

Laboratorio
Corso di Basi di Dati e Web
A.A. 2022-2023

Progetto "Piscine Comunali"

Autore:

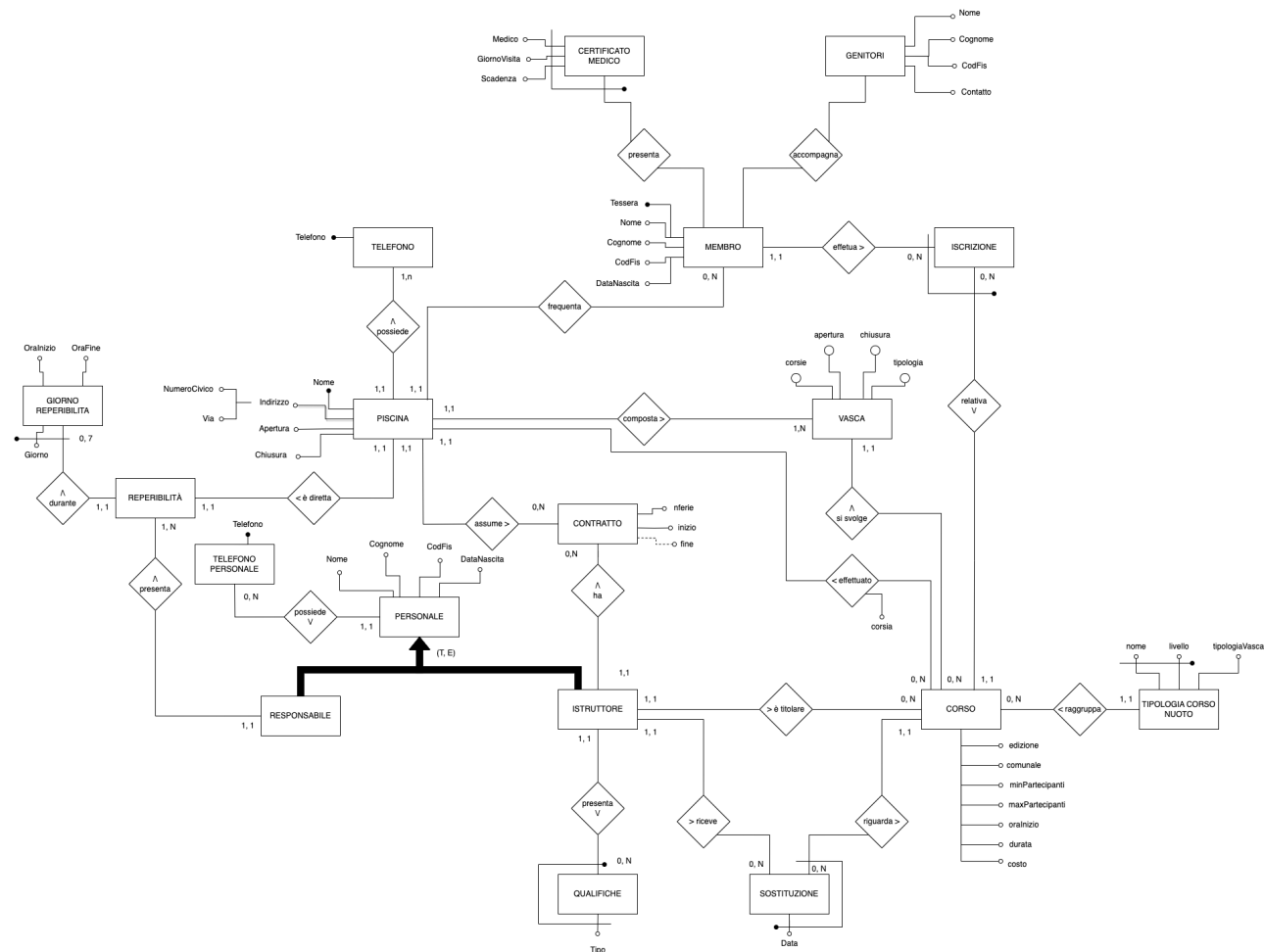
Princess Joy Padua princess.padua@studenti.unimi.it

Matricola: 980733

Data di consegna: *24/07/2023*

1. Progettazione concettuale

1.1 Schema ER



1.2 Vincoli di dominio

In questa sezione vanno riportati e numerati tutti i vincoli che devono essere garantiti nella base di dati e che non si possono direttamente desumere dallo schema ER. La numerazione dei vincoli permette una più facile identificazione nelle fasi successive della progettazione ed implementazione.

