dati e il DBMS ne impedisce la violazione
- Per i vincoli "non supportati", la responsabilità della verifica è del programmatore
 - o dell'utente, se non c'è interfaccia che controlli inserimenti, modifiche e cancellazioni

Vincoli inter-relazionali

- Informazioni in relazioni diverse sono correlate attraverso valori comuni
- In particolare, valori delle chiavi (primarie)
- Le correlazioni debbono essere "coerenti"

Infrazioni

<u>Codice</u>	Data	Vigile	Prov	Numero
34321	1/2/95	3987	MI	39548K
53524	4/3/95	3295	TO	E39548
64521	5/4/96	3295	PR	839548
73321	5/2/98	9345	PR	839548

Vigili

<u>Matricola</u>	Cognome	Nome
3987	Rossi	Luca
3295	Neri	Piero
9345	Neri	Mario
7543	Mori	Gino

Infrazioni

<u>Codice</u>	Data	Vigile	Prov	Numero
34321	1/2/95	3987	MI	39548K
53524	4/3/95	3295	TO	E39548
64521	5/4/96	3295	PR	839548
73321	5/2/98	9345	PR	839548

Auto

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

Vincolo di integrità referenziale

- Un **vincolo di integrità referenziale** (vincolo di chiave esterna, “foreign key”) fra gli attributi X di una relazione $R1$ e un'altra relazione $R2$ impone ai valori su X in $R1$ di comparire come valori della chiave primaria di $R2$

Vincolo di integrità referenziale

<u>studentId</u>	firstName	lastName	courseId
L0002345	Jim	Black	C002
L0001254	James	Harradine	A004
L0002349	Amanda	Holland	C002
L0001198	Simon	McCloud	S042

Chiave
esterna

Correlazione

Chiave
primaria

<u>courseId</u>	courseName
A004	Accounts
C002	Computing
P301	History
S042	Short Course

- Esempio

Esercizio

- Individuare le **chiavi** e i **vincoli** di integrità referenziale

PAZIENTI

Cod	Cognome	Nome
A102	Necchi	Luca
B372	Rossini	Piero
B543	Missoni	Nadia
B444	Missoni	Luigi
S555	Rossetti	Gino

REPARTI

Cod	Nome	Primario
A	Chirurgia	203
B	Medicina	574
C	Pediatria	530

RICOVERI

Paziente	Inizio	Fine	Reparto
A102	2/05/94	9/05/94	A
A102	2/12/94	2/01/95	A
S555	5/10/94	3/12/94	B
B444	1/12/94	2/01/95	B
S555	5/10/94	1/11/94	A

MEDICI

Matr	Cognome	Nome	Reparto
203	Neri	Piero	A
574	Bisi	Mario	B
431	Bargio	Sergio	B
530	Belli	Nicola	C
405	Mizzi	Nicola	A
201	Monti	Mario	A

Soluzione

- Chiavi primarie:
 - - “Cod” per la relazione PAZIENTI
 - - “Paziente” e “Inizio” per la relazione RICOVERI
 - - “Matr” per la relazione MEDICI
 - - “Cod” per la relazione REPARTI
- La scelta fatta sulla relazione RICOVERI presume che un paziente possa essere ricoverato solo una volta nello stesso giorno.

Soluzione

I vincoli di integrità sono:

- Tra l'attributo “Paziente” in RICOVERI e “Cod” in PAZIENTI

Tra “Reparto” nella relazione RICOVERI e “Cod” nella relazione REPARTI

- Tra “Primario” in REPARTI e “Matr” nella relazione MEDICI

Tra “Reparto” in MEDICI e “Cod” in REPARTI

Integrità referenziale e azioni compensative

- Esempio:
 - Viene eliminata una tupla causando una violazione
- Comportamento “standard”:
 - Rifiuto dell'operazione
- Azioni compensative:
 - Eliminazione in cascata
 - Introduzione di valori nulli o valori di default

Eliminazione in cascata

Impiegati

<u>Matricola</u>	Cognome	Progetto
34321	Rossi	IDEA
64521	Verdi	<i>NULL</i>
73032	Bianchi	IDEA

Progetti

<u>Codice</u>	Inizio	Durata	Costo
IDEA	01/2000	36	200
BOH	09/2001	24	150

Introduzione di valori nulli

Impiegati

<u>Matricola</u>	Cognome	Progetto
34321	Rossi	IDEA
53524	Neri	NULL
64521	Verdi	NULL
73032	Bianchi	IDEA

Progetti

<u>Codice</u>	Inizio	Durata	Costo
IDEA	01/2000	36	200
BOH	09/2001	24	150

Esercizio: Integrità referenziale

Definisci i vincoli di integrità referenziale

- Studenti (Matricola, Cognome, Nome, DataNasc)
- Esami (Studente, Voto, Corso)
- Corsi (Codice, Titolo, Docente)
- Vincoli di integrità referenziale fra:
 - L'attributo Corso e la relazione Corsi
 - L'attributo Studente e la relazione Studenti

Esercizio: Integrità referenziale

Definisci i vincoli di integrità referenziale

- Cliente (codFisc, telefono, città)
- Acquisto (CF, IdProd, quantità, data)
- Prodotto (Id, nome, marca, prezzo)
- Vincoli di integrità referenziale fra:
 - L'attributo CF e la relazione Cliente
 - L'attributo IdProd e la relazione Prodotto

Riepilogo: concetti chiave

- Modello basato sui valori, tre accezioni del termine "relazione", tupla, struttura posizionale e non posizionale, proprietà di tabelle e relazioni, schema di relazione e di base di dati, NULL
- Vincoli di tupla, chiave, superchiave, chiave primaria, chiave artificiale, teorema di esistenza delle chiavi, vincoli di integrità referenziale