

Basi di Dati

## Esercizi - SQL DDL

a.a. 2020-2021

### Linguaggi per basi di dati SQL Soluzione Esercizio 4.2

*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".*

**Create domain STRING as character varying (256)  
default 'sconosciuto' not null;**

## Linguaggi per basi di dati

### SQL Esercizio 4.3

Dare le definizioni SQL delle tre tabelle FONDISTA(Nome, Nazione, Età); GAREGGIA(NomeFondista, NomeGara, Piazzamento); GARA(Nome, Luogo, Nazione, Lunghezza), rappresentando in particolare i vincoli di foreign key della tabella GAREGGIA.

**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 );

**Create table GAREGGIA (**

    NomeFondista character(20) references FONDISTA(Nome),  
     NomeGara character(20),  
     Piazzamento smallint,  
     primary key (NomeFondista, NomeGara),  
     foreign key (NomeGara) references GARA(Nome) );

## Linguaggi per basi di dati

### Soluzione Esercizio 4.3 Codice SQL in POSTGRES

**CREATE TABLE public.fondista (**

    nome character(20) COLLATE pg\_catalog."default" NOT NULL,  
     nazione character(30) COLLATE pg\_catalog."default",  
     "età" smallint,  
     CONSTRAINT fondista\_pkey PRIMARY KEY (nome)

);

**CREATE TABLE public.gara (**

    nome character(20) COLLATE pg\_catalog."default" NOT NULL,  
     luogo character(20) COLLATE pg\_catalog."default",  
     nazione character(20) COLLATE pg\_catalog."default",  
     lunghezza integer,  
     CONSTRAINT gara\_pkey PRIMARY KEY (nome)

);

## Linguaggi per basi di dati

### Soluzione Esercizio 4.3 Codice SQL in POSTGRES

```
CREATE TABLE public.gareggia (
    nomefondista character(20) COLLATE pg_catalog."default" NOT NULL,
    nomegara character(20) COLLATE pg_catalog."default" NOT NULL,
    piazzamento smallint,
    CONSTRAINT gareggia_pkey PRIMARY KEY (nomefondista, nomegara),
    CONSTRAINT gareggia_nomefondista_fkey FOREIGN KEY (nomefondista)
        REFERENCES public.fondista (nome) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT gareggia_nomegara_fkey FOREIGN KEY (nomegara)
        REFERENCES public.gara (nome) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
);
```

## Linguaggi per basi di dati

### SQL Soluzione Esercizio 4.4

*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.*

```
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
);
```

## Linguaggi per basi di dati

### SQL Esercizio 4.5

*Dato lo schema dell'esercizio precedente, in particolare*  
**IN TABELLA LIBRO foreign key (NomeAutore, CognomeAutore)**  
**references AUTORE(Nome, Cognome)**

**on delete cascade**

**on update set NULL**

*spiegare cosa può capitare con l'esecuzione dei seguenti comandi di aggiornamento:*

1) *delete from AUTORE where Cognome = 'Rossi'*

- ✓ 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.

## Linguaggi per basi di dati

### SQL Soluzione Esercizio 4.5

*Dato lo schema dell'esercizio precedente,*

**AUTORE (Nome, Cognome, DataNascita, Nazionalità)**

**LIBRO (TitoloLibro, NomeAutore, CognomeAutore, Lingua)**

*spiegare cosa può capitare con l'esecuzione dei seguenti comandi di aggiornamento:*

2) *update LIBRO set Nome= 'Umberto' where Cognome= 'Eco'*

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

## Linguaggi per basi di dati

### SQL Soluzione Esercizio 4.5

*Dato lo schema dell'esercizio precedente,*

**AUTORE (Nome, Cognome, DataNascita, Nazionalità)**

**LIBRO (TitoloLibro, NomeAutore, CognomeAutore, Lingua)**

*spiegare cosa può capitare con l'esecuzione dei seguenti comandi di aggiornamento:*

