|  |  |
| --- | --- |
|  |  |

Basi di Dati e Conoscenza

Progetto A.A. 2019/2020

Sistema di gestione delle piscine comunali di Roma

0257118

Luca Santopadre

**Indice**

[1. Descrizione del Minimondo 3](#_Toc606296459)

[2. Analisi dei Requisiti 4](#_Toc1289394997)

[3. Progettazione concettuale 5](#_Toc2081466291)

[4. Progettazione logica 6](#_Toc2147004904)

[5. Progettazione fisica 8](#_Toc518560220)

[Appendice: Implementazione 9](#_Toc403811585)

# Descrizione del Minimondo

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42 | Si vuole realizzare un sistema informativo per la gestione delle piscine del comune di Roma, tenendo conto delle seguenti informazioni.  Le piscine sono identificate univocamente attraverso il nome. Per ciascuna sono inoltre noti l'indirizzo, un numero di telefono, ed il nome di un responsabile. Nel caso la piscina disponga anche di una vasca all'aperto, la base dati contiene l'informazione di quando tale vasca è utilizzabile (ad esempio da Marzo a Settembre, da Luglio ad Agosto, ecc.).  Presso le piscine sono organizzati dei corsi; lo stesso tipo di corso può essere svolto presso piscine diverse, eventualmente con modalità differenti. Ciascun corso è pertanto identificato dal nome dell'attività svolta, ad esempio aerobica, acquagym, corso di sincronizzato o corso per gestanti, e dal nome della piscina presso cui tale corso si svolge. Per ciascun corso, svolto presso una certa piscina, è noto il costo, il numero massimo e minimo di partecipanti, in quali giorni della settimana si svolge ed a che ora. Presso ciascuna piscina ogni corso può essere svolto più volte al giorno e più volte durante la settimana.  Il corpo insegnante lavora a rotazione presso le varie piscine. Per ciascun insegnante è noto il codice fiscale, che lo identifica, un nome, il numero di cellulare, se disponibile, e l'elenco delle qualifiche dell'insegnante (ad esempio istruttore di sub, istruttore di aerobica, ecc). All'interno della base dati si vuole tener traccia di tutti gli intervalli di tempo in cui un insegnante ha lavorato presso ciascuna piscina. Non si escluda che lo stesso insegnante possa aver lavorato presso una stessa piscina in intervalli di tempo diversi.  Le piscine possono essere frequentate o da persone che sono iscritte ai corsi, o secondo la modalità ad “ingresso singolo” per svolgere nuoto libero. Tutte le persone che accedono alle piscine comunali sono identificate attraverso il loro codice fiscale ed inoltre sono noti il nome, un indirizzo ed un numero arbitrario di contatti, quali numero di telefono, di cellulare, indirizzo email. Un utilizzatore del sistema di piscine comunali può accedere a più di una piscina, frequentando corsi differenti.  Le persone che sono iscritte ai corsi devono presentare un certificato medico. Pertanto, nel caso la persona sia iscritta ad un corso, il database contiene l'informazione del medico che ha redatto il certificato, la data in cui la persona ha presentato il certificato, l'età della persona, e l'elenco dei corsi a cui è iscritta. Per le persone che hanno fatto solo ingressi sono noti solo la data in cui è stato effettuato l'ultimo ingresso e presso quale piscina.  Gli insegnanti possono visualizzare dei report che mostrano l’orario e il luogo di lavoro, su base mensile/settimanale. Gli addetti alla segreteria possono gestire tutte le informazioni relative agli utenti, ma possono gestire unicamente i corsi legati alla piscina in cui lavorano. Possono inoltre variare il numero di corsi e gli orari della piscina in cui lavorano. Gli addetti comunali possono inserire/modificare le informazioni sulle piscine e sul corpo insegnante. |

# Analisi dei Requisiti

## Identificazione dei termini ambigui e correzioni possibili

|  |  |  |  |
| --- | --- | --- | --- |
| **Linea** | **Termine** | **Nuovo termine** | **Motivo correzione** |
| 13 | Partecipanti | Iscritti al corso | Chiarisco che un partecipante, è un iscritto al corso come da specifiche |
| 28 | Utilizzatore del | Persona che utilizza | Termine ambiguo utilizzatore, sostituito con la parola chiave “persona” presente già nella specifica |
| 35 | Solo ingressi | Solo ingressi singoli | Non è chiaro il concetto di ingresso singolo |
| 40 | Agli utenti | Persone che utilizzano la piscina | Un utente è una persona che utilizza la piscina |

### Specifica disambiguata

|  |
| --- |
| Si vuole realizzare un sistema informativo per la gestione delle piscine del comune di Roma, tenendo conto delle seguenti informazioni.  Le piscine sono identificate univocamente attraverso il nome. Per ciascuna sono inoltre noti l'indirizzo, un numero di telefono, ed il nome di un responsabile. Nel caso la piscina disponga anche di una vasca all'aperto, la base dati contiene l'informazione di quando tale vasca è utilizzabile (ad esempio da Marzo a Settembre, da Luglio ad Agosto, ecc.).  Presso le piscine sono organizzati dei corsi; lo stesso tipo di corso può essere svolto presso piscine diverse, eventualmente con modalità differenti. Ciascun corso è pertanto identificato dal nome dell'attività svolta, ad esempio aerobica, acquagym, corso di sincronizzato o corso per gestanti, e dal nome della piscina presso cui tale corso si svolge. Per ciascun corso, svolto presso una certa piscina, è noto il costo, il numero massimo e minimo di iscritti al corso, in quali giorni della settimana si svolge ed a che ora. Presso ciascuna piscina ogni corso può essere svolto più volte al giorno e più volte durante la settimana.  Il corpo insegnante lavora a rotazione presso le varie piscine. Per ciascun insegnante è noto il codice fiscale, che lo identifica, un nome, il numero di cellulare, se disponibile, e l'elenco delle qualifiche dell'insegnante (ad esempio istruttore di sub, istruttore di aerobica, ecc). All'interno della base dati si vuole tener traccia di tutti gli intervalli di tempo in cui un insegnante ha lavorato presso ciascuna piscina. Non si escluda che lo stesso insegnante possa aver lavorato presso una stessa piscina in intervalli di tempo diversi.  Le piscine possono essere frequentate o da persone che sono iscritte ai corsi, o secondo la modalità ad “ingresso singolo” per svolgere nuoto libero. Tutte le persone che accedono alle piscine comunali sono identificate attraverso il loro codice fiscale ed inoltre sono noti il nome, un indirizzo ed un numero arbitrario di contatti, quali numero di telefono, di cellulare, indirizzo email. Una persona che utilizza il sistema di piscine comunali può accedere a più di una piscina, frequentando corsi differenti.  Le persone che sono iscritte ai corsi devono presentare un certificato medico. Pertanto, nel caso la persona sia iscritta ad un corso, il database contiene l'informazione del medico che ha redatto il certificato, la data in cui la persona ha presentato il certificato, l'età della persona, e l'elenco dei corsi a cui è iscritta. Per le persone che hanno fatto solo ingressi singoli sono noti solo la data in cui è stato effettuato l'ultimo ingresso e presso quale piscina.  Gli insegnanti possono visualizzare dei report che mostrano l’orario e il luogo di lavoro, su base mensile/settimanale. Gli addetti alla segreteria possono gestire tutte le informazioni relative alle persone che utilizzano la piscina, ma possono gestire unicamente i corsi legati alla piscina in cui lavorano. Possono inoltre variare il numero di corsi e gli orari della piscina in cui lavorano. Gli addetti comunali possono inserire/modificare le informazioni sulle piscine e sul corpo insegnante. |

## Glossario dei Termini

|  |  |  |  |
| --- | --- | --- | --- |
| **Termine** | **Descrizione** | **Sinonimi** | **Collegamenti** |
| Piscina | La piscina è intesa come servizio. |  | Corso, Insegnante, Ingresso libero |
| Corso | Corso di nuoto svolto presso le piscine. |  | Piscina, Iscritti al corso, Lezione |
| Lezione | Lezione specifica di un corso, con giorno ed orario. |  | Corso |
| Insegnante | Insegnante che lavora presso una piscina. |  | Piscina, Qualifica |
| Qualifica | Qualifica di un’insegnante |  | Insegnante |
| Persona | Persona che utilizza I servizi della piscina. | Utilizzatore, Utente | Ingresso libero, Iscritti al corso |
| Ingresso libero | Persona che effettua un ingresso libero in piscina |  | Persona, Piscina |
| Iscritti al corso | Persona iscritta ad uno o più corsi. |  | Persona, Corso |
| Addetti alla segreteria | Personale che gestisce le persone che utilizzano la piscina solo presso la quale lavorano. |  | Persone, Iscritti al corso, Ingresso libero, Corso, Lezione |
| Addetti comunali | Personale che gestisce il corpo insegnanti e le piscine. |  | Piscina, Insegnante |

## Raggruppamento dei requisiti in insiemi omogenei

|  |
| --- |
| **Frasi relative alla Piscina** |
| Le piscine sono identificate univocamente attraverso il nome. Per ciascuna sono inoltre noti l'indirizzo, un numero di telefono, ed il nome di un responsabile. Nel caso la piscina disponga anche di una vasca all'aperto, la base dati contiene l'informazione di quando tale vasca è utilizzabile (ad esempio da Marzo a Settembre, da Luglio ad Agosto, ecc.). |

|  |
| --- |
| **Frasi relative al Corso** |
| Presso le piscine sono organizzati dei corsi; lo stesso tipo di corso può essere svolto presso piscine diverse, eventualmente con modalità differenti. Ciascun corso è pertanto identificato dal nome dell'attività svolta, ad esempio aerobica, acquagym, corso di sincronizzato o corso per gestanti, e dal nome della piscina presso cui tale corso si svolge. Per ciascun corso, svolto presso una certa piscina, è noto il costo, il numero massimo e minimo di iscritti al corso, in quali giorni della settimana si svolge ed a che ora. Presso ciascuna piscina ogni corso può essere svolto più volte al giorno e più volte durante la settimana. |

|  |
| --- |
| **Frasi relative agli Insegnanti** |
| Il corpo insegnante lavora a rotazione presso le varie piscine. Per ciascun insegnante è noto il codice fiscale, che lo identifica, un nome, il numero di cellulare, se disponibile, e l'elenco delle qualifiche dell'insegnante (ad esempio istruttore di sub, istruttore di aerobica, ecc). All'interno della base dati si vuole tener traccia di tutti gli intervalli di tempo in cui un insegnante ha lavorato presso ciascuna piscina. Non si escluda che lo stesso insegnante possa aver lavorato presso una stessa piscina in intervalli di tempo diversi. |

|  |
| --- |
| **Frasi relative alle Persone** |
| Le piscine possono essere frequentate o da persone che sono iscritte ai corsi, o secondo la modalità ad “ingresso singolo” per svolgere nuoto libero. Tutte le persone che accedono alle piscine comunali sono identificate attraverso il loro codice fiscale ed inoltre sono noti il nome, un indirizzo ed un numero arbitrario di contatti, quali numero di telefono, di cellulare, indirizzo email. Una persona che utilizza il sistema di piscine comunali può accedere a più di una piscina, frequentando corsi differenti.  Le persone che sono iscritte ai corsi devono presentare un certificato medico. Pertanto, nel caso la persona sia iscritta ad un corso, il database contiene l'informazione del medico che ha redatto il certificato, la data in cui la persona ha presentato il certificato, l'età della persona, e l'elenco dei corsi a cui è iscritta. Per le persone che hanno fatto solo ingressi singoli sono noti solo la data in cui è stato effettuato l'ultimo ingresso e presso quale piscina. |

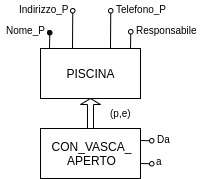
|  |
| --- |
| **Frasi relative ad insegnanti, agli addetti della segreteria e addetti comunali (UTENTI)** |
| Gli insegnanti possono visualizzare dei report che mostrano l’orario e il luogo di lavoro, su base mensile/settimanale.  Gli addetti alla segreteria possono gestire tutte le informazioni relative alle persone che utilizzano la piscina, ma possono gestire unicamente i corsi legati alla piscina in cui lavorano. Possono inoltre variare il numero di corsi e gli orari della piscina in cui lavorano.  Gli addetti comunali possono inserire/modificare le informazioni sulle piscine e sul corpo insegnante. |

# Progettazione concettuale

## Costruzione dello schema E-R

### Per la costruzione dello schema E-R ho adottato una tecnica mista, partendo comunque dai concetti base estrapolati dalla specifica e dettagliati passo passo.

Come primo passo ho evidenziato il concetto di **PISCINA**, costruendo l’entità identificata dall’attributo Nome\_P, dettagliata con I suoi attributi e aggiungendo la specializzazione parziale ed esclusiva **CON\_VASCA\_APERTO** perchè non tutti I padri hanno la specializzazione.

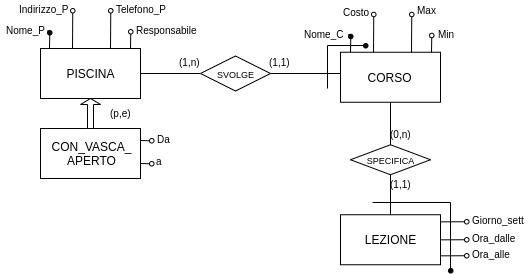


Ho aggiunto poi l’entità debole **CORSO** che ha bisogno dell’identificatore esterno Nome\_P (nome della piscina), per essere identificata unicamente.

L’entità corso ha come chiave interna il Nome\_C (nome del corso) e I suoi attributi.

La relazione **SVOLGE** con l’entità **PISCINA** è di tipo uno a molti, perchè presso una piscina possono essere svolti più corsi, ma un corso, cosi come definito con identificatore esterno, per definizione, può essere svolto solo su un unica piscina.

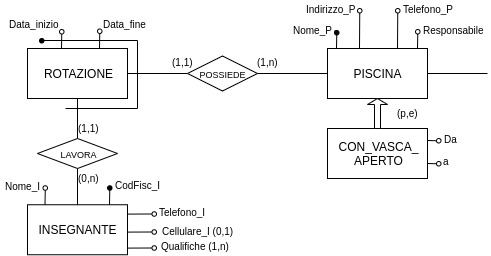
Ho inserito poi l’entità **LEZIONE** con relazione **SPECIFICA** verso CORSO di tipo uno a molti perchè una lezione è specifica di un corso ma un corso può avere più lezioni.

L’entità lezione è identificata da Giorno\_sett, Ora\_dalle, Ora\_alle, e da identificatore esterno Nome\_C e Nome\_P, questo per permettere ad un corso di essere svolto più volte durante il giorno e più volte alla settimana.

Come passo successivo ho analizzato le frasi relative agli insegnanti, ho quindi aggiunto l’entità **INSEGNANTE** indicizzata dal Codice Fiscale (CodFisc\_I) e gli attributi nome, telefono, cellulare(Nullable) e Qualifiche (1,n).

Ho quindi creato l’associazione uno a molti **LAVORA** con l’entità **ROTAZIONE** che mantiene le informazioni di Data\_inizio e Data\_fine di una rotazione di un insegnante presso una certa piscina. L’entità ROTAZIONE è in relazione uno a molti, tramite **POSSIEDE**, con l’entità PISCINA.

L’entità ROTAZIONE per essere definita univocamente ha bisogno degli identificatori esterni Nome\_P (Nome della piscina) e CodFisc\_I (Codice fiscale insegnante).



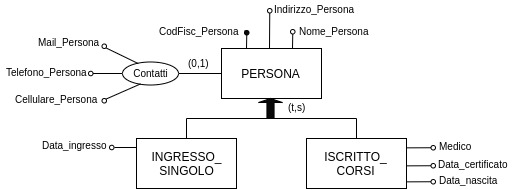
Il passo successivo è stato quello di analizzare le frasi relative alle persone che utilizzano le piscine.

Ho creato quindi l’entità **PERSONA** identificata da CodFisc\_Persona (Codice fiscale della persona) , ed ho aggiunto I suoi attributi Indirizzio\_Persona, Nome\_Persona ed ho creato un attributo composto contatti che ha cardinalità (0,1).

Per distinguere le persone iscritte ai corsi, da chi effettua ingressi singoli, ho creato le specializzazioni **INGRESSO\_SINGOLO** e **ISCRITTO\_CORSI** dell’entità PERSONAdi tipo totale ed sovrapposta. Totale perchè ogni persona che accede alle piscine è registrata nel sistema, e sovrapposta perchè ad esempio una persona iscritta ad un corso presso una piscina può comunque effettuare un ingresso libero presso una qualunque piscina del sistema.

Ho aggiunto l’attributo Data\_Ingresso alla specializzazione INGRESSO\_SINGOLO per mantenere l’ultimo ingresso effettuato da una persona.

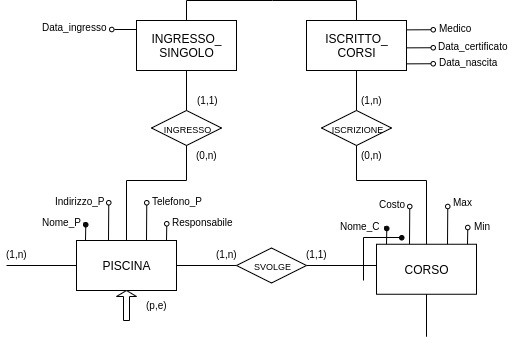
Ho aggiunto gli attributi Medico, Data\_Certificato e Data\_Nascita alla specializzazione ISCRITTO\_CORSI per mantenere informazioni relative al certificato medico.



Come ultimo passaggio ho agganciato le relazioni tra persone , piscine e corsi.

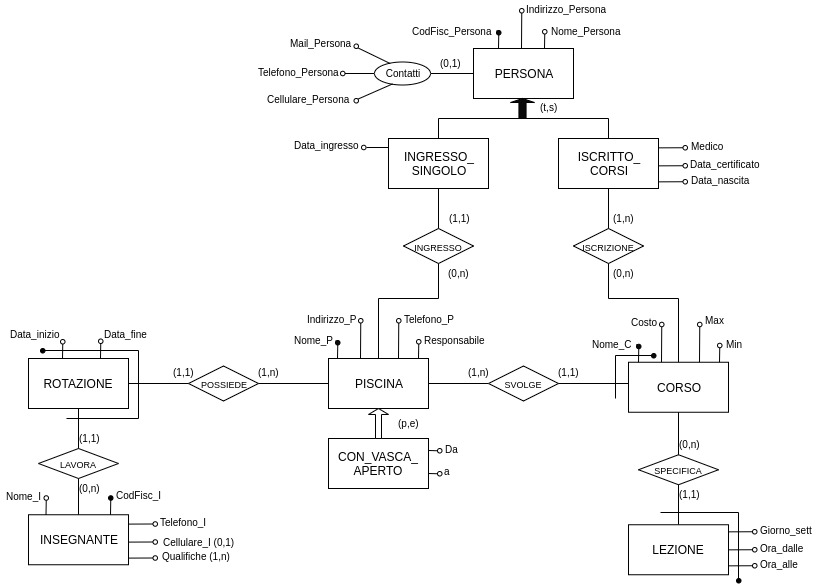
Ho creato quindi la relazione **INGRESSO** di tipo uno a molti tra INGRESSO\_SINGOLO e PISCINA. La relazione è di tipo uno a molti perchè per una persona, il database, mantiene informazioni sull’ultimo ingresso libero effettuato mentre ogni piscina può avere zero o al più n ingressi liberi.

