

## CREATE TABLE, esempio

```
CREATE TABLE Impiegato(
  Matricola CHAR(6) PRIMARY KEY,
  Nome CHAR(20) NOT NULL,
  Cognome CHAR(20) NOT NULL,
  Dipart CHAR(15),
  Stipendio NUMERIC(9) DEFAULT 0,
  FOREIGN KEY(Dipart) REFERENCES
    Dipartimento(NomeDip),
  UNIQUE (Cognome,Nome)
)
```

7

Università degli Studi di Salerno

## Domini elementari

- **Carattere**: singoli caratteri o stringhe, anche di lunghezza variabile
- **Bit**: singoli booleani o stringhe
- **Numerici**, esatti e approssimati
- **Data, ora, intervalli di tempo**
- Introdotti in SQL:1999:
  - **Boolean**
  - **BLOB, CLOB** (binary/character large object): per grandi immagini e testi

9

Università degli Studi di Salerno

## Domini in SQL2: Numeri e Stringhe

- **Numerici**
  - Interi (INTEGER o INT, SMALLINT)
  - Reali (FLOAT, REAL, DOUBLE PRECISION)
  - Numeri formattati (DECIMAL(i,j), DEC(i,j), NUMERIC(i,j))
    - i, detta precisione, indica il numero di cifre decimali
    - j, detta scala, indica il numero di cifre dopo la virgola
- **Stringhe di caratteri**
  - A lunghezza fissa (CHAR(n), CHARACTER(n))
  - A lunghezza variabile (VARCHAR(n) o CHAR VARYING(n))
    - Per default n, il numero massimo di caratteri, è 1
- **Stringhe di bit**
  - A lunghezza fissa (BIT(n))
  - A lunghezza variabile (BIT VARYING(n))

10

Università degli Studi di Salerno

## Domini in SQL2: Date e Orari

- **DATE**
  - Ha dieci posizioni, con componenti YEAR, MONTH e DAY. Formato YYYY-MM-DD
- **TIME**
  - Ha (almeno) otto posizioni con componenti HOUR, MINUTE e SECOND. Formato HH:MM:SS
- **TIME(i)**
  - i= precisione delle frazioni di secondo. Specifica i+1 posizioni aggiuntive per TIME, una per il separatore ed i per le frazioni di secondo
- **TIME WITH TIME ZONE**
  - Usa ulteriori 6 posizioni per lo spiazzamento dal GMT, con un range da +13:00 a -12:59

11

Università degli Studi di Salerno

## Definizione di nuovi domini

- Istruzione **CREATE DOMAIN**:
  - definisce un dominio (semplice), utilizzabile in definizioni di relazioni, anche con vincoli e valori di default

Esempio:

```
CREATE DOMAIN Voto
AS SMALLINT DEFAULT NULL
CHECK (value >=18 AND value <= 30)
```

13

Università degli Studi di Salerno

## Vincoli intrarelazionali

- **NOT NULL**
- **UNIQUE** definisce chiavi
- **PRIMARY KEY**: chiave primaria (una sola, implica **NOT NULL**)
- **CHECK**, permette di esprimere vincoli generici (vedremo più avanti)

14

Università degli Studi di Salerno

## PRIMARY KEY, alternative

Matricola CHAR(6) PRIMARY KEY

Matricola CHAR(6),  
...,  
PRIMARY KEY (Matricola)

17

Università degli Studi di Salerno

## Chiavi su più attributi, attenzione

Nome CHAR(20) NOT NULL,  
Cognome CHAR(20) NOT NULL,  
UNIQUE (Cognome, Nome),

Nome CHAR(20) NOT NULL UNIQUE,  
Cognome CHAR(20) NOT NULL UNIQUE,

- Non è la stessa cosa!