Corso:

- La vasca del corso deve essere della piscina a cui appartiene il corso stesso
- La corsia dove viene effettuato il corso deve essere minore uguale al numero di corsie delle vasca
- Il tipo di vasca del corso deve essere uguale alla tipologia specificata in tipo di corso
- L'istruttore deve avere un contratto con la piscina in cui si svolge il corso

Reperibilità:

- Se il responsabile dirige più piscine, allora la reperibilità deve avere dei giorni segnati

- L'istruttore sostituto può effettuare sostituzione solo se ha un contratto in quella determinata piscina per quella data

2.1 Schema ER ristrutturato



2.3 Modello relazionale

Tipologia corso nuoto (tipocorsoid, nome, livello, tipologia)

Telefonopersonale(telefono, personale)

Responsabile(idresponsabile, idpersonale)
Giornoreperibilita(idreperibilita, giorno, orainizio, orafine)

Istruttore(idistruttore, personaleid)
Contratto(idcontratto, istruttore, piscina, inizio, fine, nferie)
Qualifiche(idqualifiche, tipo, istruttore)
Sostituzione(idsostituzione, corso, data, istruttore, sostituto)

Membro(tessera, nome, cognome, codfis, datanascita, piscina)
Genitore(idgenitore, nome, cognome, codfis, contatto, minorenni)
Certificatomedico(idtessera, medico, giornovisita, scadenza)
Iscrizione(idiscrizione, membro, corso)

2.4 Codice SQL

```
--PERSONALE
CREATE TABLE Personale (
  IDPersona SERIAL PRIMARY KEY,
  Nome VARCHAR(255) NOT NULL,
  Cognome VARCHAR(255) NOT NULL,
  CodFis CHAR(20) NOT NULL UNIQUE,
  DataNascita DATE NOT NULL
);
/*
  INSERT INTO Personale (IDPersona, Nome, Cognome, CodFis, DataNascita) VALUES
  (nome, cognome, codFis, dN)
*/
CREATE TABLE TelefonoPersonale (
  Telefono VARCHAR(10) PRIMARY KEY,
  Personale INT,
  FOREIGN KEY (Personale) REFERENCES Personale(IDPersona) ON UPDATE CASCADE ON
  DELETE CASCADE
);
/*
  INSERT INTO TelefonoPersonale (Telefono, Personale) VALUES
  (Telefono, idPersonale)
*/
CREATE TABLE Responsabile (
  IDResponsabile SERIAL PRIMARY KEY,
  Personale INT NOT NULL UNIQUE,
  FOREIGN KEY (Personale) REFERENCES Personale(IDPersona) ON UPDATE CASCADE ON
  DELETE CASCADE
);
/*
  INSERT INTO Responsabile (IDResponsabile, Personale) VALUES (idPersonale)
*/
CREATE TYPE GiornoSettimana AS ENUM(
  'Lunedì',
  'Martedì',
  'Mercoledì',
  'Giovedì',
  'Venerdì',
  'Sabato',
```

```

    'Domenica'
);
CREATE TABLE Reperibilita (
    IDReperibilita SERIAL PRIMARY KEY,
    Responsabile INT NOT NULL,
    FOREIGN KEY (Responsabile) REFERENCES Responsabile(IDResponsabile) ON UPDATE
    CASCADE ON DELETE CASCADE
);
CREATE TABLE GiornoReperibilita (
    IDReperibilita INT NOT NULL,
    Giorno GiornoSettimana NOT NULL,
    OraInizio TIME NOT NULL,
    OraFine TIME NOT NULL,
    PRIMARY KEY (IDReperibilita, Giorno),
    FOREIGN KEY (IDReperibilita) REFERENCES Reperibilita(IDReperibilita) ON UPDATE
    CASCADE ON DELETE CASCADE
);
/*
    INSERT INTO Reperibilita
    (IDReperibilita,Responsabile,Giorno,OraInizio,OraFinePiscina) VALUES
    (Responsabile,Giorno,OraInizio,OraFinePiscina);
*/
CREATE TYPE Indirizzo AS (Via VARCHAR(255), NumeroCivico VARCHAR(10));
CREATE TABLE Piscina (
    --autoincrementa
    IDPiscina SERIAL PRIMARY KEY,
    Nome VARCHAR(20) UNIQUE,
    Indirizzo Indirizzo,
    Apertura DATE,
    Chiusura DATE,
    Reperibilita INT,
    FOREIGN KEY (Reperibilita) REFERENCES Reperibilita(IDReperibilita) ON UPDATE
    CASCADE ON DELETE
    SET NULL
);
/*
    INSERT INTO Piscina (IDPiscina,Nome,Indirizzo,Apertura,Chiusura,Responsabile)
    VALUES
    (Nome,Indirizzo,Apertura,Chiusura,Responsabile);
*/
CREATE TABLE Telefono (
    Telefono VARCHAR(10) PRIMARY KEY,
    Piscina INT,
    FOREIGN KEY (Piscina) REFERENCES Piscina(IDPiscina) ON UPDATE CASCADE ON
    DELETE CASCADE
);
/*
    INSERT INTO Telefono (Teledono,Piscina) VALUES (idPiscina)
*/
/*
    SELECT p.*, t.* from piscina p, telefono t where p.idpiscina=piscina
*/
CREATE TYPE TipologiaVasca AS ENUM(
    'APERTO',
    'CHIUSO',
    'OLIMPIONICA',
    'BABY',
    'NEO NATALE'
);

