



470

Meta dati

- Molti DBMS implementano i meta dati (o catalogo)
- Quasi sempre la struttura fisica di memorizzazione dei dati rispecchia il modello
- Un database ad oggetti memorizzerà e gestirà il catalogo mediante oggetti
- I meta dati differiscono da sistema a sistema

Basi di Dati + Laboratorio - Informatica Triennale - Corso A

471

Meta dati e SQL-2

- Lo standard prevede due livelli
- Definition_schema
 - Tabelle contenenti la descrizione delle strutture della base di dati
- Information_schema
 - Tabelle, viste, colonne, domini, vincoli, ecc..
 - Fanno parte integrante dello standard

Basi di Dati + Laboratorio - Informatica Triennale - Corso A

472

Inserimento di dati

- Inserimento di nuove righe

```
INSERT INTO nomeTabella ( [listaAttributi] )  
VALUES ( listaValori ) | selectSQL
```

listaAttributi := [*nomeAttributo*] [{ , *nomeAttributo* }]

Basi di Dati + Laboratorio - Informatica Triennale - Corso A

473

Esempio INSERT

```
INSERT INTO Impiegato (matricola,dipart)
VALUES ('354556','risorse umane');
```

- Come sono valorizzati i seguenti attributi?
 - nome
 - cognome
 - stipendio

Basi di Dati + Laboratorio - Informatica Triennale - Corso A

474

NON usare:

```
INSERT INTO Dipartimento(matricola,dipart)
VALUES
('354555','risorse umane'),
('354556','magazzino'),
('354557','marketing'),
('354558','fornitori'),
;
INSERT INTO Dipartimento(matricola,dipart)
VALUES '354555','risorse umane';
INSERT INTO Dipartimento(matricola,dipart)
VALUES '354556','risorse inumane';
```

Basi di Dati + Laboratorio - Informatica Triennale - Corso A

475

Insert – righe appartenenti ad altra tabella

- È possibile inserire i dati effettuando una copia totale o parziale di dati da altre tabelle

```
INSERT INTO giocatori_dilettanti  
  (n_giocatore, cognome, città, telefono)  
SELECT  
  (n_giocatore, cognome, città, telefono)  
FROM giocatori  
WHERE n_socio IS NULL
```

Basi di Dati + Laboratorio - Informatica Triennale - Corso A

476

Aggiornamento

- Update

```
UPDATE nomeTabella  
SET nomeAttributo = <espressione |  
  selectSQL | null | default>  
{, nomeAttributo = <espressione | selectSQL  
  | null | default>}  
[where condizione]
```

Basi di Dati + Laboratorio - Informatica Triennale - Corso A

477

Aggiornamento - esempio

```
UPDATE impiegati  
SET annocorso=3, nome='Mario',  
    cognome='Rossi';
```

Basi di Dati + Laboratorio - Informatica Triennale - Corso A

478

Cancellazione - esempio

```
DELETE FROM impiegati  
WHERE condizione;
```

Basi di Dati + Laboratorio - Informatica Triennale - Corso A

479

SQL, operazioni sui dati (DML)

- modifica:
 - INSERT, DELETE, UPDATE
- interrogazione:
 - SELECT

Basi di Dati + Laboratorio - Informatica Triennale - Corso A

480

Istruzione SELECT

- SELECT ListaAttributi
- FROM ListaTabelle
- [WHERE Condizione]

- Lista attributi o "target list"
- clausola FROM
- clausola WHERE

Basi di Dati + Laboratorio - Informatica Triennale - Corso A

481

Maternità	Madre	Figlio	Persone		
	Luisa	Maria	Nome	Età	Reddito
	Luisa	Luigi	Andrea	27	21
	Anna	Olga	Aldo	25	15
	Anna	Filippo	Maria	55	42
	Maria	Andrea	Anna	50	35
	Maria	Aldo	Filippo	26	30
Paternità	Padre	Figlio	Luigi	50	40
	Sergio	Franco	Franco	60	20
	Luigi	Olga	Olga	30	41
	Luigi	Filippo	Sergio	85	35
	Franco	Andrea	Luisa	75	87
	Franco	Aldo			

Basi di Dati + Laboratorio - Informatica Triennale - Corso A

482

Proiezione, senza selezione

- Nome e reddito di tutte le persone

$\text{PROJ}_{\text{Nome, Reddito}}(\text{Persone})$

SELECT nome, reddito
FROM persone

Basi di Dati + Laboratorio - Informatica Triennale - Corso A

483

Risultato

Persone

Nome	Reddito
Andrea	21
Aldo	15
Maria	42
Anna	35
Filippo	30
Luigi	40
Franco	20
Olga	41
Olga	41
Luisa	87

$\text{PROJ}_{\text{Nome, Reddito}}(\text{Persone})$

SELECT nome, reddito
FROM persone

Basi di Dati + Laboratorio - Informatica Triennale - Corso A

484

Selezione e proiezione

Nome e reddito delle persone con meno di 30 anni

$\text{PROJ}_{\text{Nome, Reddito}}(\text{SEL}_{\text{Eta} < 30}(\text{Persone}))$

SELECT nome, reddito
FROM persone
WHERE eta < 30

Basi di Dati + Laboratorio - Informatica Triennale - Corso A

485

Persone

Nome	Reddito
Andrea	21
Aldo	15
Filippo	30

Basi di Dati + Laboratorio - Informatica Triennale - Corso A

486

SELECT - sinonimi

```
SELECT nome, reddito  
FROM persone  
WHERE eta < 30
```

```
SELECT p.nome as nome, p.reddito as reddito  
FROM persone p  
WHERE p.eta < 30
```

Basi di Dati + Laboratorio - Informatica Triennale - Corso A

487

Selezione, senza proiezione

- Nome, età e reddito delle persone con meno di trenta anni

$SEL_{Eta < 30}(Persone)$

```
SELECT *  
FROM persone  
WHERE eta < 30
```

Basi di Dati + Laboratorio - Informatica Triennale - Corso A

488

SELECT - * e lista attributi equivalenti?

```
select *  
from persone  
where eta < 30
```

```
select nome, età, reddito  
from persone  
where eta < 30
```

Basi di Dati + Laboratorio - Informatica Triennale - Corso A

489

SELECT - sinonimi

Data una relazione R(A,B)

```
SELECT *  
FROM R
```

equivale (intutivamente) a

```
SELECT X.A as A, X.B as B  
FROM R X
```

Basi di Dati + Laboratorio - Informatica Triennale - Corso A

490

Target list ed espressioni

```
SELECT reddito, reddito/2 as redditoSemestrale  
FROM Persone  
WHERE Nome = 'Luigi'
```

FATTURA(ordine, prezzo, iva)

Stampare in output il prezzo e il costo per il cliente
(prezzo comprensivo di IVA).

```
SELECT prezzo, (prezzo * IVA / 100) + prezzo  
AS prezzolvato  
FROM fattura WHERE ordine=3108;
```

Basi di Dati + Laboratorio - Informatica Triennale - Corso A

491

Condizione complessa

```
SELECT *  
FROM persone  
WHERE  
    reddito > 25 AND  
    (eta > 30 AND eta < 60)
```

Basi di Dati + Laboratorio - Informatica Triennale - Corso A