La relazione **ISCRIZIONE** invece lega l’entità ISCRITTO\_CORSI e CORSO. Questa relazione è di tipo molti a molti perchè un iscritto può avere iscrizioni su più corsi ed il database vuole mantenere la lista di tutti I corsi su cui è iscritto, d’altra parte un corso può avere zero o al più n persone iscritte.



### Integrazione finale

Durante la stesura dello schema E-R sono sorti conflitti su nomi che ho risolto rendendoli univoci associandoli all’entità a riguardo. Come ad esempio nome della piscina e nome del corso che sono diventati Nome\_P e Nome\_C .



## Regole aziendali

## Un insegnante NON PUO’ avere date in rotazione che si sovrappongono.

Una persona iscritta ad almeno un corso PUO’ effettuare ingressi singoli.

## Dizionario dei dati

|  |  |  |  |
| --- | --- | --- | --- |
| **Entità** | **Descrizione** | **Attributi** | **Identificatori** |
| PISCINA | Piscina del sistema comunale di Roma. | Nome\_P, Telefono\_P, Indirizzo\_P, Responsabile | Nome\_P |
| CON\_VASCA\_APERTO | Specializzazione di Piscina | Nome\_P,  Telefono\_P,  Indirizzo\_P,  Responsabile,  Da, a | Nome\_P |
| CORSO | Corso di nuoto che si svolge presso una piscina, può avere un numero massimo e minimo di iscritti. | Nome\_C,  Costo,  Max, Min | Nome\_C,  Nome\_P |
| LEZIONE | Specifica l’orario ed il giorno della settimana in cui si svolge un corso. | Giorno\_sett,  Ora\_dalle,  Ora\_alle | Giorno\_sett,  Ora\_dalle,  Ora\_alle,  Nome\_C,  Nome\_P |
| INSEGNANTE | Insegnante di nuoto che lavora a rotazione presso le varie piscine del sistema comunale di Roma. | CodFisc\_I,  Nome\_I,  Telefono\_I,  Cellulare\_I (0,1),  Qualifiche (1,n) | CodFisc\_I |
| ROTAZIONE | Specifica una rotazione di un insegnante presso una certa piscina, con data inizio e data fine. Memorizza I periodi in cui un insegnante ha lavorato e su quale piscina. | Data\_inizio,  Data\_fine | Data\_inizio,  Data\_fine,  CodFisc\_I,  Nome\_P |
| PERSONA | Persona utilizzatore del sistema di piscine comunali. | CodFisc\_Persona,  Indirizzo\_Persona,  Nome\_Persona,  Mail\_Persona (0,1),  Telefono\_Persona (0,1),  Cellulare\_Persona (0,1) | CodFisc\_Persona |
| INGRESSO\_SINGOLO | Specializzazione di PERSONA. | CodFisc\_Persona,  Indirizzo\_Persona,  Nome\_Persona,  Mail\_Persona (0,1),  Telefono\_Persona (0,1),  Cellulare\_Persona (0,1) | CodFisc\_Persona |
| ISCRITTO\_CORSI | Specializzazione di PERSONA | CodFisc\_Persona,  Indirizzo\_Persona,  Nome\_Persona,  Mail\_Persona (0,1),  Telefono\_Persona (0,1),  Cellulare\_Persona (0,1) | CodFisc\_Persona |

|  |  |  |  |
| --- | --- | --- | --- |
| **Relazioni** | **Descrizione** | **Componenti** | **Attributi** |
| SVOLGE | Indica che in una piscina si svolgono corsi. | PISCINA,  CORSO |  |
| SPECIFICA | Sta ad indicare che un corso ha da specificare una o più lezioni. | CORSO,  LEZIONE |  |
| LAVORA | Indica che un insegnante può lavorare su una o più rotazioni delle piscine. | INSEGNANTE,  ROTAZIONE |  |
| POSSIEDE | Indica che una piscina è in possesso di una o più rotazioni | PISCINA,  ROTAZIONE |  |
| INGRESSO | Mette in relazione la piscina ed una persona che effettua ingresso singolo | PISCINA,  INGRESSO\_SINGOLO |  |
| ISCRIZIONE | Relaziona I corsi alle persone iscritte | CORSO,  ISCRITTO\_CORSI |  |

# Progettazione logica

## Volume dei dati

|  |  |  |
| --- | --- | --- |
| **Concetto nello schema** | **Tipo[[1]](#footnote-2)** | **Volume atteso** |
| PISCINA | E | 20 |
| CON\_VASCA\_APERTO | E | 5 |
| CORSO | E | 10(tipi corso)\*20(ogni piscina) = 200 |
| LEZIONE | E | 3(volte a sett.)\*200(corsi)= 600 |
| INSEGNANTE | E | 60 |
| ROTAZIONE | E | 12(1 al mese)\*20(piscine)\*60(insegnanit)= 14.400 |
| PERSONA | E | 200.000 |
| INGRESSO\_SINGOLO | E | 180.000 |
| ISCRITTO\_CORSI | E | 50.000 |
| SVOLGE | R | corso->200 |
| SPECIFICA | R | Lezione->600 |
| LAVORA | R | Rotazione->14.400 |
| POSSIEDE | R | Rotazione→14.400 |
| INGRESSO | R | Ingresso\_singolo->180.000 |
| ISCRIZIONE | R | 100.000 |

## Tavola delle operazioni

|  |  |  |
| --- | --- | --- |
| **Cod.** | **Descrizione** | **Frequenza attesa** |
| 1 | Visualizza report insegnante(mensile + settimanale) | 1000 / mese |
|  |  |  |
| 2 | Inserisci persona iscritta corso | 1000 / mese |
| 3 | Modifica / Elimina persona iscritta corso | 1000 / mese |
| 4 | Inserisci persona ingresso libero | 5000 / mese |
| 5 | Modifica / Elimina persona ingresso libero | 2000 / mese |
| 6 | Inserisci corso | 50 / mese |
| 7 | Modifica/ Elimina corso | 1000 / mese |
| 8 | Inserisci lezione | 100 / mese |
| 9 | Modifica / Elimina lezione | 2000 / mese |
|  |  |  |
| 10 | Inserisci piscina | 1 / anno |
| 11 | Modifica / Elimina piscina | 1 / mese |
| 12 | Inserisci insegnante | 10 / mese |
| 13 | Modifica / Elimina insegnante | 20 / mese |
| 14 | Inserisci rotazione | 1000 / mese |
| 15 | Modifica / Elimina rotazione | 2000 /mese |
|  |  |  |
| 16 | Modifica/ Elimina persona | 2000/mese |
|  |  |  |

## Costo delle operazioni

Cod.1 - Visualizza report insegnante

Ho un volume di 14.400 rotazioni, 60 insegnanti, quindi in media 240 rotazioni per ogni insegnante, ho poi 20 piscine e 200 corsi, quindi in media 10 corsi a piscina.

Accedo in lettura su rotazione perchè ho gia il codice fiscale insegnante. Poi accedo in lettura su piscina per trovare il luogo di lavoro.

Accedo quindi a lezione in lettura e a corso in lettura. Quindi ho 4 letture

Costo=4(L) \* 1000/mese = 4000

Cod.2 - Inserisci persona iscritta corso

Effettuo una scrittura sull’entità persona ed una su iscritto corso.

Costo=2(S) \* 1000/mese = 2\*2 \* 1000/mese = 4000

Cod.3 - Modifica / Elimina persona iscritta corso

Effettuo un accesso in lettura su iscritto corso, quindi una scrittura su iscritto corso per modificare.

Costo= ( 1(L) + 1(S) ) \* 1000/mese = ( 1+ 2) \*1000 = 4000

Cod.4 - Inserisci persona ingresso libero

Effettuo una scrittura sull’entità Persona.

Costo= 1(S) \* 5000/mese = 2 \* 5000 = 10000

Cod.5 - Modifica / Elimina persona ingresso libero

Effettuo un accesso in lettura ed uno in scrittura sull’entità Persona.

Costo= ( 1(L) + 1(S) ) \* 2000/mese = ( 1+2) \* 2000 = 6000

Cod.6 - Inserisci corso

Effettuo prima una lettura per il nome della piscina e poi scrivo:

Costo=( 1(L) + 1(S) ) \* 50/mese = ( 1+2) \* 50 = 150

Cod.7 - Modifica/ Elimina corso

Effettuo una lettura ed una scrittura

Costo=( 1(L) + 1(S) ) \* 1000/mese = ( 1+2) \* 1000 = 3000

Cod.8 - Inserisci lezione

Effettuo prima una lettura per il codice del corso e poi inserisco:

Costo=( 1(L) + 1(S) ) \* 100/mese = (1+2) \* 100 = 300

Cod.9 - Modifica / Elimina lezione

Effettuo lettura e poi scrittura per modifica

Costo=( 1(L) + 1(S) ) \* 2000/mese = (1+2) \* 2000 = 6000

Cod.10 - Inserisci piscina

Effettuo una sola scrittura:

Costo=1(S) \* 1/anno = 2 \* 1/12 = 1/6

Cod.11 - Modifica / Elimina piscina

Eseguo prima una lettura e poi una scrittura per la modifica:

Costo=( 1(L) + 1(S) ) \* 1/mese = (1+2) \* 1 = 3

Cod.12 - Inserisci insegnante

Effettuo una scrittura:

Costo=1(S) \* 10/mese = 2 \* 10 = 20

Cod.13 - Modifica / Elimina insegnante

Effettuo una lettura ed una scrittura per modificare:

Costo=( 1(L) + 1(S) ) \* 20/mese = (1+2) \* 20 = 60

Cod.14 - Inserisci rotazione

Effettuo una lettura su insegnante per trovare il codice fiscale, una per la piscina di riferimento e poi effettuo la scrittura:

Costo=( 2(L) + 1(S) ) \* 1000/mese = (2+2) \* 1000 = 4000

Cod.15 - Modifica / Elimina rotazione

Effettuo una lettura e poi una scrittura.

Costo=( 1(L) + 1(S) ) \*2000/mese = ( 1 + 2) \* 2000 = 6000

Cod.16 - Modifica / Elimina persona

Effettuo una lettura e poi una scrittura sull’entità persona.

Costo=( 1(L) + 1(S) ) \*2000/mese = ( 1 + 2) \* 2000 = 6000

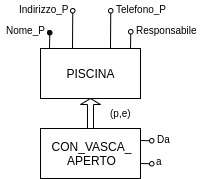
## Ristrutturazione dello schema E-R

**1) Analisi delle ridondanze**

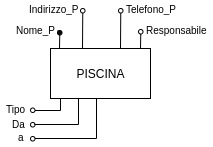
Da un’analisi dello schema E-R non risultano ridondanze su attributi e relazioni.

**2) Eliminazione delle generalizzazioni**

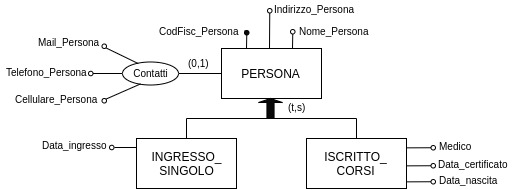
**2.1) Generalizzazione PISCINA**

****

Questa generalizzazione è parziale ed esclusiva, dato che non tutte le piscine hanno una vasca all’aperto. Su questa tabella ho un volume di dati basso elimino quindi la generalizzazione aggiungendo gli attributi dell’entità figlia, all’entità padre, e inserendo l’attributo Tipo = “con vasca all’aperto” all’entità piscina.



**2.2) Generalizzazione PERSONA**



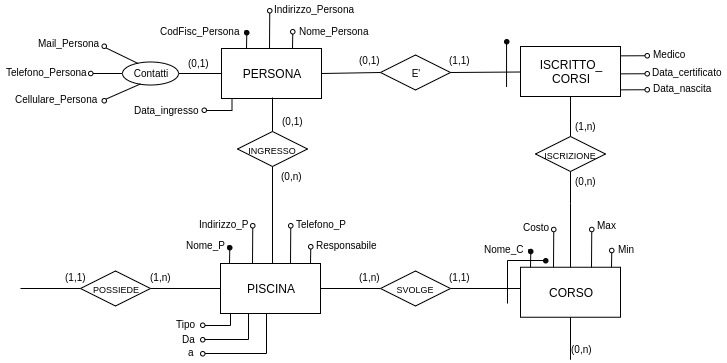
In questo caso la generalizzazione è di tipo totale e sovrapposta.

Ho eliminato questa generalizzazione adottando una strategia mista:

1)Sfruttando il fatto che il database vuole mantenere solo l’ultimo ingresso di chi effettua l’ingresso singolo ed il fatto che dall’analisi il numero di ingressi singoli è molto elevato(vicino al numero di persone totali) ho mantenuto l’entità padre PERSONA ed eliminato la generalizzazione INGRESSO\_SINGOLO, ho quindi spostato l’attributo Data\_ingresso sull’entità PERSONA.

Ho quindi associato INGRESSO all’entità PERSONA. L’associazione rimane di tipo uno a molti ma in questo caso ho cardinalità minima 0 nel verso PERSONA→PISCINA.

2)Ho mantenuto l’entità ISCRITTO\_CORSI ed ho inserito l’associazione E’ di tipo uno a uno con l’entità PERSONA.

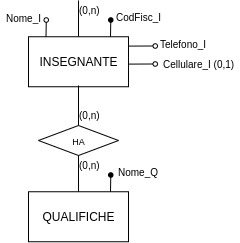


**3) Scelta identificatori primari**

* PISCINA: Nome\_P
* CORSO: Nome\_C, Nome\_P
* LEZIONE: Nome\_C, Nome\_P, Giorno\_sett, Ora\_dalle, Ora\_alle
* INSEGNANTE: CodFisc\_I
* ROTAZIONE: CodFisc\_I, Nome\_P, Data\_inizio, Data\_fine
* PERSONA: CodFisc\_Persona
* ISCRITTO\_CORSI: CodFisc\_Persona

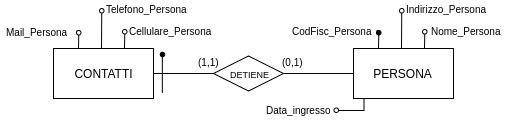
## Trasformazione di attributi e identificatori

**1) Attributo multivalore qualifiche di insegnante**

L’attributo Qualifiche dell’entità INSEGNANTE viene eliminato creando una nuova entità **QUALIFICHE** identificata da Nome\_Q (nome della qualifica). L’associazione **HA** è di tipo molti a molti perchè un insegnante può avere più qualifiche, e una qualifica può essere in possesso di più insegnanti.

**2) Attributo composto contatti di persona**

L’attributo composto dell’entità PERSONA è stato sostituito da un’opportuna nuova entità **CONTATTI**. E’ stata operata questa scelta perchè questi attributi sono arbitrari e non sono molto significativi a livello applicativo.. L’associazione adottata **DETIENE** è di tipo uno a uno come da specifiche.



## Traduzione di entità e associazioni

* **PISCINA** ( Nome\_P, Indirizzo\_P, Telefono\_P, Responsabile, Tipo, Da, a )
* **CORSO** ( Nome\_C, Nome\_Piscina, Costo, Max, Min )

Con vincolo integrità referenziale tra Nome\_Piscina e PISCINA

* **LEZIONE** ( Nome\_Corso, Nome\_Piscina\_Corso, Giorno\_sett, Ora\_dalle, Ora\_alle )

Con vincolo integrità referenziale tra (Nome\_Corso, Nome\_Piscina\_Corso) e CORSO

* **INSEGNANTE** ( CodFisc\_I, Nome\_I, Telefono\_I, Cellulare\_I )
* **QUALIFICA** ( Nome\_Q )
* **INSEGNANTE\_HA\_QUALIFICA** ( CodFisc\_Insegnante, Nome\_Qualifica )

Con vincoli di integrità referenziale tra:

* + - * + CodFisc\_Insegnante e INSEGNANTE
        + Nome\_Qualifica e QUALIFICA
* **ROTAZIONE** ( Data\_inizio, Data\_fine, CodFiscale\_Insegnante, Nome\_Piscina\_R )

Con vincoli di integrità referenziale tra:

* + - * + CodFiscale\_Insegnante e INSEGNANTE
        + Nome\_Piscina\_R e PISCINA
* **PERSONA** ( CodFisc\_Persona, Indirizzo\_Persona, Nome\_Persona, Data\_ingresso, Nome\_Piscina\_P)

Con vincoli di integrità referenziale tra Nome\_Piscina\_P e PISCINA

* **CONTATTI** ( CodFisc\_Persona\_C, Mail\_Persona, Cellulare\_Persona, Telefono\_Persona )

Con vincolo di integrità referenziale tra CodFisc\_Persona\_C e PERSONA

* **ISCRITTO\_CORSI** ( CodFisc\_Persona\_IC, Medico, Data\_certificato, Data\_nascita )

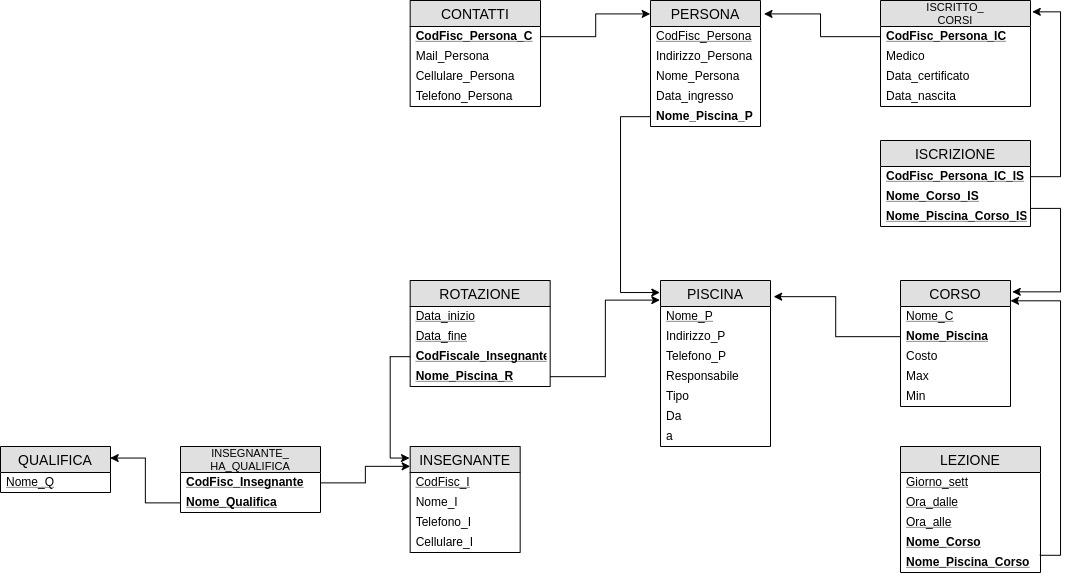
Con vincolo di integrità referenzaile tra CodFisc\_Persona\_IC e PERSONA

* **ISCRIZIONE** ( CodFisc\_Persona\_IC\_IS, Nome\_Corso\_IS, Nome\_Piscina\_Corso\_IS )

Con vincoli di integrità referenziale tra:

* + - * + CodFisc\_Persona\_IC\_IS e ISCRITTO\_CORSI
        + ( Nome\_Corso\_IS, Nome\_Piscina\_Corso\_IS) e CORSO

**Rappresentazione grafica modello relazionale completo**



Nota: **Grassetto = ForeignKey**

Sottolineato = PrimaryKey

## Normalizzazione del modello relazionale

Nel modello relazionale precedentemente illustrato non risultano dipendenze funzionali non banali.