```

```

);
CREATE TABLE Vasca (
    IDVasca SERIAL PRIMARY KEY,
    Corsie INT NOT NULL,
    Apertura DATE,
    Chiusura DATE,
    Tipologia TipologiaVasca NOT NULL,
    Piscina INT,
    FOREIGN KEY (Piscina) REFERENCES Piscina(IDPiscina) ON UPDATE CASCADE ON
DELETE CASCADE
);
/*
INSERT INTO Vasca (IDVasca,Corsie,PeriodoFruizione,TipologiaVasca) VALUES
(Corsie,PeriodoFruizione,Tipologia)
*/
CREATE TABLE Istruttore (
    IDIstruttore SERIAL PRIMARY KEY,
    PersonaleID INT NOT NULL UNIQUE,
    FOREIGN KEY (PersonaleID) REFERENCES Personale(IDPersona) ON UPDATE CASCADE ON
DELETE CASCADE
);
CREATE TABLE Contratto (
    IDContratto SERIAL PRIMARY KEY,
    Istruttore INT not null,
    Piscina int not null,
    Inizio date not null,
    fine date,
    nferie INT,
    FOREIGN KEY (Istruttore) REFERENCES Istruttore(IDIstruttore) ON UPDATE CASCADE
ON DELETE CASCADE,
    FOREIGN KEY (Piscina) REFERENCES Piscina(IDPiscina) ON UPDATE CASCADE ON
DELETE CASCADE
);
CREATE TABLE Qualifiche (
    IDQualifiche SERIAL PRIMARY KEY,
    Tipo VARCHAR(255),
    Istruttore INT NOT NULL,
    UNIQUE(Tipo, Istruttore),
    FOREIGN KEY (Istruttore) REFERENCES Istruttore(IDIstruttore) ON UPDATE CASCADE
ON DELETE CASCADE
);
CREATE TABLE TipologiaCorsoNuoto (
    TipoCorsoID SERIAL PRIMARY KEY,
    Nome VARCHAR(20) NOT NULL,
    Livello SMALLINT NOT NULL,
    Tipologia TipologiaVasca,
    UNIQUE (Nome, Livello)
);
CREATE TABLE Corso(
    IdCorso SERIAL PRIMARY KEY,
    Comunale BOOLEAN,
    TipoCorso INT,
    Edizione SMALLINT NOT NULL,
    Piscina INT NOT NULL,
    Vasca INT NOT NULL,
    Corsia SMALLINT NOT NULL,
    IstruttoreTitolare INT NOT NULL,
    Costo DECIMAL(10, 2) NOT NULL,

