



Università degli Studi di Cagliari
Corsi di Laurea in Ingegneria Chimica ed Ingegneria Meccanica

FONDAMENTI DI INFORMATICA 1

<http://people.unica.it/gianlucamarcialis>

A.A. 2018/2019

Docente: **Gian Luca Marcialis**

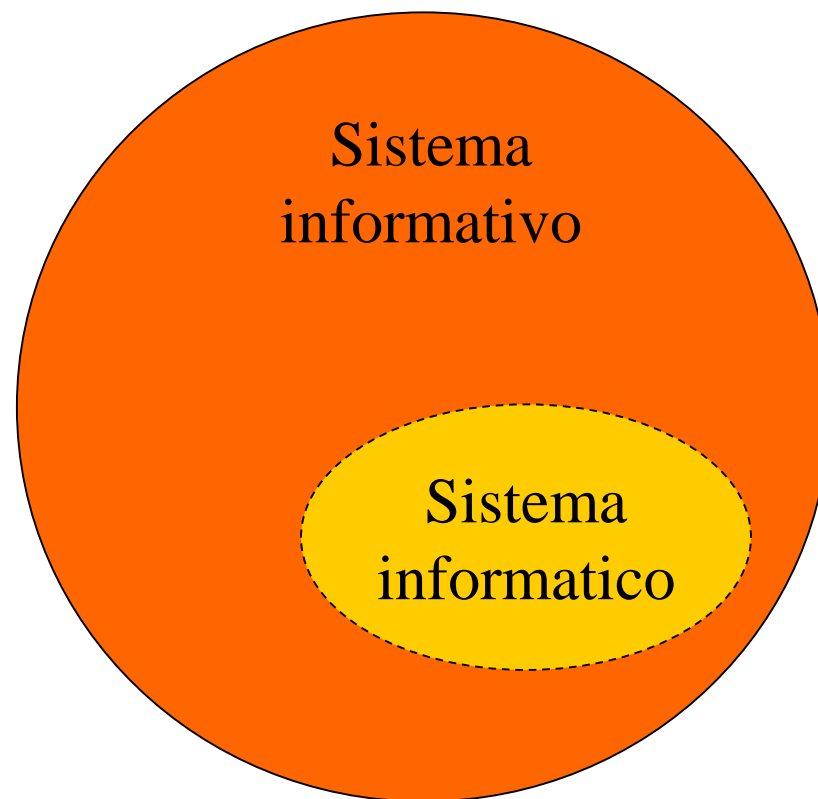
BASI DI DATI

Sommario

- Sistemi informatici e sistemi informativi
- Basi di dati: motivazioni
- Modelli dei dati
 - Gerarchico, reticolare, a oggetti, relazionale
 - DDL e DML
- Il modello relazionale
- Il linguaggio SQL

Sistemi informatici e sistemi informativi

- Un sistema informativo ha il compito di organizzare e gestire le informazioni necessarie per gli scopi dei soggetti che ne fanno uso (università, regione, comune)
- Esso è in parte **indipendente** dalla sua automatizzazione
 - E.g. l'archivio di una banca o l'archivio anagrafico
- **Un sistema informatico è la porzione automatizzata del sistema informativo**



Informazioni e dati

- **Informazione**: notizia, dato o 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
- Il dato, in informatica, può essere visto attraverso diversi livelli di astrazione
 - Come rappresentarlo in forma di bit
 - Come rappresentarlo dal punto di vista “logico”
 - Come rappresentarlo per potere essere “compreso” dall’utente

