

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

Progetto “gestione di piscine comunali”

Autore:

Francesca Ferraro - francesca.ferraro3@studenti.unimi.it

Data di consegna: 18/07/2021

Il progetto è già stato consegnato in precedenza? no

Se sì, indica tutte le date degli appelli in cui il progetto è stato consegnato:

Se sì, indica brevemente (tre righe al massimo) le modifiche apportate rispetto all'ultimo progetto consegnato:

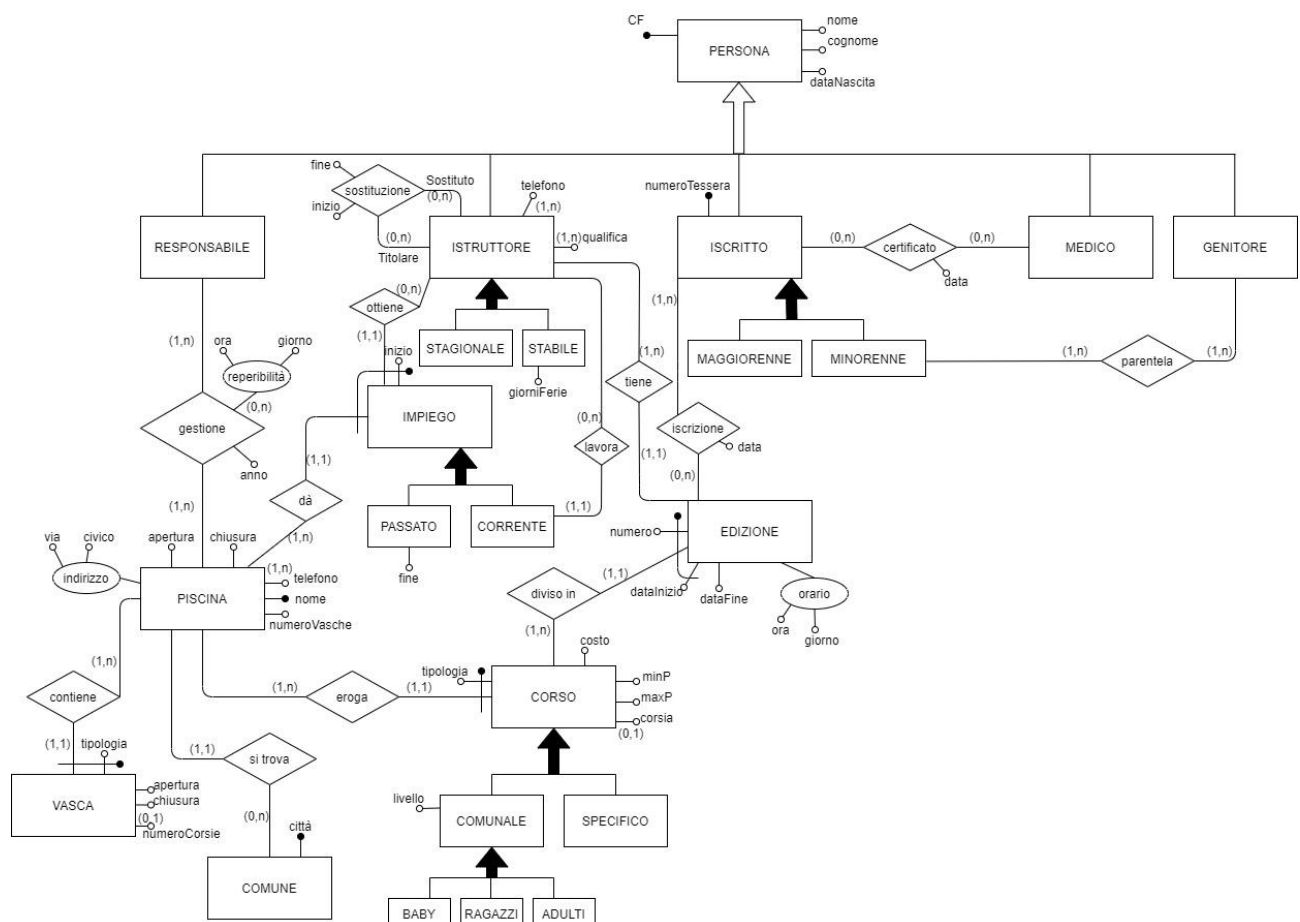
1. Progettazione concettuale

In questa sezione vanno riportate le conclusioni della progettazione concettuale. Non siamo interessati ai vari passaggi intermedi con i quali arrivate a costruire lo schema concettuale. Vogliamo lo schema concettuale e le considerazioni/vincoli finali.

