

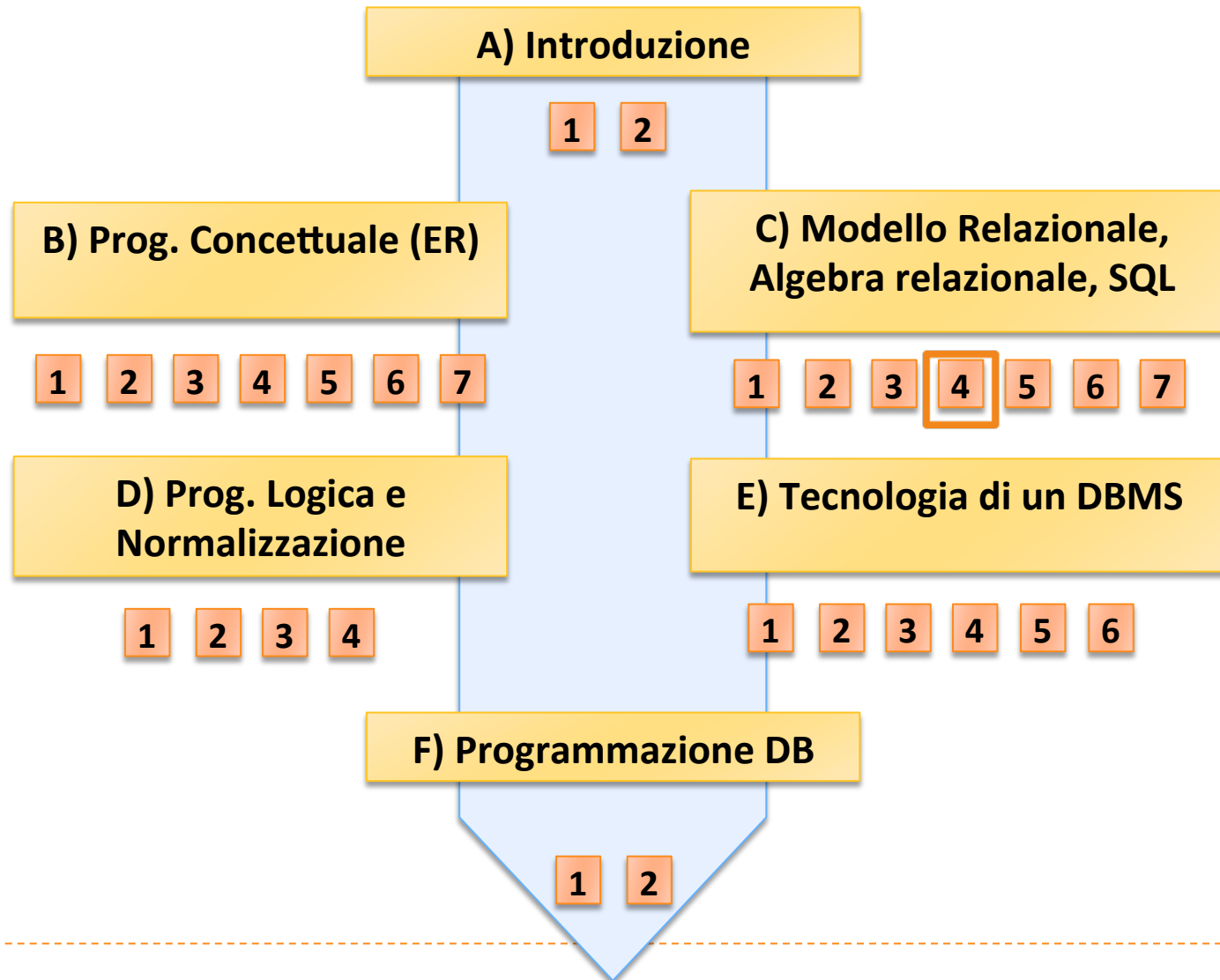


Basi di Dati



Interrogazioni in SQL

Basi di Dati – Dove ci troviamo?



Dichiaratività di SQL

- ▶ In SQL l'utente specifica **QUALE** informazione è di suo interesse ma non **COME** estrarla dai dati
- ▶ Il sistema costruisce una strategia di accesso (**QUERY OPTIMIZATION**)
- ▶ È l'aspetto più qualificante delle basi di dati relazionali

Struttura di SQL

- ▶ Basata sulla composizione di blocchi
 - ▶ SELECT
 - ▶ FROM
 - ▶ WHERE

- ▶ Ogni blocco ha il potere espressivo di una qualunque combinazione di selezioni, proiezioni e join