3) `insert into AUTORE(Nome,Cognome) values('Antonio','Bianchi')`

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

## Linguaggi per basi di dati

### SQL Soluzione Esercizio 4.5

*Dato lo schema dell'esercizio precedente,*

**AUTORE (Nome, Cognome, DataNascita, Nazionalità)**

**LIBRO (TitoloLibro, NomeAutore, CognomeAutore, Lingua)**

*spiegare cosa può capitare con l'esecuzione dei seguenti comandi di aggiornamento:*

4) `update AUTORE set Nome = 'Italo' where Cognome = 'Calvino'`

- ✓ **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.**

## Linguaggi per basi di dati

### SQL Soluzione Esercizio 4.6

Date le definizioni:

**create domain Dominio as integer default 10**

**create table Tabella (Attributo Dominio default 5)**

indicare cosa avviene in seguito ai comandi:

1. *alter table Tabella alter column Attributo drop default*
2. *alter domain Dominio drop default*
3. *drop domain Dominio*

**1. Il comando cancella il valore di default di Attributo. Il valore di default dopo il comando sarà quello impostato in Dominio, ossia 10.**

**2. Il comando cancella il valore di default di Dominio. Dopo l'operazione il valore di default sarà NULL**

**3. Il comando cancella l'intero dominio Domain. In Tabella il dominio di Attributo diventerà integer**

## Linguaggi per basi di dati

Date le definizioni:

**create domain D\_PROVA as integer default 20**

**create table T\_TEXT (A\_ESE D\_PROVA default 10)**

indicare cosa avviene in seguito ai comandi:

1. *alter table T\_TEXT alter column A\_ESE drop default*
2. *alter domain D\_PROVA drop default*
3. *drop domain D\_PROVA*

**1. Il comando cancella il valore di default di A\_ESE. Il valore di default dopo il comando sarà quello impostato in D\_PROVA, ossia 20.**

**2. Il comando cancella il valore di default del Dominio D\_PROVA. Dopo l'operazione il valore di default sarà NULL**

**3. Il comando cancella l'intero dominio D\_PROVA. In Tabella il dominio di Attributo diventerà integer ??? Soluzione del Libro di Testo.**

**EFFETTUARE UNA VERIFICA IN POSTGRES.**

## Linguaggi per basi di dati – Esercizio 2

**DATE 2 TABELLE DIPARTIMENTO (ESTERNA) E IMPIEGATO (INTERNA) RISPETTO AL VINCOLO REFERENZIALE IN TABELLA IMPIEGATO OSSERVARE IL COMPORTAMENTO DELLA POLICY ON DELETE SET NULL NEL CASO IN CUI L'ATTRIBUTO INTERESSATO SIA STATO DICHIARATO PER DEFAULT NOT NULL**

Creare una tabella **DIPARTIMENTO**

**CODICE** 4 caratteri, chiave primaria da definire con un **CONSTRAINT** malgrado sia possibile dichiararla primary key al livello di singolo attributo,

**NOME** al massimo 20 caratteri, chiave candidata ad essere anche primary key,

**DIP\_IND** al massimo 50 caratteri,

**DIP\_CITTA** al massimo 20 caratteri,

Creare una tabella **IMPIEGATO**

**MATRICOLA** 6 caratteri, chiave primaria al livello di singolo attributo,

**NOME** al massimo 20 caratteri,

**COGNOME** al massimo 20 caratteri,

**DIPART** al massimo 20 caratteri, non sono ammessi valori nulli. I valori ammessi sono solo quelli presenti nella chiave candidata **NOME** della Tabella **DIPARTIMENTO** - Policy di reazione: non permettere variazioni in caso di variazioni del valore di **NOME** in **DIPARTIMENTO** mentre in caso di cancellazione di una tupla in **DIPARTIMENTO**, inserire il valore NULL in tutte le tuple di **IMPIEGATO** coinvolte dalla cancellazione.

