Corso Basi di Dati

Dott.ssa Elisa Quintarelli

SQL

- Structured Query Language
- SQL è stato definito nel 1973 ed è oggi il linguaggio universale dei sistemi relazionali
- È un linguaggio con varie funzionalità:
  - Definizione di basi di dati (Data Definition Language DDL)
  - Linguaggio per modificare dati e interrogare (Data Manipulation Language DML)
  - Linguaggio per stabilire controlli sull'uso dei dati

Definizione Dati in SQL

- Istruzione CREATE TABLE:
  - Definisce uno schema di relazione e ne crea un'istanza vuota
  - Specifica attributi, domini, vincoli

CREATE TABLE NomeTabella
(Attributo Tipo [Valore Default][Vincolo Attributo]
{, Attributo Tipo [Valore Default][Vincolo Attributo]}
{, Vincolo Tabella})

Domini

- Domini elementari (predefiniti):
  - Carattere: singoli caratteri o stringhe anche di lunghezza variabile
  - Bit: singoli booleani (flag) o stringhe
  - Numerici, esatti e approssimati
  - Data, ora, intervalli di tempo

Dominio CARATTERE

- Permette di rappresentare singoli caratteri oppure stringhe.
- La lunghezza delle stringhe può essere fissa o variabile.

character [varying][(Lunghezza)]

character → CHAR
character varying(20) → VARCHAR(20)

Dominio BIT

- Tipicamente usato per rappresentare attributi, detti FLAG, che specificano se l'oggetto rappresentato da una tupla possiede o meno quella proprietà.
- Si può anche definire un dominio "stringa di bit".

bit [varying][(Lunghezza)]

Dominio TIPI NUMERICI ESATTI

- Permette di rappresentare valori interi o valori decimali in virgola fissa.
- SQL mette a disposizione 4 diversi tipi:
  - NUMERIC
  - DECIMAL

  } Numeri in base decimale

  - INTEGER
  - SMALLINT

  } Se non interessa avere una rappresentazione precisa della parte frazionaria

numeric [(Precisione [, Scala])]
decimal [(Precisione [, Scala])]

Numeric(4,2) ⇒ 4 cifre significative, 2 cifre dopo la virgola

Dominio TIPI NUMERICI APPROSSIMATI

- Permette di rappresentare valori numerici approssimati mediante una rappresentazione in virgola mobile.
- SQL mette a disposizione 3 diversi tipi:
  - FLOAT
  - DOUBLE PRECISION
  - REAL

float [(Precisione)]

Dominio DATA e ORA

- Permette di rappresentare istanti di tempo.
  - DATE (year, month, day)
  - TIME (hour, minute, second)
  - TIMESTAMP

time [(Precisione)][with time zone]
timestamp [(Precisione)][with time zone]

CREATE TABLE: Esempio

```
CREATE TABLE Impiegato
  (   Matricola  CHAR(6),
      Nome       VARCHAR(20),
      Cognome    VARCHAR(20),
      Qualifica  VARCHAR(20),
      Stipendio  FLOAT      )
```

Vincoli intrarelazionali

Proprietà che devono essere soddisfatte da ogni istanza della base di dati

Il soddisfacimento è definito rispetto a singole relazioni della base di dati

- NOT NULL
- UNIQUE definisce chiavi
- PRIMARY KEY: chiave primaria (una sola, implica NOT NULL)
- CHECK

NOT NULL

- Il valore nullo non è ammesso come valore dell'attributo.
  - Il valore dell'attributo deve essere specificato in fase di inserimento.

Nome VARCHAR(20) NOT NULL

UNIQUE

- Impone che i valori di un attributo (o di un insieme di attributi) siano una superchiave, quindi righe differenti della tabella non possono avere gli stessi valori.
- Si può definire su
  - Un solo attributo
  - Su un insieme di attributi

```
Matricola   CHAR(6) UNIQUE
Nome        VARCHAR(20),
Cognome     VARCHAR(20),
UNIQUE(Nome,Cognome)
```

Su più attributi: attenzione!

```
Nome        VARCHAR(20) NOT NULL,
Cognome     VARCHAR(20) NOT NULL,
UNIQUE(Nome,Cognome)
```

Impone che non ci siano due righe che abbiano uguali sia il nome che il cognome

```
Nome        VARCHAR(20) NOT NULL UNIQUE,
Cognome     VARCHAR(20) NOT NULL UNIQUE,
```

