

Laboratorio  
Corso di Basi di Dati e Web  
A.A. 2020/2021

## Progetto "Gestione di piscine comunali"

Autore:

*Veronica Mandelli - veronica.mandelli1@studenti.unimi.it*

Data di consegna: *12/09/2021*

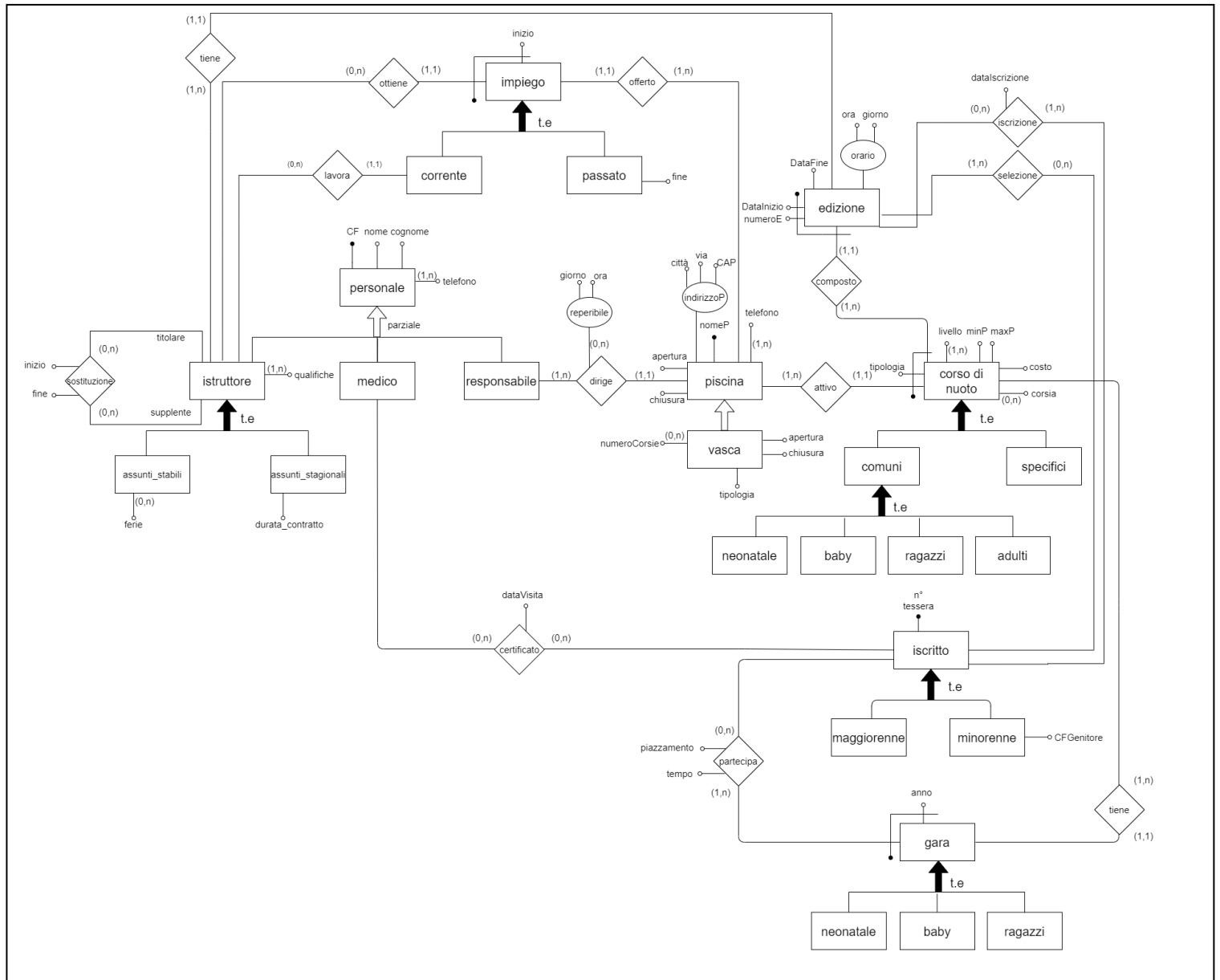
Il progetto è già stato consegnato in precedenza? SI