**UFFICIO** numerico di 3,

**STIPENDIO** numerico di 9 ed in assenza di valori inserire il valore zero,

**CITTA** al massimo 50 caratteri,

**NOME** e **COGNOME** chiave candidata ad essere anche primary key

## Linguaggi per basi di dati – Esercizio 2

**DATE 2 TABELLE DIPARTIMENTO (ESTERNA) E IMPIEGATO (INTERNA) RISPETTO AL VINCOLO REFERENZIALE IN TABELLA IMPIEGATO OSSERVARE IL COMPORTAMENTO DELLA POLICY ON DELETE SET NULL NEL CASO IN CUI L'ATTRIBUTO INTERESSATO SIA STATO DICHIARATO PER DEFAULT NOT NULL**

/\* Creare una tabella DIPARTIMENTO \*/

DROP TABLE IF EXISTS DIPARTIMENTO CASCADE;

CREATE TABLE DIPARTIMENTO (

CODICE CHAR(4) NOT NULL,

NOME VARCHAR(20) NOT NULL UNIQUE,

DIP\_IND varchar (50),

DIP\_CITTA varchar(20),

CONSTRAINT PK\_DIPARTIMENTO PRIMARY KEY (CODICE));

/\* Creare una tabella IMPIEGATO \*/

DROP TABLE IF EXISTS IMPIEGATO CASCADE;

CREATE TABLE IMPIEGATO (

MATRICOLA char(6) primary key,

NOME varchar(20),

COGNOME varchar (20),

DIPART varchar (20) NOT NULL,

UFFICIO numeric (3),

STIPENDIO numeric (9) default 0,

CITTA character varying(50), unique (cognome, nome),

CONSTRAINT FK\_DIPART foreign key (dipart) references DIPARTIMENTO(nome)  
on delete set null on update restrict );

## Linguaggi per basi di dati – Esercizio 2

**DATE 2 TABELLE DIPARTIMENTO (ESTERNA) E IMPIEGATO (INTERNA) RISPETTO AL VINCOLO REFERENZIALE IN TABELLA IMPIEGATO OSSERVARE IL COMPORTAMENTO DELLA POLICY ON DELETE SET NULL NEL CASO IN CUI L'ATTRIBUTO INTERESSATO SIA STATO DICHIARATO PER DEFAULT NOT NULL**

## /\* Inserire occorrenze in una tabella DIPARTIMENTO \*/

```

• INSERT INTO DIPARTIMENTO (CODICE, NOME, DIP_IND, DIP_CITTA)
VALUES ('AMMZ', 'Amministrazione', 'Via Tito Livio, 27', 'Milano' );
INSERT INTO DIPARTIMENTO (CODICE, NOME, DIP_IND, DIP_CITTA)
• VALUES ('PROD', 'Produzione', 'P.le Lavater, 3', 'Torino' );
INSERT INTO DIPARTIMENTO (CODICE, NOME, DIP_IND, DIP_CITTA)
• VALUES ('DIST', 'Distribuzione', 'Via Segre, 9', 'Roma' );
INSERT INTO DIPARTIMENTO (CODICE, NOME, DIP_IND, DIP_CITTA)
• VALUES ('DIRE', 'Direzione', 'Via Tito Livio, 27', 'Milano' );
INSERT INTO DIPARTIMENTO (CODICE, NOME, DIP_IND, DIP_CITTA)
• VALUES ('RICE', 'Ricerca', 'Via Venosa, 6', 'Milano' );