**1NF**

Il modello relazionale definito risulta essere in 1NF :

ogni attributo è definito su un dominio di valori atomici (L’entità PERSONA e l’entità INSEGNANTE hanno come attributi il Nome, che può essere suddiviso in due attributi Nome e Cognome, ma a scopo del progetto non risultano fondamentali, perciò ho ritenuto opportuno mantenere solamente l’attributo Nome.), ogni relazione ha una chiave che lo identifica e non ci sono nomi di attributi ripetuti.

**2NF**

L’unica relazione che ha chiave composta è CORSO. Gli attributi non chiave Costo, Max, Min dipendono funzionalmente dall’intera chiave composta Nome\_C, Nome\_Piscina. Quindi il modello relazionale descritto in precedenza si trova anche in 2NF.

**3NF**

Per la 3 forma normale è richiesta l’assenza di dipendenza transitiva, che nel modello proposto è già verificata.

# Progettazione fisica

## Utenti e privilegi

## Utente “insegnante”:

Questo utente effettua solamente dei report per visualizzare orario e luogo di lavoro, quindi ho assegnato I seguenti privilegi :

* Read-only sulle tabelle ( Insegnante, Rotazione, Piscina, Corso, Lezione ).
* Esecuzione su procedure che generano questi report.

## Utente “**addetto\_segreteria**”:

Gli addetti segreteria possono gestire tutte le informazioni relative agli utenti, ma possono gestire unicamente i corsi legati alla piscina in cui lavorano. Possono inoltre variare il numero di corsi e gli orari della piscina in cui lavorano. Pertanto ho assegnato I seguenti privilegi :

* Tutti I privilegi su tabelle ( Persona, Contatti, Iscritto\_Corsi, Iscrizione, Corso, Lezione ) perchè deve essere in grado di modificare queste tabelle.
* Read-only su tabella Piscina perchè da questa si ricava solamente la piscina in cui lavora
* Esecuzione sulle procedure per interagire con le tabelle.

## Utente “**addetto\_comunale**”:

Gli addetti comunali possono inserire/modificare le informazioni sulle piscine e sul corpo insegnante , quindi ho assegnato I seguenti privilegi:

* Tutti I privilegi su tabelle ( Qualifica, Insegnante\_Ha\_Qualifica, Insegnante, Rotazione, Piscina ) .
* Esecuzione sulle procedure per interagire con le tabelle.

## Strutture di memorizzazione

|  |  |  |
| --- | --- | --- |
| **Tabella Piscina** | | |
| **Attributo** | **Tipo di dato** | **Attributi[[2]](#footnote-3)** |
| Nome\_P | VARCHAR(50) | PK |
| Indirizzo\_P | VARCHAR(50) | NN |
| Telefono\_P | VARCHAR(20) | NN |
| Responsabile | VARCHAR(30) | NN |
| Tipo | BOOLEAN | NN |
| Da | VARCHAR(10) |  |
| a | VARCHAR(10) |  |

|  |  |  |
| --- | --- | --- |
| **Tabella Corso** | | |
| **Attributo** | **Tipo di dato** | **Attributi** |
| Nome\_C | VARCHAR(30) |  |
| Nome\_Piscina | VARCHAR(50) |  |
| Costo | DECIMAL(5,2) | NN |
| Max | INT | NN |
| Min | INT | NN |
| Nome\_C, Nome\_Piscina |  | PK |

|  |  |  |
| --- | --- | --- |
| **Tabella Lezione** | | |
| **Attributo** | **Tipo di dato** | **Attributi** |
| Giorno\_sett | INT |  |
| Ora\_dalle | TIME |  |
| Ora\_alle | TIME |  |
| Nome\_Corso | VARCHAR(30) |  |
| Nome\_Piscina\_Corso | VARCHAR(0) |  |
| Giorno\_sett, Ora\_dalle, Ora\_alle, Nome\_Corso, Nome\_Piscina\_Corso |  | PK |

|  |  |  |
| --- | --- | --- |
| **Tabella Insegnante** | | |
| **Attributo** | **Tipo di dato** | **Attributi** |
| CodFisc\_I | CHAR(16) | PK |
| Nome\_I | VARCHAR(30) | NN |
| Telefono\_I | VARCHAR(20) |  |
| Cellulare\_I | VARCHAR(20) |  |

|  |  |  |
| --- | --- | --- |
| **Tabella Insegnante\_Ha\_Qualifica** | | |
| **Attributo** | **Tipo di dato** | **Attributi** |
| CodFisc\_Insegnante | CHAR(16) |  |
| Nome\_Qualifica | VARCHAR(30) |  |
| CodFisc\_Insegnante, Nome\_Qualifica |  | PK |

|  |  |  |
| --- | --- | --- |
| **Tabella Qualifica** | | |
| **Attributo** | **Tipo di dato** | **Attributi** |
| Nome\_Q | VARCHAR(30) | PK |

|  |  |  |
| --- | --- | --- |
| **Tabella Rotazione** | | |
| **Attributo** | **Tipo di dato** | **Attributi** |
| Data\_inizio | DATE |  |
| Data\_fine | DATE |  |
| CodFiscale\_Insegnante | CHAR(16) |  |
| Nome\_Piscina\_R | VARCHAR(50) |  |
| Data\_inizio, Data\_fine, CodFiscale\_Insegnante, Nome\_Piscina\_R |  | PK |

|  |  |  |
| --- | --- | --- |
| **Tabella Persona** | | |
| **Attributo** | **Tipo di dato** | **Attributi** |
| CodFisc\_Persona | CHAR(16) | PK |
| Indirizzo\_Persona | VARCHAR(50) | NN |
| Nome\_Persona | VARCHAR(30) | NN |
| Data\_ingresso | DATE |  |
| Nome\_Piscina\_P | VARCHAR(50) |  |

|  |  |  |
| --- | --- | --- |
| **Tabella Contatti** | | |
| **Attributo** | **Tipo di dato** | **Attributi** |
| CodFisc\_Persona\_P | CHAR(16) | PK |
| Mail\_Persona | VARCHAR(50) |  |
| Cellulare\_Persona | VARCHAR(20) |  |
| Telefono\_Persona | VARCHAR(20) |  |

|  |  |  |
| --- | --- | --- |
| **Tabella Iscritto\_Corsi** | | |
| **Attributo** | **Tipo di dato** | **Attributi** |
| CodFisc\_Persona\_IC | CHAR(16) | PK |
| Medico | VARCHAR(30) | NN |
| Data\_certificato | DATE | NN |
| Data\_nascita | DATE | NN |

|  |  |  |
| --- | --- | --- |
| **Tabella Iscrizione** | | |
| **Attributo** | **Tipo di dato** | **Attributi** |
| CodFisc\_Persona\_IC\_IS | CHAR(16) |  |
| Nome\_Corso\_IS | VARCHAR(30) |  |
| Nome\_Piscina\_Corso\_IS | VARCHAR(50) |  |
| CodFisc\_Persona\_IC\_IS, Nome\_Corso\_IS,  Nome\_Piscina\_Corso\_IS |  | PK |

## Indici

|  |  |
| --- | --- |
| **Tabella Piscina** |  |
| **Indice PRIMARY** | **Tipo[[3]](#footnote-4):** |
| Nome\_P | PK |

|  |  |
| --- | --- |
| **Tabella Corso** |  |
| **Indice PRIMARY** | **Tipo:** |
| Nome\_C, Nome\_Piscina | PK |

|  |  |
| --- | --- |
| **Tabella Lezione** |  |
| **Indice PRIMARY** | **Tipo:** |
| Giorno\_sett, Ora\_dalle, Ora\_alle, Nome\_Corso, Nome\_Piscina\_Corso | PK |

|  |  |
| --- | --- |
| **Tabella Insegnante** |  |
| **Indice PRIMARY** | **Tipo:** |
| CodFisc\_I | PK |

|  |  |
| --- | --- |
| **Tabella Insegnante\_Ha\_Qualifica** |  |
| **Indice PRIMARY** | **Tipo:** |
| CodFisc\_Insegnante, Nome\_Qualifica | PK |

|  |  |
| --- | --- |
| **Tabella Qualifica** |  |
| **Indice PRIMARY** | **Tipo:** |
| Nome\_Q | PK |

|  |  |
| --- | --- |
| **Tabella Rotazione** |  |
| **Indice PRIMARY** | **Tipo:** |
| Data\_inizio, Data\_fine, CodFiscale\_Insegnante, Nome\_Piscina\_R | PK |

|  |  |
| --- | --- |
| **Tabella Persona** |  |
| **Indice PRIMARY** | **Tipo:** |
| CodFisc\_Persona | PK |

|  |  |
| --- | --- |
| **Tabella Contatti** |  |
| **Indice PRIMARY** | **Tipo:** |
| CodFisc\_Persona\_P | PK |

|  |  |
| --- | --- |
| **Tabella Iscritto\_Corsi** |  |
| **Indice PRIMARY** | **Tipo:** |
| CodFisc\_Persona\_IC | PK |

|  |  |
| --- | --- |
| **Tabella Iscrizione** |  |
| **Indice PRIMARY** | **Tipo:** |
| CodFisc\_Persona\_IC\_IS, Nome\_Corso\_IS,  Nome\_Piscina\_Corso\_IS | PK |

--------------------------------------------------------------------------------------------------------

## Trigger

Per realizzare check sul database ho utilizzato I seguenti trigger:

* **check\_insert\_ingresso\_libero\_null**
  + Trigger sull’evento before insert sulla tabella Persona.
  + Verifica se nella nuova tupla inserita, gli attributi Data\_ingresso e Nome\_Piscina\_P sono entrambi nulli o entrambi completi, per gli ingressi singoli.

CREATE TRIGGER check\_insert\_ingresso\_libero\_null BEFORE INSERT ON `Piscine-Roma-DB`.Persona

FOR EACH ROW

BEGIN

IF (NEW.Nome\_Piscina\_P IS NULL

AND NEW.Data\_ingresso IS NOT NULL)

OR

(NEW.Nome\_Piscina\_P IS NOT NULL

AND NEW.Data\_ingresso IS NULL)

THEN

SET NEW.Nome\_Piscina\_P = NULL;

SET NEW.Data\_ingresso = NULL;

SIGNAL SQLSTATE '02000' SET MESSAGE\_TEXT = 'Attenzione data ingresso e nome piscina devono essere entrambi inseriti o entrambi nulli';

END IF;

END $$

* **check\_update\_ingresso\_libero\_null**
  + Trigger sull’evento before update sulla tabella Persona.
  + Verifica se nella nuova tupla aggiornata, gli attributi Data\_ingresso e Nome\_Piscina\_P sono entrambi nulli o entrambi completi per gli ingressi singoli.

CREATE TRIGGER check\_update\_ingresso\_libero\_null BEFORE UPDATE ON `Piscine-Roma-DB`.Persona

FOR EACH ROW

BEGIN

IF (NEW.Nome\_Piscina\_P IS NULL

AND NEW.Data\_ingresso IS NOT NULL)

OR

(NEW.Nome\_Piscina\_P IS NOT NULL

AND NEW.Data\_ingresso IS NULL)

THEN

SET NEW.Nome\_Piscina\_P = NULL;

SET NEW.Data\_ingresso = NULL;

SIGNAL SQLSTATE '02000' SET MESSAGE\_TEXT = 'Attenzione data ingresso e nome piscina devono essere entrambi inseriti o entrambi nulli';

END IF;

END $$

* **check\_insert\_piscina\_tipo\_null**
  + Trigger sull’evento before insert sulla tabella Piscina.
  + Verifica nella nuova tupla inserita, se la piscina ha la vasca all’apero allora ha bisogno del periodo di disponibilità, altrimenti devono essere null .

CREATE TRIGGER check\_insert\_piscina\_tipo\_null BEFORE INSERT ON `Piscine-Roma-DB`.Piscina

FOR EACH ROW

BEGIN

IF ( NEW.Tipo = 0 AND (NEW.Da IS NOT NULL OR NEW.a IS NOT NULL) )

THEN

SET NEW.Da = NULL;

SET NEW.a = NULL;

SIGNAL SQLSTATE '02000' SET MESSAGE\_TEXT = 'Attenzione se la piscina non ha la vasca all\'aperto,non può esserci il periodo \'Da -> a\'. ';

ELSEIF ( NEW.Tipo = 1 AND (NEW.Da IS NULL OR NEW.a IS NULL) )

THEN

SIGNAL SQLSTATE '02000' SET MESSAGE\_TEXT = 'Attenzione inserire il periodo \'Da -> a\'.';

END IF;

END $$

* **check\_update\_piscina\_tipo\_null**
  + Trigger sull’evento before update sulla tabella Piscina.
  + Verifica nella nuova tupla aggiornata, se la piscina ha la vasca all’apero allora ha bisogno del periodo di disponibilità, altrimenti devono essere null .

CREATE TRIGGER check\_update\_piscina\_tipo\_null BEFORE UPDATE ON `Piscine-Roma-DB`.Piscina

FOR EACH ROW

BEGIN

IF ( NEW.Tipo = 0 AND (NEW.Da IS NOT NULL OR NEW.a IS NOT NULL) )

THEN

SET NEW.Da = NULL;

SET NEW.a = NULL;

SIGNAL SQLSTATE '02000' SET MESSAGE\_TEXT = 'Attenzione se la piscina non ha la vasca all\'aperto,non può esserci il periodo \'Da -> a\'. ';

ELSEIF ( NEW.Tipo = 1 AND (NEW.Da IS NULL OR NEW.a IS NULL) )

THEN

SIGNAL SQLSTATE '02000' SET MESSAGE\_TEXT = 'Attenzione inserire il periodo \'Da -> a\'.';

END IF;

END $$

* **check\_insert\_partecipanti\_corso**
  + Trigger sull’evento before insert sulla tabella Corso.
  + Verifica nella nuova tupla inserita, se il range di partecipanti è valido.

CREATE TRIGGER check\_insert\_partecipanti\_corso BEFORE INSERT ON `Piscine-Roma-DB`.Corso

FOR EACH ROW

BEGIN

IF ( NEW.Min > NEW.Max )

THEN

SIGNAL SQLSTATE '02000' SET MESSAGE\_TEXT = 'Attenzione inserire valido range partecipanti ';

END IF;

END $$

* **check\_update\_partecipanti\_corso**
  + Trigger sull’evento before update sulla tabella Corso.
  + Verifica nella nuova tupla aggiornata, se il range di partecipanti è valido.

CREATE TRIGGER check\_update\_partecipanti\_corso BEFORE UPDATE ON `Piscine-Roma-DB`.Corso

FOR EACH ROW

BEGIN

IF ( NEW.Min > NEW.Max )

THEN

SIGNAL SQLSTATE '02000' SET MESSAGE\_TEXT = 'Attenzione inserire valido range partecipanti ';

END IF;

END $$

* **check\_insert\_ora\_lezione**
  + Trigger sull’evento before insert sulla tabella Lezione.
  + Verifica nella nuova tupla inserita, se il range Ora\_dalle→ Ora\_alle è valido.

CREATE TRIGGER check\_insert\_ora\_lezione BEFORE INSERT ON `Piscine-Roma-DB`.Lezione

FOR EACH ROW

BEGIN

IF ( NEW.Ora\_dalle > NEW.Ora\_alle )

THEN

SIGNAL SQLSTATE '02000' SET MESSAGE\_TEXT = 'Attenzione inserire valido range per orario lezione ';

END IF;

END $$

* **check\_update\_ora\_lezione**
  + Trigger sull’evento before update sulla tabella Lezione.
  + Verifica nella nuova tupla aggiornata, se il range Ora\_dalle→ Ora\_alle è valido.

CREATE TRIGGER check\_update\_ora\_lezione BEFORE UPDATE ON `Piscine-Roma-DB`.Lezione

FOR EACH ROW

BEGIN

IF ( NEW.Ora\_dalle > NEW.Ora\_alle )

THEN

SIGNAL SQLSTATE '02000' SET MESSAGE\_TEXT = 'Attenzione inserire valido range per orario lezione ';

END IF;

END $$

* **check\_insert\_date\_rotazione**
  + Trigger sull’evento before insert sulla tabella Rotazione.
  + Verifica nella nuova tupla inserita, se la rotazione assegnata all’insegnante comprende un periodo in cui l’insegnante è già occupato in un altra piscina. (“Il corpo insegnante lavora a rotazione presso le varie piscine”)
  + Verifica anche se il range Data\_inizio → Data\_fine sia valido.