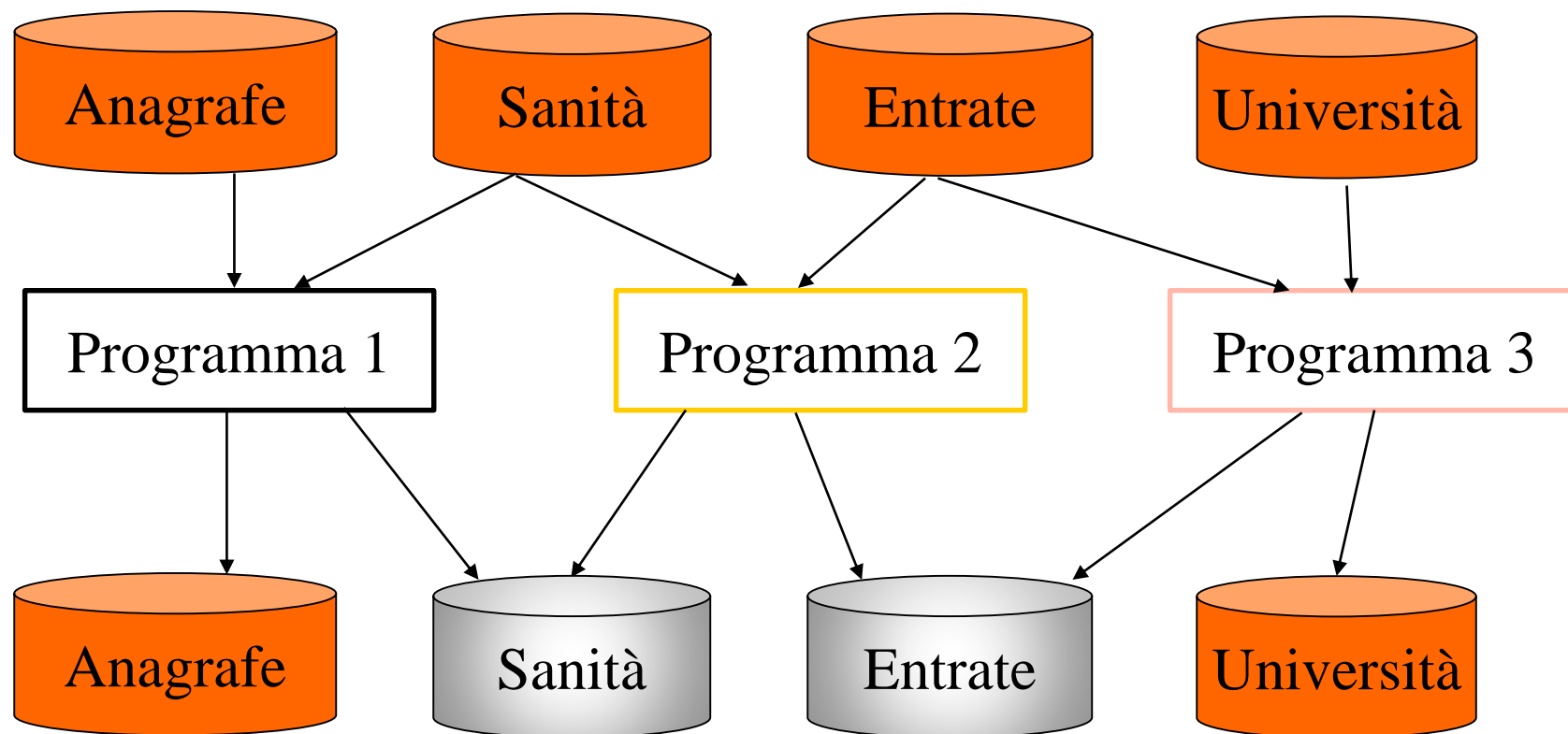
Base di dati: definizione

- **Una base di dati è una collezione di informazioni che vengono organizzate e gestite da un sistema software specifico, il Data Base Management System (DBMS)**
- Il DBMS opera al di sopra del sistema operativo
- Offre linguaggi specifici per l'organizzazione dei dati stessi...
 - Strutture per gestione file
- ...e da essi costrutti per l'estrazione delle informazioni nella forma e nel modo desiderato
 - Linguaggi di interrogazione

