SPRITZ
SECURITY & PRIVACY
RESEARCH GROUP

Basi di Dati

Esercitazione in Lab n.2

Prof. **Mauro Conti**

Dipartimento di Matematica - Università degli studi di Padova

conti@math.unipd.it - http://www.math.unipd.it/~conti

Giuseppe Cascavilla, PhD student

sites.google.com/site/gcascavilla

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Esercizio 1

Ordinare i seguenti domini in base al valore massimo rappresentable, supponendo che:

integer abbia una rappresentazione a 32 bit (unsigned)

e **smallint** a 16 bit (unsigned).

numeric(12,4), **decimal(10)**, **decimal(9)**, **integer**, **smallint**, **decimal(6,1)**.

SOLUZIONE Esercizio 1

	Dominio	Valore Massimo
1.	decimal(10)	9999999999
2.	integer	4294967295 (0 to (2^{32} -1))
3.	decimal(9)	999999999
4.	numeric(12,4)	99999999.9999
5.	decimal(6,1)	99999.9
6.	smallint	65535 (0 to (2^{16} -1))

Esercizio 2

Definire un attributo che permetta di rappresentare stringhe di lunghezza massima pari a 256 caratteri, su cui non sono ammessi valori nulli e con valore di default "sconosciuto".

HELP: create domain **?** as **? ? (?)** default **? ? ?**

SOLUZIONE Esercizio 2

create domain STRING as character varying (256) default 'sconosciuto' not null

Dare le definizioni SQL delle tre tabelle

FONDISTA(Nome, Nazione, Età)

GARA(Nome, Luogo, Nazione, Lunghezza)

GAREGGIA(NomeFondista, NomeGara, Piazzamento)

rappresentando in particolare i vincoli di foreign key della tabella GAREGGIA.

SOLUZIONE Esercizio 3 (1)

Create Table **FONDISTA**

```
(
Nome        character(20)     primary key,
Nazione     character(30),
Età         smallint
)
```

Create table **GARA**

```
(
Nome          character(20)     primary key,
Luogo         character(20),
Nazione       character(20),
Lunghezza     integer
)
```

SOLUZIONE Esercizio 3 (2)

Create table **GAREGGIA**

(

NomeFondista character(20) references **FONDISTA**(Nome),

NomeGara character(20),

Piazzamento smallint,

primary key (NomeFondista, NomeGara),

foreign key (NomeGara) references **GARA**(Nome)

)

Esercizio 4

Dare le definizioni SQL delle tabelle

AUTORE (Nome, Cognome, DataNascita, Nazionalità)

LIBRO (TitoloLibro, NomeAutore, CognomeAutore, Lingua)

Per il vincolo foreign key specificare una politica di **cascade** sulla cancellazione e di set null sulle modifiche.

SOLUZIONE Esercizio 4 (1)


```
Create table AUTORE
(
Nome          character(20),
Cognome       character(20),
DataNascita   date,
Nazionalità   character(20),
              primary key(Nome, Cognome)
)
```

SOLUZIONE Esercizio 4 (2)


```
Create table LIBRO
(
TitoloLibro          character(30)        primary key,
NomeAutore           character(20),
CognomeAutore        character(20),
Lingua               character(20),

foreign key (NomeAutore, CognomeAutore)
references AUTORE(Nome, Cognome)
        on delete cascade
        on update set NULL
)
```

Al primo argomento di **Foreign Key** corrisponde il primo attr. argomento di **References**

Tutte le righe della Tab. interna corrispondenti alla riga cancellata vengono cancellate

Esercizio 5 (1)

Dato lo schema dell'esercizio 4, spiegare cosa può capitare con l'esecuzione dei seguenti comandi di aggiornamento:

a. **delete** from AUTORE where Cognome = 'Rossi'

SOLUZIONE Esercizio 5 (1)