Il Progetto è stato consegnato in precedenza il 18/07/2021

modifiche apportate: ho aggiunto le richieste dell'appello di luglio

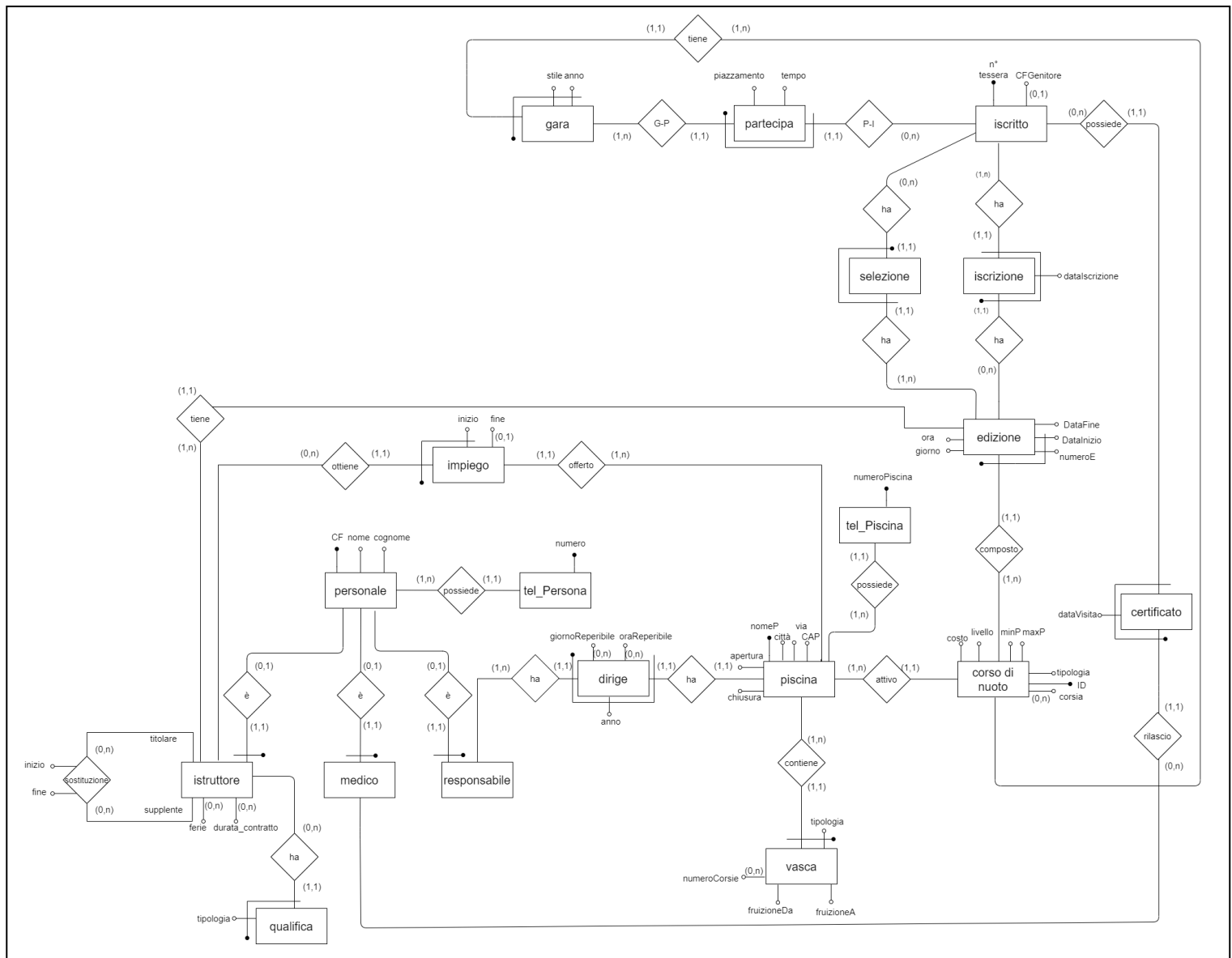
# 1. Progettazione concettuale

## 1.1 Schema ER



## 1.2 Vincoli di dominio

1. Alcuni corsi (baby o nuoto neonatale) possono essere attivati esclusivamente nelle strutture con le apposite vasche
2. Livelli dei corsi di nuoto (da 1 a 3)
3. il costo dei corsi non deve assumere valori negativi
4. deve essere fissato un numero di partecipanti massimo al Corso
5. la data di inizio impiego deve essere antecedente a quella di fine
6. l'attributo "numero" di Edizione deve essere Maggiore di 0;
7. l'attributo "giorno" può assumere valori: lunedì, martedì, mercoledì, giovedì, venerdì, sabato, Domenica.



