Consorzio per la formazione e la ricerca in Ingegneria dell'Informazione

Structured Query Language *parte 1*

Come interrogare una base di dati relazionale…ed avere la risposta esatta

Docente:

Gennaro Pepe

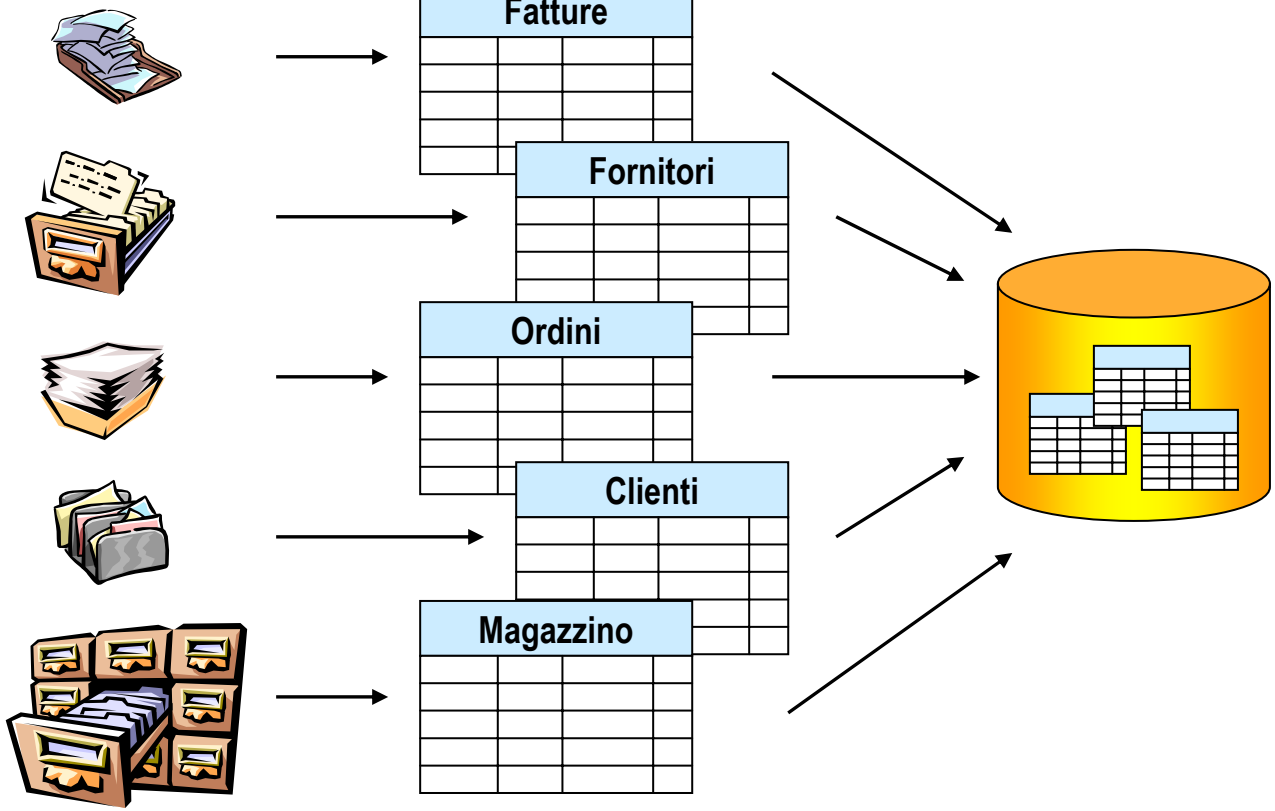
CEFRIEL

Gennaro.pepe@cefriel.it

http://www.cefriel.it/Item

Riassumendo...le basi dati(1)

- Fatture
- Schedario dei numeri di telefono dei fornitori
- Ordini dei clienti
- Indirizzi dei clienti
- Archivio prodotti in magazzino