CREATE TRIGGER check\_insert\_date\_rotazione BEFORE INSERT ON `Piscine-Roma-DB`.Rotazione

FOR EACH ROW

BEGIN

IF (

SELECT EXISTS(

SELECT \*

FROM Rotazione

WHERE NEW.CodFiscale\_Insegnante = Rotazione.CodFiscale\_Insegnante

AND(

(NEW.Data\_inizio >= Rotazione.Data\_inizio

AND NEW.Data\_inizio <= Rotazione.Data\_fine

)

OR

(NEW.Data\_fine >= Rotazione.Data\_inizio

AND NEW.Data\_fine <= Rotazione.Data\_fine

)

OR

(NEW.Data\_inizio < Rotazione.Data\_inizio

AND NEW.Data\_fine > Rotazione.Data\_fine

)

)

)

)

THEN

SIGNAL SQLSTATE '02000' SET MESSAGE\_TEXT = 'Attenzione esiste già una rotazione assegnata al periodo scelto per questo insegnante';

END IF;

IF ( NEW.Data\_inizio > NEW.Data\_fine )

THEN

SIGNAL SQLSTATE '02000' SET MESSAGE\_TEXT = 'Attenzione la data di inizio non può essere successiva alla data di fine. ';

END IF;

END $$

* **check\_update\_date\_rotazione**
  + Trigger sull’evento before update sulla tabella Rotazione.
  + Verifica nella tupla aggiornata, se la rotazione assegnata all’insegnante comprende un periodo in cui l’insegnante è già occupato in un altra piscina. (“Il corpo insegnante lavora a rotazione presso le varie piscine”)
  + Verifica anche se il range Data\_inizio → Data\_fine sia valido.

CREATE TRIGGER check\_update\_date\_rotazione BEFORE UPDATE ON `Piscine-Roma-DB`.Rotazione

FOR EACH ROW

BEGIN

IF (

SELECT EXISTS(

SELECT \*

FROM Rotazione

WHERE NEW.CodFiscale\_Insegnante = Rotazione.CodFiscale\_Insegnante

AND(

(NEW.Data\_inizio >= Rotazione.Data\_inizio

AND NEW.Data\_inizio <= Rotazione.Data\_fine

)

OR

(NEW.Data\_fine >= Rotazione.Data\_inizio

AND NEW.Data\_fine <= Rotazione.Data\_fine

)

OR

(NEW.Data\_inizio < Rotazione.Data\_inizio

AND NEW.Data\_fine > Rotazione.Data\_fine

)

)

)

)

THEN

SIGNAL SQLSTATE '02000' SET MESSAGE\_TEXT = 'Attenzione esiste già una rotazione assegnata al periodo scelto per questo insegnante';

END IF;

IF ( NEW.Data\_inizio > NEW.Data\_fine )

THEN

SIGNAL SQLSTATE '02000' SET MESSAGE\_TEXT = 'Attenzione la data di inizio non può essere successiva alla data di fine. ';

END IF;

END $$

## Eventi

## Per la realizzazione del DB non ho usato eventi.

## Viste

## Per la realizzazione del DB non ho usato viste.

## Stored Procedures e transazioni

* **is\_CF\_insegnante\_valid(IN CF CHAR(16))**
  + Verifica l’esistenza del codice fiscale nella tabella Insegannte durante la procedura di accesso come utente Insegnante.

CREATE PROCEDURE `Piscine-Roma-DB`.`is\_CF\_insegnante\_valid`(IN CF CHAR(16))

SELECT EXISTS( SELECT CodFisc\_I FROM `Piscine-Roma-DB`.Insegnante i

WHERE i.CodFisc\_I = CF ) as is\_valid\_cf ;

* **report\_insegnante(startDate DATE, giorni INT, CF\_I CHAR(16))**
  + Procedura per implementare il report insegnante sia settimanale che mensile, prende in ingresso la data di inizio report, I giorni dalla data di inizio, ed il codice fiscale dell’insegnante che è acceduto.
  + Inizialmente creo una tabella temporanea Calendario per mantenere tutti I giorni in formato date da startDate a fine report.
  + Successivamente con una SELECT su calendario e JOIN tra le tabelle Insegnante, Rotazione, Piscina, Corso e Lezione eseguo il report.
  + Elimino in fine la tabella temporanea Calendario.

CREATE PROCEDURE `Piscine-Roma-DB`.`report\_insegnante`(startDate DATE, giorni INT, CF\_I CHAR(16))

BEGIN

### INIZIALLIZZO DATI

DECLARE counter INT DEFAULT 1;

DECLARE data\_var DATE DEFAULT startDate;

#DECLARE giorni INT DEFAULT (SELECT DATEDIFF(endDate,startDate)) + 1;

### CREO TABELLA TEMPORANEA CALENDARIO

DROP TABLE IF EXISTS calendario;

CREATE TEMPORARY TABLE calendario(

data\_calendario DATE,

PRIMARY KEY(data\_calendario)

);

### INSERISCO IN CALENDARIO,DATE TRA IL RANGE [startDate,endDate]

WHILE counter <= giorni DO

INSERT INTO calendario(data\_calendario) VALUES(data\_var);

SET counter = counter + 1;

SET data\_var = DATE\_ADD(data\_var,INTERVAL 1 day);

END WHILE;

### GENERA REPORT

SELECT data\_calendario as 'data',DAYNAME(data\_calendario) as 'giorno',

DAY(data\_calendario) as 'giorno\_n',

MONTHNAME(data\_calendario) as 'mese',

YEAR(data\_calendario) as 'anno',

Ora\_dalle as 'dalle',Ora\_alle as 'alle',

Nome\_P as 'piscina',

Indirizzo\_P as 'indirizzo',

Nome\_C as 'corso'

FROM calendario,

Insegnante

INNER JOIN Rotazione ON

Insegnante.CodFisc\_I = Rotazione.CodFiscale\_Insegnante

INNER JOIN Piscina ON

Rotazione.Nome\_Piscina\_R = Piscina.Nome\_P

INNER JOIN Corso ON

Piscina.Nome\_P = Corso.Nome\_Piscina

INNER JOIN Lezione ON

Corso.Nome\_C = Lezione.Nome\_Corso AND

Corso.Nome\_Piscina = Lezione.Nome\_Piscina\_Corso

WHERE

DAYOFWEEK(calendario.data\_calendario) = Lezione.Giorno\_sett

AND

Insegnante.CodFisc\_I=CF\_I

AND

data\_calendario BETWEEN Rotazione.Data\_inizio AND Rotazione.Data\_fine

ORDER BY data\_calendario

;

#### ELIMINA TABELLA TEMPORANEA CALENDARIO

DROP TABLE calendario;

END

* **persona\_inserisci(IN CF\_P CHAR(16), IN indirizzo VARCHAR(50), IN nome VARCHAR(30))**
  + Prende in input il codice fiscale, l’indirizzo e il nome , ed inserisce una nuova persona.

CREATE PROCEDURE `Piscine-Roma-DB`.persona\_inserisci(IN CF\_P CHAR(16), IN indirizzo VARCHAR(50), IN nome VARCHAR(30))

INSERT INTO `Piscine-Roma-DB`.Persona VALUES(CF\_P, indirizzo, nome, NULL, NULL);

* **persona\_modifica(IN CF\_P CHAR(16), IN indirizzo VARCHAR(50), IN nome VARCHAR(30))**
  + Prende in input il codice fiscale, l’indirizzo e il nome , modifica I dati della persona identificata dal codice fiscale.

CREATE PROCEDURE `Piscine-Roma-DB`.persona\_modifica(IN CF\_P CHAR(16), IN indirizzo VARCHAR(50), IN nome VARCHAR(30))

UPDATE `Piscine-Roma-DB`.Persona

SET Indirizzo\_Persona = indirizzo,

Nome\_Persona = nome

WHERE CodFisc\_Persona = CF\_P;

* **persona\_elimina(IN CF\_P CHAR(16))**
  + Prende in input il codice fiscale ed elimina la persona.

CREATE PROCEDURE `Piscine-Roma-DB`.persona\_elimina(IN CF\_P CHAR(16))

DELETE FROM `Piscine-Roma-DB`.Persona

WHERE CodFisc\_Persona = CF\_P;

* **persona\_inserisci\_contatti(IN CF\_P CHAR(16), IN mail VARCHAR(30), IN cellulare VARCHAR(20), IN telefono VARCHAR(20))**
  + Prende in input il codice fiscale della persona, la mail , il cellulare ,il telefono ed inserisce I contatti della persona identificata dal codice fiscale

CREATE PROCEDURE `Piscine-Roma-DB`.persona\_inserisci\_contatti(IN CF\_P CHAR(16), IN mail VARCHAR(30), IN cellulare VARCHAR(20), IN telefono VARCHAR(20))

INSERT INTO `Piscine-Roma-DB`.Contatti

VALUES(CF\_P, mail, cellulare, telefono);

* **persona\_modifica\_contatti(IN CF\_P CHAR(16), IN mail VARCHAR(30), IN cellulare VARCHAR(20), IN telefono VARCHAR(20))**
  + Prende in input il codice fiscale della persona, la mail , il cellulare ,il telefono modifica I contatti della persona identificata dal codice fiscale.

CREATE PROCEDURE `Piscine-Roma-DB`.persona\_modifica\_contatti(IN CF\_P CHAR(16), IN mail VARCHAR(30), IN cellulare VARCHAR(20), IN telefono VARCHAR(20))

UPDATE `Piscine-Roma-DB`.Contatti

SET Mail\_Persona = mail,

Cellulare\_Persona = cellulare,

Telefono\_Persona = telefono

WHERE CodFisc\_Persona\_P = CF\_P;

* **persona\_elimina\_contatti(IN CF\_P CHAR(16))**
  + Prende in input il codice fiscale della persona ed elimina I contatti associati.

CREATE PROCEDURE `Piscine-Roma-DB`.persona\_elimina\_contatti(IN CF\_P CHAR(16))

DELETE FROM `Piscine-Roma-DB`.Contatti

WHERE CodFisc\_Persona\_P = CF\_P;

* **persona\_effettua\_ingresso\_libero(IN CF\_P CHAR(16), IN data\_ingr DATE, IN nome\_piscina VARCHAR(50))**
  + Prende in input il codice fiscale della persona , la data di ingresso in piscina, il nome della piscina, ed effettua un ingresso libero, inserendo I valori nella tabella Persona con update.

CREATE PROCEDURE `Piscine-Roma-DB`.persona\_effettua\_ingresso\_libero(IN CF\_P CHAR(16), IN data\_ingr DATE, IN nome\_piscina VARCHAR(50))

UPDATE `Piscine-Roma-DB`.Persona

SET Data\_ingresso = data\_ingr,

Nome\_Piscina\_P = nome\_piscina

WHERE CodFisc\_Persona = CF\_P;

* **persona\_inserisci\_iscritta\_corsi(IN CF\_P CHAR(16), IN medico VARCHAR(30), IN data\_cert DATE, IN data\_nasc DATE)**
  + Prende in input il codice fiscale della persona , il medico, la data del certificato e la data di nascita, e inserisce I valori nella tabella IscrittoCorsi

CREATE PROCEDURE `Piscine-Roma-DB`.persona\_inserisci\_iscritta\_corsi(IN CF\_P CHAR(16), IN medico VARCHAR(30), IN data\_cert DATE, IN data\_nasc DATE)

INSERT INTO `Piscine-Roma-DB`.Iscritto\_Corsi

VALUES(CF\_P, medico, data\_cert, data\_nasc);

* **persona\_modifica\_iscritta\_corsi(IN CF\_P CHAR(16), IN medico VARCHAR(30), IN data\_cert DATE, IN data\_nasc DATE)**
  + Prende in input il codice fiscale della persona , il medico, la data del certificato e la data di nascita, e modifica nella tabella IscrittoCorsi identificata dal codice fiscale.

CREATE PROCEDURE `Piscine-Roma-DB`.persona\_modifica\_iscritta\_corsi(IN CF\_P CHAR(16), IN medico VARCHAR(30), IN data\_cert DATE, IN data\_nasc DATE)

UPDATE `Piscine-Roma-DB`.Iscritto\_Corsi

SET Medico = medico,

Data\_certificato = data\_cert,

Data\_nascita = data\_nasc

WHERE CodFisc\_Persona\_IC = CF\_P;

* **persona\_elimina\_iscritta\_corsi(IN CF\_P CHAR(16))**
  + Prende in input il codice fiscale della persona , ed elimina nella tabella Iscritto\_Corsi, la persona identificata dal codifìce fiscale

CREATE PROCEDURE `Piscine-Roma-DB`.persona\_elimina\_iscritta\_corsi(IN CF\_P CHAR(16))

DELETE FROM `Piscine-Roma-DB`.Iscritto\_Corsi

WHERE CodFisc\_Persona\_IC = CF\_P;

* **persona\_effettua\_iscrizione\_corso(IN CF\_P CHAR(16), IN corso VARCHAR(30), IN piscina VARCHAR(50))**
  + Prende in input il codice fiscale della persona ,il corso, la piscina ed effettua un iscrizione al corso, inserendo I valori nella tabella creata dall’associazione molti a molti tra Corso e Iscritto\_Corsi.

CREATE PROCEDURE `Piscine-Roma-DB`.persona\_effettua\_iscrizione\_corso(IN CF\_P CHAR(16), IN corso VARCHAR(30), IN piscina VARCHAR(50))

INSERT INTO `Piscine-Roma-DB`.Iscrizione

VALUES(CF\_P, corso, piscina);

* **persona\_cancella\_iscrizione\_corso(IN CF\_P CHAR(16), IN corso VARCHAR(30), IN piscina VARCHAR(50))**
  + Prende in input il codice fiscale della persona ,il corso, la piscina e cancella un iscrizione ad un corso per la persona identificata dal codice fiscale.

CREATE PROCEDURE `Piscine-Roma-DB`.persona\_cancella\_iscrizione\_corso(IN CF\_P CHAR(16), IN corso VARCHAR(30), IN piscina VARCHAR(50))

DELETE FROM `Piscine-Roma-DB`.Iscrizione

WHERE CodFisc\_Persona\_IC\_IS = CF\_P

AND Nome\_Corso\_IS = corso

AND Nome\_Piscina\_Corso\_IS = piscina;

* **corso\_inserisci(IN nome CHAR(30), IN piscina VARCHAR(50), IN costo DECIMAL(5,2), IN massimo INT, IN minimo INT)**
  + Prende in input gli attributi della tabella corso ed effettua un inserimento nella stessa.

CREATE PROCEDURE `Piscine-Roma-DB`.corso\_inserisci(IN nome CHAR(30), IN piscina VARCHAR(50), IN costo DECIMAL(5,2), IN massimo INT, IN minimo INT)

INSERT INTO `Piscine-Roma-DB`.Corso VALUES(nome, piscina, costo, massimo, minimo);

* **corso\_modifica(IN nome CHAR(30), IN piscina VARCHAR(50), IN costo DECIMAL(5,2), IN massimo INT, IN minimo INT)**
  + Prende in input gli attributi della tabella corso e modifica I valori della tupla identificata dal nome corso e nome della piscina.

CREATE PROCEDURE `Piscine-Roma-DB`.corso\_modifica(IN nome CHAR(30), IN piscina VARCHAR(50), IN costo DECIMAL(5,2), IN massimo INT, IN minimo INT)

UPDATE `Piscine-Roma-DB`.Corso

SET Costo = costo,

Max = massimo,

Min = minimo

WHERE Nome\_C = nome AND Nome\_Piscina = piscina;

* **corso\_elimina(IN nome CHAR(30), IN piscina VARCHAR(50))**
  + Prende in input il nome del corso e la piscina, ed elimina la tupla corrispettiva.

CREATE PROCEDURE `Piscine-Roma-DB`.corso\_elimina(IN nome CHAR(30), IN piscina VARCHAR(50))

DELETE FROM `Piscine-Roma-DB`.Corso

WHERE Nome\_C = nome AND Nome\_Piscina = piscina;

* **corso\_inserisci\_lezione(IN giorno INT, IN dalle TIME, IN alle TIME, IN corso VARCHAR(30), IN piscina VARCHAR(50))**
  + Prende in input gli attributi della tabella Lezione ed inserisce una lezione di un corso presso la piscina.

CREATE PROCEDURE `Piscine-Roma-DB`.corso\_inserisci\_lezione(IN giorno INT, IN dalle TIME, IN alle TIME, IN corso VARCHAR(30), IN piscina VARCHAR(50))

INSERT INTO `Piscine-Roma-DB`.Lezione VALUES(giorno, dalle, alle, corso, piscina);

* **corso\_modifica\_lezione(IN giorno INT, IN old\_dalle TIME, IN old\_alle TIME, IN corso VARCHAR(30), IN piscina VARCHAR(50), IN new\_dalle TIME, IN new\_alle TIME)**
  + Prende in input gli attributi della tabella Lezione, e la nuova data, orario lezione e modifica la lezione.

CREATE PROCEDURE `Piscine-Roma-DB`.corso\_modifica\_lezione(IN giorno INT, IN old\_dalle TIME, IN old\_alle TIME, IN corso VARCHAR(30), IN piscina VARCHAR(50), IN new\_dalle TIME, IN new\_alle TIME)

UPDATE `Piscine-Roma-DB`.Lezione

SET Ora\_dalle = new\_dalle,

Ora\_alle = new\_alle

WHERE (Giorno\_sett,Ora\_dalle,Ora\_alle,Nome\_Corso,Nome\_Piscina\_Corso) = (giorno,old\_dalle,ora\_alle,corso,piscina);

* **corso\_elimina\_lezione(IN giorno INT, IN dalle TIME, IN alle TIME, IN corso VARCHAR(30), IN piscina VARCHAR(50))**
  + Prende in input gli attributi della tabella Lezione ed elimina la tupla identificata da questi attributi.

CREATE PROCEDURE `Piscine-Roma-DB`.corso\_elimina\_lezione(IN giorno INT, IN dalle TIME, IN alle TIME, IN corso VARCHAR(30), IN piscina VARCHAR(50))

DELETE FROM `Piscine-Roma-DB`.Lezione

WHERE (Giorno\_sett,Ora\_dalle,Ora\_alle,Nome\_Corso,Nome\_Piscina\_Corso) = (giorno,dalle,alle,corso,piscina);

* **piscina\_inserisci(IN nome CHAR(50), IN indirizzo VARCHAR(50), IN telefono VARCHAR(20), IN responsabile VARCHAR(30), IN tipo BOOLEAN,IN da\_mese VARCHAR(10),IN a\_mese VARCHAR(10))**
  + Prende in input gli attributi della tabella Piscina ed effetua un inserimento della tupla passata come input alla procedura.

CREATE PROCEDURE `Piscine-Roma-DB`.piscina\_inserisci(IN nome CHAR(50), IN indirizzo VARCHAR(50), IN telefono VARCHAR(20), IN responsabile VARCHAR(30), IN tipo BOOLEAN,IN da\_mese VARCHAR(10),IN a\_mese VARCHAR(10))

INSERT INTO `Piscine-Roma-DB`.Piscina VALUES(nome, indirizzo, telefono, responsabile, tipo, da\_mese, a\_mese);

* **piscina\_modifica(IN nome CHAR(50), IN indirizzo VARCHAR(50), IN telefono VARCHAR(20), IN responsabile VARCHAR(30), IN tipo BOOLEAN,IN da\_mese VARCHAR(10),IN a\_mese VARCHAR(10))**
  + Prende in input gli attributi della tabella Piscina ed effetua un aggiornamento sulla tupla identificata dal nome della piscina.