19

Università degli Studi di Salerno

Basi di dati – Prof. G. Polese Capitolo 4

### Infrazioni

Codice	Data	Vigile	Prov	Numero
34321	1/2/95	3987	MI	39548K
53524	4/3/95	3295	TO	E39548
64521	5/4/96	3295	PR	839548
73321	5/2/98	9345	PR	839548

**Vigili**

Matricola	Cognome	Nome
3987	Rossi	Luca
3295	Neri	Piero
9345	Neri	Mario
7543	Mori	Gino

21 Università degli Studi di Salerno

Basi di dati – Prof. G. Polese Capitolo 4

### Infrazioni

Codice	Data	Vigile	Prov	Numero
34321	1/2/95	3987	MI	39548K
53524	4/3/95	3295	TO	E39548
64521	5/4/96	3295	PR	839548
73321	5/2/98	9345	PR	839548

**Auto**

Prov	Numero	Cognome	Nome
MI	39548K	Rossi	Mario
TO	E39548	Rossi	Mario
PR	839548	Neri	Luca

22 Università degli Studi di Salerno

Basi di dati – Prof. G. Polese Capitolo 4

### CREATE TABLE, esempio

```
CREATE TABLE Infrazioni(
  Codice CHAR(6) PRIMARY KEY,
  Data DATE NOT NULL,
  Vigile INTEGER
    REFERENCES Vigili(Matricola)
    ON UPDATE CASCADE ON DELETE SET NULL,
  Provincia CHAR(2),
  Numero CHAR(6) ,
  FOREIGN KEY(Provincia, Numero)
    REFERENCES Auto(Provincia, Numero)
    ON UPDATE CASCADE ON DELETE CASCADE
)
```

23 Università degli Studi di Salerno

Basi di dati – Prof. G. Polese Capitolo 4

### Modifiche degli schemi

```
ALTER DOMAIN
ALTER TABLE
DROP DOMAIN
DROP TABLE
...
```

24 Università degli Studi di Salerno

## Alter Table: aggiunta di un attributo

- Vogliamo aggiungere l'email di un impiegato nella tabella Impiegato:
  - ALTER TABLE IMPIEGATO  
ADD EMAIL VARCHAR(12);
- Il valore di EMAIL o si specifica di default o sarà null.
- Con la ALTER TABLE non è permessa la clausola NOT NULL

25

Università degli Studi di Salerno

## ALTER TABLE: Eliminazione attributo

- Quando si elimina una colonna occorre scegliere l'opzione CASCADE o RESTRICT
  - Con CASCADE vincoli e viste che referenziano la colonna sono eliminati dallo schema
  - Con RESTRICT il comando ha successo solo se nessun vincolo o vista referencia la colonna.
- Esempio: rimuovere la colonna INDIRIZZO dalla tabella IMPIEGATO
  - ALTER TABLE IMPIEGATO DROP INDIRIZZO  
CASCADE;

26

Università degli Studi di Salerno

## ALTER TABLE: Modifica vincoli

- Modifica di una colonna eliminando una clausola di default o definendone una nuova  
Esempi:
  - ALTER TABLE IMPIEGATO ALTER DIPART  
DROP DEFAULT;
  - ALTER TABLE INFRAZIONE ALTER VIGILE  
SET DEFAULT "3334";
- E' possibile eliminare un vincolo solo se gli si è dato un nome nella CREATE TABLE tramite la keyword CONSTRAINT

27

Università degli Studi di Salerno

## SQL, operazioni sui dati

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

30

Università degli Studi di Salerno

Basi di dati – Prof. G. Polese

Capitolo 4

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

Università degli Studi di Salerno

Basi di dati – Prof. G. Polese

Capitolo 4

## Selezione e proiezione

- Nome e reddito delle persone con meno di trenta anni

$PROJ_{Nome, Reddito}(SEL_{Eta < 30}(Persone))$

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

Università degli Studi di Salerno

Basi di dati – Prof. G. Polese

Capitolo 4

## Condizione “LIKE”

- Le persone che hanno un nome che inizia per 'A' e ha una 'd' come terza lettera

```
select *
from persone
where nome like 'A_d%'
```

Università degli Studi di Salerno

Basi di dati – Prof. G. Polese

Capitolo 4

- Gli impiegati la cui età è o potrebbe essere maggiore di 40

$SEL_{Eta > 40 \text{ OR } Eta \text{ IS NULL}}(Impiegati)$

```
select *
from impiegati
where eta > 40 or eta is null
```