informazioni **tabelle** **base di dati**

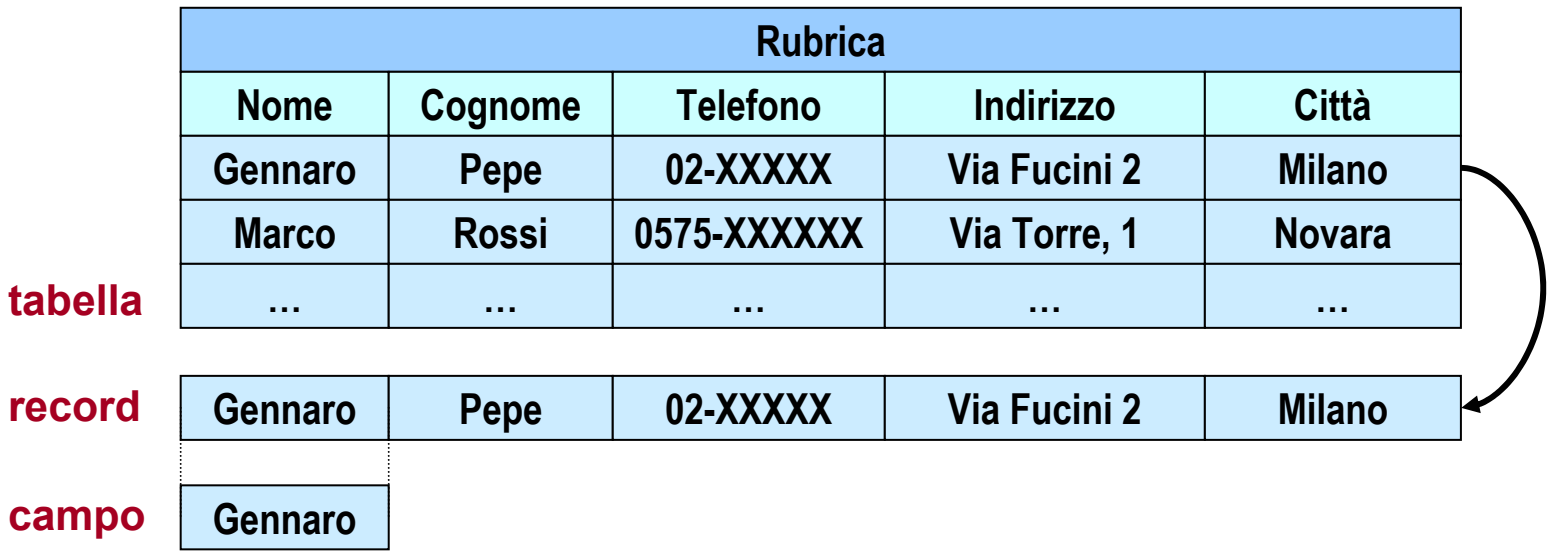
Riassumendo...le basi di dati(2)

Tabelle, record e campi

I dati sono organizzati in **tabelle** chiamate relazioni.

Ogni relazione è definita dalle colonne di cui è composta e contiene tutte le tuple (o righe) dello stesso tipo contenute nella base di dati.

Le colonne della tabella rappresentano i **campi** o proprietà della tabella; ogni riga corrisponde a un **record** e raggruppa i valori che queste proprietà assumono in una tupla.



Cos'è SQL?

- SQL (Structured Query Language) è il linguaggio di riferimento per le basi di dati relazionali
  - Data Definition Language (permette di definire lo schema di una b.d.r.)
  - Data Manipulation Language (permette di modificare le istanze delle b.d.r.)
  - Data Control Language (permette di definire controllo e autorizzazioni in una b.d.r)

Comandi SQL (Structured Query Language)		
DDL	DML	DCL
CREATE	INSERT	GRANT
DROP	UPDATE	REVOKE
	DELETE	
	SELECT	

Come viene usato SQL?

- SQL esprime le interrogazioni dichiarando l'obiettivo dell'interrogazione stessa e non il modo per raggiungerlo
- Tale interrogazione viene passata ad un interprete che la analizza e formula una interrogazione equivalente utilizzando il linguaggio procedurale specifico del DBMS
- Rappresenta uno strumento per l'accesso ai dati alternativo a quelli che prevedono l'uso di linguaggi di sviluppo ad hoc e/o interfacce amichevoli personalizzate per il particolare DB

- NOTA. Anche se sono diffuse interfacce di tipo grafico, la conoscenza di SQL rimane molto importante per lavorare con i DB (soprattutto per chi deve sviluppare SW)

Le tappe della storia

1974 : prima proposta Structured English Query Language - SEQUEL (IBM Research)

1981 : prima implementazione in SQL/DS (IBM)

dal 1983 ca.: “standard di fatto”

1986 : prima versione standard dell’ANSI

1989 : estensione dello standard → SQL-89

1992 : **SQL-92 o SQL2**

1999 : SQL3, ancora non implementato da nessuno, orientato agli aspetti più avanzati, quali le applicazioni web, XML, il paradigma ad oggetti

Selezione semplice

- Per visualizzare o estarre i dati da una tabella occorre utilizzare le query di selezione, basate sull'uso del comando **SELECT**
- Come dice il nome, una tale interrogazione permette di 'selezionare' dei particolari dati di una tabella e visualizzarli
- La selezione viene fatta sulla base dei nomi delle colonne della tabella
- La parola chiave **FROM** permette di scegliere la tabella del database da cui si vogliono estarre i dati

```
select ListaCampi
from Tabella
```

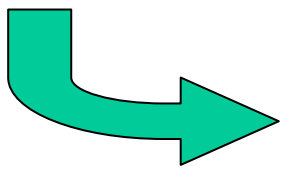
Selezione semplice – es.