## 2.2 Vincoli di dominio

1. Alcuni corsi (baby o nuoto neonatale) possono essere attivati esclusivamente nelle strutture con le apposite vasche
2. Livelli dei corsi di nuoto (da 1 a 3)
3. il costo dei corsi non deve assumere valori negativi
4. deve essere fissato un numero di partecipanti massimo al Corso
5. la data di inizio impiego deve essere antecedente a quella di fine
6. l'attributo "numero" di Edizione deve essere Maggiore di 0;
7. l'attributo "giorno" può assumere valori: lunedì, martedì, mercoledì, giovedì, venerdì, sabato, Domenica.
8. in Edizione l'attributo "dataInizio" deve sempre avere un valore minore o uguale di "dataFine";
9. l'attributo "numeroCorsie" di Vasca e di Corso può assumere valore 0 per vasche che non sono divise in corsie e per i corsi di nuoto che si tengono nelle piscine che non sono divise in corsie
10. l'attributo "tipologia" di Vasca può assumere i seguenti valori: all'aperto, al chiuso, olimpionica, nuoto baby, nuoto neonatale;
11. per poter frequentare un edizione di un Corso, l'iscritto deve possedere un certificate medico.
12. l'attributo "dataVisita" nell'associazione certificate non può essere successiva alla data corrente;
13. l'attributo "inizio" nell'associazione sostituzione e nell'entità impiego deve essere minore o uguale di "fine";
14. Un iscritto è minorenne se la cardinalità minima in CFGenitore è = a 0
15. nella cardinalità da edizione verso selezione, il massimo è 3 (che è anche il minimo)

## 2.3 Modello relazionale

PISCINA (nomeP, città, via, CAP, apertura, chiusura)

CORSONUOTO (ID, tipologia, costo, minP, maxP, livello, corsia, nomeP)

VASCA (tipologia, nomeP, numeroCorsie, fruizioneDa, fruizioneA)

PERSONALE(CF, nome, cognome)

TEL\_PERSONALE(numero, CF)

TEL\_PISCINA (numeroPiscina, nomeP)

RESPONSABILE (CF)

DIRIGE (CFresponsabile, nomeP, anno, giornoReperibile, oraReperibile)

MEDICO(CF)

ISTRUTTORE(CF,*ferie*,*durataContratto*)

QUALIFICA(*tipologia*, CFIstruttore)

SOSTITUZIONE(CFtitolare,CFsupplente,inizio,fine)

IMPIEGO(CFIstruttore, inizio,*fine*, nomeP)

ISCRITTO (numTessera, *CFGenitore*)

ISCRIZIONE(IDcorso , *numTessera*, dataIscrizione)

SELEZIONE(IDcorso,numTesseraIscritto)

EDIZIONE (IDCorso, numEdizione, dataInizio, dataFine, ora, giorno, CFistruttore)

CERTIFICATO(CFMedico,*numTessera*,*dataVisita*)

PARTECIPA(numTesseraIscritto, *stile*, *anno*, IDcorso, piazzamento, tempo)

GARA(IDcorso, *stile*, *anno*)

Sono state attuate due ottimizzazioni:

- nella relazione “Iscrizione” sono mantenute solo tre istanze (IDcorso, numTessera, dataIscrizione) che fanno da chiave primaria, perchè le altre (relative all’Edizione) sono ricavabili implicitamente.

- nella relazione “Selezione” sono mantenute solo due istanze (IDcorso, numTessera) che fanno da chiave primaria, perchè le altre (relative all’Edizione) sono ricavabili implicitamente.