```

```

MinPartecipanti SMALLINT NOT NULL,
MaxPartecipanti SMALLINT NOT NULL,
OraInizio TIME NOT NULL,
Durata INTERVAL HOUR TO MINUTE NOT NULL,
FOREIGN KEY (TipoCorso) REFERENCES TipologiaCorsoNuoto(TipoCorsoID),
FOREIGN KEY (Piscina) REFERENCES Piscina(IDPiscina) ON UPDATE CASCADE ON
DELETE CASCADE,
FOREIGN KEY (Vasca) REFERENCES Vasca(IDVasca) ON UPDATE CASCADE ON DELETE
CASCADE,
FOREIGN KEY (IstruttoreTitolare) REFERENCES Istruttore(IDIstruttore)
);
---MEMBRI ISCRITTI ---
CREATE TABLE Membro(
    Tessera VARCHAR(10) PRIMARY KEY,
    Nome VARCHAR(255) NOT NULL,
    Cognome VARCHAR(255) NOT NULL,
    CodFis CHAR(20) NOT NULL,
    DataNascita DATE NOT NULL,
    Piscina INT NOT NULL,
    FOREIGN KEY (Piscina) REFERENCES Piscina(IDPiscina) ON UPDATE CASCADE ON
DELETE CASCADE
);
CREATE TABLE Genitore(
    IDGenitore SERIAL PRIMARY KEY,
    Nome VARCHAR(255) NOT NULL,
    Cognome VARCHAR(255) NOT NULL,
    CodFis CHAR(20) NOT NULL,
    Contatto varchar(10),
    Minorenne varchar(10),
    FOREIGN KEY (Minorenne) REFERENCES Membro(Tessera) ON UPDATE CASCADE ON DELETE
CASCADE
);
CREATE TABLE CertificatoMedico(
    IDTessera VARCHAR(10) NOT NULL,
    Medico VARCHAR(255) NOT NULL,
    GiornoVisita DATE NOT NULL,
    Scadenza DATE NOT NULL,
    UNIQUE(IDTessera, Medico, GiornoVisita, Scadenza),
    FOREIGN KEY (IDTessera) REFERENCES Membro(Tessera) ON UPDATE CASCADE ON DELETE
CASCADE
);
CREATE TABLE Iscrizione(
    IdIscrizione SERIAL PRIMARY KEY,
    Membro VARCHAR(10) NOT NULL,
    Corso INT NOT NULL,
    FOREIGN KEY (Membro) REFERENCES Membro(Tessera) ON UPDATE CASCADE ON DELETE
CASCADE,
    FOREIGN KEY (Corso) REFERENCES Corso(IDCorso) ON UPDATE CASCADE ON DELETE
CASCADE,
    UNIQUE(Corso, Membro)
);
CREATE TABLE Sostituzione(
    IdSostituzione SERIAL PRIMARY KEY,
    Corso INT NOT NULL,
    Data DATE NOT NULL,
    IstruttoreSostituto INT NOT NULL,
    FOREIGN KEY (Corso) REFERENCES Corso(IdCorso) ON UPDATE CASCADE ON DELETE
CASCADE,

```

```

    FOREIGN KEY (IstruttoreSostituto) REFERENCES Istruttore(IDIstruttore) ON
UPDATE CASCADE ON DELETE CASCADE,
    UNIQUE (Corso, Data)
);