Nome	Età	Reddito
Angelo	27	48
Marco	26	45
Piero	19	50

persone

Individuare tutti i *nomi* e i rispettivi *redditi*

```
select Nome, Reddito
from Persone;
```



Nome	Reddito
Angelo	48
Marco	45
Piero	50

Ordinare i dati

- Per avere in output un insieme di righe ordinate secondo un certo campo, si può utilizzare il comando **Order by**

```
select ListaCampi
from Tabella
order by campox
```

- **Esempio**

persone

Nome	Età	Reddito

Individuare nomi e redditi ordinati per Reddito

```
select Nome,Reddito
from persone
order by Reddito;
```

Filtrare i dati

- La clausola **WHERE** permette di specificare una condizione che deve essere soddisfatta dai dati estratti
- Su ciascuna riga vengono valutati i valori dei campi coinvolti nella condizione: solo se essi verificano la condizione i campi di tale riga specificati dalla SELECT vengono selezionati

```
select ListaCampi
from Tabella
where Condizione
```

La condizione è un predicato (o più) che usa gli operatori =, <>, <, >, <=, >=

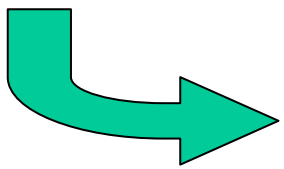
Filtrare i dati - es.

Nome	Età	Reddito
Angelo	27	48
Marco	26	45
Piero	19	50

persone

Individuare nomi e redditi delle persone che guadagnano più di 45

```
select Nome, Reddito
from Persone
where Reddito>45;
```



Nome	Reddito
Angelo	48
Piero	50

Filtrare i dati - wildcard

- Per specificare i criteri di selezione su dei campi di tipo testo è possibile utilizzare la funzione **LIKE**
- L'uso dei segni **%** o ***** permette di selezionare più stringhe che si corrispondono solo in parte
- I segni **_** o **?** spesso permette di sostituire invece un singolo carattere

```
select ListaCampi
from Tabella
where campox like "text*"
```

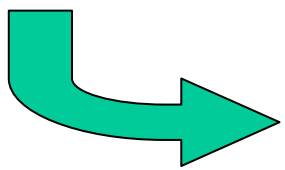
Filtrare i dati – wildcard – es.

Nome	Età	Reddito
Angelo	27	48
Marco	26	45
Martina	23	52
Piero	19	50

persone

Individuare l'età di tutte le persone con un nome che inizia per "Mar".

```
select Età
from persone
where nome like "Mar*";
```



Età
26
23

Filtraggio avanzato – operatori logici

- Attraverso gli operatori **AND** ed **OR** è possibile combinare più criteri di selezione
- L'operatore **NOT** è utilizzabile per creare delle condizioni che sono il negato di altre

Filtraggio avanzato – es.(1)

Nome	Età	Reddito
Angelo	27	48
Marco	26	45
Martina	23	52
Piero	19	50

persone

Individuare nome e età di tutte le persone con un nome che inizia per "Mar" e un reddito maggiore di 50.

Individuare nome e età di tutte le persone con un nome che non inizia per "Mar".

```
select nome, età
from persone
where nome not like 'Mar*';
```

Nome	Età
Angelo	27
Piero	19

```
select nome, età
from persone
where nome like 'Mar*' and
Reddito>50;
```

Nome	Età
Martina	23

Filtraggio avanzato..altri operatori…

- L'operatore **IN** permette di specificare la condizione di appartenenza ad un certo insieme di valori (eventualmente risultato dell'esecuzione di un'altra query)
- Per verificare se un attributo è **null** si usa l'operatore **IS**
  - Es. *Attributo is null* oppure *Attributo is not null*

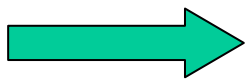
Filtraggio avanzato – es.(2)

Nome	Età	Reddito
Angelo	27	48
Marco	26	45
Martina	23	52
Piero	19	50

persone

Individuare nome e età di tutte le persone con un nome che non è Angelo, Giovanni e Marco.

```
select nome, età
from persone
where nome not in
  ('Angelo',
'Giovanni',
'Marco');
```