```

## Linguaggi per basi di dati – Esercizio 2

## /\* Inserire occorrenze in una tabella IMPIEGATO \*/

```

• INSERT INTO impiegato (matricola, nome, cognome, dipart, ufficio, stipendio, citta)
VALUES ('000001', 'Mario', 'Rossi', 'Amministrazione', 10, 45, 'Milano');
• INSERT INTO impiegato (matricola, nome, cognome, dipart, ufficio, stipendio, citta)
VALUES ('000002', 'Carlo', 'Bianchi', 'Produzione', 20, 36, 'Torino');
• INSERT INTO impiegato (matricola, nome, cognome, dipart, ufficio, stipendio, citta)
VALUES ('000003', 'Giovanni', 'Verdi', 'Amministrazione', 20, 40, 'Roma');
• INSERT INTO impiegato (matricola, nome, cognome, dipart, ufficio, stipendio, citta)
VALUES ('000004', 'Franco', 'Neri', 'Distribuzione', 16, 45, 'Napoli');
• INSERT INTO impiegato (matricola, nome, cognome, dipart, ufficio, stipendio, citta)
VALUES ('000005', 'Carlo', 'Rossi', 'Direzione', 14, 80, 'Milano');
• INSERT INTO impiegato (matricola, nome, cognome, dipart, ufficio, stipendio, citta)
VALUES ('000006', 'Lorenzo', 'Gialli', 'Direzione', 7, 73, 'Genova');
• INSERT INTO impiegato (matricola, nome, cognome, dipart, ufficio, stipendio, citta)
VALUES ('000007', 'Paola', 'Rosati', 'Amministrazione', 75, 40, 'Venezia');
• INSERT INTO impiegato (matricola, nome, cognome, dipart, ufficio, stipendio, citta)
VALUES ('000008', 'Marco', 'Franco', 'Produzione', 20, 46, 'Roma');

```



## Linguaggi per basi di dati – Esercizio 2

```
SELECT *
FROM IMPIEGATO;
```

Data Output Explain Messages Notifications

	matricola [PK] character (6)	nome character varying (20)	cognome character varying (20)	dipart character varying (20)	ufficio numeric (3)	stipendio numeric (9)	citta character varying (50)
1	000001	Mario	Rossi	Amministrazione	10	45	Milano
2	000002	Carlo	Bianchi	Produzione	20	36	Torino
3	000003	Giovanni	Verdi	Amministrazione	20	40	Roma
4	000004	Franco	Neri	Distribuzione	16	45	Napoli
5	000005	Carlo	Rossi	Direzione	14	80	Milano
6	000006	Lorenzo	Gialli	Direzione	7	73	Genova
7	000007	Paola	Rosati	Amministrazione	75	40	Venezia
8	000008	Marco	Franco	Produzione	20	46	Roma

## Linguaggi per basi di dati – Esercizio 2

**DATE 2 TABELLE DIPARTIMENTO (ESTERNA) E IMPIEGATO (INTERNA) RISPETTO AL VINCOLO REFERENZIALE IN TABELLA IMPIEGATO OSSERVARE IL COMPORTAMENTO DELLA POLICY ON DELETE SET NULL NEL CASO IN CUI L'ATTRIBUTO INTERESSATO SIA STATO DICHIARATO PER DEFAULT NOT NULL**

OSSERVARE COSA ACCADE QUANDO SI EFFETTUA LA CANCELLAZIONE DALLA TABELLA DIPARTIMENTO DELLA TUPLA CHE HA COME VALORE DEL CAMPO NOME *'Direzione'*

```
DELETE FROM DIPARTIMENTO
WHERE DIPARTIMENTO.NOME='Direzione';
```

```
SELECT *
FROM DIPARTIMENTO;
```

## Linguaggi per basi di dati – Esercizio 2

OSSERVARE COSA ACCADE QUANDO QUANDO CANCELLO DALLA TABELLA DIPARTIMENTO LA TUPLA CHE COME VALORE DEL CAMPO NOME 'Direzione'

```
DELETE FROM DIPARTIMENTO
WHERE DIPARTIMENTO.NOME='Direzione';
SELECT *
FROM DIPARTIMENTO;
```

**ERROR: ERROR: valori null nella colonna "dipart" violano il vincolo non-null** *DETAIL: La riga in errore contiene (000005, Carlo, Rossi, null, 14, 80, Milano).* **CONTEXT:** istruzione SQL "UPDATE ONLY "public"."impiegato" SET "dipart" = NULL WHERE \$1::pg\_catalog.text OPERATOR(pg\_catalog.=) "dipart"::pg\_catalog.text" SQL state: 23502