```

```

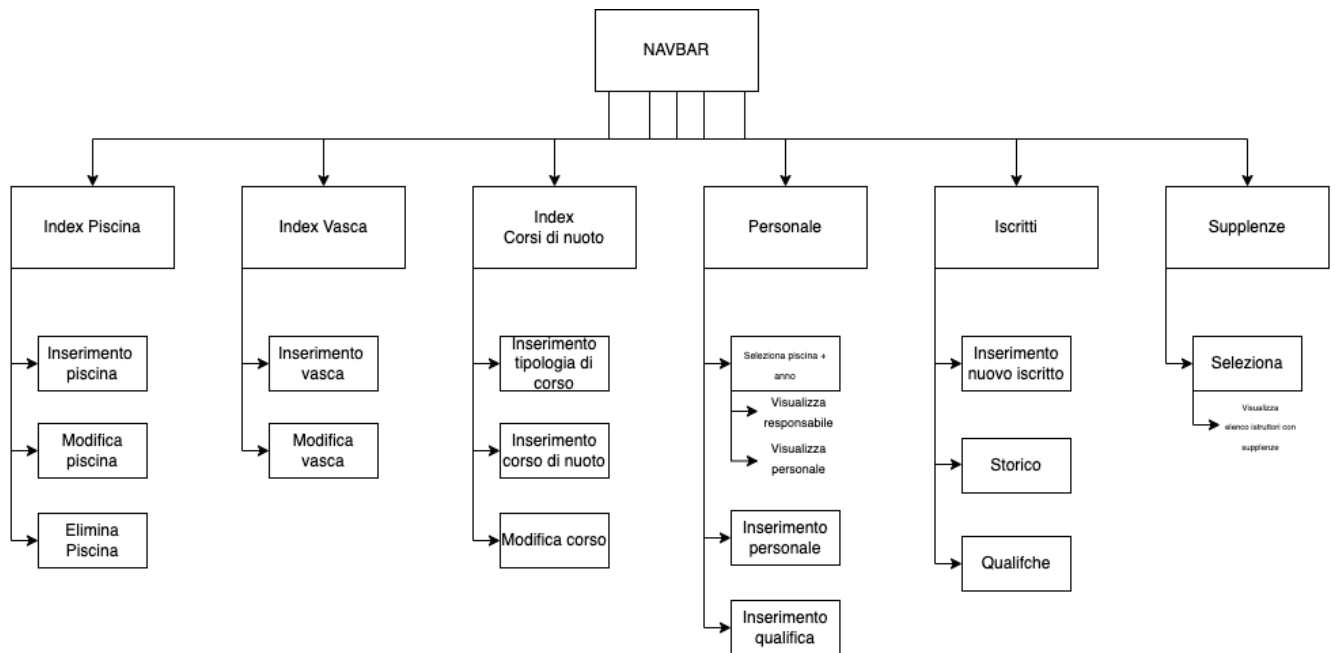
-- a. Determinare gli istruttori supplenti che
--      hanno esattamente una supplenza nella stagione corrente
SELECT s.istruttoresostituto,
       p.nome as nomepersona,
       p.cognome as cognomepersona
from sostituzione s
       join corso c on s.corso = c.idcorso
       join istruttore i on s.istruttoresostituto = i.idistruttore
       join personale p on i.personaleid = p.idpersona
WHERE c.edizione = EXTRACT(YEAR FROM NOW())
group by s.istruttoresostituto,
         nomepersona,
         cognomepersona
having count(*) = 1;

-- b. Determinare gli istruttori supplenti che
--      hanno almeno due supplenze nella stagione corrente
SELECT s.istruttoresostituto,
       p.nome as nomepersona,
       p.cognome as cognomepersona
from sostituzione s
       join corso c on s.corso = c.idcorso
       join istruttore i on s.istruttoresostituto = i.idistruttore
       join personale p on i.personaleid = p.idpersona
WHERE c.edizione = EXTRACT(YEAR FROM NOW())
group by s.istruttoresostituto,
         nomepersona,
         cognomepersona
having count(*) >= 2;

-- c. Determinare gli istruttori supplenti che
--      hanno non più di due supplenze nella stagione corrente
SELECT s.istruttoresostituto,
       p.nome as nomepersona,
       p.cognome as cognomepersona
from sostituzione s
       join corso c on s.corso = c.idcorso
       join istruttore i on s.istruttoresostituto = i.idistruttore
       join personale p on i.personaleid = p.idpersona
WHERE c.edizione = EXTRACT(YEAR FROM NOW())
group by s.istruttoresostituto,
         nomepersona,
         cognomepersona
having count(*) <= 2;

```


3. Progettazione del sito



Nome del database PostgreSQL: piscine

Nome della cartella sotto htdocs: piscina