

Soluzione DDL ed Algebra Relazionale

Banca del Tempo

a) Istruzioni DDL (realizzate con Toad Data Modeler)

```
-- CREATE TABLES SECTION -----
```

```
-- Table Zona
```

```
CREATE TABLE Zona(  
  ID Integer NOT NULL,  
  Descrizione Varchar2(25 CHAR) NOT NULL,  
  Immagine Blob NOT NULL  
)
```

```
-- Add keys for table Zona
```

```
ALTER TABLE Zona ADD CONSTRAINT PKDitta PRIMARY KEY (ID)
```

```
-- Table Socio
```

```
CREATE TABLE Socio(  
  ID Integer NOT NULL,  
  Nome Varchar2(25 CHAR) NOT NULL,  
  Cognome Varchar2(25 CHAR) NOT NULL,  
  Via Varchar2(30 CHAR) NOT NULL,  
  NumeroCivico Varchar2(7 CHAR) NOT NULL,  
  Citta Varchar2(20 CHAR) NOT NULL,  
  RecapitoTelefonico Varchar2(10 CHAR),  
  FKZona Integer NOT NULL  
)
```

```
-- Add keys for table Socio
```

```
ALTER TABLE Socio ADD CONSTRAINT PKSocio PRIMARY KEY (ID)
```

```
-- Table Categoria
```

```
CREATE TABLE Categoria(  
  ID Integer NOT NULL,  
  Descrizione Varchar2(30 CHAR) NOT NULL  
)
```

```
-- Add keys for table Categoria
```

```
ALTER TABLE Categoria ADD CONSTRAINT PKCategoria PRIMARY KEY (ID)
```

-- Table Servizio

```
CREATE TABLE Servizio(  
  ID Integer NOT NULL,  
  Data Date NOT NULL,  
  NumeroOre Integer NOT NULL,  
  FKPrestazione Integer NOT NULL,  
  FKSocioOfferente Integer NOT NULL,  
  FKSocioRicevente Integer NOT NULL  
)
```

-- Add keys for table Servizio

```
ALTER TABLE Servizio ADD CONSTRAINT PKServizi PRIMARY KEY (ID)
```

-- Table Prestazione

```
CREATE TABLE Prestazione(  
  ID Integer NOT NULL,  
  Descrizione Varchar2(30 CHAR) NOT NULL,  
  FKCategoria Integer NOT NULL  
)
```

-- Add keys for table Prestazione

```
ALTER TABLE Prestazione ADD CONSTRAINT PKPrestazione PRIMARY KEY (ID)
```

-- Table Competenza

```
CREATE TABLE Competenza(  
  FKSocio Integer NOT NULL,  
  FKPrestazione Integer NOT NULL  
)
```

-- Add keys for table Competenza

```
ALTER TABLE Competenza ADD CONSTRAINT Key1 PRIMARY KEY (FKSocio,FKPrestazione)
```

-- CREATE SEQUENCES SECTION -----

```
CREATE SEQUENCE SeqZona  
  INCREMENT BY 1  
  START WITH 1
```

```
CREATE SEQUENCE SeqSocio  
  INCREMENT BY 1  
  START WITH 1
```

```
CREATE SEQUENCE SeqServizio  
  INCREMENT BY 1  
  START WITH 1
```

```
CREATE SEQUENCE SeqPrestazione
```

```
INCREMENT BY 1  
START WITH 1
```

```
CREATE SEQUENCE SeqCategoria  
INCREMENT BY 1  
START WITH 1
```

-- CREATE RELATIONSHIPS SECTION -----

```
ALTER TABLE Prestazione ADD CONSTRAINT Categoria-Prestazione FOREIGN KEY  
(FKCategoria) REFERENCES Categoria (ID)
```

```
ALTER TABLE Servizio ADD CONSTRAINT Prestazione-Servizio FOREIGN KEY  
(FKPrestazione) REFERENCES Prestazione (ID)
```

```
ALTER TABLE Socio ADD CONSTRAINT Zona-Socio FOREIGN KEY (FKZona) REFERENCES Zona  
(ID)
```

```
ALTER TABLE Servizio ADD CONSTRAINT Socio Offerente-Servizio FOREIGN KEY  
(FKSocioOfferente) REFERENCES Socio (ID)
```

```
ALTER TABLE Servizio ADD CONSTRAINT Socio Ricevente-Servizio FOREIGN KEY  
(FKSocioRicevente) REFERENCES Socio (ID)
```

```
ALTER TABLE Competenza ADD CONSTRAINT Socio-Competenza FOREIGN KEY (FKSocio)  
REFERENCES Socio (ID)
```

```
ALTER TABLE Competenza ADD CONSTRAINT Prestazione-Competenza FOREIGN KEY  
(FKPrestazione) REFERENCES Prestazione (ID)
```

-- CREATE TRIGGER SECTION -----

-- trigger for sequence seqzona for column id in table zona -----

```
CREATE OR REPLACE TRIGGER t_SeqZona BEFORE INSERT  
ON Zona FOR EACH ROW  
BEGIN  
    SELECT SeqZona.nextval INTO :new.ID FROM DUAL;  
END;
```