1.1 Schema ER

Lo schema concettuale proposto deve utilizzare al meglio i costrutti presenti nel modello ER (Entità, associazioni, gerarchie di generazioni, identificazione interne ed esterne, attributi multi-valore e composti). Lo schema ER deve essere facilmente leggibile. “A colpo d’occhio” deve essere chiaro quello che lo schema vuole rappresentare. Entità e associazioni devono avere nomi distinti e di facile comprensione.

Lo schema ER deve contenere tutte le entità ed associazioni che verranno utilizzate nel modello logico e poi nell'implementazione vera e propria. Non è ammesso che nelle fasi successive vengano aggiunte nuove informazioni che non siano motivate dall'applicazione corretta delle regole viste a lezione/laboratorio. Se in una fase successiva del progetto vi accorgete di esservi "dimenticati qualcosa", dovete rimettere mano allo schema concettuale.



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.

1. il valore dell'attributo "livello" in Corso può assumere valori da 1 a 3;
2. l'attributo "giorno" può assumere valori: lunedì, martedì, mercoledì, giovedì, venerdì, sabato e domenica;
3. in Edizione, l'attributo "datainizio" deve sempre avere un valore minore o uguale di "datafine";
4. l'attributo "numerosvasche" di Piscina deve sempre assumere valori maggiori di 0;
5. l'attributo "inizio" nell'associazione Sostituzione non può essere successivo alla data di fine;
6. in Impiego, l'attributo "inizio" dev'essere minore o uguale di "fine";
7. in Corso, l'attributo "costo" dev'essere maggiore di 0;
8. l'attributo "numero" di Edizione dev'essere maggiore di 0;
9. l'attributo "data" nell'associazione Certificato non può essere successivo alla data corrente;
10. un iscritto è Minorenne se l'anno dell'attributo "dataNascita" (in Persona) ad esso associato è minore di (*anno_corrente*-18);
11. l'attributo "tipologia" di Vasca può assumere i seguenti valori: all'aperto, al chiuso, olimpionica, nuoto baby, nuoto neonatale;
12. alcuni corsi (nuoto baby e neonatale) possono essere attivati esclusivamente nelle strutture che possiedono le apposite vasche;
13. per poter frequentare un'Edizione di un Corso, l'Iscritto deve possedere un certificato medico.

2. Progettazione logica

2.1 Schema ER ristrutturato

In questa sezione va riportato lo schema ER ristrutturato secondo la metodologia presentata a lezione e contenuta nel libro di testo. Questo schema, a parte la ristrutturazione, deve essere equivalente a quello presentato nel Paragrafo 1.1.

1. il valore dell'attributo "livello" in Corso può assumere valori da 1 a 3;
2. l'attributo "giorno" può assumere valori: lunedì, martedì, mercoledì, giovedì, venerdì, sabato e domenica;
3. in Edizione, l'attributo "datainizio" deve sempre avere un valore minore o uguale di "datafine";
4. l'attributo "numerosvasche" di Piscina deve sempre assumere valori maggiori di 0;
5. l'attributo "inizio" nell'associazione Sostituzione non può essere successivo alla data di fine;
6. in Impiego, l'attributo "inizio" dev'essere minore o uguale di "fine";
7. in Corso, l'attributo "costo" dev'essere maggiore di 0;
8. l'attributo "numero" di Edizione dev'essere maggiore di 0;
9. l'attributo "data" nell'entità Certificato non può essere successivo alla data corrente;
10. un iscritto è Minorenne se l'anno dell'attributo "dataNascita" (in Persona) ad esso associato è minore di (*anno_corrente*-18);
11. l'attributo "tipologia" di Vasca può assumere i seguenti valori: all'aperto, al chiuso, olimpionica, nuoto baby, nuoto neonatale;

12. alcuni corsi (nuoto baby e neonatale) possono essere attivati esclusivamente nelle strutture che possiedono le apposite vasche;
13. per poter frequentare un'Edizione di un Corso, l'Iscritto deve possedere un certificato medico;
14. (aggiunto in fase di progettazione logica) l'entità Telefono non può essere contemporaneamente associata a entrambe le entità Piscina e Istruttore, ma può averne solo una alla volta mentre l'altra assume valore nullo.

2.3 Modello relazionale

Riportare il modello relazionale e gli eventuali vincoli da considerare. Eventuali ottimizzazioni devono essere prodotte a partire da questa versione, analogamente a quanto visto durante le lezioni di laboratorio.