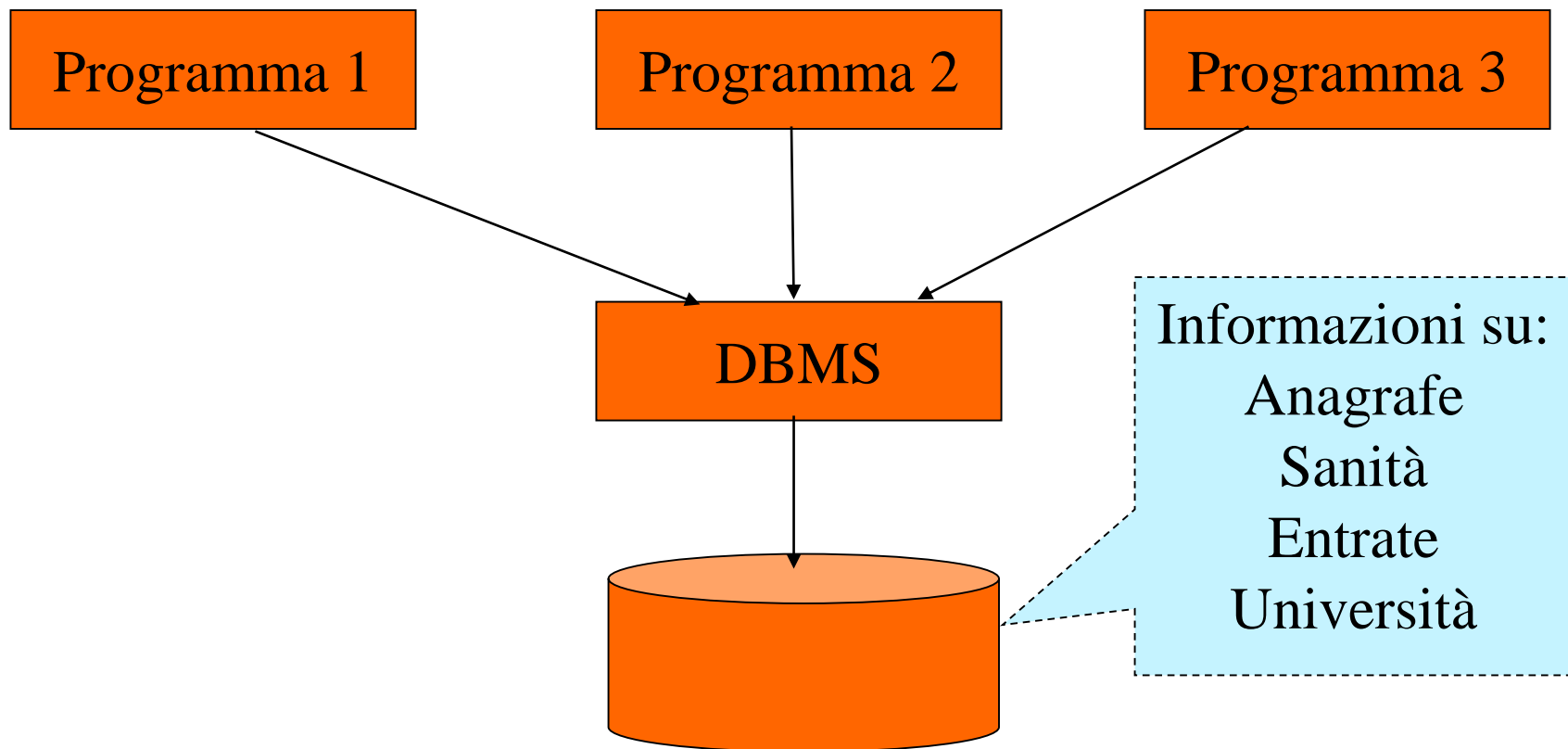
DBMS sul mercato (o freeware)

- DB2
- Oracle
- Microsoft SQL Server
- MySQL (freeware)
- PostgreSQL (freeware)
- Microsoft Access

Archivi separati o DB?



...forse è meglio un DBMS



Vantaggi di un DBMS

- Evita inconsistenza e ridondanza dei dati
 - Potenziale duplicazione dei dati → rischio inconsistenza
 - Es. indirizzo di un correntista su un file dei conti, e su un altro file con dei depositi
 - Nei DBMS I dati hanno una sola rappresentazione
- Garantisce la riservatezza dei dati
 - Protezione differenziata dei dati
 - Es. accesso di un cliente ad un conto non suo
- Garantisce l'integrità dei dati
 - Definizione di vincoli di consistenza (integrità)
 - Es. un assegno non può essere pagato se il suo incasso rende negativo l'ammontare del conto corrente
- Permette la concorrenza tra più programmi
 - Più programmi accedono contemporaneamente agli stessi dati → rischio inconsistenza
 - Es. più programmi sullo stesso conto corrente
 - Mutua esclusione

Linguaggi di un DBMS

- Data Definition Language (DDL)
 - Definizione dello schema del DB
 - Istruzioni e costrutti simili alla definizione dei tipi di dati strutturati nei linguaggi (es. istruzioni di definizione di grafi ed alberi)
 - I tipi definiti, che caratterizzano lo schema, vengono inseriti nel “Dizionario dei dati” (*Data dictionary*)
- Data Manipulation Language (DML)
 - Manipolare il DB
 - Interrogazioni o query → estrazione informazioni
 - Modificare il contenuto del DB
 - In questo corso: SQL

Basi di dati e transazioni: proprietà “acide”

➤ Transazioni

- Stato Iniziale DB → Transazione → Stato finale DB (diverso dal precedente)

➤ Proprietà “ACID”