QUESTO PERCHE' **DIPART NON ACCETTA VALORI NON NULLI** MA LA POLITICA DI REAZIONE DEFINITA

CONSTRAINT FK\_DIPART foreign key (dipart) references DIPARTIMENTO(nome) **on delete set null on update restrict )**

IMPLICA INVECE CHE IL DBMS DEVE INSERIRE NULL (set null) IN DIPART DELLA TABELLA IMPIEGATO, QUANDO LA TUPLA ESTERNA CONTENENTE IL VALORE 'Direzione' IN DIPARTIMENTO VIENE CANCELLATA.

## Linguaggi per basi di dati – Esercizio 3

Creare un dominio denominato **DOMINIO\_VOTO** con riferimento al Dominio di Base **SMALLINT**. **DOMINIO\_VOTO**, deve garantire come valore di default il valore zero e non deve permettere valori  $\geq$  ZERO e  $\leq$  30.

Utilizzare il Dominio definito ovvero **DOMINIO\_VOTO** per definire l'attributo **VOTO** di una Tabella **STUDENTE** così articolata:

**matricola** 10 caratteri, *chiave primaria al livello di singolo attributo*,

**cognome** al massimo 20 caratteri,

**nome** al massimo 20 caratteri,

**voto** che assume come dominio di riferimento **DOMINIO VOTO** ma per default assume il valore 18

## Linguaggi per basi di dati – Esercizio 3

```
DROP DOMAIN IF EXISTS dominio_voto CASCADE;
DROP TABLE IF EXISTS studente CASCADE;
```

```
CREATE DOMAIN dominio_voto AS SMALLINT DEFAULT 0
CONSTRAINT check_voto check(VALUE >=0 and VALUE <=30);
```

```
CREATE TABLE studente(
    matricola CHAR(10) PRIMARY KEY,
    cognome VARCHAR(20),
    nome VARCHAR(20),
    voto dominio_voto DEFAULT 18 );
```

```
INSERT INTO studente(matricola, cognome, nome, voto)
VALUES('0612700232', 'D''Aniello', 'Giuseppe', 26);
```

```
INSERT INTO studente(matricola, cognome, nome, voto)
VALUES('0612700647', 'Rossi', 'Mario', 28);
```

## Linguaggi per basi di dati – Esercizio 3

LA TABELLA Studente è:

```
matricola CHAR(10) PRIMARY KEY,
cognome VARCHAR(20),
nome VARCHAR(20),
voto dominio_voto DEFAULT 18
```

```
CON DOMAIN dominio_voto AS SMALLINT DEFAULT 0
CONSTRAINT check_voto check(VALUE >=0 and VALUE <=30);
```

## 1) Effettuare una Query semplice

```
SELECT * FROM STUDENTE;
```

	matricola [PK] character (10)	cognome character varying (20)	nome character varying (20)	voto smallint
1	0612700232	D'Aniello	Giuseppe	26
2	0612700647	Rossi	Mario	28

## 2) Cancellare il DEFAULT 18 dalla colonna VOTO e verificare cosa succede con una query

## Linguaggi per basi di dati – Esercizio 3

IL DOMINIO è: dominio\_voto AS SMALLINT DEFAULT 0  
 CONSTRAINT check\_voto check(VALUE >=0 and VALUE <=30);

2) Cancellare il DEFAULT 18 dalla colonna VOTO,  
 ALTER TABLE studente ALTER COLUMN voto DROP DEFAULT;  
 INSERT INTO studente(matricola, nome, cognome)  
 VALUES('001234123', 'Verdi', 'Giuseppe');  
 SELECT \*  
 FROM STUDENTE;