Impone che non ci siano due righe che abbiano lo stesso nome o lo stesso cognome

PRIMARY KEY

- Specifica la chiave primaria della relazione
  - Si usa una sola volta per tabella
  - Implica una definizione di NOT NULL
- Due forme:
  - Nella definizione di un attributo, se forma da solo la chiave
  - Come elemento separato

```
Matricola      CHAR(6) PRIMARY KEY

Nome           VARCHAR(20),
Cognome        VARCHAR(20),
PRIMARY KEY(Nome,Cognome)
```

CREATE TABLE: Esempio

```
CREATE TABLE Impiegato
   (   Matricola    CHAR(6)  PRIMARY KEY,
       Nome         VARCHAR(20) NOT NULL,
       Cognome      VARCHAR(20) NOT NULL,
       Qualifica    VARCHAR(20),
       Stipendio    FLOAT  DEFAULT 0.0,
       UNIQUE(Cognome, Nome)
```

Il valore che deve assumere l'attributo quando viene inserita una riga nella tabella senza che sia specificato un valore per l'attributo stesso. Se non specificato, si assume come valore di default null

CREATE TABLE: esempio Check

```
CREATE TABLE Impiegato
  (   Matricola  CHAR(6)  PRIMARY KEY,
      Nome       VARCHAR(20) NOT NULL,
      Cognome    VARCHAR(20) NOT NULL,
      Qualifica  VARCHAR(20),
      Stipendio  FLOAT  DEFAULT 100.0,
      UNIQUE(Cognome, Nome),
      CHECK (Stipendio>= 100)    )
```

INSERT

- Come popolare una tabella (inserimento righe):

```
INSERT INTO
  NomeTabella[(ElencoAttributi)] VALUES
  (Elenco di Valori);

INSERT INTO
  Impiegato(Matricola,Nome,Cognome)
  VALUES ('A00001', 'Mario', 'Rossi');
```

Vincoli interrelazionali

Vincoli che coinvolgono più relazioni.

I più significativi sono i vincoli di integrità referenziale o vincoli di riferimento.

- In SQL il costrutto per la definizione dei vincoli di riferimento è dato dal vincolo FOREIGN KEY (chiave esterna).

FOREIGN KEY

Crea un legame tra i valori dell'attributo della tabella corrente (*interna*) e i valori dell'attributo di un'altra tabella (*esterna*).

- Impone che per ogni riga della tabella interna il valore dell'attributo, se diverso dal valore nullo, sia presente tra i valori di un attributo della tabella esterna.
- ATTENZIONE: L'attributo della tabella esterna a cui si fa riferimento deve essere soggetto a vincolo UNIQUE (o PRIMARY KEY).

FOREIGN KEY

- Nel vincolo possono essere coinvolti più attributi, ad esempio quando la chiave della tabella esterna è costituita da un insieme di attributi.
  - Si confrontano insiemi di valori invece che singoli valori.
- Può essere definito in due modi:
  - Uso costrutto REFERENCES
  - Uso costrutto FOREIGN KEY

USO REFERENCES

- Si usa il costrutto REFERENCES quando il vincolo è definito su un unico attributo.
- Con REFERENCES (nella tabella interna) si specificano la tabella esterna e l'attributo della tabella esterna con il quale l'attributo della tabella interna deve essere legato.

CREATE TABLE: REFERENCES

Tabella Interna

```
CREATE TABLE Impiegato(
   Matricola       CHAR(6)  PRIMARY KEY,
   Nome            VARCHAR(20) NOT NULL,
   Cognome         VARCHAR(20) NOT NULL,
   NomeDipartimentoVARCHAR(15)
       REFERENCES
   Dipartimento(NomeDip));
```

Tabella Esterna

Attributo Chiave

CREATE TABLE: REFERENCES

Tabella Esterna

Vincolo di UNIQUE o PRIMARY KEY

```
CREATE TABLE Dipartimento(
   NomeDip         VARCHAR(15)
       PRIMARY KEY,
   Sede       VARCHAR(20)      NOT NULL,
   Telefono   VARCHAR(15));
```

ESEMPIO

Tabella Interna: IMPIEGATO

Matricola	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	Vendite

Tabella Esterna: DIPARTIMENTO

Vincolo UNIQUE o PRIMARY KEY

NomeDip	Sede	Telefono
Acquisti	Verona	045/8008080
Vendite	Milano	02/8007070

USO FOREIGN KEY

- Si usa il costrutto FOREIGN KEY quando il vincolo è definito su un insieme di attributi.
- Con FOREIGN KEY (nella tabella interna) si elencano gli attributi della tabella interna coinvolti nel legame e con REFERENCES si specificano la tabella esterna e gli attributi della tabella esterna con il quale gli attributi della tabella interna devono essere legati.