CREATE PROCEDURE `Piscine-Roma-DB`.piscina\_modifica(IN nome CHAR(50), IN indirizzo VARCHAR(50), IN telefono VARCHAR(20), IN responsabile VARCHAR(30), IN tipo BOOLEAN,IN da\_mese VARCHAR(10),IN a\_mese VARCHAR(10))

UPDATE `Piscine-Roma-DB`.Piscina

SET Indirizzo\_P = indirizzo,

Telefono\_P = telefono,

Responsabile = responsabile,

Tipo = tipo,

Da = da\_mese,

a = a\_mese

WHERE Nome\_P = nome;

* **piscina\_elimina(IN nome CHAR(50))**
  + Prende in input il nome della piscina ed elimina la tupla corrispondente.

CREATE PROCEDURE `Piscine-Roma-DB`.piscina\_elimina(IN nome CHAR(50))

DELETE FROM `Piscine-Roma-DB`.Piscina

WHERE Nome\_P = nome ;

* **insegnante\_inserisci(IN CF\_Ins CHAR(16), IN nome VARCHAR(50), IN telefono VARCHAR(20), IN cellulare VARCHAR(20))**
  + Prende in input gli attributi della tabella insegnante ed inserisce la tupla.

CREATE PROCEDURE `Piscine-Roma-DB`.insegnante\_inserisci(IN CF\_Ins CHAR(16), IN nome VARCHAR(50), IN telefono VARCHAR(20), IN cellulare VARCHAR(20))

INSERT INTO `Piscine-Roma-DB`.Insegnante VALUES(CF\_Ins, nome, telefono, cellulare);

* **insegnante\_modifica(IN CF\_Ins CHAR(16), IN nome VARCHAR(50), IN telefono VARCHAR(20), IN cellulare VARCHAR(20))**
  + Prende in input gli attributi della tabella insegnante e modifica la tupla identificata dal codice fiscale dell’insegnante.

CREATE PROCEDURE `Piscine-Roma-DB`.insegnante\_modifica(IN CF\_Ins CHAR(16), IN nome VARCHAR(50), IN telefono VARCHAR(20), IN cellulare VARCHAR(20))

UPDATE `Piscine-Roma-DB`.Insegnante

SET Nome\_I = nome,

Telefono\_I = telefono,

Cellulare\_I = cellulare

WHERE CodFisc\_I = CF\_Ins;

* **insegnante\_elimina(IN CF\_Ins CHAR(16))**
  + Prende in input il codice fiscale dell’insegnante ed elimina la tupla corrispondente.

CREATE PROCEDURE `Piscine-Roma-DB`.insegnante\_elimina(IN CF\_Ins CHAR(16))

DELETE FROM `Piscine-Roma-DB`.Insegnante

WHERE CodFisc\_I = CF\_Ins;

* **qualifica\_inserisci(IN nome VARCHAR(30))**
  + Prende in input il nome della qualifica ed inserisce nella tabella

CREATE PROCEDURE `Piscine-Roma-DB`.qualifica\_inserisci(IN nome VARCHAR(30))

INSERT INTO `Piscine-Roma-DB`.Qualifica VALUES(nome);

* **qualifica\_modifica(IN old\_Nome\_Q VARCHAR(30), IN new\_Nome\_Q VARCHAR(30))**
  + Prende in input il nome della qualifica esistente, il nuovo ed aggiorna nella tabella

CREATE PROCEDURE `Piscine-Roma-DB`.qualifica\_modifica(IN old\_Nome\_Q VARCHAR(30), IN new\_Nome\_Q VARCHAR(30))

UPDATE `Piscine-Roma-DB`.Qualifica

SET Nome\_Q = new\_Nome\_Q

WHERE Nome\_Q = old\_Nome\_Q;

* **qualifica\_elimina(IN nome VARCHAR(30))**
  + Prende in input il nome della qualifica esistente e la elimina dalla tabella

CREATE PROCEDURE `Piscine-Roma-DB`.qualifica\_elimina(IN nome VARCHAR(30))

DELETE FROM `Piscine-Roma-DB`.Qualifica

WHERE Nome\_Q = nome;

* **insegnante\_assegna\_qualifica(IN CF\_I CHAR(16), IN nome\_qual VARCHAR(30))**
  + Prende in input il nome della qualifica, il codice fiscale dell’insegnante ed inserisce nella tabella.

CREATE PROCEDURE `Piscine-Roma-DB`.insegnante\_assegna\_qualifica(IN CF\_I CHAR(16), IN nome\_qual VARCHAR(30))

INSERT INTO `Piscine-Roma-DB`.Insegnante\_Ha\_Qualifica

VALUES(CF\_I, nome\_qual);

* **insegnante\_cancella\_qualifica(IN CF\_I CHAR(16), IN nome\_qual VARCHAR(30))**
  + Prende in input il nome della qualifica, il codice fiscale dell’insegnante e cancella la qualifica da quell’insegnante.

CREATE PROCEDURE `Piscine-Roma-DB`.insegnante\_cancella\_qualifica(IN CF\_I CHAR(16), IN nome\_qual VARCHAR(30))

DELETE FROM `Piscine-Roma-DB`.Insegnante\_Ha\_Qualifica

WHERE CodFisc\_Insegnante = CF\_I

AND Nome\_Qualifica = nome\_qual;

* **rotazione\_inserisci(IN inizio DATE,IN fine DATE,IN CF\_I CHAR(16),IN piscina VARCHAR(50))**
  + Prende in input la data di inizio,di fine, il codice fiscale dell’insegnante , la piscina ed inserisce nella tabella Rotazione

CREATE PROCEDURE `Piscine-Roma-DB`.rotazione\_inserisci(IN inizio DATE,IN fine DATE,IN CF\_I CHAR(16),IN piscina VARCHAR(50))

INSERT INTO `Piscine-Roma-DB`.Rotazione VALUES(inizio,fine,CF\_I,piscina);

* **rotazione\_modifica(IN old\_inizio DATE,IN old\_fine DATE,IN CF\_I CHAR(16),IN piscina VARCHAR(50), IN new\_inizio DATE,IN new\_fine DATE)**
  + Prende in input la data di inizio,di fine, il codice fiscale dell’insegnante , la piscina, la nuova data di inizio e la nuova data di fine e modifica nella tabella Rotazione

CREATE PROCEDURE `Piscine-Roma-DB`.rotazione\_modifica(IN old\_inizio DATE,IN old\_fine DATE,IN CF\_I CHAR(16),IN piscina VARCHAR(50), IN new\_inizio DATE,IN new\_fine DATE)

UPDATE `Piscine-Roma-DB`.Rotazione

SET Data\_inizio = new\_inizio,

Data\_fine = new\_fine

WHERE (Data\_inizio,Data\_fine,CodFiscale\_Insegnante,Nome\_Piscina\_R) = (old\_inizio,old\_fine,CF\_I,piscina);

* **rotazione\_elimina(IN inizio DATE,IN fine DATE,IN CF\_I CHAR(16),IN piscina VARCHAR(50))**
  + Prende in input la data di inizio,di fine, il codice fiscale dell’insegnante , la piscina, ed elimina la tupla corrispondente dalla tabella Rotazione.

CREATE PROCEDURE `Piscine-Roma-DB`.rotazione\_elimina(IN inizio DATE,IN fine DATE,IN CF\_I CHAR(16),IN piscina VARCHAR(50))

DELETE FROM `Piscine-Roma-DB`.Rotazione

WHERE (Data\_inizio,Data\_fine,CodFiscale\_Insegnante,Nome\_Piscina\_R) = (inizio,fine,CF\_I,piscina);

* **piscina\_visualizza\_lista()**
  + Visualizza una lista di tutte le piscine

CREATE PROCEDURE `Piscine-Roma-DB-test`.piscina\_visualizza\_lista()

SELECT \*

FROM `Piscine-Roma-DB-test`.Piscina ;

* **insegnante\_visualizza\_lista()**
  + Visualizza una lista di tutte gli insegnanti

CREATE PROCEDURE `Piscine-Roma-DB-test`.insegnante\_visualizza\_lista()

SELECT \*

FROM `Piscine-Roma-DB-test`.Insegnante ;

* **rotazione\_visualizza\_lista\_per\_piscina(IN piscina CHAR(50))**
  + Visualizza una lista delle rotazioni esistenti in una piscina

CREATE PROCEDURE `Piscine-Roma-DB-test`.rotazione\_visualizza\_lista\_per\_piscina(IN piscina CHAR(50))

SELECT \*

FROM `Piscine-Roma-DB-test`.Rotazione

WHERE Nome\_Piscina\_R = piscina;

* **rotazione\_visualizza\_lista\_per\_insegnante(IN CF\_I CHAR(16))**
  + Visualizza una lista delle rotazioni esistenti per un insegnante

CREATE PROCEDURE `Piscine-Roma-DB-test`.rotazione\_visualizza\_lista\_per\_insegnante(IN CF\_I CHAR(16))

SELECT \*

FROM `Piscine-Roma-DB-test`.Rotazione

WHERE CodFiscale\_Insegnante = CF\_I;

* **corso\_visualizza\_lista\_iscritti(IN corso VARCHAR(30), IN piscina VARCHAR(50))**
  + Visualizza una lista degli iscritti ad un corso presso una piscina

CREATE PROCEDURE `Piscine-Roma-DB-test`.corso\_visualizza\_lista\_iscritti(IN corso VARCHAR(30), IN piscina VARCHAR(50))

SELECT \*

FROM `Piscine-Roma-DB-test`.Iscrizione

WHERE Nome\_Corso\_IS = corso AND Nome\_Piscina\_Corso\_IS = piscina;

* **corso\_visualizza\_lista\_per\_piscina(IN piscina VARCHAR(50))**
  + Visualizza una lista dei corsi esistenti in una piscina

CREATE PROCEDURE `Piscine-Roma-DB-test`.corso\_visualizza\_lista\_per\_piscina(IN piscina VARCHAR(50))

SELECT \*

FROM `Piscine-Roma-DB-test`.Corso

WHERE Nome\_Piscina = piscina;

* **corso\_visualizza\_lista\_lezioni(IN corso VARCHAR(30), IN piscina VARCHAR(50))**
  + Visualizza una lista delle lezioni di un corso presso una piscina

CREATE PROCEDURE `Piscine-Roma-DB-test`.corso\_visualizza\_lista\_lezioni(IN corso VARCHAR(30), IN piscina VARCHAR(50))

SELECT \*

FROM `Piscine-Roma-DB-test`.Lezione

WHERE (Nome\_Corso,Nome\_Piscina\_Corso) = (corso,piscina);

# Appendice: Implementazione

## Codice SQL per instanziare il database

######## CREAO DATABASE ##########

CREATE DATABASE `Piscine-Roma-DB` CHARACTER SET utf8 COLLATE utf8\_general\_ci;

####### SETTO NOMI DELLE DATE IN ITALIANO ########

SET GLOBAL lc\_time\_names = 'it\_CH';

####### PISCINA ########

CREATE TABLE `Piscine-Roma-DB`.`Piscina` (

`Nome\_P` VARCHAR(50) NOT NULL,

`Indirizzo\_P` VARCHAR(50) NOT NULL,

`Telefono\_P` VARCHAR(50) NOT NULL,

`Responsabile` VARCHAR(30) NOT NULL,

`Tipo` BOOLEAN NOT NULL COMMENT 'Tipo = Con vasca all\'aperto ? true : false',

`Da` VARCHAR(10) NULL,

`a` VARCHAR(10) NULL,

PRIMARY KEY (`Nome\_P`));

####### CORSO #########

CREATE TABLE `Piscine-Roma-DB`.`Corso` (

`Nome\_C` VARCHAR(30) NOT NULL,

`Nome\_Piscina` VARCHAR(50) NOT NULL,

`Costo` DECIMAL(5,2) NOT NULL,

`Max` INT NOT NULL,

`Min` INT NOT NULL,

PRIMARY KEY (`Nome\_C`, `Nome\_Piscina`));

####### LEZIONE #######

CREATE TABLE `Piscine-Roma-DB`.`Lezione` (

`Giorno\_sett` INT NOT NULL COMMENT 'Range [1, ... ,7]',

`Ora\_dalle` TIME NOT NULL,

`Ora\_alle` TIME NOT NULL,

`Nome\_Corso` VARCHAR(30) NOT NULL,

`Nome\_Piscina\_Corso` VARCHAR(50) NOT NULL,

PRIMARY KEY (`Giorno\_sett`, `Ora\_dalle`, `Ora\_alle`, `Nome\_Corso`, `Nome\_Piscina\_Corso`));

######I INSEGNANTE #########

CREATE TABLE `Piscine-Roma-DB`.`Insegnante` (

`CodFisc\_I` CHAR(16) NOT NULL,

`Nome\_I` VARCHAR(30) NOT NULL,

`Telefono\_I` VARCHAR(20) NULL,

`Cellulare\_I` VARCHAR(20) NULL,

PRIMARY KEY (`CodFisc\_I`));

###### INSEGNANTE HA QUALIFICA #######

CREATE TABLE `Piscine-Roma-DB`.`Insegnante\_Ha\_Qualifica` (

`CodFisc\_Insegnante` CHAR(16) NOT NULL,

`Nome\_Qualifica` VARCHAR(30) NOT NULL,

PRIMARY KEY (`CodFisc\_Insegnante`, `Nome\_Qualifica`));

####### QUALIFICA #########

CREATE TABLE `Piscine-Roma-DB`.`Qualifica` (

`Nome\_Q` VARCHAR(30) NOT NULL,

PRIMARY KEY (`Nome\_Q`));

######## ROTAZIONE ########

CREATE TABLE `Piscine-Roma-DB`.`Rotazione` (

`Data\_inizio` DATE NOT NULL,

`Data\_fine` DATE NOT NULL,

`CodFiscale\_Insegnante` CHAR(16) NOT NULL,

`Nome\_Piscina\_R` VARCHAR(50) NOT NULL,

PRIMARY KEY (`Data\_inizio`, `Data\_fine`, `CodFiscale\_Insegnante`, `Nome\_Piscina\_R`));

###### PERSONA #######

CREATE TABLE `Piscine-Roma-DB`.`Persona` (

`CodFisc\_Persona` CHAR(16) NOT NULL,

`Indirizzo\_Persona` VARCHAR(50) NOT NULL,

`Nome\_Persona` VARCHAR(30) NOT NULL,

`Data\_ingresso` DATE NULL,

`Nome\_Piscina\_P` VARCHAR(50) NULL,

PRIMARY KEY (`CodFisc\_Persona`));

###### CONTATTI ######

CREATE TABLE `Piscine-Roma-DB`.`Contatti` (

`Codfisc\_Persona\_P` CHAR(16) NOT NULL,

`Mail\_Persona` VARCHAR(50) NULL,

`Cellulare\_Persona` VARCHAR(20) NULL,

`Telefono\_Persona` VARCHAR(20) NULL,

PRIMARY KEY (`Codfisc\_Persona\_P`));

###### ISCRITTO\_CORSI ########

CREATE TABLE `Piscine-Roma-DB`.`Iscritto\_Corsi` (

`CodFisc\_Persona\_IC` CHAR(16) NOT NULL,

`Medico` VARCHAR(30) NOT NULL,

`Data\_certificato` DATE NOT NULL,

`Data\_nascita` DATE NOT NULL,

PRIMARY KEY (`CodFisc\_Persona\_IC`));

####### ISCRIZIONE ########

CREATE TABLE `Piscine-Roma-DB`.`Iscrizione` (

`CodFisc\_Persona\_IC\_IS` CHAR(16) NOT NULL,

`Nome\_Corso\_IS` VARCHAR(30) NOT NULL,

`Nome\_Piscina\_Corso\_IS` VARCHAR(50) NOT NULL,

PRIMARY KEY (`CodFisc\_Persona\_IC\_IS`, `Nome\_Corso\_IS`, `Nome\_Piscina\_Corso\_IS`));

ALTER TABLE `Piscine-Roma-DB`.Corso ADD CONSTRAINT Corso\_FK FOREIGN KEY (Nome\_Piscina) REFERENCES `Piscine-Roma-DB`.Piscina(Nome\_P)

ON DELETE CASCADE ON UPDATE CASCADE;

######## FOREIGN KEYS ##############

ALTER TABLE `Piscine-Roma-DB`.Lezione ADD CONSTRAINT Lezione\_FK FOREIGN KEY (Nome\_Corso,Nome\_Piscina\_Corso) REFERENCES `Piscine-Roma-DB`.Corso(Nome\_C,Nome\_Piscina)

ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `Piscine-Roma-DB`.Rotazione ADD CONSTRAINT Rotazione\_FK FOREIGN KEY (Nome\_Piscina\_R) REFERENCES `Piscine-Roma-DB`.Piscina(Nome\_P)

ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `Piscine-Roma-DB`.Rotazione ADD CONSTRAINT Rotazione\_FK\_1 FOREIGN KEY (CodFiscale\_Insegnante) REFERENCES `Piscine-Roma-DB`.Insegnante(CodFisc\_I)

ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `Piscine-Roma-DB`.Insegnante\_Ha\_Qualifica ADD CONSTRAINT Insegnante\_Ha\_Qualifica\_FK FOREIGN KEY (CodFisc\_Insegnante) REFERENCES `Piscine-Roma-DB`.Insegnante(CodFisc\_I)

ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `Piscine-Roma-DB`.Insegnante\_Ha\_Qualifica ADD CONSTRAINT Insegnante\_Ha\_Qualifica\_FK\_1 FOREIGN KEY (Nome\_Qualifica) REFERENCES `Piscine-Roma-DB`.Qualifica(Nome\_Q)

ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `Piscine-Roma-DB`.Persona ADD CONSTRAINT Persona\_FK FOREIGN KEY (Nome\_Piscina\_P) REFERENCES `Piscine-Roma-DB`.Piscina(Nome\_P)

ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `Piscine-Roma-DB`.Contatti ADD CONSTRAINT Contatti\_FK FOREIGN KEY (Codfisc\_Persona\_P) REFERENCES `Piscine-Roma-DB`.Persona(CodFisc\_Persona)

ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `Piscine-Roma-DB`.Iscritto\_Corsi ADD CONSTRAINT Iscritto\_Corsi\_FK FOREIGN KEY (CodFisc\_Persona\_IC) REFERENCES `Piscine-Roma-DB`.Persona(CodFisc\_Persona)

ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `Piscine-Roma-DB`.Iscrizione ADD CONSTRAINT Iscrizione\_FK FOREIGN KEY (CodFisc\_Persona\_IC\_IS) REFERENCES `Piscine-Roma-DB`.Iscritto\_Corsi(CodFisc\_Persona\_IC)

ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `Piscine-Roma-DB`.Iscrizione ADD CONSTRAINT Iscrizione\_FK\_1 FOREIGN KEY (Nome\_Corso\_IS,Nome\_Piscina\_Corso\_IS) REFERENCES `Piscine-Roma-DB`.Corso(Nome\_C,Nome\_Piscina)

ON DELETE CASCADE ON UPDATE CASCADE;

######## DATI ##########

INSERT INTO `Piscine-Roma-DB`.`Piscina` (`Nome\_P`,`Indirizzo\_P`,`Telefono\_P`,`Responsabile`,`Tipo`,`Da`,`a`) VALUES ("Piscina Delle Rose","viale America, 20","0592-193147","Sara De Simone","1","Maggio","Settembre");

INSERT INTO `Piscine-Roma-DB`.`Piscina` (`Nome\_P`,`Indirizzo\_P`,`Telefono\_P`,`Responsabile`,`Tipo`,`Da`,`a`) VALUES ("AquaTibur","via del frantoio, 8","0626-133997","Erika De Simone","0",NULL,NULL);

INSERT INTO `Piscine-Roma-DB`.`Piscina` (`Nome\_P`,`Indirizzo\_P`,`Telefono\_P`,`Responsabile`,`Tipo`,`Da`,`a`) VALUES ("Octopus Acquatic Club","via della Tenuta di Torrenova, 128","0761-239856","Giovanni Cattaneo","0",NULL,NULL);

INSERT INTO `Piscine-Roma-DB`.`Piscina` (`Nome\_P`,`Indirizzo\_P`,`Telefono\_P`,`Responsabile`,`Tipo`,`Da`,`a`) VALUES ("Olimpia 1","via Giuseppe Messina, 41","0328-876266","Dario Morelli","1","Giugno","Agosto");

INSERT INTO `Piscine-Roma-DB`.`Piscina` (`Nome\_P`,`Indirizzo\_P`,`Telefono\_P`,`Responsabile`,`Tipo`,`Da`,`a`) VALUES ("Crawl 2000","via del Gran Paradiso, 93","0974-862708","Nicolò Valente","0",NULL,NULL);

INSERT INTO `Piscine-Roma-DB`.`Piscina` (`Nome\_P`,`Indirizzo\_P`,`Telefono\_P`,`Responsabile`,`Tipo`,`Da`,`a`) VALUES ("Delta","via mazzantini, 15","0614-651436","Irene Battaglia","0",NULL,NULL);

INSERT INTO `Piscine-Roma-DB`.`Piscina` (`Nome\_P`,`Indirizzo\_P`,`Telefono\_P`,`Responsabile`,`Tipo`,`Da`,`a`) VALUES ("Juventus Nuoto","via bravetta, 539","0882-180032","Salvatore Sorrentino","0",NULL,NULL);

INSERT INTO `Piscine-Roma-DB`.`Piscina` (`Nome\_P`,`Indirizzo\_P`,`Telefono\_P`,`Responsabile`,`Tipo`,`Da`,`a`) VALUES ("Roma Uno","largo Ascianghi, 4","0671-113567","Christian Fusco","1","Giugno","Settembre");

INSERT INTO `Piscine-Roma-DB`.`Piscina` (`Nome\_P`,`Indirizzo\_P`,`Telefono\_P`,`Responsabile`,`Tipo`,`Da`,`a`) VALUES ("CSI Roma Est","via montona, 17","0558-539154","Giorgia Monti","0",NULL,NULL);

INSERT INTO `Piscine-Roma-DB`.`Piscina` (`Nome\_P`,`Indirizzo\_P`,`Telefono\_P`,`Responsabile`,`Tipo`,`Da`,`a`) VALUES ("Azzurra","via manduria, 21","0955-073831","Jessica Pace","1","Maggio","Agosto");

INSERT INTO `Piscine-Roma-DB`.`Piscina` (`Nome\_P`,`Indirizzo\_P`,`Telefono\_P`,`Responsabile`,`Tipo`,`Da`,`a`) VALUES ("Roma 12","via dei pini, 1","0179-387008","Michele Marra","0",NULL,NULL);

INSERT INTO `Piscine-Roma-DB`.Corso (Nome\_C,Nome\_Piscina,Costo,Max,Min) VALUES

('acquagym','AquaTibur',55.00,10,5)

,('acquagym','Azzurra',50.00,15,8)

,('acquagym','Crawl 2000',50.00,10,5)

,('acquagym','CSI Roma Est',65.00,15,5)

,('acquagym','Juventus Nuoto',40.00,15,5)

,('acquagym','Octopus Acquatic Club',55.00,10,5)

,('acquagym','Piscina Delle Rose',55.00,8,4)

,('acquagym','Roma 12',50.00,8,5)

,('acquagym','Roma Uno',45.00,8,4)

,('dorso','AquaTibur',40.00,10,3)

;

INSERT INTO `Piscine-Roma-DB`.Corso (Nome\_C,Nome\_Piscina,Costo,Max,Min) VALUES

('dorso','Azzurra',50.00,8,4)

,('dorso','Crawl 2000',35.00,15,5)

,('dorso','Delta',35.00,5,2)

,('dorso','Octopus Acquatic Club',40.00,10,3)

,('dorso','Olimpia 1',45.00,8,2)

,('dorso','Roma 12',30.00,5,2)

,('hydrobike','Azzurra',150.00,5,4)

,('hydrobike','Juventus Nuoto',65.00,8,5)

,('hydrobike','Roma Uno',60.00,5,5)

,('nuoto neonatale','Crawl 2000',35.00,4,1)

;

INSERT INTO `Piscine-Roma-DB`.Corso (Nome\_C,Nome\_Piscina,Costo,Max,Min) VALUES

('pallanuoto','AquaTibur',150.00,14,7)

,('pallanuoto','CSI Roma Est',100.00,10,5)

,('pallanuoto','Olimpia 1',150.00,14,7)

,('pallanuoto','Piscina Delle Rose',90.00,14,7)

,('pallanuoto','Roma Uno',80.00,15,7)

,('rana','AquaTibur',40.00,10,3)

,('rana','Crawl 2000',40.00,15,5)

,('rana','Octopus Acquatic Club',40.00,10,3)

,('rana','Roma 12',30.00,5,2)

,('sincronizzato','AquaTibur',150.00,5,4)

;

INSERT INTO `Piscine-Roma-DB`.Corso (Nome\_C,Nome\_Piscina,Costo,Max,Min) VALUES

('sincronizzato','Piscina Delle Rose',80.00,5,4)

,('stile libero','AquaTibur',30.00,10,3)

,('stile libero','Azzurra',40.00,8,4)

,('stile libero','Crawl 2000',35.00,15,5)

,('stile libero','Delta',30.00,10,3)

,('stile libero','Octopus Acquatic Club',40.00,10,3)

,('stile libero','Olimpia 1',35.00,8,2)

,('stile libero','Roma 12',30.00,5,2)

,('stile libero','Roma Uno',55.00,10,6)

,('tuffi','AquaTibur',200.00,1,1)

;

INSERT INTO `Piscine-Roma-DB`.Corso (Nome\_C,Nome\_Piscina,Costo,Max,Min) VALUES

('tuffi','Delta',95.00,4,2)

,('tuffi','Piscina Delle Rose',80.00,4,2)

;

INSERT INTO `Piscine-Roma-DB`.Lezione (Giorno\_sett,Ora\_dalle,Ora\_alle,Nome\_Corso,Nome\_Piscina\_Corso) VALUES

(2,'10:00:00','11:00:00','acquagym','AquaTibur')

,(2,'14:00:00','16:00:00','acquagym','AquaTibur')

,(4,'17:00:00','19:00:00','acquagym','AquaTibur')

,(3,'08:00:00','10:00:00','acquagym','Azzurra')

,(5,'18:00:00','20:00:00','acquagym','Azzurra')

,(4,'15:00:00','16:00:00','acquagym','Crawl 2000')

,(6,'12:00:00','13:00:00','acquagym','Crawl 2000')

,(7,'11:00:00','13:00:00','acquagym','Crawl 2000')

,(2,'14:00:00','16:00:00','acquagym','CSI Roma Est')

,(2,'20:00:00','21:00:00','acquagym','CSI Roma Est')

;

INSERT INTO `Piscine-Roma-DB`.Lezione (Giorno\_sett,Ora\_dalle,Ora\_alle,Nome\_Corso,Nome\_Piscina\_Corso) VALUES

(5,'16:00:00','17:00:00','acquagym','CSI Roma Est')

,(3,'11:00:00','12:00:00','acquagym','Juventus Nuoto')

,(4,'15:00:00','16:00:00','acquagym','Juventus Nuoto')

,(5,'18:00:00','19:00:00','acquagym','Juventus Nuoto')

,(2,'10:00:00','11:00:00','acquagym','Octopus Acquatic Club')

,(4,'18:00:00','20:00:00','acquagym','Octopus Acquatic Club')

,(3,'15:00:00','17:00:00','acquagym','Piscina Delle Rose')

,(6,'10:00:00','12:00:00','acquagym','Piscina Delle Rose')

,(2,'16:00:00','17:00:00','acquagym','Roma 12')

,(2,'20:00:00','21:00:00','acquagym','Roma 12')

;

INSERT INTO `Piscine-Roma-DB`.Lezione (Giorno\_sett,Ora\_dalle,Ora\_alle,Nome\_Corso,Nome\_Piscina\_Corso) VALUES

(6,'18:00:00','19:00:00','acquagym','Roma 12')

,(2,'12:00:00','13:00:00','acquagym','Roma Uno')

,(3,'16:00:00','17:00:00','acquagym','Roma Uno')

,(7,'10:00:00','11:00:00','acquagym','Roma Uno')

,(2,'12:00:00','13:00:00','dorso','AquaTibur')

,(3,'10:00:00','11:00:00','dorso','AquaTibur')

,(2,'18:00:00','20:00:00','pallanuoto','AquaTibur')

,(6,'20:00:00','22:00:00','pallanuoto','AquaTibur')

,(5,'14:00:00','16:00:00','sincronizzato','AquaTibur')

,(6,'14:00:00','16:00:00','sincronizzato','AquaTibur')

;

INSERT INTO `Piscine-Roma-DB`.Lezione (Giorno\_sett,Ora\_dalle,Ora\_alle,Nome\_Corso,Nome\_Piscina\_Corso) VALUES

(3,'15:00:00','17:00:00','stile libero','AquaTibur')

,(5,'10:00:00','12:00:00','stile libero','AquaTibur')

;

INSERT INTO `Piscine-Roma-DB`.Persona (CodFisc\_Persona,Indirizzo\_Persona,Nome\_Persona,Data\_ingresso,Nome\_Piscina\_P) VALUES

('AOHPPJ62P18J665T','Ap #530-4512 Nullam Avenue','Sara Franco','2019-02-24','Crawl 2000')

,('APFBSK84C89K742F','P.O. Box 351, 1981 Leo. Road','Vincenzo Ruggiero','2020-01-28','Crawl 2000')

,('APFERG35F48U254F','P.O. Box 265, 185 Rutrum Ave','Samuele Bruni','2019-11-24','Olimpia 1')

,('APLXTO83V31A695Q','865-2825 Pede, Road','Luigi Calabrese','2019-04-02','Delta')

,('AWCNED46N79H485Y','3505 Nisl. St.','Viola Silvestri','2019-08-20','Olimpia 1')

,('BCMVTD99E81V769T','Ap #115-6538 Arcu. Road','Mattia Ceccarelli','2018-12-03','Roma 12')

,('BEMJZF28C73G445W','308-9208 Natoque Rd.','Filippo Ruggeri','2018-07-16','Piscina Delle Rose')

,('BHEPWE78S83X958V','Ap #181-3461 Amet, Rd.','Sofia Pozzi',NULL,NULL)

,('BIVVPC82X53O492J','440-260 Dolor Street','Cristiano Lombardi','2019-01-18','Olimpia 1')

,('BVIXSR94H68T423J','7058 Quisque Avenue','Augusto Piazza',NULL,NULL)

;

INSERT INTO `Piscine-Roma-DB`.Persona (CodFisc\_Persona,Indirizzo\_Persona,Nome\_Persona,Data\_ingresso,Nome\_Piscina\_P) VALUES

('BZPNQH41V29N288H','3923 Augue Ave','Irene Santini','2018-12-18','Roma 12')

,('CCBETP53Z92V992S','204-7599 Tincidunt Rd.','Emanuele Guerra','2018-01-20','Olimpia 1')

,('CLPPFK67Q43X994I','9682 Libero. St.','Stefania Longo','2018-07-01','AquaTibur')

,('CVFFGJ23O96R727K','Ap #748-5112 Ac Rd.','Mirko Caruso',NULL,NULL)

,('DBNCEN17G33O274R','Ap #287-2187 Consectetuer Av.','Jacopo Catalano','2019-03-02','Azzurra')

,('DJWKIH87G71J681X','P.O. Box 876, 5583 Lectus. Av.','Mirko Vitale','2020-01-28','Crawl 2000')

,('DTVFRK28H41E323E','629-7272 Arcu Rd.','Gaia Santini','2018-09-10','Piscina Delle Rose')

,('DZCNZU31G98V547Y','P.O. Box 625, 5426 Egestas Av.','Ilaria Silvestri','2019-02-03','AquaTibur')

,('EFHOWM88W52S685I','199 Nunc. St.','Davide Ricci','2020-01-15','Piscina Delle Rose')

,('ELWJCY27T39Y121W','P.O. Box 318, 3880 Felis Street','Cristian Amato','2019-09-05','Roma Uno')

;

INSERT INTO `Piscine-Roma-DB`.Persona (CodFisc\_Persona,Indirizzo\_Persona,Nome\_Persona,Data\_ingresso,Nome\_Piscina\_P) VALUES

('ENVZBV11A74D638E','P.O. Box 532, 4138 Morbi Av.','Laura Lombardi',NULL,NULL)

,('ETESXL36F68I554T','Ap #220-9725 Sagittis. Ave','Simone Ferraro','2019-07-09','Roma Uno')

,('EUILXM17S59C346D','Ap #887-3020 A, Rd.','Stefano Mancini','2019-08-27','Octopus Acquatic Club')

,('FEVOLP39M29B984I','P.O. Box 317, 2844 Felis. Ave','Lorenzo Monaco','2018-02-18','Piscina Delle Rose')

,('FGNQBQ21H74Y112N','466-6403 Non St.','Alex Mariani','2018-07-19','Roma 12')

,('FKRURM97R96P714F','1447 Non Rd.','Giada Mazza','2018-06-12','Roma Uno')

,('FNYKRA94A84T312B','5147 Mauris. Road','Marta Mancini',NULL,NULL)

,('FRPJDA55E53L754V','Ap #990-6178 Pellentesque St.','Federica Milani','2018-03-17','CSI Roma Est')

,('FUBMUL51Y56M796U','Ap #466-7384 Metus Rd.','Jessica Vitali','2020-01-06','Octopus Acquatic Club')

,('FUHHGU63J93V624Q','Ap #969-6212 Elementum Av.','Camilla Giuliani','2018-08-11','Crawl 2000')

;

INSERT INTO `Piscine-Roma-DB`.Persona (CodFisc\_Persona,Indirizzo\_Persona,Nome\_Persona,Data\_ingresso,Nome\_Piscina\_P) VALUES

('FVSCXW98F83B572R','3121 Aliquam St.','Manuela Sorrentino','2019-09-24','Azzurra')

,('FYUAYW29I96E672F','Ap #283-1313 Cras Road','Filippo Fiore','2018-05-06','CSI Roma Est')

,('GIHUBU53T52K291T','5029 Dictum St.','Jacopo Calabrese','2019-04-21','Delta')

,('GJHOPO67K52J454K','829-8842 Lorem, Ave','Augusto Pozzi','2019-11-24','Roma 12')

,('HDYJYD93L99U962B','Ap #822-1179 In Rd.','Marco Piccolo',NULL,NULL)

,('HFLYUA68B48O757J','Ap #131-554 Vitae, St.','Emanuele Fabbri','2019-11-16','Piscina Delle Rose')

,('HGMRWX83B16A252T','P.O. Box 568, 581 Sit Street','Dario Olivieri','2019-04-29','Olimpia 1')

,('HIWBYT94K48Z642M','P.O. Box 146, 5481 Dui. St.','Mirko Conti',NULL,NULL)

,('HOVVKI67I54O292X','Ap #133-4336 Mollis Av.','Erika Piccolo','2018-11-14','CSI Roma Est')

,('HVBNKW55R92J425A','582-207 Nascetur St.','Giulio Mariani','2019-02-27','Delta')

;

INSERT INTO `Piscine-Roma-DB`.Persona (CodFisc\_Persona,Indirizzo\_Persona,Nome\_Persona,Data\_ingresso,Nome\_Piscina\_P) VALUES

('HXGVSN17V15J346D','1169 Tempus Avenue','Gianpaolo Conti',NULL,NULL)

,('HXGZMN37X43K484K','140-1748 Felis St.','Silvia Fusco',NULL,NULL)

,('IDOAUL69X82I963Y','Ap #756-190 Hendrerit St.','Samuel Monti','2018-11-07','Azzurra')

,('INNGZZ73T54M967N','P.O. Box 918, 7021 Rutrum Road','Elena Coppola','2018-06-22','Octopus Acquatic Club')

,('IVAGEV54J45K421Y','P.O. Box 541, 7223 Nam Ave','Giulia Carbone',NULL,NULL)

,('IXKFWJ54J38X341I','230-7429 Et, Street','Federica Conti','2019-03-27','Octopus Acquatic Club')

,('IXTUIY19O35Y472J','P.O. Box 470, 3974 Consectetuer Rd.','Cristiano Cavallo','2018-04-16','Azzurra')

,('IYDUWI39L66A253C','8340 Cras Rd.','Giuseppe Montanari',NULL,NULL)

,('JIATHY15U72F495B','894-7893 Lobortis Rd.','Veronica Palmieri','2018-11-04','Piscina Delle Rose')

,('JJRATK56U92O644K','P.O. Box 422, 579 Pede. Street','Gianni Conte','2019-03-14','Octopus Acquatic Club')

;

INSERT INTO `Piscine-Roma-DB`.Persona (CodFisc\_Persona,Indirizzo\_Persona,Nome\_Persona,Data\_ingresso,Nome\_Piscina\_P) VALUES

('JNFZEI46L53H962L','Ap #402-5685 Vulputate, St.','Giada De Angelis',NULL,NULL)