	Data Output	Explain	Messages	Notifications
	matricola [PK] character (10)	cognome character varying (20)	nome character varying (20)	voto smallint
1	0612700232	D'Aniello	Giuseppe	26
2	0612700647	Rossi	Mario	28
3	001234123	Giuseppe	Verdi	0

VOTO ASSUME IL VALORE DI DEFAULT DI DOMINIO\_VOTO E NON IL VALORE 18 DEFAULT DELL'ATTRIBUTO VOTO

3) Cancellare il DEFAULT 0 DI DOMINIO\_VOTO

## Linguaggi per basi di dati – Esercizio 3

IL DOMINIO è: dominio\_voto AS SMALLINT DEFAULT 0  
 CONSTRAINT check\_voto check(VALUE >=0 and VALUE <=30);

3) Cancellare il DEFAULT 0 DI DOMINIO\_VOTO  
 ALTER DOMAIN dominio\_voto DROP DEFAULT;  
 INSERT INTO studente(matricola, nome, cognome)  
 VALUES('001123123', 'Atzeni', 'Paolo');  
 SELECT \*  
 FROM STUDENTE;

	Data Output	Explain	Messages	Notifications
	matricola [PK] character (10)	cognome character varying (20)	nome character varying (20)	voto smallint
1	0612700232	D'Aniello	Giuseppe	26
2	0612700647	Rossi	Mario	28
3	001234123	Giuseppe	Verdi	0
4	001123123	Paolo	Atzeni	[null]

VOTO ASSUME IL VALORE NULL, VALORE DI DEFAULT DI BASE

4) Cancellare il DOMINIO dominio\_voto

## Linguaggi per basi di dati – Esercizio 3

IL DOMINIO è: dominio\_voto AS SMALLINT DEFAULT 0  
 CONSTRAINT check\_voto check(VALUE >=0 and VALUE <=30);

4) Cancellare il DOMINIO dominio\_voto  
 DROP DOMAIN dominio\_voto;

*ERROR: ERRORE: non è possibile eliminare tipo dominio\_voto perché altri oggetti dipendono da esso DETAIL: colonna voto di tabella studente dipende da tipo dominio\_voto HINT: Usa DROP ... CASCADE per eliminare anche gli oggetti dipendenti. SQL state: 2BP01*

IL SISTEMA GENERA UN ERRORE IN QUANDO LA COLONNA VOTO DELLA TABELLA STUDENTE DIPENDE DA DOMINIO VOTO

5) Cancellare il DOMINIO dominio\_voto CON L'OPZIONE CASCADE

## Linguaggi per basi di dati – Esercizio 3

5) Cancellare il DOMINIO dominio\_voto CON L'OPZIONE CASCADE  
 DROP DOMAIN dominio\_voto CASCADE ;  
 INSERT INTO studente(matricola, nome, cognome)  
 VALUES('0044545123', 'Ceri', 'Stefano');  
 INSERT INTO studente(matricola, nome, cognome)  
 VALUES('12312784', 'Fraternali', 'Piero');  
 Select \* from impiegati;

	matricola [PK] character (10)	cognome character varying (20)	nome character varying (20)
1	0612700232	D'Aniello	Giuseppe
2	0612700647	Rossi	Mario
3	001234123	Giuseppe	Verdi
4	001123123	Paolo	Atzeni
5	0044545123	Stefano	Ceri
6	12312784	Piero	Fraternali

COMANDO ESEGUITO IN POSTGRES - SI OSSERVI CHE LA COLONNA VOTO E' STATA ELIMINATA DALLA TABELLA STUDENTI IN QUANDO IL SISTEMA HA ESEGUITO IL DROP CASCADE DI DOMINIO\_VOTO E DI TUTTI GLI OGGETTI COLLEGATI – SI CONFRONTI QUESTA AZIONE CON QUANTO DESCRITTO A PAG. 106 CAPITOLO 4 DEL LIBRO PARAGRAFO 4.2.8. IN MERITO AL DROP: ESEMPIO StringaLunga

## Linguaggi per basi di dati – Esercizio 4

**CREARE UNA TABELLA PERSONE, COSI ARTICOLATA:**

nome al massimo 20 caratteri e chiave primaria,  
 reddito integer,  
 eta smallint,  
 sesso char.