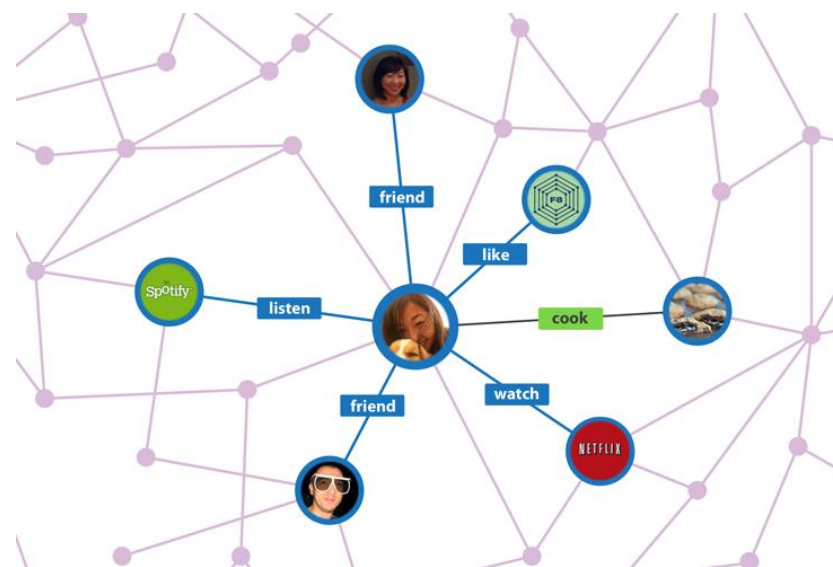
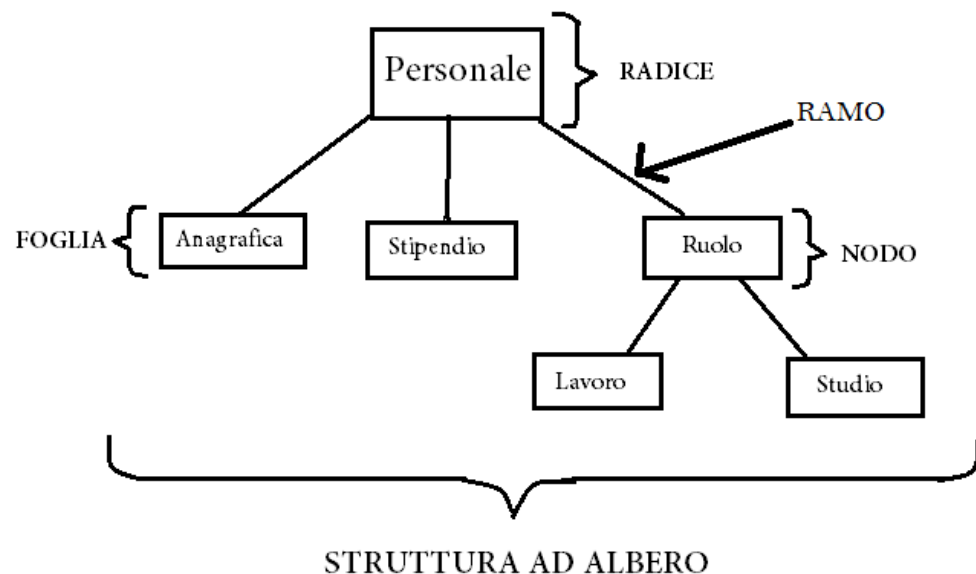
- **A**tomicità: termine con insuccesso (**rollback**) o successo (**commit**)
 - Nel caso di rollback, viene ripristinato lo stato iniziale cancellando eventuali modifiche parziali
- **C**onsistenza: nessuna violazione dell'integrità dei dati
 - In caso contrario, rollback parziale (si eliminano solo i passaggi che violano l'integrità) o totale, imposto dal DBMS
- **I**solamento: più transazioni lavorano l'una indipendentemente dall'altra, in modo che l'eseguirsi di una non determini errori nell'esecuzione dell'altra (dati modificati o erronei rispetto alla transazione, ma che prima erano corretti)
- **D**urabilità o persistenza: l'effetto di una transazione è permanente

Modelli dei dati

➤ Specifica collezione di costruttori di **tipo** per organizzare i dati in un DBMS

➤ Tipi di dati utilizzati:

- Alberi: modello gerarchico (1960)
- Grafi: modello reticolare (Codasyl, 1973, 1978)
- Insiemi e tabelle: modello relazionale (1970)
- Oggetti: modello a oggetti (1985)