## 2.4 Codice SQL

### - CREAZIONE TABELLE

```
CREATE TABLE Piscina (  
    nomeP varchar(20) PRIMARY KEY,  
    citta varchar(20) NOT NULL,  
    via varchar(20) NOT NULL,  
    CAP integer NOT NULL,  
    apertura date NOT NULL,  
    chiusura date NOT NULL  
);
```

```
CREATE TABLE CorsoDiNuoto (  
    ID varchar(20) PRIMARY KEY,  
    tipologia varchar(20) NOT NULL,  
    costo decimal(8,2) NOT NULL DEFAULT 0 CHECK(costo>=0),  
    minP integer NOT NULL,  
    maxP integer NOT NULL CHECK(maxP<=20),  
    livello integer NOT NULL CHECK(livello>=1 AND livello<=3),  
    corsia integer NOT NULL,  
    nomeP varchar(20) NOT NULL REFERENCES Piscina(nomeP) ON UPDATE CASCADE ON  
    DELETE CASCADE  
);
```

```
CREATE TABLE Vasca (  
    tipologia varchar(30),  
    nomeP varchar(20) REFERENCES Piscina(nomeP) ON UPDATE CASCADE ON DELETE  
    CASCADE,  
    corsia integer NULL,  
    FruizioneDA date NOT NULL,  
    FruizioneA date NOT NULL,  
    PRIMARY KEY(tipologia, nomeP)  
);
```

```
CREATE TABLE Personale (  
    CF varchar(20) PRIMARY KEY,  
    nome varchar(20) NOT NULL,  
    cognome varchar(20) NOT NULL  
);
```

```
CREATE TABLE Tel_Personale (  
    numero integer PRIMARY KEY,  
    CF varchar(20) REFERENCES Personale(CF) ON UPDATE CASCADE ON DELETE  
    CASCADE  
);
```

```
CREATE TABLE Tel_piscina (  
    numero integer PRIMARY KEY,  
    nomeP varchar(20) REFERENCES Piscina(NomeP) ON UPDATE CASCADE ON DELETE  
    CASCADE  
);
```

```
CREATE TABLE Responsabile (  
    CF varchar(20) PRIMARY KEY,  
    FOREIGN KEY(CF) REFERENCES Personale(CF) ON UPDATE CASCADE ON DELETE  
    CASCADE  
);
```

```
CREATE TABLE Dirige (  
    CFResponsabile varchar(20) REFERENCES Responsabile(CF) ON UPDATE CASCADE ON  
    DELETE CASCADE,  
    nomeP varchar(20) NOT NULL,  
    anno integer NOT NULL,  
    giornoReperibile varchar(20) NULL,  
    oraReperibile time NULL,  
    PRIMARY KEY(CFResponsabile, nomeP, anno),  
    FOREIGN KEY(nomeP) REFERENCES Piscina(nomeP) ON UPDATE CASCADE ON  
    DELETE CASCADE  
);
```

```
CREATE TABLE Medico (  
    CF varchar(20) PRIMARY KEY,  
    FOREIGN KEY(CF) REFERENCES Personale(CF) ON UPDATE CASCADE ON DELETE  
    CASCADE  
);
```

```
CREATE TABLE Istruttore(  
    CF varchar(20) PRIMARY KEY references Personale(CF) ON UPDATE CASCADE ON  
    DELETE CASCADE,  
    ferie numeric NULL CHECK(ferie>=0),  
    durataContratto varchar(20) NULL  
);
```

```
CREATE TABLE Qualifica(  
    tipologia varchar(20),  
    CFIstruttore varchar(20) REFERENCES Personale(CF) ON UPDATE CASCADE ON DELETE  
    CASCADE,  
    PRIMARY KEY(tipologia, CFIstruttore)  
);
```

```
CREATE TABLE Sostituzione(  
    CFTitolare varchar(20) REFERENCES Personale(CF) ON UPDATE CASCADE ON DELETE  
    CASCADE,  
    CFsupplente varchar(20) REFERENCES Personale(CF) ON UPDATE CASCADE ON  
    DELETE CASCADE,  
    inizio date NOT NULL,  
    fine date NOT NULL,  
    PRIMARY KEY(CFTitolare, CFsupplente, inizio),  
    CHECK(inizio<=fine)  
);
```

```
CREATE TABLE Impiego(  
    CFIstruttore char(20) REFERENCES Personale(CF) ON UPDATE CASCADE ON DELETE  
    CASCADE,  
    inizio date NOT NULL CHECK(inizio<fine),  
    fine date NULL CHECK(fine>inizio),  
    nomeP varchar(20) NOT NULL,  
    PRIMARY KEY (CFIstruttore, inizio),  
    FOREIGN KEY(nomeP) REFERENCES Piscina(nomeP) ON UPDATE CASCADE ON  
    DELETE CASCADE  
);
```

```
CREATE TABLE Iscritto (  
    numTessera varchar(20) PRIMARY KEY,  
    CFGenitore varchar(20) NULL  
);
```

```
CREATE TABLE Edizione (  
    IDCorso varchar(20) REFERENCES CorsoDiNuoto(ID) ON UPDATE CASCADE ON  
    DELETE CASCADE,  
    numEdizione numeric NOT NULL,  
    dataInizio date NOT NULL,  
    dataFine date NOT NULL,  
    ora time NOT NULL,  
    giorno varchar(20) NOT NULL,  
    CFistruttore varchar(20),  
    PRIMARY KEY(IDCorso, numEdizione, dataInizio),  
    FOREIGN KEY(CFistruttore) references Istruttore(CF) ON UPDATE CASCADE ON DELETE  
    CASCADE  
);
```

```
CREATE TABLE Iscrizione (  
    IDCorso varchar(20) REFERENCES CorsoDiNuoto(ID) ON UPDATE CASCADE ON  
    DELETE CASCADE,  
    numTessera varchar(20),  
    dataIscrizione date,  
    PRIMARY KEY(IDCorso, numTessera, dataIscrizione),  
    FOREIGN KEY(numTessera) references Iscritto(numTessera) ON UPDATE CASCADE ON  
    DELETE CASCADE  
);
```



```
CREATE TABLE Selezione (
    numTesseraiScritto varchar(20) REFERENCES Iscritto(numTesserai) ON UPDATE CASCADE
    ON DELETE CASCADE,
    IDcorso varchar(20),
    PRIMARY KEY(numTesseraiScritto, IDcorso),
    FOREIGN KEY(IDcorso) references CorsodiNuoto(ID) ON UPDATE CASCADE ON DELETE
    CASCADE
);
```

```
CREATE TABLE Certificato (
    CFMedico varchar(20) NOT NULL REFERENCES Medico(CF) ON UPDATE CASCADE ON
    DELETE CASCADE,
    numTesserai varchar(20) NOT NULL,
    dataVisita date NOT NULL,
    PRIMARY KEY(CFMedico, numTesserai, dataVisita),
    FOREIGN KEY(numTesserai) REFERENCES Iscritto(numTesserai) ON UPDATE CASCADE
    ON DELETE CASCADE
);
```

```
CREATE TABLE Partecipa (
    numTesseraiScritto varchar(20) REFERENCES Iscritto(numTesserai) ON UPDATE CASCADE
    ON DELETE CASCADE,
    stile varchar(20) NOT NULL,
    anno varchar(20) NOT NULL,
    IDcorso varchar(20) NOT NULL,
    piazzamento numeric NOT NULL,
    tempo time NOT NULL,
    PRIMARY KEY(numTesseraiScritto, stile, anno, IDcorso),
    FOREIGN KEY(IDcorso) REFERENCES CorsodiNuoto(ID) ON UPDATE CASCADE ON
    DELETE CASCADE
);
```

```
CREATE TABLE Gara (
    IDcorso varchar(20) NOT NULL REFERENCES CorsodiNuoto(ID) ON UPDATE CASCADE
    ON DELETE CASCADE,
    stile varchar(20) NOT NULL,
    anno varchar(20) NOT NULL,
    PRIMARY KEY(IDcorso, stile, anno)
);
```