,('JNQVIR21S86J866L','Ap #334-6753 Elementum, Avenue','Sofia Donati','2019-06-23','Roma 12')

,('JQAAQG74G31C959X','475-3102 Egestas Road','Aurora Mancini','2019-07-11','Juventus Nuoto')

,('JRBKMN51D23S875J','P.O. Box 919, 8386 Pharetra Av.','Samuel Rossetti',NULL,NULL)

,('JXEHJV64F82G136U','P.O. Box 677, 700 Orci. Street','Valerio Rinaldi','2019-10-16','CSI Roma Est')

,('KGPOYT36R58H886U','433-8670 Mauris Street','Matteo Marini',NULL,NULL)

,('KLXZSB19H59Q863I','Ap #938-9024 Sit Street','Gabriele Bellini','2018-12-10','Crawl 2000')

,('KPIESM18W54D778K','Ap #990-6632 Nunc Rd.','Marta Conti','2019-09-08','Olimpia 1')

,('LQDATV27F33W749V','444-1850 Lacus. St.','Daniele Agostini','2018-01-18','Roma 12')

,('LTUPYT21T52K518V','P.O. Box 168, 3499 Et Road','Margherita Sartori',NULL,NULL)

;

INSERT INTO `Piscine-Roma-DB`.Persona (CodFisc\_Persona,Indirizzo\_Persona,Nome\_Persona,Data\_ingresso,Nome\_Piscina\_P) VALUES

('LXPPNS32M56U463F','Ap #828-3391 Ac Rd.','Irene Amato','2019-05-15','Piscina Delle Rose')

,('LYGTVF49G59H298G','P.O. Box 281, 122 Blandit. Rd.','Giovanni Grimaldi','2019-11-15','AquaTibur')

,('MDJKPJ26N79R392N','P.O. Box 920, 4733 Eu Avenue','Enrico Marchi',NULL,NULL)

,('MUGGKO16K16A261X','616-6951 Interdum. St.','Ilaria Ferri','2019-12-21','AquaTibur')

,('MXHUQJ32W81G431W','P.O. Box 846, 9898 Posuere Av.','Emanuele Lombardo','2019-07-11','Azzurra')

,('NCMMDE85I79F666K','622-5606 Sem. Avenue','Augusto Volpe','2019-05-03','Azzurra')

,('NFFJRG51G15X437V','Ap #829-7498 Mattis Avenue','Vanessa Fontana',NULL,NULL)

,('NJZGAK58I87I191T','679-2645 Est Av.','Sofia Ruggiero','2018-09-15','Crawl 2000')

,('NKDRMO58R74Z392N','Ap #860-6753 Posuere Av.','Christian Greco','2019-08-19','Octopus Acquatic Club')

,('NVMXTA37O62S522P','P.O. Box 586, 6796 Tellus Rd.','Ginevra Mariani','2019-09-20','Olimpia 1')

;

INSERT INTO `Piscine-Roma-DB`.Persona (CodFisc\_Persona,Indirizzo\_Persona,Nome\_Persona,Data\_ingresso,Nome\_Piscina\_P) VALUES

('NYQEHL66L73E271C','Ap #140-5929 Non Rd.','Alex Orlando','2019-08-04','CSI Roma Est')

,('NZNGRZ32I73F841A','261-6630 Pede. St.','Christian Fiore','2019-02-27','Delta')

,('OJMPTX13P77B569K','7014 A Rd.','Chiara Gallo','2019-01-21','Roma 12')

,('OXNRHU44D81J895M','506-3453 Vel, St.','Christian Barbieri','2019-02-06','Crawl 2000')

,('OYHNUC61N78P638T','804-2931 Dui. Street','Emma Santini',NULL,NULL)

,('OYNSHY15I47R935V','5892 Fusce Rd.','Mario Bellini','2018-04-12','Roma 12')

,('PILNRI72V79D314V','Ap #459-1654 Quis Rd.','Jessica Rizzo','2018-03-22','Crawl 2000')

,('PRIFZK51G16A986H','Ap #183-6672 Morbi Avenue','Domenico Pagano','2018-01-16','Octopus Acquatic Club')

,('PVTGPR11B78E271H','P.O. Box 409, 1672 Libero Ave','Giuseppe Leone',NULL,NULL)

,('QDJDRR83O84P934V','Ap #881-4748 Mi Street','Debora Cattaneo','2018-03-26','Azzurra')

;

INSERT INTO `Piscine-Roma-DB`.Persona (CodFisc\_Persona,Indirizzo\_Persona,Nome\_Persona,Data\_ingresso,Nome\_Piscina\_P) VALUES

('QIOOGS54E81S383X','Ap #577-3717 Feugiat Avenue','Alice Rinaldi',NULL,NULL)

,('QPWLCB96X48O615D','2689 Lorem Av.','Lisa Castelli','2019-03-30','Olimpia 1')

,('QTFKKI21J52G273U','5631 Elit, St.','Nicoletta Ceccarelli','2019-10-30','Olimpia 1')

,('QYTQUM39C98F652Y','5039 Sodales Av.','Gabriele Garofalo','2019-01-03','CSI Roma Est')

,('QZUQCQ82M52Y434Z','Ap #348-2909 Amet Avenue','Lorenzo Ferri','2018-09-29','Crawl 2000')

,('RCMOSJ89M27V386B','Ap #363-424 Proin Road','Emanuele Piazza',NULL,NULL)

,('RIFVVN57X88O323D','3595 Luctus Rd.','Davide Valentini','2019-08-02','Olimpia 1')

,('RIOABI86I63Q318L','1525 Aliquet Rd.','Gaia Farina','2019-08-16','Octopus Acquatic Club')

,('RJQSKK32O56V421N','P.O. Box 928, 5259 Scelerisque Ave','Erica Villa','2020-01-12','Piscina Delle Rose')

,('RLXSIC98X78X927M','3269 In Avenue','Giuseppe Rossetti','2019-12-16','Olimpia 1')

;

INSERT INTO `Piscine-Roma-DB`.Persona (CodFisc\_Persona,Indirizzo\_Persona,Nome\_Persona,Data\_ingresso,Nome\_Piscina\_P) VALUES

('RVHQEQ13U82N947G','2911 Aliquet, St.','Matteo Aiello',NULL,NULL)

,('RYVCIV61Y83R329V','473-8749 Rutrum Ave','Mirko Grasso','2018-11-04','Roma Uno')

,('SKAVNG32D94B523Z','Ap #153-8833 A Avenue','Alessia Caputo','2019-12-16','CSI Roma Est')

,('SLXBBC34G73T957X','562-2172 Nibh. Ave','Matilde Greco','2018-12-07','Octopus Acquatic Club')

,('SOCGZI99P23F635O','P.O. Box 337, 6514 Arcu St.','Federico Rossetti','2018-06-07','Piscina Delle Rose')

,('SUCFSY44S46T328Q','320-3650 Lorem Avenue','Angelo Santoro','2019-08-11','Azzurra')

,('SXHYLN71P12I256G','1258 Nonummy Street','Cristina Costa',NULL,NULL)

,('TDDCAU61P52C191C','374-969 Vulputate Street','Giorgia Galli','2019-01-19','Olimpia 1')

,('TKIJBB37D75Z835R','1830 Ligula. Rd.','Giulietta Cattaneo','2019-08-06','Juventus Nuoto')

,('TLCOQP65G52I656M','458-2272 A Av.','Antonio Conti','2019-03-05','Roma 12')

;

INSERT INTO `Piscine-Roma-DB`.Persona (CodFisc\_Persona,Indirizzo\_Persona,Nome\_Persona,Data\_ingresso,Nome\_Piscina\_P) VALUES

('TTBEAK47H91N897X','P.O. Box 964, 9626 Ut, St.','Sara Lombardi','2018-11-04','CSI Roma Est')

,('TVXQJA58K47Y764C','Ap #531-2481 Pretium Avenue','Claudia Leone','2019-12-26','Roma 12')

,('UELVJZ49S54V944Q','9232 Lobortis Rd.','Roberta Cattaneo','2018-05-03','Azzurra')

,('UHCSEW63R41H139Y','3031 Pellentesque St.','Tommaso Basile',NULL,NULL)

,('UNHIZK73P22S187I','817-3301 Elit Avenue','Beatrice Ceccarelli','2018-04-22','Crawl 2000')

,('UOBLNT11E74M412Z','P.O. Box 877, 9992 Lacinia. Road','Elena Valente','2019-11-23','Piscina Delle Rose')

,('URDZRZ88C28Q887F','455-2541 Mi St.','Roberta Fabbri','2018-02-13','Roma Uno')

,('UZZKVE86J29O861N','P.O. Box 898, 5753 Curabitur Street','Vincenzo Rossi','2018-10-10','Roma Uno')

,('VYXBHD38Q59Q142D','Ap #462-3136 Tristique Ave','Luigi Donati','2019-10-20','Olimpia 1')

,('WCFPHP74E53S828J','7065 Sit Rd.','Matilde Ruggeri',NULL,NULL)

;

INSERT INTO `Piscine-Roma-DB`.Persona (CodFisc\_Persona,Indirizzo\_Persona,Nome\_Persona,Data\_ingresso,Nome\_Piscina\_P) VALUES

('WDJBWH36C82F319S','P.O. Box 314, 3868 Leo. St.','Antonio Arena','2018-09-17','Juventus Nuoto')

,('WLIGYV23L83Z159K','960-548 Dictum. St.','Alberto Caruso','2019-07-04','Delta')

,('WPQVWE88O77D398F','665-4118 Vulputate St.','Irene Volpe','2019-07-31','Juventus Nuoto')

,('WQSYVA66A76X719I','P.O. Box 838, 9989 Sociis Av.','Alessia Martini',NULL,NULL)

,('XAKYSD32T75P685Z','P.O. Box 819, 2677 Sodales Av.','Filippo Baldi','2019-06-17','Juventus Nuoto')

,('XEFSNP43O47J957Z','807-4382 Sem. Road','Giovanni Mariani','2019-07-24','Roma 12')

,('XIQJLG42Y74T777E','Ap #687-6121 Ullamcorper. Avenue','Erica Gentile','2018-08-08','Piscina Delle Rose')

,('XOJBYB51T51G998E','Ap #335-5442 Congue Avenue','Alice Ferri','2019-03-11','CSI Roma Est')

,('XVGJMZ21J52I896U','P.O. Box 588, 2701 Ullamcorper Rd.','Samuel Proietti','2018-12-03','Azzurra')

,('YAVMON79U14D859T','Ap #261-5205 Lectus. Rd.','Francesco Volpe',NULL,NULL)

;

INSERT INTO `Piscine-Roma-DB`.Persona (CodFisc\_Persona,Indirizzo\_Persona,Nome\_Persona,Data\_ingresso,Nome\_Piscina\_P) VALUES

('YZAAMP76B23U272I','666-6530 Ipsum St.','Monica Cavallo','2018-10-09','AquaTibur')

,('YZZPYW46B85J423T','7831 Consequat Av.','Giorgia Pozzi','2018-10-16','AquaTibur')

,('ZAQSBI56T19X653W','Ap #164-591 Nulla Ave','Claudio Ferraro','2019-01-31','Roma Uno')

,('ZBYXYU86R97F573U','P.O. Box 314, 9809 Id, Road','Salvatore Martini',NULL,NULL)

,('ZCAGDG93X13P578K','234-8583 Tristique Road','Camilla Mele',NULL,NULL)

,('ZFWHZG21K62Z878H','678-4626 Odio Road','Marcello Monti','2018-10-16','Juventus Nuoto')

,('ZJTGCS68E58C366M','421-5720 Amet Rd.','Valeria Ruggeri','2019-07-12','CSI Roma Est')

,('ZSEBUP21K37O194I','500-3847 Mauris Road','Roberta Volpe','2019-01-27','Azzurra')

,('ZUXXME35L42Z588H','P.O. Box 252, 8856 Cursus St.','Dario Lombardo',NULL,NULL)

,('ZZEZJQ23O31R997S','8334 Malesuada Rd.','Chiara Baldi','2018-06-12','Octopus Acquatic Club')

;

INSERT INTO `Piscine-Roma-DB`.`Contatti` (`CodFisc\_Persona\_P`,`Mail\_Persona`,`Cellulare\_Persona`,`Telefono\_Persona`) VALUES ("IVAGEV54J45K421Y","ante@ettristiquepellentesque.net","377-3480848","0344-843707");

INSERT INTO `Piscine-Roma-DB`.`Contatti` (`CodFisc\_Persona\_P`,`Mail\_Persona`,`Cellulare\_Persona`,`Telefono\_Persona`) VALUES ("JRBKMN51D23S875J","fringilla.porttitor@metus.ca","398-2947253","0326-168030");

INSERT INTO `Piscine-Roma-DB`.`Contatti` (`CodFisc\_Persona\_P`,`Mail\_Persona`,`Cellulare\_Persona`,`Telefono\_Persona`) VALUES ("NFFJRG51G15X437V","diam.lorem.auctor@ametluctus.co.uk","388-1591651","0878-514683");

INSERT INTO `Piscine-Roma-DB`.`Contatti` (`CodFisc\_Persona\_P`,`Mail\_Persona`,`Cellulare\_Persona`,`Telefono\_Persona`) VALUES ("OYHNUC61N78P638T","Donec@Donec.co.uk","343-5588814","0251-225369");

INSERT INTO `Piscine-Roma-DB`.`Contatti` (`CodFisc\_Persona\_P`,`Mail\_Persona`,`Cellulare\_Persona`,`Telefono\_Persona`) VALUES ("QPWLCB96X48O615D","Aliquam.gravida.mauris@Aliquamornare.net","372-9419490","0252-710981");

INSERT INTO `Piscine-Roma-DB`.`Contatti` (`CodFisc\_Persona\_P`,`Mail\_Persona`,`Cellulare\_Persona`,`Telefono\_Persona`) VALUES ("LYGTVF49G59H298G","euismod.est.arcu@venenatisvel.com","394-6473894","0291-402167");

INSERT INTO `Piscine-Roma-DB`.`Contatti` (`CodFisc\_Persona\_P`,`Mail\_Persona`,`Cellulare\_Persona`,`Telefono\_Persona`) VALUES ("IXTUIY19O35Y472J","vitae@rutrum.org","342-8309620","0389-138040");

INSERT INTO `Piscine-Roma-DB`.`Contatti` (`CodFisc\_Persona\_P`,`Mail\_Persona`,`Cellulare\_Persona`,`Telefono\_Persona`) VALUES ("GJHOPO67K52J454K","id.ante.Nunc@Nuncullamcorper.com","335-9583196","0766-668956");

INSERT INTO `Piscine-Roma-DB`.`Contatti` (`CodFisc\_Persona\_P`,`Mail\_Persona`,`Cellulare\_Persona`,`Telefono\_Persona`) VALUES ("LXPPNS32M56U463F","Ut.semper@condimentumDonecat.ca","359-4136641","0382-801899");

INSERT INTO `Piscine-Roma-DB`.`Contatti` (`CodFisc\_Persona\_P`,`Mail\_Persona`,`Cellulare\_Persona`,`Telefono\_Persona`) VALUES ("OYNSHY15I47R935V","nisi.nibh@ametrisus.com","335-3094095","0814-859572");

INSERT INTO `Piscine-Roma-DB`.`Contatti` (`CodFisc\_Persona\_P`,`Mail\_Persona`,`Cellulare\_Persona`,`Telefono\_Persona`) VALUES ("KPIESM18W54D778K","ipsum@nisimagna.co.uk","383-2827480","0729-228027");

INSERT INTO `Piscine-Roma-DB`.`Contatti` (`CodFisc\_Persona\_P`,`Mail\_Persona`,`Cellulare\_Persona`,`Telefono\_Persona`) VALUES ("MDJKPJ26N79R392N","imperdiet.ornare@eueratsemper.ca","325-5599679","0518-189359");

INSERT INTO `Piscine-Roma-DB`.`Contatti` (`CodFisc\_Persona\_P`,`Mail\_Persona`,`Cellulare\_Persona`,`Telefono\_Persona`) VALUES ("APLXTO83V31A695Q","mollis.dui.in@odio.com","329-0116606","0239-874667");

INSERT INTO `Piscine-Roma-DB`.`Contatti` (`CodFisc\_Persona\_P`,`Mail\_Persona`,`Cellulare\_Persona`,`Telefono\_Persona`) VALUES ("DJWKIH87G71J681X","purus.accumsan@ornare.net","384-6032065","0759-955090");