Certificati(iscritto, medico, data)

Comuni(città)

Corsi(id, tipologia, costo, max, min, livello, corsia, piscina)

Edizioni(numero, dataInizio, corso, dataFine, ora, giorno, istruttore)

Genitori(cf)

Gestioni(responsabile, giornoReperibilità, oraReperibilità, piscina, anno)

Impieghi(cf, piscina, inizio, fine)

Iscritti(tessera, cf)

Iscrizioni(corso, iscritto, data)

Istruttori(cf, giorniFerie)

Medici(cf)

Parentela(genitore, figlio)

Persone(cf, nome, cognome, dataNascita)

Piscine(nome, città, via, civico, apertura, chiusura, numeroVasche)

Qualifiche(tipologia, istruttore)

Responsabili(cf)

Sostituzioni(titolare, sostituto, inizio, fine)

Telefoni(numero, istruttore, piscina)

Vasche(piscina, tipologia, apertura, chiusura, corsie)

Sono state attuate due ottimizzazioni:

- alla relazione "Corsi" è stato aggiunto un id numerico per l'identificazione.
- nella relazione "Iscrizioni" sono mantenute solo tre istanze (corso, iscritto, data) che fanno da chiave primaria, perchè le altre (relative all'Edizione) sono ricavabili implicitamente.

2.4 Codice SQL

Riportare il codice SQL di creazione e alterazione delle tabelle per attuare i vincoli individuati (in particolare vincoli di primary e foreign key, not null, unique e check) e delle interrogazioni richieste nel testo del progetto.