- INTERROGAZIONI

1. Trovare tutti gli istruttori qualificati nella tipologia “pallanuoto”

```
SELECT CFistruttore
FROM qualifica
WHERE tipologia='pallanuoto'
```

2. trovare il periodo di apertura della piscina di Milano

```
SELECT apertoDa, finoA
```

```
FROM Piscina
WHERE città = 'Milano'
```

3. trovare l'ID dei corsi di nuoto che si svolgono nella 2° corsia

```
SELECT ID
FROM CorsoDiNuoto
WHERE corsia = '2'
```

4. trovare il numero di tessera di tutti gli iscritti maggiorenni

```
SELECT numTessera
FROM Iscritto
WHERE CFGenitore = 'null'
```

5. Determinare gli istruttori supplenti che hanno esattamente una supplenza nella stagione corrente

```
SELECT CFsupplente
FROM sostituzione
WHERE inizio >= '2021-01-01'
GROUP BY CFsupplente
HAVING COUNT(inizio) = 1
```

6. Determinare gli istruttori supplenti che hanno almeno due supplenze nella stagione corrente

```
SELECT CFsupplente
FROM sostituzione
WHERE inizio >= '2021-01-01'
GROUP BY CFsupplente
HAVING COUNT(inizio) >= 2
```

7. Determinare gli istruttori supplenti che hanno non più di due supplenze nella stagione corrente

```
SELECT CFsupplente
FROM sostituzione
WHERE inizio >= '2021-01-01'
GROUP BY CFsupplente
HAVING COUNT(inizio) <= 2
```