CREATE TABLE: FOREIGN KEY E REFERENCES

Tabella Interna

```
CREATE TABLE Impiegato(
   Matricola  CHAR(6)          PRIMARY KEY,
   Nome       VARCHAR(20)      NOT NULL,
   Cognome VARCHAR(20)         NOT NULL,
   NomeDipartimento      VARCHAR(15)
        REFERENCES Dipartimento(NomeDip),
   FOREIGN KEY(Nome,Cognome)
        REFERENCES Anagrafica(Nome,Cognome));
```

Tabella Esterna

Attributi Chiave (ordinati)

CREATE TABLE: FOREIGN KEY E REFERENCES

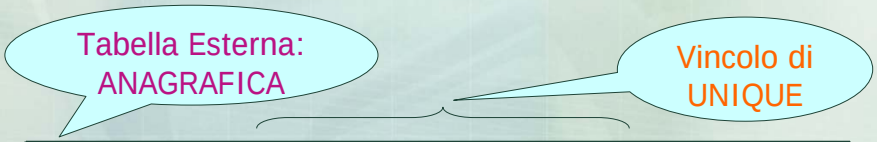
Tabella Esterna

```
CREATE TABLE Anagrafica(
   CodFisc      CHAR(11)        PRIMARY KEY,
   Nome         VARCHAR(20)     NOT NULL,
   Cognome      VARCHAR(20)     NOT NULL,
   Indirizzo    VARCHAR(30),
   UNIQUE(Nome,Cognome)
);
```

Vincolo di UNIQUE

ESEMPIO

Matricola	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	Vendite



CodFisc	Nome	Cognome	Indirizzo
RSSMRA…	Mario	Rossi	Via X
VRDPAO…	Paolo	Verdi	Via Y

Violazione vincoli e politiche

- È possibile associare ad un vincolo di integrità referenziale una politica di reazione alle violazioni
  - SQL permette di decidere quale reazione adottare
- Per gli altri vincoli, in presenza di violazione, l'aggiornamento viene rifiutato.

Violare i vincoli operando sulla tabella interna

- Si possono introdurre violazioni modificando il contenuto della tabella interna solo in due modi:
  - Modificando il valore dell'attributo referente
  - Inserendo una nuova riga
- Per queste operazioni SQL non offre nessun supporto:
  - Le operazioni vengono semplicemente impedite

ESEMPIO

Tabella Interna: IMPIEGATO

Matricola	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	Vendite

A00003	Marco	Bianchi	Marketing

Tentativo di inserimento che causa VIOLAZIONE!!!

Tabella Esterna: DIPARTIMENTO

NomeDip	Sede	Telefono
Acquisti	Verona	045/8008080
Vendite	Milano	02/8007070

ESEMPIO

Tabella Interna: IMPIEGATO

Matricola	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	Vendite
~~A00003~~	~~Marco~~	~~Bianchi~~	~~Marketing~~

L'inserimento viene impedito

Matricola	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	Vendite

Violare i vincoli operando sulla tabella esterna

- Diverse alternative per rispondere a violazioni generate da modifiche sulla tabella esterna (o tabella Master).
- La tabella interna (o tabella Slave) deve adeguarsi alle modifiche che avvengono sulla tabella Master.
- Le violazioni possono avvenire per:
  - Modifiche dell'attributo riferito (MA)
  - Cancellazione righe dalla tabella Master (CR).

Politiche di reazione per modifica attributo riferito

- **Cascade**: il nuovo valore dell'attributo della tabella esterna viene riportato su tutte le corrispondenti righe della tabella interna.
- Esempio: modifica di un valore dell'attributo NomeDip nella tabella DIPARTIMENTO.
  - DIPARTIMENTO: Da Vendite a VenditeMI
  - IMPIEGATO: Da Vendite a VenditeMI

CASCADE (MA)