**CONTROLLARE LA LUNGHEZZA DI CHAR QUANDO NON SI SPECIFICA IL NUMERO DI CARATTERI.****CREARE UNA TABELLA GENITORI, COSI ARTICOLATA:**

figlio al massimo 20 caratteri, attributo in chiave primaria  
 genitore al massimo 20 caratteri, attributo in chiave primaria  
 Inoltre dobbiamo essere CERTI che se inseriamo un INDIVIDUO nella tabella GENITORI, conosciamo il GENITORE e soprattutto all'atto dell'inserimento il suo COGNOME e quello del GENITORE devono già essere presenti nella tabella PERSONE.

## Linguaggi per basi di dati – Esercizio 4

Definire la POLICY di reazione al fine di garantire l'integrità referenziale, se una riga di PERSONE viene CANCELLATA (ON DELETE) oppure MODIFICATA (ON UPDATE), per entrambe è possibile scegliere tra:

**SET DEFAULT** (*all'attributo referenziante - tabella interna - viene assegnato il valore di default al posto del valore modificato nella tabella referenziata – tabella esterna*);

**CASCADE** (*partendo dal valore della tabella referenziata – tabella esterna, aggiorna i valori nella referenziante - tabella interna*);

**SET NULL** (*all'attributo referenziante - tabella interna - viene assegnato il valore di NULL al posto del valore modificato nella tabella referenziata – tabella esterna*);

**NO ACTION** (*se l'azione sia essa modifica o cancellazione viola i vincoli di Foreign Key non viene permessa*).

Linguaggi per basi di dati – Esercizio 4

```
DROP TABLE IF EXISTS persone cascade;  
DROP TABLE IF EXISTS genitori cascade;
```

```
CREATE TABLE persone(  
    nome varchar(20) PRIMARY KEY,  
    reddito integer,  
    eta smallint,  
    sesso char  
);
```

```
CREATE TABLE genitori(  
    figlio varchar(20),  
    genitore varchar(20),  
    PRIMARY KEY (figlio, genitore),  
    FOREIGN KEY (figlio) REFERENCES persone(nome),  
    FOREIGN KEY (genitore) REFERENCES persone(nome)  
);
```

Linguaggi per basi di dati – Esercizio 4

#### POPOLAMENTO TABELLA PERSONE

```
INSERT INTO persone(nome, reddito, eta, sesso)  
VALUES('Mario', 15, 80, 'M');  
INSERT INTO persone(nome, reddito, eta, sesso)  
VALUES('Carlo', 25, 24, 'M');  
INSERT INTO persone(nome, reddito, eta, sesso)  
VALUES('Giuseppe', 30, 45, 'M');  
INSERT INTO persone(nome, reddito, eta, sesso)  
VALUES('Maria', 76, 43, 'F');  
INSERT INTO persone(nome, reddito, eta, sesso)  
VALUES('Gianni', 60, 50, 'M');  
INSERT INTO persone(nome, reddito, eta, sesso)  
VALUES('Francesca', 18, 26, 'F');  
INSERT INTO persone(nome, reddito, eta, sesso)  
VALUES('Paola', 45, 60, 'F');  
INSERT INTO persone(nome, reddito, eta, sesso)  
VALUES('Marco', 80, 35, 'M');
```

Linguaggi per basi di dati – Esercizio 4

### POPOLAMENTO TABELLA GENITORI

```
INSERT INTO genitori(figlio, genitore) VALUES('Paola', 'Mario');
INSERT INTO genitori(figlio, genitore) VALUES('Marco', 'Paola');
INSERT INTO genitori(figlio, genitore) VALUES('Carlo', 'Gianni');
INSERT INTO genitori(figlio, genitore) VALUES('Carlo', 'Maria');
INSERT INTO genitori(figlio, genitore) VALUES('Francesca', 'Giuseppe');
INSERT INTO genitori(figlio, genitore) VALUES('Marco', 'Giuseppe');
```

ESERCITARSI CON DIVERSE TIPOLOGIE DI SELECT DI BASE, TRA CUI QUELLA DI FARE ELENCHI ORDINATI IN VARI MODI (Order By attributo asc e desc).