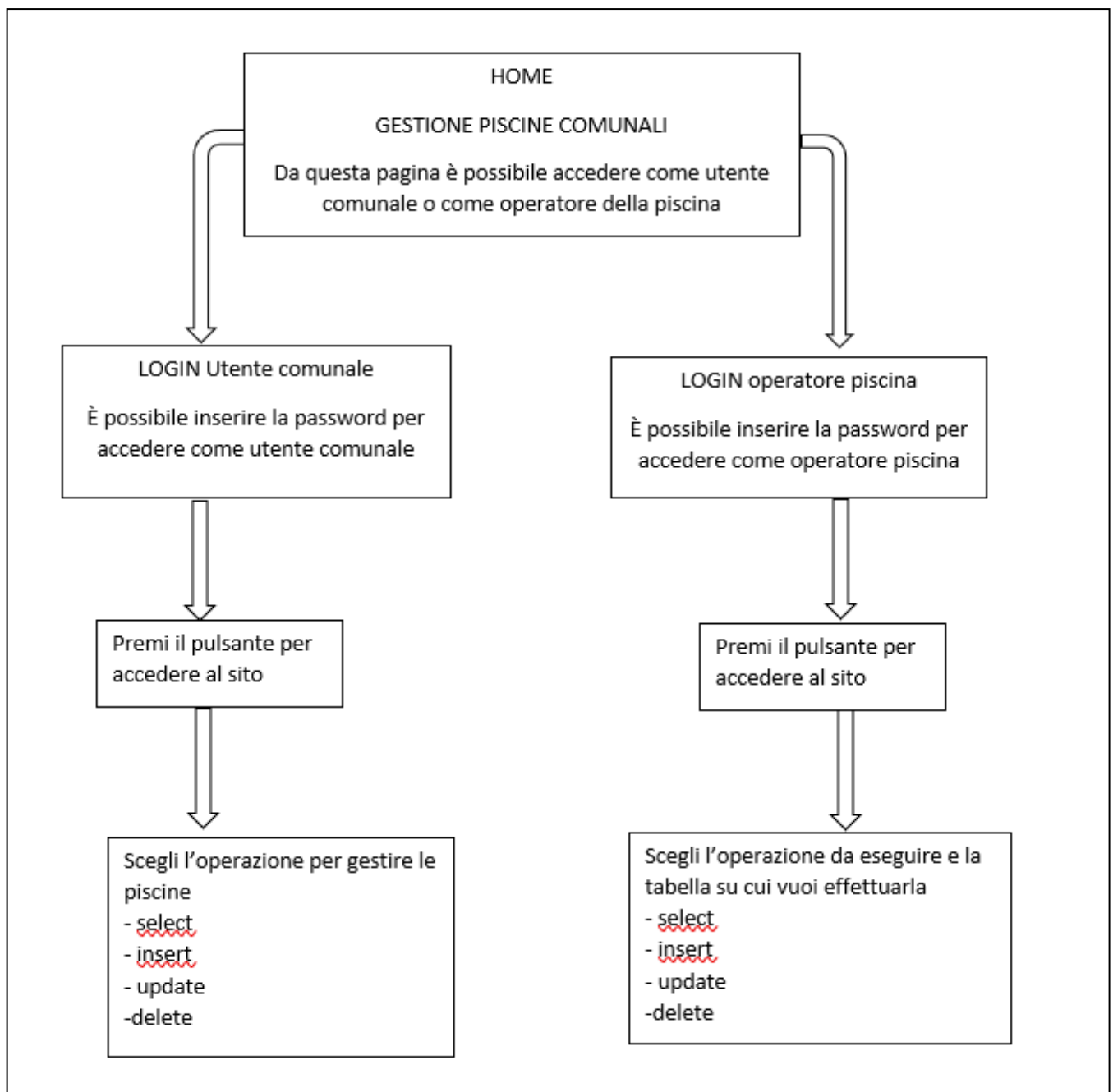
8. restituire per ogni corso il numero di partecipanti alle gare di fine corso per l'anno corrente

```
SELECT DISTINCT idcorso, COUNT (numTesseraIscritto)
FROM partecipa
WHERE anno='2021'
GROUP BY idcorso
```

9. Restituire le informazioni del podio di gare relative allo stile rana per l'anno corrente.

```
SELECT IDcorso, numTesseraIscritto, piazzamento
FROM partecipa
WHERE stile='rana' AND anno='2021' AND piazzamento BETWEEN 1 AND 3
ORDER BY IDcorso, piazzamento
```

### 3. Progettazione del sito



Nome del database PostgreSQL: *project\_piscine*

Nome della cartella sotto htdocs: *lab/project\_piscine*