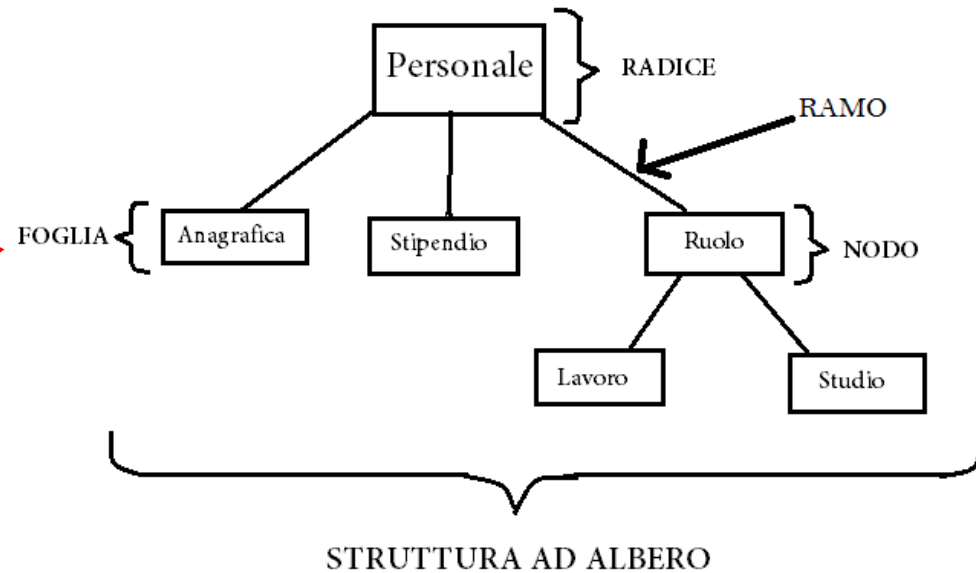


Struttura a grafo

Schemi e istanze

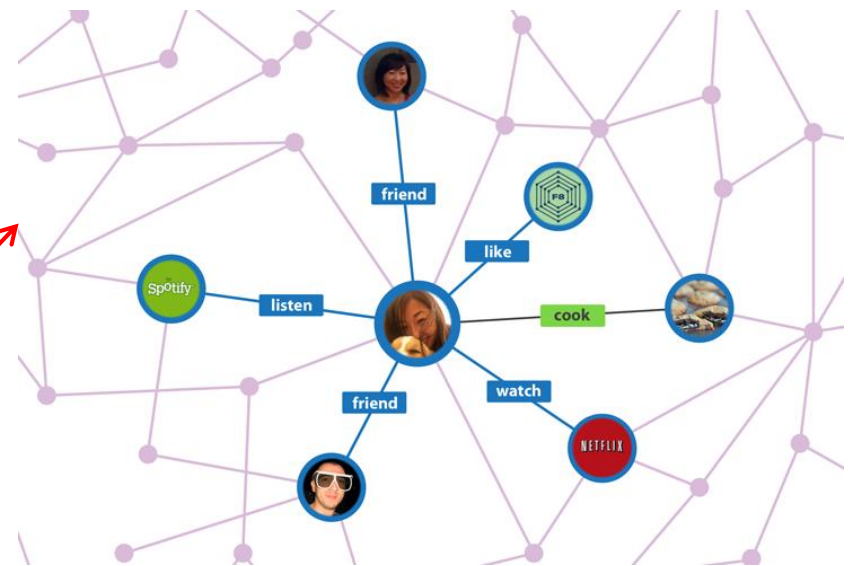
➤ Schema di un DB

- Descrizione di dati di uno specifico contesto applicativo conforme al modello dei dati presente nel DBMS
- Definito in fase di «Database design»



➤ Istanza o occorrenza di un DB

- “Valore” (insieme di valori) assunti ad un determinato istante



Struttura a grafo

Il modello relazionale

- Un DB relazionale è una collezione di “relazioni”
- Rappresentate attraverso tabelle
 - Numero fisso di colonne (**grado**), chiamate **attributi**
 - Numero fisso di righe (**cardinalità**), chiamate **tuple**
- Ogni attributo è definito su un dominio
 - Es. attributo “età” definito sul dominio intero senza segno
 - Per estensione possiamo pensare ai tipi semplici del C
- Schema di una relazione
 - Descrizione della struttura di una relazione
- Istanza di una relazione
 - Configurazione ad un certo istante della relazione stessa (il suo contenuto)

Esempio di istanza di DB relazionale

Corsi

Corso	Docente	Aula
Basi di dati	10006	B1
Fondamenti	10006	Z
Reti	9000	Z
Calcolatori	10005	V

Aule

Nome	Edificio	Piano
B1	DIEE-B	1
Z	Parcheeggi	0
V	Parcheeggi	0

Docenti

Matricola	Nome	Tel.	Ufficio
10000	Marcialis	5893	A
10005	Roli	5779	B
10006	Giacinto	5752	C
9000	Fumera	5754	C

Corrispondente schema logico dell'esempio

Relation Corsi

(Corso: **char**(30),
Docente: **char**(20),
Aula: **char**(4)).

Relation Aule

(Nome: **char**(4),
Edificio: **char**(10),
Piano: **char**(4)).

Relation Docenti

(Matricola: **integer**,
Nome: **char**(20),
Tel.: **integer**,
Ufficio: **char**(2)).

Informazioni (viste) estraibili dal DB: rappresentazioni esterne

- Elenco corsi professati nell'aula Z
- Elenco corsi professati da Marcialis
- Elenco delle aule che si trovano nel padiglione
Parcheggi e dove si tengono corsi professati da Roli
- ...

