

# Corso di Fondamenti di Informatica



## Lezione 12

**Nicola Capuano**

Dipartimento di Scienze Aziendali, Management  
& Innovation Systems

[ncapuano@unisa.it](mailto:ncapuano@unisa.it)

## Prossime Lezioni

Date	Aule	Orari
Lunedì 4 Dicembre	Laboratorio	14.30-16.30
Mercoledì 6 Dicembre	Aula 6	14.30-16.30
Giovedì 7 Dicembre	Laboratorio	8.30-10.30
Giovedì 7 Dicembre	Laboratorio	12.30-14.30

# Programma del Corso

## Modulo 2 – Laboratorio di tecnologie dell'informazione e della comunicazione (aggiornamento)

- Algoritmi e Flow Chart
- Sistemi informativi e DBMS

## Interrogare un Database

Una **tabella** può contenere **centinaia di campi** (colonne) e **milioni di record** (righe)

- In genere un utente è interessato solo ad una **piccola parte** di tutta questa informazione
- **Esempi:** i CD di un certo artista, i voti di un certo studente, ecc.

Una **Query** (interrogazione) è una frase scritta in un linguaggio speciale che **permette di estrarre** da una o più tabelle **solo i dati che interessano**

- Una Query **crea una tabella più piccola** che contiene solo i dati che interessano



# Interrogare un Database

Le principali modalità di interrogazione di un DB sono le seguenti:

- **QBE Query by Example** (interfaccia grafica)
  - Modalità **semplice e diretta**
  - Adatta per **interrogazioni semplici**
  - Modalità non standard, spesso legata al DBMS
- **SQL Structured Query Language** (linguaggio)
  - Modalità **più complessa** della precedente
  - Valida per **tutti i tipi di interrogazione**
  - Linguaggio universale, adottato da tutti i DBMS



## Sintassi SQL

**Sintassi di base** di una query SQL:

```
SELECT [nome campi separati da una virgola, * indica tutti i campi]  
FROM [nome tabella]  
WHERE [condizioni di ricerca]
```

Perché una query riesca ad **estrarre solo i dati che interessano** occorre specificare delle **condizioni di ricerca** che sono **soddisfatte dai record che interessano**

Le **condizioni di ricerca** possono essere espresse tramite:

- **operatori di confronto**: =, <, >, <=, >=, <> (diverso)
- **operatori logici**: AND, OR, NOT

# Esempi di Interrogazione

## STUDENTI

Matricola	Cognome	Nome	Data di Nascita
486545	Rossi	Maria	25/11/1971
205745	Neri	Anna	23/04/1972
200768	Verdi	Fabio	12/02/1972
487614	Rossi	Luca	10/10/1971
207653	Bruni	Mario	01/12/1971

Ottenere tutte le informazioni sullo **studente Fabio Verdi**

```
SELECT *  
FROM STUDENTI  
WHERE Nome = "Fabio" AND Cognome = "Verdi"
```

Risultato

Matricola	Cognome	Nome	Data di Nascita
200768	Verdi	Fabio	12/02/1972

# Esempi di Interrogazione

## STUDENTI

Matricola	Cognome	Nome	Data di Nascita
486545	Rossi	Maria	25/11/1971
205745	Neri	Anna	23/04/1972
200768	Verdi	Fabio	12/02/1972
487614	Rossi	Luca	10/10/1971
207653	Bruni	Mario	01/12/1971

Ottenere **nome, cognome e matricola** di tutti gli **studenti** con numero di **matricola** che inizia per **48**

```
SELECT Cognome, Nome, Matricola  
FROM STUDENTI  
WHERE Matricola >= 480000 AND Matricola <= 489999
```

Risultato:

Cognome	Nome	Matricola
Rossi	Maria	486545
Rossi	Luca	487614

# Ordinamento

La clausola **ORDER BY** si usa per **ordinare i risultati di una query**

## Sintassi:

```
SELECT [nome campi separati da una virgola, * indica tutti i campi]
FROM [nome tabella]
WHERE [condizioni di ricerca]
ORDER BY [nome campi separati da una virgola] [DESC]
```

- Quando ORDER BY indica **più campi** allora si ordina in base al primo, poi (in caso di valori identici) al secondo, ecc.
- La parola chiave **opzionale DESC** si usa per ordinare in senso inverso

# Ordinamento

## STUDENTI

Matricola	Cognome	Nome	Data di Nascita
486545	Rossi	Maria	25/11/1971
205745	Neri	Anna	23/04/1972
200768	Verdi	Fabio	12/02/1972
487614	Rossi	Luca	10/10/1971
207653	Bruni	Mario	01/12/1971

Ottenere **cognome, nome e matricola** di tutti gli studenti **ordinati** prima per **cognome** e poi per **nome**

```
SELECT Cognome, Nome, Matricola
FROM STUDENTI
ORDER BY Cognome, Nome
```

Risultato

Cognome	Nome	Matricola
Bruni	Mario	207653
Neri	Anna	205745
Rossi	Luca	487614
Rossi	Maria	486545
Verdi	Fabio	200768

# Ordinamento

## STUDENTI

Matricola	Cognome	Nome	Data di Nascita
486545	Rossi	Maria	25/11/1971
205745	Neri	Anna	23/04/1972
200768	Verdi	Fabio	12/02/1972
487614	Rossi	Luca	10/10/1971
207653	Bruni	Mario	01/12/1971

Ottenere tutte le informazioni sugli studenti con **cognome Rossi** ordinate dal **meno giovane al più giovane**

```
SELECT * FROM STUDENTI  
WHERE Cognome = "Rossi"  
ORDER BY [Data di Nascita] DESC
```

Risultato

Matricola	Cognome	Nome	Data di Nascita
486545	Rossi	Maria	25/11/1971
487614	Rossi	Luca	10/10/1971

# Estrazione per Similitudine

L'operatore **LIKE** consente di estrarre da una tabella dati testuali che **rispettano un certo schema**

- È un **operatore di confronto** utilizzabile nella clausola **WHERE** (come =, <, <=, >, >=, <>)
- Utilizza il formato: **[Nome Campo] LIKE "schema"**

All'interno dello **schema**:

- Il **simbolo ?** indica una **lettera arbitraria**
- Il **simbolo \*** indica un **numero arbitrario di lettere** (anche 0)
- I **simboli []** racchiudono un **insieme di lettere possibili**

# Estrazione per Similitudine

All'interno dello **schema**:

- Il simbolo **?** indica una **lettera arbitraria**
- Il simbolo **\*** indica un **numero arbitrario di lettere (anche 0)**
- I simboli **[]** racchiudono un **insieme di lettere possibili**

## Esempi:

```
SELECT * FROM STUDENTI WHERE Cognome LIKE "R*"
```

- Estrae gli studenti che hanno il cognome che **inizia per R**

```
SELECT * FROM STUDENTI WHERE Cognome LIKE "[RN]*"
```

- Estrae gli studenti che hanno il cognome che **inizia R o per N**

# Estrazione per Similitudine

All'interno dello **schema**:

- Il simbolo **?** indica una **lettera arbitraria**
- Il simbolo **\*** indica un **numero arbitrario di lettere (anche 0)**
- I simboli **[]** racchiudono un **insieme di lettere possibili**

## Esempi:

```
SELECT * FROM STUDENTI WHERE Cognome LIKE "*n*"
```

- Estrae gli studenti che hanno il cognome che **include una n**

```
SELECT * FROM STUDENTI WHERE Cognome LIKE "**[Nn]*"
```

- Estrae gli studenti che hanno il Cognome che **include una n o una N**

# Estrazione per Similitudine

All'interno dello **schema**:

- Il simbolo **?** indica una **lettera arbitraria**
- Il simbolo **\*** indica un **numero arbitrario di lettere (anche 0)**
- I simboli **[]** racchiudono un **insieme di lettere possibili**

## Esempi:

```
SELECT * FROM STUDENTI WHERE Cognome LIKE "*ssi"
```

- Estrae gli studenti che hanno il cognome che **finisce per ssi**

```
SELECT * FROM STUDENTI WHERE Cognome LIKE "R*si"
```

- Estrae gli studenti che hanno il cognome che **inizia per R e finisce per si**

# Estrazione per Similitudine

All'interno dello **schema**:

- Il simbolo **?** indica una **lettera arbitraria**
- Il simbolo **\*** indica un **numero arbitrario di lettere (anche 0)**
- I simboli **[]** racchiudono un **insieme di lettere possibili**

## Esempi:

```
SELECT * FROM STUDENTI WHERE Cognome LIKE "R??si"
```

- Estrae gli studenti il cui cognome **inizia per R seguito da due lettere qualsiasi e finisce per si**

```
SELECT * FROM STUDENTI WHERE Cognome LIKE "?[aeiou]*"
```

- Estrae gli studenti il cui cognome ha per **seconda lettera una vocale**

# Estrazione per Similitudine

## STUDENTI

Matricola	Cognome	Nome	Data di Nascita
486545	Rossi	Maria	25/11/1971
205745	Neri	Anna	23/04/1972
200768	Verdi	Fabio	12/02/1972
487614	Rosi	Marco	10/10/1971
207653	Bruni	Mario	01/12/1971

Ottenere tutte le informazioni sugli studenti il cui **nome inizia M** e il cui **cognome inizia per R** ordinati prima per **Cognome** e poi per **Nome**

```
SELECT * FROM STUDENTI
WHERE Nome LIKE "M*" AND COGNOME LIKE "R*"
ORDER BY Cognome, Nome
```

Risultato

Matricola	Cognome	Nome	Data di Nascita
487614	Rosi	Marco	10/10/1971
486545	Rossi	Maria	25/11/1971

# Estrazione per Similitudine

## STUDENTI

Matricola	Cognome	Nome	Data di Nascita
486545	Rossi	Maria	25/11/1971
205745	Neri	Anna	23/04/1972
200768	Verdi	Fabio	12/02/1972
487614	Rosi	Adele	10/10/1971
207653	Bruni	Mario	01/12/1971

Ottenere **Nome** e **Cognome** di tutti gli studenti il cui nome **finisce per a oppure e**

```
SELECT Nome, Cognome
FROM STUDENTI
WHERE Nome LIKE "**[ae]"
```

Risultato

Nome	Cognome
Maria	Rossi
Anna	Neri
Adele	Rosi

## Esercizi

Data la tabella **Film** che include i seguenti record:

Titolo	Anno	Cognome Regista	Nome Regista	Tipo Film	Valutazione
Addams Family	1998	Paine	Dave	Commedia	3
Al di là dei sogni	1998	Ward	Vincent	Drammatico	5
Così è la vita	1998	Venier	Massimo	Commedia	7
Il mostro	1995	Benigni	Roberto	Commedia	6
La banda degli onesti	1950	Mastrocinque	Camillo	Comico	8

Scrivere la query SQL per **selezionare i titoli dei film usciti prima del 1996 ordinati per anno di uscita**

```
SELECT Titolo FROM Film  
WHERE Anno < 1996 ORDER BY Anno
```

## Esercizi

Data la tabella **Film** che include i seguenti record:

Titolo	Anno	Cognome Regista	Nome Regista	Tipo Film	Valutazione
Addams Family	1998	Paine	Dave	Commedia	3
Al di là dei sogni	1998	Ward	Vincent	Drammatico	5
Così è la vita	1998	Venier	Massimo	Commedia	7
Il mostro	1995	Benigni	Roberto	Commedia	6
La banda degli onesti	1950	Mastrocinque	Camillo	Comico	8