Tabella Esterna: DIPARTIMENTO

Tabella Interna: IMPIEGATO

NomeDip	Sede	Telefono
Acquisti	Verona	045/8008080
Vendite	Milano	02/8007070

NomeDip	Sede	Telefono
Acquisti	Verona	045/8008080
VenditeMI	Milano	02/8007070

Matricola	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	Vendite

Matricola	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	VenditeMI

CASCADE (MA): risultato

NomeDip	Sede	Telefono
Acquisti	Verona	045/8008080
VenditeMI	Milano	02/8007070

Tabella Esterna: DIPARTIMENTO

Tabella Interna: IMPIEGATO

Matricola	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	VenditeMI

Politiche di reazione per modifica attributo riferito

- **Set null**: all'attributo referente (tabella interna) viene assegnato valore nullo al posto del valore modificato nella tabella esterna.
- Esempio: modifica di un valore dell'attributo NomeDip nella tabella DIPARTIMENTO.
  - DIPARTIMENTO: Da Vendite a VenditeMI
  - IMPIEGATO: Da Vendite a NULL

SET NULL (MA)

Tabella Esterna: DIPARTIMENTO

Tabella Interna: IMPIEGATO

NomeDip	Sede	Telefono
Acquisti	Verona	045/8008080
Vendite	Milano	02/8007070

NomeDip	Sede	Telefono
Acquisti	Verona	045/8008080
VenditeMI	Milano	02/8007070

Matricola	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	Vendite

Matricola	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	NULL

Politiche di reazione per modifica attributo riferito

- **Set default**: all'attributo referente viene assegnato un valore di default al posto del valore modificato nella tabella esterna.
- Esempio: modifica di un valore dell'attributo NomeDip nella tabella DIPARTIMENTO supponendo che il valore di default sia DipVendite.
  - DIPARTIMENTO: Da Vendite a VenditeMI
  - IMPIEGATO: Da Vendite a DipVendite

SET DEFAULT (MA)

Tabella Esterna: DIPARTIMENTO

Tabella Interna: IMPIEGATO

NomeDip	Sede	Telefono
Acquisti	Verona	045/8008080
DipVendite	Roma	06/8003838
Vendite	Milano	02/8007070

NomeDip	Sede	Telefono
Acquisti	Verona	045/8008080
DipVendite	Roma	06/8003838
VenditeMI	Milano	02/8007070

Matricola	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	Vendite

Matricola	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	DipVendite

Politiche di reazione per modifica attributo riferito

- **No action**: viene impedita la modifica della tabella esterna.
- Esempio: modifica di un valore dell'attributo NomeDip nella tabella DIPARTIMENTO.
  - DIPARTIMENTO: Da Vendite a VenditeMI (Viene Impedito)

NO ACTION (MA)

Tabella Esterna: DIPARTIMENTO (rimane immutata)

Tabella Interna: IMPIEGATO

NomeDip	Sede	Telefono
Acquisti	Verona	045/8008080
Vendite	Milano	02/8007070

NomeDip	Sede	Telefono
Acquisti	Verona	045/8008080
~~VenditeMI~~	~~Milano~~	~~02/8007070~~

Matricola	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	Vendite

Matricola	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	Vendite

Politiche di reazione per cancellazione riga tabella esterna

- SQL mette a disposizione le stesse politiche di reazione:
  - **Cascade**: tutte le righe della tabella interna corrispondenti alla riga cancellata vengono cancellate.
  - **Set null**: all'attributo referente viene assegnato il valore nullo al posto del valore presente nella riga cancellata dalla tabella esterna.
  - **Set default**: all'attributo referente viene assegnato un valore di default.
  - **No action**: viene impedita la cancellazione.

Politiche di reazione per cancellazione attributo riferito

- **Cascade**: tutte le righe della tabella interna corrispondenti alla riga cancellata vengono cancellate.
- Esempio: cancellazione dalla tabella DIPARTIMENTO della riga che ha NomeDip uguale a Vendite.

CASCADE (CD)

Tabella Esterna: DIPARTIMENTO

Tabella Interna: IMPIEGATO

NomeDip	Sede	Telefono
Acquisti	Verona	045/8008080
Vendite	Milano	02/8007070

NomeDip	Sede	Telefono
Acquisti	Verona	045/8008080

Matricola	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	Vendite

Matricola	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti

Politiche di reazione per cancellazione attributo riferito

- **Set null**: all'attributo referente viene assegnato il valore nullo al posto del valore presente nella riga cancellata dalla tabella esterna.
- Esempio: cancellazione dalla tabella DIPARTIMENTO della riga che ha NomeDip uguale a Vendite.