● 2.4.1 creazione delle tabelle

```
CREATE TABLE Persone (  
cf char(16) PRIMARY KEY,  
nome varchar(50) NOT NULL,  
cognome varchar(50) NOT NULL,  
datanascita date NOT NULL  
);
```

```
CREATE TABLE Responsabili (  
cf char(16) NOT NULL REFERENCES Persone (cf) ON UPDATE CASCADE ON DELETE  
CASCADE,  
PRIMARY KEY(cf)  
);
```

```
CREATE TABLE Istruttori (  
cf char(16) NOT NULL REFERENCES Persone (cf) ON UPDATE CASCADE ON DELETE  
CASCADE,  
giorniferie decimal(3),  
CHECK(giorniferie>=0),  
PRIMARY KEY(cf)  
);
```

```
CREATE TABLE Qualifiche (  
cf char(16) NOT NULL REFERENCES Istruttori (cf) ON UPDATE CASCADE ON DELETE  
CASCADE,  
tipologia varchar(50),  
PRIMARY KEY(cf, tipologia)  
);
```

```
CREATE TABLE Comuni (  
citta varchar(50) PRIMARY KEY  
);
```

```
CREATE TABLE Piscine (  
nome varchar(30) PRIMARY KEY,  
citta varchar(50) NOT NULL REFERENCES Comuni (citta) ON UPDATE CASCADE ON DELETE  
CASCADE,  
via varchar(30) NOT NULL,  
civico decimal(3) NOT NULL,  
apertura varchar(30) NOT NULL,  
chiusura varchar(30) NOT NULL,  
numeroVasche decimal(3) NOT NULL,  
CHECK(numeroVasche>0),  
UNIQUE(citta, via, civico)  
);
```

```
CREATE TABLE Telefoni (  
numero decimal(10) PRIMARY KEY,
```

```
istruttore char(16) DEFAULT NULL REFERENCES Istruttori (cf) ON UPDATE CASCADE ON
DELETE CASCADE,
piscina varchar(30) DEFAULT NULL REFERENCES Piscine (nome) ON UPDATE CASCADE ON
DELETE CASCADE
);
```

```
CREATE TABLE Sostituzioni (
titolare char(16) NOT NULL REFERENCES Istruttori (cf) ON UPDATE CASCADE ON DELETE
CASCADE,
sostituto char(16) NOT NULL REFERENCES Istruttori (cf) ON UPDATE CASCADE ON DELETE
CASCADE,
inizio date NOT NULL,
fine date NOT NULL,
PRIMARY KEY(titolare, sostituto, inizio),
CHECK(inizio<=fine)
);
```

```
CREATE TABLE Impieghi (
cf char(16) NOT NULL REFERENCES Istruttori (cf) ON UPDATE CASCADE ON DELETE
CASCADE,
piscina varchar(30) NOT NULL REFERENCES Piscine (nome) ON UPDATE CASCADE ON
DELETE CASCADE,
inizio date NOT NULL,
fine date,
CHECK(inizio<fine),
PRIMARY KEY(inizio, cf, piscina)
);
```

```
CREATE TABLE Corsi (
id numeric(10) PRIMARY KEY,
tipologia varchar(50) NOT NULL,
costo decimal(3) NOT NULL,
max decimal(3) NOT NULL,
min decimal(3) NOT NULL,
livello decimal(1),
corsia decimal(3),
piscina varchar(30) NOT NULL REFERENCES Piscine (nome) ON UPDATE CASCADE ON
DELETE CASCADE,
CHECK(costo>0 AND livello<=3 AND livello>=1)
);
```

```
CREATE TABLE Edizioni (
numero numeric(3) NOT NULL,
datainizio date NOT NULL,
corso decimal(10) NOT NULL REFERENCES Corsi (id) ON UPDATE CASCADE ON DELETE
CASCADE,
datafine date NOT NULL,
ora char(5) NOT NULL,
giorno char(10) NOT NULL,
```

```
istruttore char(16) NOT NULL REFERENCES Istruttori (cf) ON UPDATE CASCADE ON DELETE CASCADE,  
PRIMARY KEY(numero, datainizio, corso),  
CHECK(dataInizio<dataFine AND numero>0),  
UNIQUE(datainizio, datafine, istruttore, numero)  
);
```

```
CREATE TABLE Gestioni (  
responsabile char(16) NOT NULL REFERENCES Responsabili (cf) ON UPDATE CASCADE ON DELETE CASCADE,  
giornoReperibilita varchar(10) NOT NULL,  
oraReperibilita char(5) NOT NULL,  
anno decimal(10) NOT NULL,  
piscina varchar(30) NOT NULL REFERENCES Piscine (nome) ON UPDATE CASCADE ON DELETE CASCADE,  
PRIMARY KEY(responsabile, piscina, anno)  
);
```

```
CREATE TABLE Vasche (  
piscina varchar(30) NOT NULL REFERENCES Piscine (nome) ON UPDATE CASCADE ON DELETE CASCADE,  
tipologia varchar(30) NOT NULL,  
apertura varchar(30) NOT NULL,  
chiusura varchar(30) NOT NULL,  
corsie decimal(3),  
PRIMARY KEY(piscina, tipologia)  
);
```

```
CREATE TABLE Iscritti (  
tessera decimal(10) PRIMARY KEY,  
cf char(16) NOT NULL REFERENCES Persone (cf) ON UPDATE CASCADE ON DELETE CASCADE  
);
```

```
CREATE TABLE Iscrizioni (  
corso numeric(10) NOT NULL REFERENCES Corsi (id) ON UPDATE CASCADE ON DELETE CASCADE,  
iscritto decimal(10) NOT NULL REFERENCES Iscritti (tessera) ON UPDATE CASCADE ON DELETE CASCADE,  
data date NOT NULL,  
PRIMARY KEY(corso, iscritto, data)  
);
```

```
CREATE TABLE Medici (  
cf char(16) NOT NULL REFERENCES Persone (cf) ON UPDATE CASCADE ON DELETE CASCADE,  
PRIMARY KEY(cf)  
);
```



```
CREATE TABLE Certificati (  
iscritto decimal(10) NOT NULL REFERENCES Iscritti (tessera) ON UPDATE CASCADE ON  
DELETE CASCADE,  
medico char(16) NOT NULL REFERENCES Medici (cf) ON UPDATE CASCADE ON DELETE  
CASCADE,  
data date NOT NULL,  
PRIMARY KEY(iscritto, medico, data),  
CHECK(data<=current_date)  
);
```

```
CREATE TABLE Genitori (  
cf char(16) NOT NULL REFERENCES Persone (cf) ON UPDATE CASCADE ON DELETE  
CASCADE,  
PRIMARY KEY(cf)  
);
```

```
CREATE TABLE Parentela (  
genitore char(16) NOT NULL REFERENCES Genitori (cf) ON UPDATE CASCADE ON DELETE  
CASCADE,  
figlio decimal(10) NOT NULL REFERENCES Iscritti (tessera) ON UPDATE CASCADE ON  
DELETE CASCADE,  
PRIMARY KEY(genitore, figlio)  
);
```

- **2.4.1 interrogazioni**

--

--Interrogazione 1: Determinare gli istruttori supplenti che hanno esattamente una supplenza nella stagione corrente

--

```
SELECT sostituto
FROM sostituzioni
WHERE inizio>='2020-09-01'
GROUP BY(sostituto)
HAVING COUNT(inizio)=1
```

--

--Interrogazione 2: Determinare gli istruttori supplenti che hanno almeno due supplenze nella stagione corrente