Linguaggi per basi di dati – Esercizio 4

### DISCUTERE LE CRITICITA' E VERIFICARE IN AMBIENTE POSTGRES NELL'IPOTESI DI UNA DEFINIZIONE DEL SEGUENTE TIPO:

```
CREATE TABLE persone(
    nome varchar(20) PRIMARY KEY,
    reddito integer,
    eta smallint,
    sesso char
);
CREATE TABLE genitori(
    figlio varchar(20),
    genitore varchar(20),
    PRIMARY KEY (figlio, genitore),
    FOREIGN KEY (figlio) REFERENCES persone(nome),
    ON DELETE SET DEFAULT ON UPDATE CASCADE;
    FOREIGN KEY (genitore) REFERENCES persone(nome)
    ON DELETE NO ACTION ON UPDATE NO ACTION;
```



## Linguaggi per basi di dati – Esercizio 4

**DOMANDE (Errori tipici in seduta di esame)**

- 1) Una Tabella può avere due PRIMARY KEY ?
- 2) Posso a livello di attributo definire una PRIMARY KEY composta da due attributi ?
- 3) Posso nel CONSTRAINT così come faccio per PRIMARY KEY e UNIQUE indicare un CONSTRAINT NULL, NOT NULL o DEFAULT su uno o più attributi ? PROVARE IN POSTGRES
- 4) Premesso che è possibile utilizzare il CHECK a livello di definizione di un dominio, posso fare lo stesso a livello di Tabella su un attributo ? PROVARE IN POSTGRES
- 5) Perché quando realizzo il legame tra un attributo della tabella corrente (interna) oltre alla Tabella è corretto indicare anche l'attributo della tabella (esterna) e non solo la Tabella (nell'ipotesi che il sistema RDBMS accetti questa scorciatoia) PROVARE IN POSTGRES
- 6) Posso creare un legame tra un attributo della tabella corrente (interna) e un attributo qualsiasi di un'altra tabella (esterna) PROVARE IN POSTGRES

## Linguaggi per basi di dati – Esercizio 4

- 7) Date due Tabelle Interna ed Esterna oggetto di un vincolo di integrità referenziale cosa accade se cancello una riga della tabella Interna ? **DISCUTERE IN PRESENZA DI VARIE POLICY DI CANCELLAZIONE E MODIFICA DI RIGHE NELLA TABELLA INTERNA. ESEGUIRE SIMULAZIONI IN POSTGRES**
- 8) date due Tabelle Interna ed Esterna la violazione di un vincolo di integrità referenziale può avvenire a livello di tabella interna (inserimento, modifica) e a livello di tabella esterna (modifica o cancellazione). Cosa accade a livello di Tabella Interna QUANDO CANCELLO una Tupla e cosa accade a livello di tabella esterna QUANDO INSERISCO una Tupla .
- 9) Individuare gli errori nella seguente Tabella e discutere su cosa accade in termini di azioni di inserimento, modifica e cancellazione con una tabella impiegato così definita che referencia una Tabella Ufficio. PROVARE AD UTILIZZARE POSTGRES.  
 Create Impiegato (  
     matricola char() primary key,  
     dipart varchar (20) DEFAULT NOT NULL,  
     CONSTRAINT FK\_DIPART foreign key (dipart) references ufficio(nome)  
     on delete set null on update restrict);  
 Dopo aver effettuate le eventuali correzioni cosa accade allorché viene cancella una riga della tabella esterna ufficio?
- 10) Indicare, se vi sono le differenza tra DROP Dipartimento e DELETE Dipartimento