Scrivere la query SQL per **selezionare titolo, anno e valutazione dei film di genere commedia o comico**

```
SELECT Titolo, Anno, Valutazione  
FROM Film WHERE [Tipo Film] = "Commedia"  
OR [Tipo Film] = "Comico"
```

## Esercizi

Data la tabella **Film** che include i seguenti record:

Titolo	Anno	Cognome Regista	Nome Regista	Tipo Film	Valutazione
Addams Family	1998	Paine	Dave	Commedia	3
Al di là dei sogni	1998	Ward	Vincent	Drammatico	5
Così è la vita	1998	Venier	Massimo	Commedia	7
Il mostro	1995	Benigni	Roberto	Commedia	6
La banda degli onesti	1950	Mastrocinque	Camillo	Comico	8

Scrivere la query SQL per **selezionare titolo e anno dei film che hanno un titolo che contiene la parola “banda”**

```
SELECT Titolo, Anno FROM Film  
WHERE Titolo LIKE “*banda*”
```

## Esercizi

Data la tabella **Film** che include i seguenti record:

Titolo	Anno	Cognome Regista	Nome Regista	Tipo Film	Valutazione
Addams Family	1998	Paine	Dave	Commedia	3
Al di là dei sogni	1998	Ward	Vincent	Drammatico	5
Così è la vita	1998	Venier	Massimo	Commedia	7
Il mostro	1995	Benigni	Roberto	Commedia	6
La banda degli onesti	1950	Mastrocinque	Camillo	Comico	8

Scrivere la query SQL per **selezionare tutte le informazioni dei film di Roberto Benigni, ordinati per anno di uscita**

```
SELECT * FROM Film  
WHERE [Nome Regista] = “Roberto”  
AND [Cognome Regista] = “Benigni”  
ORDER BY Anno
```

## Esercizi

Data la tabella **Film** che include i seguenti record:

Titolo	Anno	Cognome Regista	Nome Regista	Tipo Film	Valutazione
Addams Family	1998	Paine	Dave	Commedia	3
Al di là dei sogni	1998	Ward	Vincent	Drammatico	5
Così è la vita	1998	Venier	Massimo	Commedia	7
Il mostro	1995	Benigni	Roberto	Commedia	6
La banda degli onesti	1950	Mastrocinque	Camillo	Comico	8

Scrivere la query SQL per **selezionare tutte le informazioni sui film girati da registi con cognome che inizia per P o W ordinati in ordine decrescente di valutazione**

```
SELECT * FROM Film  
WHERE [Cognome Regista] LIKE "[PW]*"  
ORDER BY Valutazione DESC
```

## Esercizi

Data la tabella **Film** che include i seguenti record:

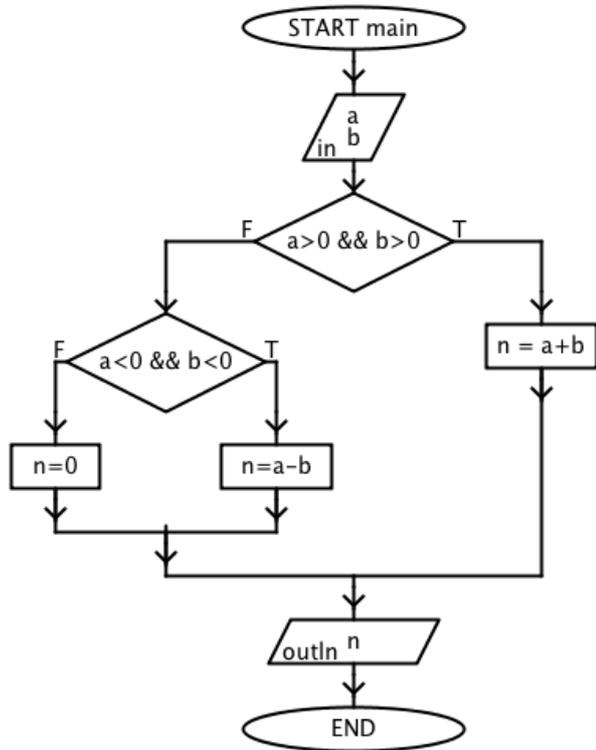
Titolo	Anno	Cognome Regista	Nome Regista	Tipo Film	Valutazione
Addams Family	1998	Paine	Dave	Commedia	3
Al di là dei sogni	1998	Ward	Vincent	Drammatico	5
Così è la vita	1998	Venier	Massimo	Commedia	7
Il mostro	1995	Benigni	Roberto	Commedia	6
La banda degli onesti	1950	Mastrocinque	Camillo	Comico	8