Università degli Studi di Salerno

## Selezione, proiezione e join

- I padri di persone che guadagnano più di venti milioni

```

PROJPadre(paternita
  JOINFiglio=Nome
  SELReddito>20(persone))

select distinct padre
from persone, paternita
where figlio = nome and reddito > 20
  
```

54

Università degli Studi di Salerno

## Join naturale

- Padre e madre di ogni persona

paternita JOIN maternita

```

select paternita.figlio, padre, madre
from maternita, paternita
where paternita.figlio = maternita.figlio
  
```

55

Università degli Studi di Salerno

- Le persone che guadagnano più dei rispettivi padri; mostrare nome, reddito e reddito del padre

```

PROJNome, Reddito, RP(SELReddito>RP
  (RENNP,EP,RP ← Nome, Eta, Reddito (persone)
    JOINNP=Padre
    (paternita JOINFiglio=Nome persone)))

select f.nome, f.reddito, p.reddito
from persone p, paternita, persone f
where p.nome = padre and
      figlio = f.nome and f.reddito > p.reddito
  
```

56

Università degli Studi di Salerno

## SELECT, con ridenominazione del risultato

```

select figlio, f.reddito as reddito,
      p.reddito as redditoPadre
from persone p, paternita, persone f
where p.nome = padre and figlio = f.nome
and f.reddito > p.reddito
  
```

57

Università degli Studi di Salerno

## Join esplicito

- Padre e madre di ogni persona

```
select paternita.figlio, padre, madre
from maternita, paternita
where paternita.figlio = maternita.figlio
```

```
select madre, paternita.figlio, padre
from maternita join paternita on
    paternita.figlio = maternita.figlio
```

58

Università degli Studi di Salerno

- Le persone che guadagnano più dei rispettivi padri; mostrare nome, reddito e reddito del padre

```
select f.nome, f.reddito, p.reddito
from persone p, paternita, persone f
where p.nome = padre and
    figlio = f.nome and
    f.reddito > p.reddito
```

```
select f.nome, f.reddito, p.reddito
from persone p join paternita on p.nome = padre
join persone f on figlio = f.nome
where f.reddito > p.reddito
```

60

Università degli Studi di Salerno

## Ulteriore estensione: join naturale (meno diffuso)

```
PROJFiglio, Padre, Madre(
    paternita JOINFiglio = Nome RENNome <- Figlio(maternita))
```

```
paternita JOIN maternita
```

```
select madre, paternita.figlio, padre
from maternita join paternita on
    paternita.figlio = maternita.figlio
```

```
select madre, paternita.figlio, padre
from maternita natural join paternita
```

61

Università degli Studi di Salerno

## Join esterno: "outer join"

- Padre e, se nota, madre di ogni persona

```
select paternita.figlio, padre, madre
from paternita left join maternita
on paternita.figlio = maternita.figlio
```

```
select paternita.figlio, padre, madre
from paternita left outer join maternita
on paternita.figlio = maternita.figlio
```

- outer e' opzionale

62

Università degli Studi di Salerno

## Outer join

```
select paternita.figlio, padre, madre
from maternita join paternita
on maternita.figlio = paternita.figlio
```

```
select paternita.figlio, padre, madre
from maternita left outer join paternita
on maternita.figlio = paternita.figlio
```

```
select paternita.figlio, padre, madre
from maternita full outer join paternita
on maternita.figlio = paternita.figlio
```

63

Università degli Studi di Salerno

## Ordinamento del risultato

- Nome e reddito delle persone con meno di trenta anni **in ordine alfabetico inverso**

```
select nome, reddito
from persone
where eta < 30
order by nome DESC
```

- La clausola di default è ASC

64

Università degli Studi di Salerno

## Operatori aggregati: COUNT

- Il numero di figli di Franco

```
select count(*) as NumFigliDiFranco
from Paternita
where Padre = 'Franco'
```

- l'operatore aggregato (**count**) viene applicato al risultato dell'interrogazione:

```
select *
from Paternita
where Padre = 'Franco'
```