Nome	Età
Martina	23
Piero	19

Duplicati

- In SQL si possono avere più righe uguali in una stessa tabella
- L'eliminazione dei duplicati non viene fatta automaticamente perché è una operazione costosa e spesso non necessaria
- Se non si vogliono i duplicati è necessario esplicitarlo tramite la parola chiave distinct

persone

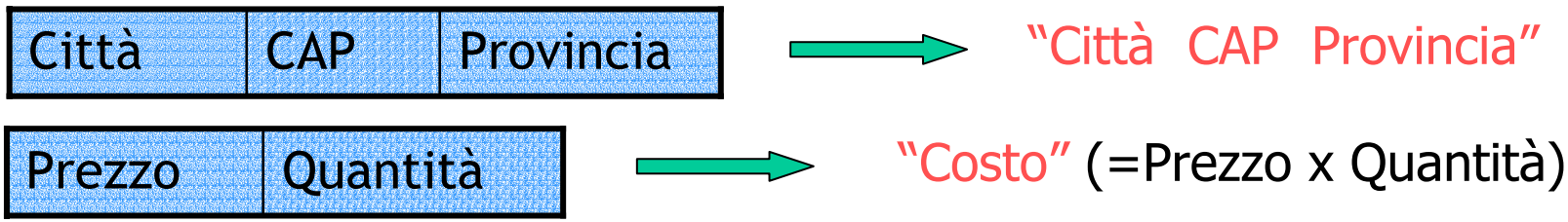
Nome	Età	Reddito

Individuare i nomi distinti della tabella *persone*

```
select distinct Nome
from persone;
```

I campi calcolati

- Spesso i dati presenti in una base di dati non sono nella forma di cui hanno bisogno le applicazioni che li utilizzeranno:



- Utilizzando operatori quali **||**, **+** , **&** è possibile concatenare più stringhe
- Utilizzando gli operatori aritmetici è possibile associare a dei campi del risultato delle espressioni
- Utilizzando il comando **AS** è possibile specificare un alias per un campo o un'espressione

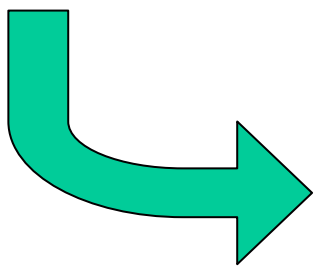
I campi calcolati – es.

Nome	Compenso_mensile	Mesi_di_lavoro
Angelo	3	3
Marco	2,5	5
Martina	4	6

persone

Selezionare il nome e il compenso totale di ogni persona

```
select nome, compenso_mensile*mesi_di_lavoro as compenso
from persone;
```



Nome	Compenso
Angelo	9
Marco	12,5
Martina	24

Uso di funzioni

- Al pari di altri linguaggi di programmazione, SQL supporta l'uso di funzioni per la manipolazione dei dati
- Tali funzioni spesso sono molto specifiche al DBMS
- Esse si possono dividere in quattro grosse categorie:
  - manipolazione di stringhe di testo
  - operazioni matematiche
  - manipolazione di valori di date ed orari
  - estrazione di informazioni di sistema

Uso di funzioni – es.

- Funzioni di manipolazioni di date:

MySQL	Microsoft SQL Server	ORACLE
Year(data)	**Datepart**(yyyy, data)	**to_char**(data,'YY')
Month(data)	mm	MM
Dayofmonth(data)	dd	DD

- Funzioni matematiche:

ABS(x)	valore assoluto
SQRT(x)	radice quadrata
EXP(x)	esponenziale
LOG x)	
...	

Esercizio

- Film (Codice, Titolo, Data_uscita, CasaProduttrice, Regista, Sceneggiatore, Costo)

1. Trovare i registi ed I titoli precisi dei film della serie Rocky…
2. Trovare i film che hanno come regista lo sceneggiatore
3. Trovare i film diretti da Spielberg ed usciti tra il 1990 ed il 2000

Soluzione

1.
```
select Titolo, Regista
from Film
where Film.Titolo LIKE “Rocky*”
```

2.
```
select Titolo
from Film
where Film.Regista=Film.Sceneggiatore
```

3.
```
select Titolo, Anno
from Film
where (Film.Regista LIKE “Spielberg”) AND
( Year(Film.Data_uscita)>1990 ) AND
( Year(Film.Data_uscita)<2000 )
```

Bibliografia

- Paolo Atzeni, Stefano Ceri,
  Stefano Paraboschi, Riccardo Torlone
  Basi di Dati - Seconda edizione
  settembre 1999 - McGraw Hill
  - **Capitolo 4.2**
- Teach Yourself SQL in 10 minutes
  second edition
  Ben Forta
  2001, SAMS