Scrivere la query SQL per **selezionare tutte le informazioni sui film usciti dopo il 1990 che includono nel titolo la parola "vita" ordinati per cognome del regista**

```
SELECT * FROM Film  
WHERE Anno > 1990 AND Titolo LIKE "*vita*"  
ORDER BY [Cognome Regista]
```

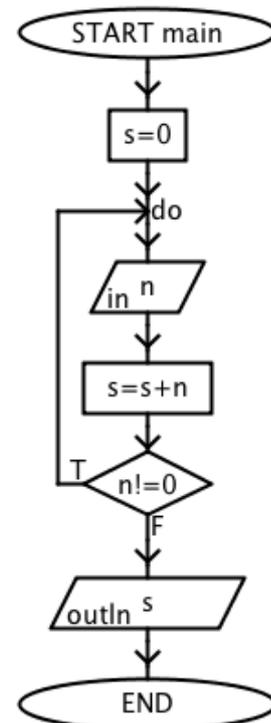
## Flow Chart (Esercizi)

Descrivere, mediante diagramma di flusso, un algoritmo che chiede in input 2 valori **a** e **b**. Se entrambi sono maggiori di **0** visualizza la **somma a+b**, se entrambi sono minori di **0** visualizza la **differenza a-b**, altrimenti visualizza **0**.



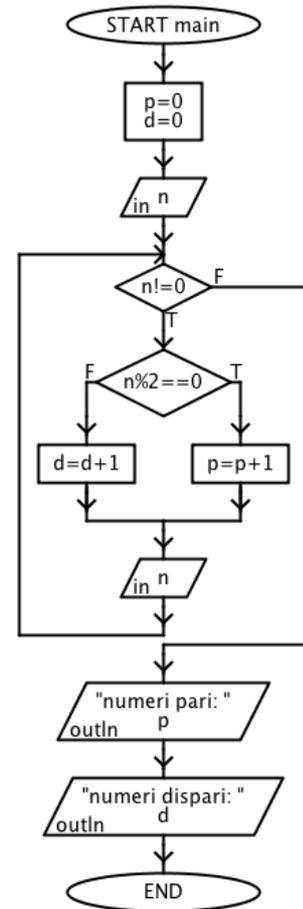
## Flow Chart (Esercizi)

Descrivere, mediante diagramma di flusso, un algoritmo che continua a leggere valori in input finché l'utente non inserisce il **valore 0**. L'algoritmo restituisce in output la **somma dei valori inseriti** dall'utente.



## Flow Chart (Esercizi)

Descrivere, mediante diagramma di flusso, un algoritmo che continua a leggere valori in input finché l'utente non inserisce il **valore 0**. L'algoritmo restituisce in output il **conteggio dei numeri pari e dispari** inseriti dall'utente.



## Flow Chart (Esercizio per casa)

Descrivere, mediante diagramma di flusso, un algoritmo che chiede in input un numero intero **n** e ne visualizza in output i **primi 10 multipli** (ovvero  $n*1$ ,  $n*2$ , ...  $n*10$ ).