SET NULL (CR)

Tabella Esterna: DIPARTIMENTO

Tabella Interna: IMPIEGATO

NomeDip	Sede	Telefono
Acquisti	Verona	045/8008080
Vendite	Milano	02/8007070

NomeDip	Sede	Telefono
Acquisti	Verona	045/8008080

Matricola	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	Vendite

Matricola	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	NULL

Politiche di reazione per cancellazione attributo riferito

- **Set default**: all'attributo referente viene assegnato valore di default.
- Esempio: cancellazione dalla tabella DIPARTIMENTO della riga che ha NomeDip uguale a Vendite supponendo che il valore di default sia DipVendite.
- Attenzione: il valore di default deve essere presente nella tabella esterna.

SET DEFAULT (CD)

NomeDip	Sede	Telefono
Acquisti	Verona	045/8008080
DipVendite	Roma	06/8003838
Vendite	Milano	02/8007070

NomeDip	Sede	Telefono
Acquisti	Verona	045/8008080
DipVendite	Roma	06/8003838

Matricola	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	Vendite

Matricola	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	DipVendite

Politiche di reazione per cancellazione attributo riferito

- **No action**: viene impedita la cancellazione nella tabella Esterna
- Esempio: cancellazione dalla tabella DIPARTIMENTO della riga che ha NomeDip uguale a Vendite (viene impedita).

NO ACTION (CR)

Tabella Esterna: DIPARTIMENTO (rimane invariata)

Tabella Interna: IMPIEGATO

NomeDip	Sede	Telefono
Acquisti	Verona	045/8008080
Vendite	Milano	02/8007070

NomeDip	Sede	Telefono
Acquisti	Verona	045/8008080

Matricola	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	Vendite

Matricola	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	Vendite

Vincoli di integrità: sommario

- Vincoli su attributi (controlla)
  - Vincolo Attributo:=
    [NOT NULL [UNIQUE]] | [CHECK (Condizione)]
    [REFERENCES Tabella [(Attributo {, Attributo})]]
    [ON {DELETE|UPDATE} {NO ACTION | CASCADE |
    SET NULL | SET DEFAULT}]
- Vincoli su tabella
  - Vincolo Tabella:= UNIQUE(Attributo {, Attributo})
    | CHECK(Condizione) |
    | PRIMARY KEY [Nome] Attributo {, Attributo})
    | FOREIGN KEY [Nome] Attributo {, Attributo})
    REFERENCES Tabella [(Attributo {, Attributo})]
    [ON {DELETE|UPDATE} {NO ACTION | CASCADE |
    SET NULL | SET DEFAULT}]

CREATE TABLE: esempio

```
CREATE TABLE Impiegato(
    Matricola           CHAR(6)          PRIMARY
    KEY,
    Nome                VARCHAR(20)      NOT NULL,
    Cognome             VARCHAR(20)      NOT NULL,
    NomeDipartimento VARCHAR(15)
        REFERENCES Dipartimento(NomeDip),
    FOREIGN KEY(Nome,Cognome)
        REFERENCES Anagrafica(Nome,Cognome)
    ON DELETE CASCADE
    ON UPDATE NO ACTION);
```

Esempi esercizi DDL (temi d'esame)

Si consideri il seguente schema di base di dati di una casa editrice:

bozza (id., n°_capitolo, ISBN_libro, c.f._correttore, data_invio, data_restit.)

capitolo(n°_capitolo, titolo, ISBN_libro, n°_pagine)

libro(ISBN, titolo, autore, n°_pagine, data_prevista)

correttore(c.f., nome, n°_telefono)

Esempi esercizi DDL (temi d'esame)

Si consideri il seguente schema di base di dati di una casa editrice:

bozza (id., n°_capitolo, ISBN_libro, c.f._correttore, data_invio, data_restit.)

capitolo(n°_capitolo, titolo, ISBN_libro, n°_pagine)

libro(ISBN, titolo, autore, n°_pagine, data_prevista)

correttore(c.f., nome, n°_telefono)

Esercizi DDL

```
create table BOZZA
(  id          char(6)  primary key,
   n_capitolo  smallint,
   ISBN_libro  char(10),
   c.f._correttore char(16),
   data_invio    date not null,
  data_restit.   date,
   foreign key (ISBN_libro,n_capitolo)
      references CAPITOLO
  (ISBN_libro,n_capitolo)
          on delete cascade
           on update cascade)
```