Esempio: gestione degli esami universitari

studente

MATR	NOME	CITTA'	C-DIP
123	Carlo	Bologna	Inf
415	Paola	Torino	Inf
702	Antonio	Roma	Log

esame

MATR	COD-CORSO	DATA	VOTO
123	1	2004-09-07	30
123	2	2005-01-08	28
702	2	2004-09-07	20

corso

COD-CORSO	TITOLO	DOCENTE
1	matematica	Barozzi
2	informatica	Natali

Interrogazioni semplici

```
SELECT *  
FROM STUDENTE
```

MATR	NOME	CITTA'	C-DIP
123	Carlo	Bologna	Inf
415	Paola	Torino	Inf
702	Antonio	Roma	Log

```
SELECT *  
FROM STUDENTE  
WHERE C-DIP = 'Log'
```

MATR	NOME	CITTA'	C-DIP
702	Antonio	Roma	Log

Sintassi nella clausola **SELECT**

- ▶ **SELECT** *
- ▶ **SELECT** NOME, CITTÀ
- ▶ **SELECT DISTINCT** CITTÀ
- ▶ **SELECT** CITTÀ **AS** LUOGO-DI-RESIDENZA
- ▶ **SELECT** REDDITO-CATASTALE * 0.05
AS TASSA-ICI
- ▶ **SELECT SUM** (SALARIO)

Sintassi della clausola FROM

- ▶ FROM STUDENTE
- ▶ FROM STUDENTE AS X
- ▶ FROM STUDENTE X
- ▶ FROM STUDENTE, ESAME
- ▶ FROM STUDENTE JOIN ESAME
ON STUDENTE.MATR=ESAME.MATR

Sintassi del predicato di selezione

Espressione booleana di predicati semplici

- ▶ operazioni booleane :
 - ▶ AND (P1 AND P2)
 - ▶ OR (P1 OR P2)
 - ▶ NOT (NOT P1)
- ▶ predicati semplici :
 - ▶ TRUE, FALSE
 - ▶ termine comparatore
termine
- ▶ comparatore :
 - ▶ =, !=, <, <=, >, >=
- ▶ termine :
 - ▶ costante, attributo
 - ▶ espressione aritmetica di costanti e attributi

Sintassi della clausola **WHERE**

- ▶ Espressione Booleana di predicati semplici (come in algebra)
- ▶ Alcuni predicati aggiuntivi:
 - ▶ **BETWEEN:**
 - ▶ DATA **BETWEEN** 1997-01-01 **AND** 2006-12-31
 - ▶ **LIKE:**
 - ▶ C-DIP LIKE 'log%' → **stringa arbitraria**
 - ▶ TARGA LIKE 'E_777CX' → **carattere arbitrario**

Sintassi della clausola WHERE

- ▶ WHERE NOME LIKE 'B%'
 - ▶ BOFFI
 - ▶ BUCCHI
 - ▶ BIANCHI
 - ▶ BIFFI
 - ▶ BONFATTI

- ▶ WHERE NOME LIKE 'BI%'
 - ▶ BIANCHI
 - ▶ BIFFI

- ▶ Operatori aritmetici nel WHERE:
 - ▶ WHERE SALARIO + STRAORD > 18
 - ▶ WHERE STRAORD + 5 > SALARIO

Selezione

```
SELECT *  
FROM STUDENTE  
WHERE NOME='Paola'
```

- ▶ Il risultato è una tabella (priva di nome) con schema:
 - ▶ lo stesso schema di **STUDENTE**
- ▶ istanze :
 - ▶ le tuple di **STUDENTE** che soddisfano il predicato di selezione

MATR	NOME	CITTA'	C-DIP
415	Paola	Torino	Inf

Esempio di selezione

```
SELECT *  
FROM STUDENTE  
WHERE (CITTÀ='Torino') OR  
((CITTÀ='Roma') AND NOT (C-DIP='log'))
```

MATR	NOME	CITTÀ	C-DIP
123	Carlo	Bolegna	Inf
415	Paola	Torino	Inf
702	Antonio	Roma	Log

Gestione duplicati (proiezione)

**SELECT DISTINCT C-DIP
FROM STUDENTE**

C-DIP
Inf
Log

**SELECT C-DIP
FROM STUDENTE**

C-DIP
Inf
Inf
Log

Valori nulli

```
SELECT *  
FROM STUDENTE  
WHERE CITTÀ IS [NOT] NULL
```

- ▶ Attenzione :
 - ▶ se CITTÀ ha valore NULL il risultato per (CITTÀ = 'Milano') ha valore 'UNKNOWN'

Composizione di predicati con valore nullo

Logica a tre valori (V,F,U)
(Vero, Falso, Unknown)

V AND U = U

V OR U = V

F AND U = F

F OR U = U

NOT U = U

P =
(CITTÀ IS NOT NULL)
AND (C-DIP LIKE '%Inf')


CITTA'	C-DIP	P	TUPLA SELEZ.
Milano	Inf	V	si
Milano	NULL	U	no
NULL	Inf	F	no
Milano	Log	F	no

Join di due tabelle

```
SELECT NOME
FROM STUDENTE, ESAME
WHERE STUDENTE.MATR = ESAME.MATR
AND C-DIP LIKE 'In%' AND VOTO = 30
```

Variante sintattica:


```
SELECT NOME
FROM STUDENTE JOIN ESAME
    ON STUDENTE.MATR = ESAME.MATR
WHERE C-DIP LIKE 'In%' AND VOTO = 30
```



NOME
Carlo

Join di tre tabelle

```
SELECT NOME  
FROM STUDENTE, ESAME, CORSO  
WHERE STUDENTE.MATR = ESAME.MATR  
AND CORSO.COD-CORSO = ESAME.COD-CORSO  
AND TITOLO LIKE 'info%' AND VOTO < 24
```



NOME
Antonio

Interrogazioni con variabili relazionali

► Es: chi sono i dipendenti “non-pendolari”?