67

Università degli Studi di Salerno

## Altri operatori aggregati

- SUM, AVG, MAX, MIN**
- Media dei redditi dei figli di Franco

```
select avg(reddito)
from persone join paternita on nome=figlio
where padre='Franco'
```

70

Università degli Studi di Salerno



## Operatori aggregati e target list

- un'interrogazione scorretta:

```
select nome, max(reddito)
from persone
```

- di chi sarebbe il nome? La target list deve essere omogenea

```
select min(eta), avg(reddito)
from persone
```

72

Università degli Studi di Salerno

## Operatori aggregati e raggruppamenti

- Il numero di figli di ciascun padre

```
select padre, count(*) AS NumFigli
from paternita
group by Padre
```

paternita

Padre	Figlio
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo

Padre	NumFigli
Sergio	1
Luigi	2
Franco	2

74

Università degli Studi di Salerno

## Raggruppamenti e target list

scorretta

```
select padre, avg(f.reddito), p.reddito
from persone f join paternita on figlio = nome join
     persone p on padre =p.nome
group by padre
```

corretta

```
select padre, avg(f.reddito), p.reddito
from persone f join paternita on figlio = nome join
     persone p on padre =p.nome
group by padre, p.reddito
```

76

Università degli Studi di Salerno

## Condizioni sui gruppi

- I padri i cui figli hanno un reddito medio maggiore di 25

```
select padre, avg(f.reddito)
from persone f join paternita on figlio = nome
group by padre
having avg(f.reddito) > 25
```

77

Università degli Studi di Salerno

## WHERE o HAVING?

- I padri i cui figli sotto i 30 anni hanno un reddito medio maggiore di 25

```
select padre, avg(f.reddito)
from persone f join paternita on figlio = nome
where eta < 30
group by padre
having avg(f.reddito) > 25
```

78

Università degli Studi di Salerno

## Sintassi, riassumiamo++

### SelectSQL ::=

```
select ListaAttributiOEspressioni
from ListaTabelle
[ where CondizioniSemplici ]
[ group by
    ListaAttributiDiRaggruppamento
  [ having CondizioniAggregate ]
  [ order by ListaAttributiDiOrdinamento ]
```

79

Università degli Studi di Salerno

## Notazione posizionale, 2

```
select padre, figlio
from paternita
union
select figlio, madre
from maternita
```

```
select padre, figlio
from paternita
union
select madre, figlio
from maternita
```

83

Università degli Studi di Salerno

## Intersezione

```
select Nome
from Impiegato
intersect
select Cognome as Nome
from Impiegato
```

- equivale a

```
select I.Nome
from Impiegato I, Impiegato J
where I.Nome = J.Cognome
```

86

Università degli Studi di Salerno

- nome e reddito del padre di Franco

```
select Nome, Reddito
from Persone, Paternita
where Nome = Padre and Figlio = 'Franco'
```

```
select Nome, Reddito
from Persone
where Nome = ( select Padre
               from Paternita
               where Figlio = 'Franco')
```

88

Università degli Studi di Salerno

- Nome e reddito dei padri di persone che guadagnano più di 20 milioni

```
select distinct P.Nome, P.Reddito
from Persone P, Paternita, Persone F
where P.Nome = Padre and Figlio = F.Nome
and F.Reddito > 20
```

```
select Nome, Reddito
from Persone
where Nome in (select Padre
               from Paternita
               where Figlio = any (select Nome
                                   from Persone
                                   where Reddito > 20))
```

90

Università degli Studi di Salerno

- Nome e reddito dei padri di persone che guadagnano più di 20 milioni

```
select distinct P.Nome, P.Reddito
from Persone P, Paternita, Persone F
where P.Nome = Padre and Figlio = F.Nome
and F.Reddito > 20
```

```
select Nome, Reddito
from Persone
where Nome in (select Padre
               from Paternita, Persone
               where Figlio = Nome
               and Reddito > 20)
```

91

Università degli Studi di Salerno

- Nome e reddito dei padri di persone che guadagnano più di 20 milioni, con indicazione del reddito del figlio

```
select distinct P.Nome, P.Reddito, F.Reddito
from Persone P, Paternita, Persone F
where P.Nome = Padre and Figlio = F.Nome
and F.Reddito > 20
```

```
select Nome, Reddito, ???
from Persone
where Nome in (select Padre
               from Paternita
               where Figlio = any (select Nome
                                   from Persone
                                   where Reddito > 20))
```