Esercizi DDL

```
create table CAPITOLO
(  n_capitolo   smallint,
   titolo    varchar(30) not null,
   ISBN_libro   char(10),
   n_pagine     int not null,
   primary key(n_capitolo,ISBN_libro),
   foreign key ISBN_libro
      references LIBRO(ISBN)
                 on delete cascade
                 on update cascade)
```

Esercizi DDL

Si consideri il seguente schema di base di dati che vuole tenere traccia delle rappresentazioni di un gruppo di compagnie teatrali:

COMPAGNIA (nome, cfdirettore, città_sede)

TEATRO (id_teatro, nome, città, numero di telefono)

CALENDARIO (id_teatro, giorno, titolo, compagnia, ora_inizio, prezzo_biglietto)

SPETTACOLO (titolo, compagnia, genere, durata)

PERSONA (CF, Nome, Cognome, datadinascita, cittànascita, telefono)

Esercizi DDL

Si consideri il seguente schema di base di dati che vuole tenere traccia delle rappresentazioni di un gruppo di compagnie teatrali:

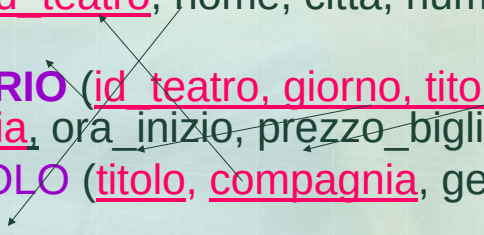
COMPAGNIA (nome, cfdirettore, città_sede)

TEATRO (id_teatro, nome, città, numero di telefono)

CALENDARIO (id_teatro, giorno, titolo, compagnia, ora_inizio, prezzo_biglietto)

SPETTACOLO (titolo, compagnia, genere, durata)

PERSONA (CF, Nome, Cognome, datadinascita, cittànascita, telefono)



Esercizi DDL

```
create table TEATRO
( id_teatro    char(6) primary key,
  nome         varchar (20),
  città        varchar(15),
 numero_telefono varchar(15),
);
```

Esercizi DDL

```
create table CALENDARIO
( id_teatro    char(6) references TEATRO (id_teatro)
                   on delete cascade
                   on update cascade,
  giorno       date,
  titolo       varchar(20),
  compagnia    varchar(20),
  ora_inizio   time,
  prezzo_biglietto integer,
  primary key (id_teatro, giorno, titolo, compagnia)
  foreign key (titolo, compagnia)
      references SPETTACOLO (titolo, compagnia)
                  on delete cascade
                  on update cascade);
```

Esercizi DDL

Si consideri il seguente schema di base di dati che vuole tenere traccia dei dati di un campionato di pallacanestro (non vengono memorizzate informazioni di tipo storico):

SQUADRA (Nome, Città, Logo, NomeAllenatore, CognomeAllenatore)
PARTITA (Giornata, SquadraCasa, SquadraOspite, PuntiCasa, PuntiOspite)
CLASSIFICA (Giornata, Squadra, Punti)
GIOCATORE(NomeGiocatore, CognomeGiocatore, NomeSquadra, Ruolo, Nazionalità)

Esercizi DDL

Si consideri il seguente schema di base di dati che vuole tenere traccia dei dati di un campionato di pallacanestro (non vengono memorizzate informazioni di tipo storico):

SQUADRA (Nome, Città, Logo, NomeAllenatore, CognomeAllenatore)
PARTITA (Giornata, SquadraCasa, SquadraOspite, PuntiCasa, PuntiOspite)
CLASSIFICA (Giornata, Squadra, Punti)
GIOCATORE(NomeGiocatore, CognomeGiocatore, NomeSquadra, Ruolo, Nazionalità)

Esercizi DDL

```
create table Squadra
(  NomeSquadra      varchar(30)
  primary key,
  Città             varchar(30),
  Logo              varchar(30),
  NomeAllenatore    varchar (30),
  CognomeAllenatore varchar(30)
    )
```

Esercizi DDL

```
create table Partita
(  Giornata    integer,
   SquadraCasa varchar(30) references
            Squadra (NomeSquadra)
                  on update cascade
                  on delete no action,
  SquadraOspite varchar(30) references
            Squadra (NomeSquadra)
                  on update cascade
                  on delete no action,
  PuntiCasa       integer,
  PuntiOspite     integer,
  primary key(Giornata, SquadraCasa)
)
```