--

```
SELECT sostituto
FROM sostituzioni
WHERE inizio>='2020-09-01'
GROUP BY(sostituto)
HAVING COUNT(inizio)>=2
```

--

--Interrogazione 3: Determinare gli istruttori supplenti che hanno non più di due supplenze nella stagione corrente

--

```
SELECT sostituto
FROM sostituzioni
WHERE inizio>='2020-09-01'
GROUP BY(sostituto)
HAVING COUNT(inizio)<=2
```

3. Progettazione del sito

Rappresentare uno schema che descriva il flusso di navigazione all'interno del sito. Ogni pagina deve essere rappresentata da un rettangolo che contenga il nome (univoco) della pagina e l'elenco delle funzionalità principali. Indicare come l'utente può spostarsi tra una pagina e l'altra.

Nome del database PostgreSQL: piscine

Nome della cartella sottohtdocs: progetto_piscine

| HOME |
|----------------------------|
| - accesso all'area comuni |
| - accesso all'area piscine |

| LOGIN COMUNI |
|---|
| (accesso tramite button "Area Comuni" in Home) |
| - inserimento password ("abcd") |
| - possibilità di tornare a Home tramite button in alto a sx |

| INSERIMENTO PISCINE |
|---|
| (accesso tramite password in Login comuni o navbar in alto alla pagina) |
| - form per l'inserimento di una nuova struttura da parte di un comune |

| CANCELLAZIONE PISCINE |
|--|
| (accesso tramite navbar in alto alla pagina) |
| - form per l'eliminazione di una struttura da parte di un comune |

| AGGIORNAMENTO PISCINE |
|---|
| (accesso tramite navbar in alto alla pagina) |
| - form per l'aggiornamento di una struttura da parte di un comune |

| VISUALIZZAZIONE PISCINE |
|--|
| (accesso tramite navbar in alto alla pagina) |
| - visualizzazione tabellare dei dati delle Piscine |

| LOGIN PISCINE |
|---|
| (accesso tramite button "Area Piscine" in Home) |
| - inserimento password ("1234") |
| - possibilità di tornare a Home tramite button in alto a sx |

| INSERIMENTO CORSI |
|---|
| (accesso tramite password in Login comuni o navbar in alto alla pagina) |
| - form per l'inserimento di un nuovo corso da parte di una piscina |

| CANCELLAZIONE CORSI |
|---|
| (accesso tramite navbar in alto alla pagina) |
| - form per l'eliminazione di un corso da parte di una piscina |

| AGGIORNAMENTO CORSI |
|--|
| (accesso tramite navbar in alto alla pagina) |
| - form per l'aggiornamento di un corso da parte di una piscina |

| VISUALIZZAZIONE CORSI |
|--|
| (accesso tramite navbar in alto alla pagina) |
| - visualizzazione tabellare dei dati dei Corsi |

| INSERIMENTO EDIZIONI |
|--|
| (accesso tramite navbar in alto alla pagina) |
| - form per l'inserimento di una nuova edizione da parte di una piscina |

| CANCELLAZIONE EDIZIONI |
|--|
| (accesso tramite navbar in alto alla pagina) |
| - form per l'eliminazione di un'edizione da parte di una piscina |

| AGGIORNAMENTO EDIZIONI |
|---|
| (accesso tramite navbar in alto alla pagina) |
| - form per l'aggiornamento di un'edizione da parte di una piscina |

| VISUALIZZAZIONE EDIZIONI |
|---|
| (accesso tramite navbar in alto alla pagina) |
| - visualizzazione tabellare dei dati delle Edizioni |

| INSERIMENTO ISCRITTI |
|---|
| (accesso tramite navbar in alto alla pagina) |
| - form per l'inserimento di un nuovo iscritto da parte di una piscina |

| CANCELLAZIONE ISCRITTI |
|--|
| (accesso tramite navbar in alto alla pagina) |
| - form per l'eliminazione di un iscritto da parte di una piscina |

| AGGIORNAMENTO ISCRITTI |
|---|
| (accesso tramite navbar in alto alla pagina) |
| - form per l'aggiornamento di un iscritto da parte di una piscina |

| VISUALIZZAZIONE ISCRITTI |
|--|
| (accesso tramite navbar in alto alla pagina) |
| - visualizzazione tabellare dei dati degli Iscritti con le rispettive Iscrizioni |

| VISUALIZZAZIONE SELETTIVA PERSONALE |
|---|
| (accesso tramite navbar in alto alla pagina) |
| - visualizzazione tabellare dei dati degli istruttori e dei responsabili di una data piscina, ordinati in base al CF e all'anno |