-- trigger for sequence seqsocio for column id in table socio -----

```
CREATE OR REPLACE TRIGGER t_SeqSocio BEFORE INSERT  
ON Socio FOR EACH ROW  
BEGIN  
    SELECT SeqSocio.nextval INTO :new.ID FROM DUAL;  
END;
```

```
-- trigger for sequence seqcategoria for column id in table categoria -----  
  
CREATE OR REPLACE TRIGGER t_SeqCategoria BEFORE INSERT  
ON Categoria FOR EACH ROW  
BEGIN  
    SELECT SeqCategoria.nextval INTO :new.ID FROM DUAL;  
END;  
  
-- trigger for sequence seqservizio for column id in table servizio -----  
  
CREATE OR REPLACE TRIGGER t_SeqServizio BEFORE INSERT  
ON Servizio FOR EACH ROW  
BEGIN  
    SELECT SeqServizio.nextval INTO :new.ID FROM DUAL;  
END;  
  
-- trigger for sequence seqprestazione for column id in table prestazione -----  
  
CREATE OR REPLACE TRIGGER t_SeqPrestazione BEFORE INSERT  
ON Prestazione FOR EACH ROW  
BEGIN  
    SELECT SeqPrestazione.nextval INTO :new.ID FROM DUAL;  
END;
```

b) Interrogazioni in Algebra Relazionale:

1) per ogni zona, visualizzare la descrizione e la relativa immagine;

$$\pi_{Descrizione, Immagine} (Zona)$$

2) mostrare nome e cognome dei soci di Rovereto;

$$\pi_{Nome, Cognome} (\sigma_{Citta="Rovereto"} (Socio))$$

3) visualizzare le informazioni relative alle prestazioni di tipo culinario;

Scomponiamo l'interrogazione in una serie di operazioni, che indicheremo con opportune sigle:

$$A1 := \rho_{ID Prestazione, Descrizione Prestazione} \leftarrow ID, Descrizione (Prestazione)$$

$$A2 := \sigma_{Descrizione="Culinario"} (Categoria)$$

$$A3 := A1 \triangleright \triangleleft_{Categoria.ID=FKCategoria} A2$$

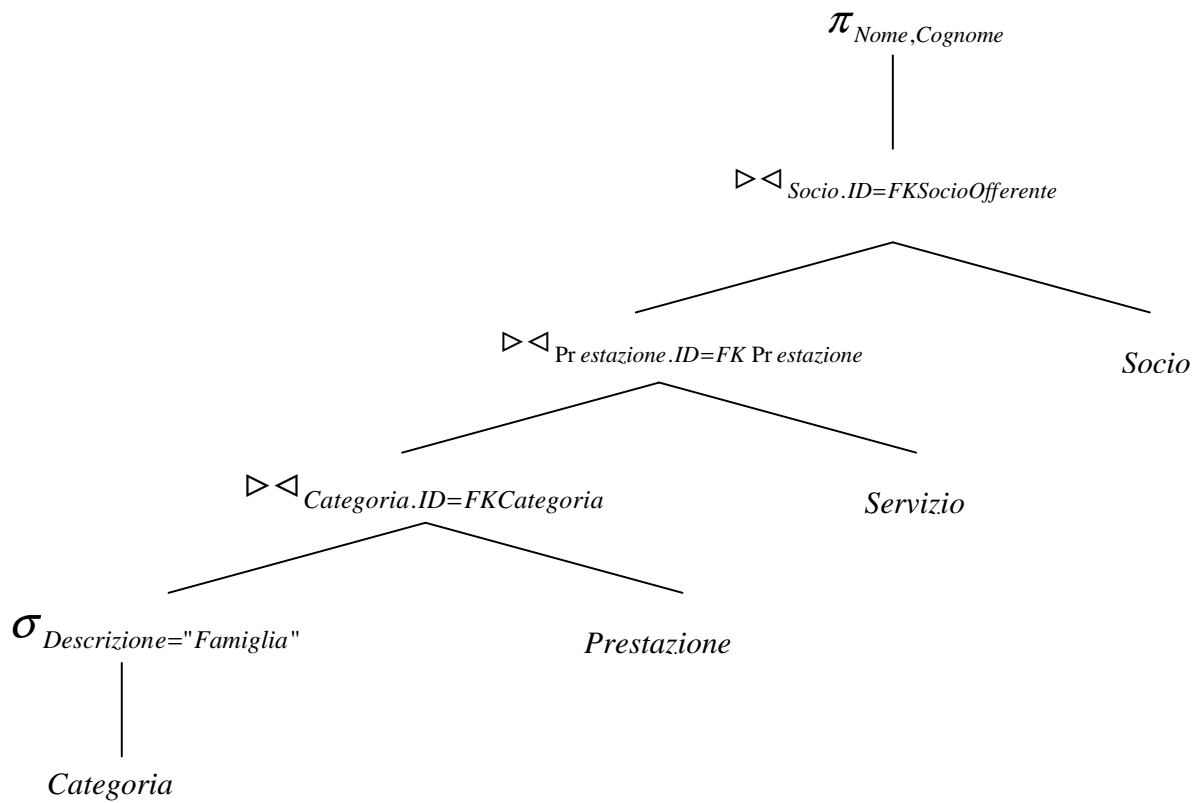
$$A4 := \pi_{ID Prestazione, Descrizione Prestazione} (A3)$$

4) nome e cognome dei soci che hanno erogato servizi nel mese di gennaio 2009;

$$\pi_{Nome, Cognome} (Socio \triangleright \triangleleft_{Socio.ID=FKSocioOfferente} (\sigma_{(Data \geq 01/01/2009 \text{ AND } Data \leq 31/01/2009)} (Servizio))))$$

5) nome e cognome dei soci che hanno erogato servizi di tipo famiglia;

Per ragioni di spazio, rappresentiamo l'interrogazione con una struttura ad albero.



6) nome e cognome dei soci che sono disponibili ad effettuare servizi di tipo sport.

Per ragioni di spazio, rappresentiamo l'interrogazione con una struttura ad albero.