MATR	NOME	CITTÀ	SALARIO	MATR-MGR
1	Piero	BO	1500 €	2
2	Giorgio	MO	2000 €	4
3	Giovanni	FE	1000 €	2
....

impiegato

DNO	NOME	CITTÀ	MATR-MGR
1	AMMINISTRAZIONE	BO	2
2	SPEDIZIONI	FE	7
...

dipartimento

Variabili relazionali

```
SELECT X.NOME
FROM IMPIEGATO AS X, DIPARTIMENTO AS Y
WHERE X.MATR-MGR = Y.MATR-MGR
AND Y.CITTÀ = X.CITTÀ
```



X.NOME
Piero

```
AND Y.CITTÀ != X.CITTÀ
```



X.NOME
Giovanni

Variabili relazionali (self-join)

- ▶ Es: Chi sono i dipendenti di Giorgio?

```
SELECT X.NOME  
FROM IMPIEGATO AS X, IMPIEGATO AS Y  
WHERE X.MATR-MGR = Y.MATR  
AND Y.NOME = 'Giorgio'
```

X.NOME
Piero
Giovanni

Blocchi SQL per la modifica

- ▶ Tre operazioni elementari:
 - ▶ Cancellazione: DELETE
 - ▶ inserimento: INSERT
 - ▶ modifica: UPDATE

- ▶ Cancellazione
 - ▶ `DELETE FROM STUDENTE WHERE MATR = '678678'`

Inserimento

```
INSERT INTO STUDENTE
```

```
VALUES
```

```
('456878', 'Giorgio Rossi', 'Bologna', 'Logistica e Produzione')
```

```
[ , ( ... ) ]
```

```
INSERT INTO BOLOGNESI
```

```
( SELECT *
```

```
FROM STUDENTE
```

```
WHERE CITTÀ = 'Bologna'
```

```
)
```

Modifica

```
UPDATE ESAME
```

```
SET VOTO = 30
```

```
WHERE DATA = 2004-04-01
```

```
UPDATE ESAME
```

```
SET VOTO = VOTO + 1
```

```
WHERE MATR = '787989'
```


Esercizi

- ▶ Dato un DB per la gestione del personale esprimere in SQL le **interrogazioni** seguenti:
 - ▶ in quali tipi di progetti lavora Giovanni?
 - ▶ chi è il manager di Piero?
 - ▶ in quali progetti lavora Piero?
 - ▶ quali impiegati lavorano nel progetto “IDEA”?
 - ▶ quali impiegati lavorano al 100% del loro tempo nel progetto “WIDE”?
- ▶ E le **modifiche**:
 - ▶ inserire la tupla <4,Luca,2004-01-01,2M,1>
 - ▶ modificare il salario di Piero in 3000€
 - ▶ aumentare il salario di Giorgio del 5%
 - ▶ cancellare i dati di Giovanni

Esempio : gestione personale

impiegato

MATR	NOME	DATA-ASS	SALARIO	MATR-MGR
1	Piero	2002-01-01	1500 €	2
2	Giorgio	2004-01-01	2000 €	null
3	Giovanni	2003-07-01	1000 €	2

assegnamento

MATR	NUM-PROG	PERC
1	3	50
1	4	50
2	3	100
3	4	100

progetto

NUM-PROG	TITOLO	TIPO
3	Idea	Esprit
4	Wide	Esprit

Esercizi

- **in quali tipi di progetti lavora Giovanni?**

```
SELECT TIPO
FROM IMPIEGATO AS I, ASSEGNAMENTO AS A,
     PROGETTO AS P
WHERE I.MATR=A.MATR
AND A.NUM-PROG=P.NUM-PROG
AND NOME='Giovanni'
```

TIPO
Esprit

- **chi è il manager di Piero?**

```
SELECT Y.NOME
FROM IMPIEGATO AS X, IMPIEGATO AS Y
WHERE X.MATR-MGR=Y.MATR
AND X.NOME='Piero'
```

NOME
Giorgio

Esercizi

- **modificare il salario di Piero in 3000 €**

```
UPDATE IMPIEGATO  
SET SALARIO = 3000  
WHERE NOME='Piero'
```

- **aumentare il salario di Giorgio del 5%**

```
UPDATE IMPIEGATO  
SET SALARIO = SALARIO * 1.05  
WHERE NOME='Giorgio'
```