- Provate ad indicare grado e cardinalità di ciascuna delle
precedenti relazioni

Interpretare i seguenti schemi di relazione

Relation Conti-Correnti
(Numero-CC: **integer**,
Nome: **char**(20),
Indirizzo: **char**(20),
Saldo: **decimal**(14,2))

Relation Movimenti
(Numero-CC: **integer**,
Data-Mov: **date**,
Numero-Mov: **integer**,
Importo: **decimal**(14,2),
Causale: **char**(1))

Indicare grado e cardinalità delle relazioni di cui sopra.

Definizione: restrizioni e chiavi

- Si definisce **restrizione** di una tupla t sugli attributi A_1, \dots, A_n della relazione R , ed indicata con $t[A_1, \dots, A_n]$, la lista di valori assunti dalla tupla
- Si definisce **super-chiave**, un'insieme di attributi A_1, \dots, A_n tale che, estratte tutte le corrispondenti restrizioni da R , indicate con t_1, \dots, t_c , si ha $t_i \neq t_j$ per $i \neq j$
- Si definisce **chiave** una super-chiave minimale, tale che il suo numero di attributi caratterizzanti è minimo

Proprietà delle chiavi

- Data una chiave K su R
- Unicità
 - In una qualunque istanza di R , non possono esistere due tuple distinte di R la cui restrizione su K sia uguale
- Minimalità
 - Non è possibile sottrarre a K un attributo senza che la condizione di unicità cessi di valere
- Chiave primaria
 - Tra tutte le chiavi, quella che viene usata più spesso per accedere ai dati

Il linguaggio SQL

- SQL = Structured Query Language
- Standardizzato dall'ANSI e dall'ISO, nel 1992
- Cosa si può fare:
 - Creazione di relazioni (DDL)
 - Definizione di chiavi (DDL)
 - Anche chiavi primarie, con obbligo di indicazione del valore dell'attributo (NOT NULL)
 - Definizione di vincoli (DDL)
 - Vincoli di attributo, di tupla, di integrità referenziale
 - Aggiunta dati (DML)
 - **Interrogazioni (DML)**
 - **Selezione, proiezione, join, unione e differenza (algebra relazionale)**

Interrogazioni in SQL: costruito base del DML

```
SELECT [DISTINCT] nomi di attributi  
FROM nomi di relazioni  
WHERE condizioni di ricerca sui dati;
```

Questo costruito restituisce le **restrizioni** sulle tuple delle relazioni indicate, rispettando le condizioni di ricerca seguenti la parola-chiave "where"

La clausola "distinct" rimuove dalla selezione le tuple per cui i valori degli attributi risultano uguali.

Siano date le seguenti istanze...

Conti-Correnti

Numero-CC	Nome	Indirizzo	Saldo
1	Marcialis	Via Rossi	1,500
2	Roli	Via Bianchi	4,000
3	Giacinto	Via Neri	2,500
4	Fumera	Via Verdi	1,800

Movimenti

Numero-CC	Data-Mov	Numero-Mov	Importo	Causale
1	14-1-03	1	+200	V
1	14-1-03	2	-500	P
1	27-1-03	1	+2,700	S
4	27-1-03	1	+1,850	S
3	25-1-03	1	-650	A

Operazioni elementari sulle tabelle

➤ Proiezione

- Mostra tutte le istanze corrispondenti ad un sottoinsieme di attributi

➤ Selezione

- Proiezione limitata a determinati vincoli sugli attributi (non necessariamente quelli proiettati)

➤ Join

- Prodotto cartesiano di due tabelle
- Join+Selezione permette la connessione logica di più dati e l'estrazione di informazioni da dati «incrociati»

Risultato degli operatori

- Gli operatori non generano nuove relazioni ma «viste», ovvero tabelle derivate dalle relazioni definite
- Esercizio
 - Date due relazioni R_1 e R_2 di grado N_1 e N_2 e cardinalità M_1 e M_2 , dire grado e cardinalità delle viste generate da una proiezione e da una join

Esercizi

Scrivere l'output delle seguenti interrogazioni e indicare grado e cardinalità delle viste ottenute:

Proiezione:

```
SELECT NOME, INDIRIZZO  
FROM CONTI-CORRENTI;
```

Selezione:

```
SELECT SALDO  
FROM CONTI-CORRENTI  
WHERE NUMERO-CC=2;
```

Condizioni di selezione

- Vale l'utilizzo degli operatori booleani:
 - AND, OR
- Anche gli operatori di diversità ed uguaglianza
 - =, <>, >, <
- Nella selezione degli attributi, laddove numerici, si possono utilizzare:
 - +, -, *, /
- Le costanti numeriche e carattere si indicano
 - 'a', 1000

Prodotto cartesiano

```
SELECT *
```

```
FROM CONTI-CORRENTI, MOVIMENTI
```

- Genera una vista data dal prodotto cartesiano delle relazioni `Conti-Correnti` e `Movimenti` e la proietta con tutti gli attributi.
- Grado e cardinalità?