Il comando cancella dalla tabella AUTORE tutte le tuple con Cognome = 'Rossi'. A causa della politica cascade anche le tuple di LIBRO con CognomeAutore = 'Rossi' verranno eliminate.

```
Create table AUTORE
(
Nome                character(20),
Cognome             character(20),
DataNascita         date,
Nazionalità         character(20),
                    primary key(Nome, Cognome)
)

Create table LIBRO
(
TitoloLibro                 character(30)       primary key,
NomeAutore                  character(20),
CognomeAutore               character(20),
Lingua                      character(20),
foreign key (NomeAutore, CognomeAutore)
references AUTORE(Nome, Cognome)
            on delete cascade
            on update set NULL
)
```

Esercizio 5 (2)

Dato lo schema dell'esercizio 4, spiegare cosa può capitare con l'esecuzione dei seguenti comandi di aggiornamento:

b. **update** LIBRO set NomeAutore= 'Umberto'

where CognomeAutore = 'Eco'

SOLUZIONE Esercizio 5 (2)

Il comando non è corretto: Nome e Cognome sono attributi della tabella AUTORE e non della tabella LIBRO

```
Create table AUTORE
(
Nome                    character(20),
Cognome                 character(20),
DataNascita             date,
Nazionalità             character(20),
                        primary key(Nome, Cognome)
)

Create table LIBRO
(
TitoloLibro                     character(30)           primary key,
NomeAutore                      character(20),
CognomeAutore                   character(20),
Lingua                          character(20),
foreign key (NomeAutore, CognomeAutore)
references AUTORE(Nome, Cognome)
            on delete cascade
            on update set NULL
)
```

Esercizio 5 (3)

Dato lo schema dell'esercizio 4, spiegare cosa può capitare con l'esecuzione dei seguenti comandi di aggiornamento:

c. **insert** into AUTORE(Nome,Cognome) values ('Antonio','Bianchi')

SOLUZIONE Esercizio 5 (3)

Il comando aggiunge una nuova tupla alla tabella AUTORE. Non ha alcun effetto sulla tabella LIBRO

```
Create table AUTORE
(
Nome                    character(20),
Cognome                 character(20),
DataNascita             date,
Nazionalità             character(20),
                        primary key(Nome, Cognome)
)

Create table LIBRO
(
TitoloLibro                     character(30)           primary key,
NomeAutore                      character(20),
CognomeAutore                   character(20),
Lingua                          character(20),
foreign key (NomeAutore, CognomeAutore)
references AUTORE(Nome, Cognome)
                on delete cascade
                on update set NULL
)
```

Esercizio 5 (4)

Dato lo schema dell'esercizio 4, spiegare cosa può capitare con l'esecuzione dei seguenti comandi di aggiornamento:

d. **update** AUTORE set Nome = 'Italo' where Cognome = 'Calvino'

SOLUZIONE Esercizio 5 (4)

Le tuple di AUTORE con Cognome = Calvino vengono aggiornate a Nome = Italo. A causa della politica set null gli attributi NomeAutore e CognomeAutore delle tuple di Libro con CognomeAutore = Calvino vengono posti a NULL.

```
Create table AUTORE
(
Nome                character(20),
Cognome             character(20),
DataNascita         date,
Nazionalità         character(20),
                    primary key(Nome, Cognome)
)

Create table LIBRO
(
TitoloLibro                 character(30)       primary key,
NomeAutore                  character(20),
CognomeAutore               character(20),
Lingua                      character(20),
foreign key (NomeAutore, CognomeAutore)
references AUTORE(Nome, Cognome)
            on delete cascade
            on update set NULL
)
```

Esercizi Query SQL e SOLUZIONI

Esercizi Query SQL - SOLUZIONE (1)

1. Trovare la citta'(city) dove ha sede il cliente Frau da Collezione.

```
SELECT city
FROM   customers
WHERE  customerName='Frau da Collezione';
```

Esercizi Query SQL - SOLUZIONE (2)