93

Università degli Studi di Salerno

## Interrogazioni nidificate, commenti, 3

- regole di visibilità:
  - non è possibile fare riferimenti a variabili definite in blocchi più interni
  - se un nome di variabile è omissso, si assume riferimento alla variabile più “vicina”
- in un blocco si può fare riferimento a variabili definite in blocchi più esterni; la semantica base (prodotto cartesiano, selezione, proiezione) non funziona più, vedremo presto

94

Università degli Studi di Salerno

- Le persone che hanno almeno un figlio

```
select *
from Persone
where exists (
    select *
    from Paternita
    where Padre = Nome) or
    exists (
    select *
    from Maternita
    where Madre = Nome)
```

96

Università degli Studi di Salerno

- I padri i cui figli guadagnano tutti più di venti milioni

```
select distinct Padre
from Paternita Z
where not exists (
    select *
    from Paternita W, Persone
    where W.Padre = Z.Padre
    and W.Figlio = Nome
    and Reddito <= 20)
```

97

Università degli Studi di Salerno

## Visibilità

- scorretta:

```
select *
from Impiegato
where Dipart in (select Nome
    from Dipartimento D1
    where Nome = 'Produzione') or
    Dipart in (select Nome
    from Dipartimento D2
    where D2.Citta = D1.Citta)
```

99

Università degli Studi di Salerno

## Disgiunzione e unione (ma non sempre)

```
select * from Persone where Reddito > 30
union
select F.*
from Persone F, Paternita, Persone P
where F.Nome = Figlio and Padre = P.Nome
and P.Reddito > 30

select *
from Persone F
where Reddito > 30 or
exists (select *
        from Paternita, Persone P
        where F.Nome = Figlio and Padre = P.Nome
        and P.Reddito > 30)
```

100

Università degli Studi di Salerno

## Differenza e nidificazione

```
select Nome from Impiegato
except
select Cognome as Nome from Impiegato

select Nome
from Impiegato I
where not exists (select *
                 from Impiegato
                 where Cognome = I.Nome)
```

101

Università degli Studi di Salerno

## Massimo e nidificazione

- La persona (o le persone) con il reddito massimo

```
select *
from persone
where reddito = (select max(reddito)
                from persone)
```

102

Università degli Studi di Salerno

## Operazioni di aggiornamento

- operazioni di
  - inserimento: **insert**
  - eliminazione: **delete**
  - modifica: **update**
- di una o più ennuple di una relazione
- sulla base di una condizione che può coinvolgere anche altre relazioni

103

Università degli Studi di Salerno

## Inserimento, esempi

```
INSERT INTO Persone VALUES ('Mario',25,52)
```

```
INSERT INTO Persone(Nome, Eta, Reddito)
VALUES('Pino',25,52)
```

```
INSERT INTO Persone(Nome, Reddito)
VALUES('Lino',55)
```

```
INSERT INTO Persone ( Nome )
SELECT Padre
FROM Paternita
WHERE Padre NOT IN (SELECT Nome
                     FROM Persone)
```

105

Università degli Studi di Salerno

## Eliminazione, esempi

```
DELETE FROM Persone
WHERE Eta < 35
```

```
DELETE FROM Paternita
WHERE Figlio NOT in ( SELECT Nome
                     FROM Persone)
```

```
DELETE FROM Paternita
```

108

Università degli Studi di Salerno

## Modifica di ennuple

```
UPDATE NomeTabella
SET Attributo = < Espressione |
                SELECT ... |
                NULL |
                DEFAULT >
[ WHERE Condizione ]
```

110

Università degli Studi di Salerno

## Aggiornamento, esempi

```
UPDATE Persone SET Reddito = 45
WHERE Nome = 'Piero'
```

```
UPDATE Persone
SET Reddito = Reddito * 1.1
WHERE Eta < 30
```

111

Università degli Studi di Salerno