L'operazione di “Join”

- Estrarre nome e indirizzo dei correntisti che hanno un movimento in data 27-1-03. Indicare grado e cardinalità della vista ottenuta.

- Soluzione

```
SELECT NOME, INDIRIZZO
```

```
FROM CONTI-CORRENTI, MOVIMENTI
```

```
WHERE DATA-MOV=27-1-03
```

```
AND
```

```
CONTI-CORRENTI.NUMERO-CC = MOVIMENTI.NUMERO-CC
```

Esercizio

- Descrivere i singoli passi della precedente query

- Soluzione
 - Generazione della vista «prodotto cartesiano» delle tabelle date
 - Selezione sulla vista in base ad i vincoli fra gli attributi
 - Due sono lo stesso attributo ==> **chiave** di una delle relazioni
 - Visualizzazione della vista ottenuta

Riprendiamo le istanze...

Conti-Correnti

Numero-CC	Nome	Indirizzo	Saldo
1	Marcialis	Via Rossi	1,500
2	Roli	Via Bianchi	4,000
3	Giacinto	Via Neri	2,500
4	Fumera	Via Verdi	1,800

Movimenti

Numero-CC	Data-Mov	Numero-Mov	Importo	Causale
1	14-1-03	1	+200	V
1	14-1-03	2	-500	P
1	27-1-03	1	+2,700	S
4	27-1-03	1	+1,850	S
3	25-1-03	1	-650	A

Generazione del prodotto cartesiano

Numero-CC	Nome	Indirizzo	Saldo	Numero-CC	Data-Mov	Numero-Mov	Importo	Causale
1	Marcialis	Via Rossi	1,500	1	14-1-3	1	+200	V
1	Marcialis	Via Rossi	1,500	1	14-1-3	2	-500	P
1	Marcialis	Via Rossi	1,500	1	27-1-3	1	+2,700	S
1	Marcialis	Via Rossi	1,500	4	27-1-3	1	+1,850	S
1	Marcialis	Via Rossi	1,500	3	25-1-3	1	-650	A
2	Roli	Via Bianchi	4,000	1	14-1-3	1	+200	V
2	Roli	Via Bianchi	4,000	1	14-1-3	2	-500	P
...
4	Fumera	Via Verdi	1,800	4	27-1-3	1	+1,850	S
4	Fumera	Via Verdi	1,800	3	25-1-3	1	-650	A

Selezione delle tuple in base ai vincoli

Numero-CC	Nome	Indirizzo	Saldo	Numero-CC	Data-Mov	Numero-Mov	Importo	Causale
1	Marcialis	Via Rossi	1,500	1	14-1-3	1	+200	V
1	Marcialis	Via Rossi	1,500	1	14-1-3	2	-500	P
1	Marcialis	Via Rossi	1,500	1	27-1-3	1	+2,700	S
1	Marcialis	Via Rossi	1,500	4	27-1-3	1	+1,850	S
1	Marcialis	Via Rossi	1,500	3	25-1-3	1	-650	A
2	Roli	Via Bianchi	4,000	1	14-1-3	1	+200	V
2	Roli	Via Bianchi	4,000	1	14-1-3	2	-500	P
...
4	Fumera	Via Verdi	1,800	4	27-1-3	1	+1,850	S
4	Fumera	Via Verdi	1,800	3	25-1-3	1	-650	A

Visualizzazione della vista

Nome	Indirizzo
Marcialis	Via Rossi
Fumera	Via Verdi

Esercizi (1)

➤ Scrivere l'interrogazione "estrarre nome del correntista, importo e causale dei movimenti per tutti i correntisti con un saldo maggiore di 2000 euro e con un movimento che sia versamento (V) o un accredito di stipendio (S)"

➤ Soluzione

```
SELECT NOME, IMPORTO, CAUSALE
FROM MOVIMENTI, CONTI-CORRENTI
WHERE SALDO>2000
AND (CAUSALE='V' OR CAUSALE='S')
AND          MOVIMENTI.NUMERO-CC=CONTI-
CORRENTI.NUMERO-CC
```

Esercizi (2)

- Esprimere il risultato della seguente query in SQL:

```
SELECT    NUMERO-CC,    DATA-MOV,    NUMERO-MOV,  
           IMPORTO*0.1  
FROM    MOVIMENTI  
WHERE    CAUSALE='S'
```

- Soluzione:
- Significa: estrarre numero del conto e data, numero di movimento e 10% dell'importo dei movimenti dovuti ad un accredito di stipendio