2. Trovare il Nome (*firstname*), Cognome (*lastname*), e la email del presidente (**cioe' con jobTitle="president"**).

```
SELECT firstName,lastName,email
FROM   employees
WHERE  jobTitle='president';
```

Esercizi Query SQL - SOLUZIONE (3)

3. Trovare il Nome e il limite di credito (**creditLimit**) di tutti i clienti che hanno un limite di credito > di 100000.

```
SELECT customerName,creditLimit
FROM   customers
WHERE  creditLimit>100000;
```

Esercizi Query SQL - SOLUZIONE (4)

4. Trovare tutte le linee di prodotti (ogni linea deve comparire una sola volta)

```
SELECT distinct productline
FROM   productlines;
```

Esercizi Query SQL - SOLUZIONE (5)

5. Restituire il Nome (firstName) e Cognome (lastName) dei dipendenti in ordine alfabetico (prima sul cognome e poi sul nome).

```
SELECT lastName,firstName
FROM   employees
ORDER BY lastName,firstName;
```

Esercizi Query SQL - SOLUZIONE (6)

6. Trovare per ogni dipendente il suo Cognome e quello delle persone che dirige.

```
SELECT dipendente.lastName AS Dipendente, dirigente.
lastName AS Dirige
FROM   employees As dirigente, employees As  dipendente
WHERE  dirigente.reportsTo=dipendente.employeeNumber;
```

Esercizi Query SQL - SOLUZIONE (7)

7. Trovare il prezzo minimo, massimo e medio di ciascuna linea di prodotti

```
SELECT   productLine, min(buyPrice) AS Minimo,
                avg(buyPrice) as Medio,
                max(buyPrice) as Massimo
FROM     products
GROUP BY productLine;
```

Esercizi Query SQL - SOLUZIONE (8)

8. Restituire il nome (customerName) e il limite di credito (creditLimit) dei primi dieci clienti in ordine di limite di credito decrescente (usare LIMIT)

```
SELECT customerName AS Nome, creditLimit AS credito
FROM   customers
ORDER BY creditLimit DESC
LIMIT 10;
```

Esercizi Query SQL - SOLUZIONE (9)

9. Capire se esiste un dipendente con il ruolo di dirigente (Sale Manager (EMEA)) a Paris

```
SELECT  lastName
FROM    employees JOIN offices ON (employees.
officeCode=offices.officecode)
WHERE  employees.jobTitle='Sale Manager (EMEA)' AND
offices.city='Paris';
```

Esercizi Query SQL - SOLUZIONE (10)

10. Il nome delle citta' con sedi con almeno 3 dipendenti.

```
SELECT city
FROM   employees JOIN offices ON (employees.
officeCode=offices.officecode)
GROUP BY city
HAVING COUNT(city) >3;
```

Esercizi Query SQL - SOLUZIONE (11)

11. Trovare il nome di tutti i clienti della sede di Tokyo

```
SELECT customerName
FROM   offices,customers,employees
WHERE employees.officeCode=offices.officecode
    AND employees.employeeNumber=customers.
salesRepEmployeeNumber
    AND offices.city='Tokyo';
```

Esercizi Query SQL - SOLUZIONE (12)

12. Trovare il nome dei clienti che hanno ordinato prodotti della linea planes

```
SELECT distinct customerName
FROM   customers,orders,orderdetails,products
WHERE customers.customerNumber=orders.customerNumber
    AND orders.orderNumber=orderdetails.orderNumber
    AND orderdetails.productCode=products.productCode
    AND productline='planes';
```

Esercizi Query SQL - SOLUZIONE (13)

13. Elencare il numero identificativo dei dipendenti e il numero di clienti gestito da ciascuno

```
SELECT employees.employeeNumber AS number, count(*) AS
norders
FROM   employees JOIN customers ON (employees.
employeeNumber=customers.salesRepEmployeeNumber)
Group BY employeeNumber;
```