Esercizi (3)

- Estrarre nome del correntista, importo del saldo e della causale per tutti i movimenti corrispondenti al pagamento di assegni il cui importo supera il saldo di 50 euro

- Soluzione

```
SELECT NOME, SALDO, IMPORTO, CAUSALE
FROM MOVIMENTI, CONTI-CORRENTI
WHERE IMPORTO>SALDO+50
AND CAUSALE='A'
AND CONTI-CORRENTI.NUMERO-CC=MOVIMENTI.NUMERO-CC
```

Esercizi

➤ Facciamo riferimento al seguente schema logico

Relation Studenti

(Matricola: char(5) PRIMARY KEY NOT NULL,
Nome: char(20) NOT NULL,
Data-N: date,
Anno-Corso: integer,
Corso-Laurea: char (1));

Relation Corsi

(Cod-Corso: char(6) PRIMARY KEY NOT NULL,
Titolo: char(50),
Docente: char(20));

Relation Esami

(Cod-Corso: char(5) NOT NULL,
Matricola: char(5) NOT NULL,
Data-E: date NOT NULL,
Voto: integer NOT NULL,
PRIMARY KEY (Cod-Corso, Matricola));

➤ Domande preliminari:

- Creare le tabelle relative usando gli opportuni costrutti SQL
- Popolare ciascuna tabella con le seguenti tuple:

Corsi:

(1, Fondamenti di Informatica, Paperino)
(2, Analisi 1, Topolino)

Studenti:

(10010, Pippo, 13-1-87, 1, Ingegneria Meccanica)

Esami:

(1, 10010, 20-9-15, 21)
(2, 10010, 16-7-15, 22)

Scrivere le seguenti interrogazioni

- Estrarre codici di corso e docenti di Fondamenti di Informatica e Analisi 1.
- Estrarre nome e matricola degli studenti che hanno sostenuto Analisi 1 il 10-1-04 prendendo un voto superiore a 28.
- Estrarre matricola, nome e voto degli studenti che hanno sostenuto l'esame di un corso insegnato da Paperino o Topolino.
- Estrarre i docenti dei corsi i cui esami sono stati superati da Pippo con voto superiore a 25.

Soluzioni

- Estrarre codici di corso e docenti di Fondamenti di Informatica e Analisi 1.

```
SELECT COD-CORSO, DOCENTE
FROM CORSI
WHERE
    TITOLO= 'FONDAMENTI DI INFORMATICA'
    OR
    TITOLO= 'ANALISI 1'
```


Soluzioni

- Estrarre nome e matricola degli studenti che hanno sostenuto Analisi 1 il 10-1-04 prendendo un voto superiore a 28.

```
SELECT NOME, MATRICOLA
FROM STUDENTI, ESAMI
WHERE STUDENTI.MATRICOLA=ESAMI.MATRICOLA
AND COD-CORSO=1
AND VOTO>28
```

Soluzione più articolata

```
SELECT NOME, STUDENTI.MATRICOLA
FROM STUDENTI, ESAMI, CORSI
WHERE STUDENTI.MATRICOLA=ESAMI.MATRICOLA
AND
        CORSI.COD-CORSO=ESAMI.COD-CORSO
AND
        TITOLO='ANALISI 1'
AND
        VOTO>28
```

Soluzioni

- Estrarre matricola, nome e voto degli studenti che hanno sostenuto l'esame di un corso insegnato da Paperino o Topolino.

```
SELECT DISTINCT STUDENTI.MATRICOLA, NOME, VOTO  
FROM STUDENTI, CORSI, ESAMI  
WHERE STUDENTI.MATRICOLA=ESAMI.MATRICOLA  
AND CORSI.COD-CORSO=ESAMI.COD-CORSO  
AND (DOCENTE='PAPERINO' OR DOCENTE='TOPOLINO')
```

Soluzioni

- Estrarre i docenti dei corsi i cui esami sono stati superati da Pippo con voto superiore a 25.

```
SELECT DOCENTE  
FROM CORSI, STUDENTI, ESAMI  
WHERE STUDENTI.MATRICOLA=ESAMI.MATRICOLA  
AND CORSO.COD-CORSO=ESAMI.COD-CORSO  
AND NOME='PIPPO'  
AND VOTO>25
```

Per saperne di più...

➤ Archivi e basi di dati

- Ceri, Mandriola, Sbattella, “Informatica – arte e mestiere”, McGraw-Hill 1999, ISBN 88 386 0804-0, Capitolo 14

➤ Approfondimento:

- Atzeni, Ceri, Paraboschi, Torlone, “Basi di dati – concetti, linguaggi, architetture”, McGraw-Hill 1996, ISBN 88 386 0724-9