INSERT INTO `Piscine-Roma-DB`.`Contatti` (`CodFisc\_Persona\_P`,`Mail\_Persona`,`Cellulare\_Persona`,`Telefono\_Persona`) VALUES ("BEMJZF28C73G445W","[aliquet.metus@tempusmauriserat.edu](mailto:aliquet.metus@tempusmauriserat.edu)","366-6473004","0185-271430");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("BHEPWE78S83X958V","Roberto Piazza","2019-03-04","1971-01-29");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("BVIXSR94H68T423J","Enrico Ferrara","2019-07-29","1987-07-08");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("CVFFGJ23O96R727K","Veronica Antonelli","2019-01-25","1983-08-05");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("ENVZBV11A74D638E","Stefano Mazza","2019-09-19","1969-05-10");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("FNYKRA94A84T312B","Noemi Mariani","2019-10-09","1963-07-21");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("HDYJYD93L99U962B","Alice Aiello","2019-03-30","1976-08-06");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("HIWBYT94K48Z642M","Elena Neri","2019-07-14","1989-11-12");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("HXGVSN17V15J346D","Gianni Bianco","2019-06-03","1977-04-21");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("HXGZMN37X43K484K","Alberto Valentini","2019-04-03","1985-10-23");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("IVAGEV54J45K421Y","Simona Costantini","2019-04-05","1996-10-12");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("IYDUWI39L66A253C","Veronica Proietti","2019-12-05","1969-04-30");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("JNFZEI46L53H962L","Angelo Calabrese","2019-04-08","1985-05-22");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("JRBKMN51D23S875J","Giulietta Mele","2019-05-04","1981-09-17");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("KGPOYT36R58H886U","Noemi Garofalo","2019-02-15","2000-11-29");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("LTUPYT21T52K518V","Arianna Costantini","2019-10-12","1992-01-21");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("MDJKPJ26N79R392N","Alessio Giorgi","2019-03-22","1989-07-18");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("NFFJRG51G15X437V","Vittoria Sorrentino","2019-03-15","1973-11-05");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("OYHNUC61N78P638T","Davide Rossetti","2019-09-12","1970-08-06");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("PVTGPR11B78E271H","Cristina Bruno","2019-09-13","1983-06-29");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("QIOOGS54E81S383X","Luigi Conte","2019-07-28","1960-01-23");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("RCMOSJ89M27V386B","Samuele Caruso","2019-08-29","1962-05-09");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("RVHQEQ13U82N947G","Daniele Ricci","2019-08-22","1963-02-19");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("SXHYLN71P12I256G","Vincenzo Martinelli","2019-04-16","1986-06-15");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("UHCSEW63R41H139Y","Emma Pagano","2019-04-13","1989-10-27");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("WCFPHP74E53S828J","Claudia Orlando","2019-02-05","1962-11-30");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("WQSYVA66A76X719I","Giorgia Marra","2019-04-03","1976-02-08");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("YAVMON79U14D859T","Matteo Rinaldi","2019-11-17","1971-12-15");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("ZBYXYU86R97F573U","Luigi Bellini","2019-10-07","1983-09-01");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("ZCAGDG93X13P578K","Valerio Cirillo","2019-03-15","1966-11-10");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("ZUXXME35L42Z588H","Jessica Monti","2019-03-31","1973-12-26");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("APFBSK84C89K742F","Alberto Bellini","2019-06-01","1992-10-24");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("DJWKIH87G71J681X","Giulio Fumagalli","2019-10-09","1971-12-18");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("EFHOWM88W52S685I","Valerio Calabrese","2019-06-08","1985-08-02");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("RJQSKK32O56V421N","Silvia Pellegrini","2019-09-26","1995-02-10");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("FUBMUL51Y56M796U","Giorgia Ferrante","2019-09-03","1964-03-03");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("TVXQJA58K47Y764C","Federico Poli","2019-01-19","1972-10-24");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("MUGGKO16K16A261X","Mirko Riva","2019-12-02","1992-07-25");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("RLXSIC98X78X927M","Fabio Ferrara","2019-01-16","1961-04-26");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("SKAVNG32D94B523Z","Silvia Milani","2019-02-11","1996-02-02");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("APFERG35F48U254F","Francesca Grassi","2019-12-31","1976-04-14");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("GJHOPO67K52J454K","Sara Ferretti","2019-11-24","2000-02-07");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("UOBLNT11E74M412Z","Simona Mancini","2019-07-06","1988-10-23");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("HFLYUA68B48O757J","Elena Palmieri","2019-10-24","1973-03-22");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("BIVVPC82X53O492J","Christian Romano","2019-12-23","1978-04-27");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("QTFKKI21J52G273U","Gianluca Morelli","2019-10-28","1976-10-24");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("VYXBHD38Q59Q142D","Alessio Greco","2019-06-11","1977-07-26");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("JXEHJV64F82G136U","Nicoletta Fabbri","2019-12-14","1976-07-22");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("FVSCXW98F83B572R","Sara Ferrara","2019-06-08","1960-02-22");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("NVMXTA37O62S522P","Domenico Conti","2019-02-26","1993-09-12");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("KPIESM18W54D778K","Samuele Serra","2019-03-09","1983-06-20");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("ELWJCY27T39Y121W","Cristian Vitali","2019-06-21","1996-01-31");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("KLXZSB19H59Q863I","Stefania Proietti","2019-06-03","1962-06-04");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("HOVVKI67I54O292X","Giulio Sorrentino","2019-09-10","1991-11-22");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("IXTUIY19O35Y472J","Irene Costa","2019-07-11","1960-06-14");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("FEVOLP39M29B984I","Giacomo Ruggeri","2019-02-05","1997-01-03");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("CCBETP53Z92V992S","Alessia Piras","2019-07-12","1990-06-19");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("TDDCAU61P52C191C","Martina Caruso","2019-09-25","1986-01-10");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("CLPPFK67Q43X994I","Monica Fontana","2019-07-12","1962-04-28");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("SOCGZI99P23F635O","Gianpaolo Romano","2019-11-04","1982-12-05");

INSERT INTO `Piscine-Roma-DB`.`Iscritto\_Corsi` (`CodFisc\_Persona\_IC`,`Medico`,`Data\_certificato`,`Data\_nascita`) VALUES ("PILNRI72V79D314V","Vanessa Santoro","2019-12-29","1972-05-29");

INSERT INTO `Piscine-Roma-DB`.Iscrizione (CodFisc\_Persona\_IC\_IS,Nome\_Corso\_IS,Nome\_Piscina\_Corso\_IS) VALUES

('APFBSK84C89K742F','acquagym','AquaTibur')

,('BVIXSR94H68T423J','acquagym','AquaTibur')

,('HOVVKI67I54O292X','acquagym','CSI Roma Est')

,('JNFZEI46L53H962L','acquagym','CSI Roma Est')

,('APFERG35F48U254F','acquagym','Piscina Delle Rose')

,('CVFFGJ23O96R727K','acquagym','Roma 12')

,('FUBMUL51Y56M796U','dorso','AquaTibur')

,('BIVVPC82X53O492J','dorso','Azzurra')

,('DJWKIH87G71J681X','dorso','Octopus Acquatic Club')

,('SXHYLN71P12I256G','dorso','Olimpia 1')

;

INSERT INTO `Piscine-Roma-DB`.Iscrizione (CodFisc\_Persona\_IC\_IS,Nome\_Corso\_IS,Nome\_Piscina\_Corso\_IS) VALUES

('CVFFGJ23O96R727K','dorso','Roma 12')

,('BIVVPC82X53O492J','hydrobike','Azzurra')

,('FEVOLP39M29B984I','hydrobike','Juventus Nuoto')

,('GJHOPO67K52J454K','hydrobike','Roma Uno')

,('ELWJCY27T39Y121W','nuoto neonatale','Crawl 2000')

,('APFBSK84C89K742F','pallanuoto','AquaTibur')

,('CLPPFK67Q43X994I','pallanuoto','AquaTibur')

,('HOVVKI67I54O292X','pallanuoto','CSI Roma Est')

,('BHEPWE78S83X958V','pallanuoto','Olimpia 1')

,('PVTGPR11B78E271H','pallanuoto','Roma Uno')

;

INSERT INTO `Piscine-Roma-DB`.Iscrizione (CodFisc\_Persona\_IC\_IS,Nome\_Corso\_IS,Nome\_Piscina\_Corso\_IS) VALUES

('CCBETP53Z92V992S','rana','Crawl 2000')

,('NVMXTA37O62S522P','rana','Octopus Acquatic Club')

,('FUBMUL51Y56M796U','sincronizzato','AquaTibur')

,('APFERG35F48U254F','sincronizzato','Piscina Delle Rose')

,('APFBSK84C89K742F','stile libero','AquaTibur')

,('IXTUIY19O35Y472J','stile libero','Crawl 2000')

,('ELWJCY27T39Y121W','stile libero','Delta')

,('ZBYXYU86R97F573U','stile libero','Roma Uno')

,('BVIXSR94H68T423J','tuffi','AquaTibur')

,('JRBKMN51D23S875J','tuffi','AquaTibur')

;

INSERT INTO `Piscine-Roma-DB`.Iscrizione (CodFisc\_Persona\_IC\_IS,Nome\_Corso\_IS,Nome\_Piscina\_Corso\_IS) VALUES

('GJHOPO67K52J454K','tuffi','Piscina Delle Rose')

;

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("WOPLZK22V67I829S","Lucia Martinelli","0956-465015","396-6677072");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("OLFMMR86E63X476P","Alessandro Vitale","0565-209747","366-8088379");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("TYPXUV58S87U782H","Francesco Caruso","0921-289985","342-1022351");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("QKVIFD71S16X358B","Emanuele Cattaneo","0881-977695","321-2023994");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("JZREDD42E18K447E","Debora Piazza","0971-412867","326-6383968");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("FHTRVE32U94E696P","Martina Ceccarelli","0552-277737","311-1425628");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("INDACE69M79X514T","Gianni Grimaldi","0273-603212","393-6044591");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("HONYJD31V17I559Y","Daniele Zanetti","0397-379790","385-9007661");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("ZUGQCE81X85U678V","Vanessa Benedetti","0654-022419","352-1505168");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("DCNAUY95X57Q238F","Valeria Villa","0332-605274","347-8823046");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("FTUQFA32L41M478L","Christian Martinelli","0547-731151","345-3390117");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("QZIXUH81G76S742N","Manuel Pastore","0858-052433","315-3165929");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("BEPESP91N32S144A","Emanuele Ceccarelli","0666-300655","311-5005494");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("QMBSMZ85G62B462M","Camilla Guidi","0526-404923","372-9874645");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("VELIJK79Y69J435L","Stefano Valentini","0843-556881","397-0532374");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("WXAXML68Q84K615X","Nicole Valentini","0882-780727","317-3654280");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("UEAHQA52Z35T869C","Federico Martino","0196-547662","389-8120988");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("HHCFTK35D84V552H","Giulia Testa","0781-567720","371-3933022");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("REXHUV69U11F915M","Diego Gallo","0718-559645","322-3108377");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("GXNPMP69Z93N455R","Francesca Villa","0188-664189","377-0140461");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("UAKFFL84E66A692X","Leonardo Bellini","0911-285208","392-1213534");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("DJGWFN63B48T835T","Beatrice Pellegrino","0355-915270","377-6682103");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("AUJZXD85R36S337X","Marta Ferrari","0178-430064","356-0592975");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("ABXIOT95T83K775C","Stefania Rizzo","0999-911187","314-7642994");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("GADEYD81U12F744T","Vittoria Riva","0261-036358","397-8487403");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("EBMQIM21F91I343R","Lucia Guidi","0479-506322","350-7532364");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("ENXXWF33P88K189U","Francesco Marino","0612-210527","339-2787783");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("FXYYXX43R81F461R","Angelo Caruso","0127-357871","379-6513166");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("KDAYSP74O24A626G","Ilaria Santini","0891-147523","331-8935994");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("LXPDAP14F89A931V","Marco Martinelli","0524-230707","352-1243544");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("BCBKNH16I57M188A","Maria Leone","0789-600823","386-1251899");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("QIBKSS16L51T413V","Lorenzo Marchetti","0942-241767","368-2583945");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("DTQNHJ81W46V681O","Edoardo Lombardo","0353-519442","328-1095845");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("IFJHTX14P27M254U","Giovanni Marchi","0119-611206","369-5723516");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("KBDXSO41O71T671C","Sara Villani","0221-678503","370-1073449");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("URKLXT94I87T744M","Luigi Morelli","0723-790104","312-2962987");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("QUECRN86W55X891L","Tommaso Pagano","0763-275560","356-6143371");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("UNDWWO82L94K127C","Mattia Morelli","0316-129469","332-8985111");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("BPRSDV62C49X657D","Cristina Grasso","0611-717971","351-7285939");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("ZEVRXD85K24Y584S","Angela Palmieri","0358-526020","352-3406442");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("UPNLDS46I97S916A","Elisa Martinelli","0295-006565","393-9349786");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("ZHMEJC68S91U221O","Paolo Pepe","0623-899017","360-8831797");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("LLWDWH29L38A112H","Riccardo Palmieri","0258-040710","385-2585211");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("FCEQGH61S26G974F","Arianna Benedetti","0793-984703","366-4367810");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("SDUPBY48V64Q767U","Vanessa Marra","0131-145108","394-1099872");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("UHEOOL17R85Y717H","Paolo Pepe","0967-173659","378-1520353");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("HCJJMW23E95E221X","Dario Piras","0616-559583","392-0221635");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("OAUUXU79T91U458K","Alex Barone","0925-785866","396-2048507");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("CVBXTB61U69W276X","Augusto Fiore","0448-711123","376-2306666");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("FLMIWQ69U58Z996Q","Giacomo Proietti","0357-214534","371-5828841");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("WRBAQE13H26L783O","Claudia Pinna","0278-587839","391-9745368");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("XTKVCI15N93J675Q","Claudio Gatti","0516-661211","328-3977268");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("JASCEI89G76X687E","Giacomo Monti","0233-026820","325-8810869");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("PAAEFC57D63D862F","Maria Riva","0749-030136","380-1623124");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("PCVDOC27J88H386I","Dario Calabrese","0681-387583","393-6011652");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("JQZZRC92H71I377F","Monica Orlando","0891-819859","368-4472970");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("LOUNVR17U71W894G","Domenico Bruno","0566-695865","384-5809364");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("SMOBCU53B92E547P","Alessandro Barbieri","0182-402306","369-6364494");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("UKWARY15Y65I922K","Mirko Mele","0633-034018","353-5005798");

INSERT INTO `Piscine-Roma-DB`.`Insegnante` (`CodFisc\_I`,`Nome\_I`,`Telefono\_I`,`Cellulare\_I`) VALUES ("OXSDAD44N25E329B","Samuele Santini","0768-141184","332-3875613");

INSERT INTO `Piscine-Roma-DB`.Qualifica (Nome\_Q) VALUES

('acquagym')

,('nuoto neonatale')

,('pallanuoto')

,('salvamento')

,('sincronizzato')

,('sub')

,('tuffi')

,('tuffi sincronizzati')

;

INSERT INTO `Piscine-Roma-DB`.Insegnante\_Ha\_Qualifica (CodFisc\_Insegnante,Nome\_Qualifica) VALUES

('ABXIOT95T83K775C','acquagym')

,('JQZZRC92H71I377F','acquagym')

,('BEPESP91N32S144A','nuoto neonatale')

,('DCNAUY95X57Q238F','nuoto neonatale')

,('FHTRVE32U94E696P','nuoto neonatale')

,('BEPESP91N32S144A','pallanuoto')

,('DCNAUY95X57Q238F','pallanuoto')

,('LLWDWH29L38A112H','pallanuoto')

,('LOUNVR17U71W894G','pallanuoto')

,('OXSDAD44N25E329B','pallanuoto')

;

INSERT INTO `Piscine-Roma-DB`.Insegnante\_Ha\_Qualifica (CodFisc\_Insegnante,Nome\_Qualifica) VALUES

('QMBSMZ85G62B462M','pallanuoto')

,('HONYJD31V17I559Y','salvamento')

,('JASCEI89G76X687E','salvamento')

,('UKWARY15Y65I922K','salvamento')

,('ABXIOT95T83K775C','sincronizzato')

,('CVBXTB61U69W276X','sincronizzato')

,('WXAXML68Q84K615X','sincronizzato')

,('ABXIOT95T83K775C','sub')

,('ENXXWF33P88K189U','sub')

,('LLWDWH29L38A112H','sub')

;

INSERT INTO `Piscine-Roma-DB`.Insegnante\_Ha\_Qualifica (CodFisc\_Insegnante,Nome\_Qualifica) VALUES

('FLMIWQ69U58Z996Q','tuffi')

,('AUJZXD85R36S337X','tuffi sincronizzati')

,('FCEQGH61S26G974F','tuffi sincronizzati')

;

INSERT INTO `Piscine-Roma-DB`.Rotazione (Data\_inizio,Data\_fine,CodFiscale\_Insegnante,Nome\_Piscina\_R) VALUES

('2020-01-01','2020-01-31','ABXIOT95T83K775C','AquaTibur')

,('2020-02-01','2020-02-29','ABXIOT95T83K775C','Azzurra')

,('2020-03-01','2020-03-31','ABXIOT95T83K775C','Crawl 2000')

,('2020-01-01','2020-01-15','AUJZXD85R36S337X','Azzurra')

,('2020-01-16','2020-01-31','BCBKNH16I57M188A','Azzurra')

,('2020-02-01','2020-02-10','BEPESP91N32S144A','Azzurra')

,('2020-02-21','2020-02-29','BEPESP91N32S144A','AquaTibur')

,('2020-02-11','2020-03-31','BPRSDV62C49X657D','Azzurra')

,('2020-02-01','2020-02-20','DCNAUY95X57Q238F','AquaTibur')

;

######### USERS #############

########### INSEGNANTE #############

CREATE USER 'insegnante'@'localhost' IDENTIFIED BY 'insegnante123';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.report\_insegnante TO 'insegnante'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.is\_CF\_insegnante\_valid TO 'insegnante'@'localhost';

########### ADDETTO SEGRETERIA #############

CREATE USER 'addetto\_segreteria'@'localhost' IDENTIFIED BY 'asegreteria123';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.is\_piscina\_addetto\_valid TO 'addetto\_segreteria'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.persona\_inserisci TO 'addetto\_segreteria'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.persona\_modifica TO 'addetto\_segreteria'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.persona\_elimina TO 'addetto\_segreteria'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.persona\_inserisci\_contatti TO 'addetto\_segreteria'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.persona\_modifica\_contatti TO 'addetto\_segreteria'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.persona\_elimina\_contatti TO 'addetto\_segreteria'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.persona\_effettua\_ingresso\_libero TO 'addetto\_segreteria'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.persona\_inserisci\_iscritta\_corsi TO 'addetto\_segreteria'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.persona\_modifica\_iscritta\_corsi TO 'addetto\_segreteria'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.persona\_elimina\_iscritta\_corsi TO 'addetto\_segreteria'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.persona\_effettua\_iscrizione\_corso TO 'addetto\_segreteria'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.persona\_cancella\_iscrizione\_corso TO 'addetto\_segreteria'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.corso\_visualizza\_lista\_iscritti TO 'addetto\_segreteria'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.corso\_inserisci TO 'addetto\_segreteria'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.corso\_modifica TO 'addetto\_segreteria'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.corso\_elimina TO 'addetto\_segreteria'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.corso\_visualizza\_lista\_per\_piscina TO 'addetto\_segreteria'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.corso\_inserisci\_lezione TO 'addetto\_segreteria'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.corso\_modifica\_lezione TO 'addetto\_segreteria'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.corso\_elimina\_lezione TO 'addetto\_segreteria'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.corso\_visualizza\_lista\_lezioni TO 'addetto\_segreteria'@'localhost';

########### ADDETTO COMUNALE #############

CREATE USER 'addetto\_comunale'@'localhost' IDENTIFIED BY 'acomunale123';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.piscina\_inserisci TO 'addetto\_comunale'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.piscina\_modifica TO 'addetto\_comunale'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.piscina\_elimina TO 'addetto\_comunale'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.piscina\_visualizza\_lista TO 'addetto\_comunale'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.insegnante\_inserisci TO 'addetto\_comunale'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.insegnante\_modifica TO 'addetto\_comunale'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.insegnante\_elimina TO 'addetto\_comunale'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.insegnante\_visualizza\_lista TO 'addetto\_comunale'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.qualifica\_inserisci TO 'addetto\_comunale'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.qualifica\_modifica TO 'addetto\_comunale'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.qualifica\_elimina TO 'addetto\_comunale'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.insegnante\_assegna\_qualifica TO 'addetto\_comunale'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.insegnante\_cancella\_qualifica TO 'addetto\_comunale'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.rotazione\_inserisci TO 'addetto\_comunale'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.rotazione\_modifica TO 'addetto\_comunale'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.rotazione\_elimina TO 'addetto\_comunale'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.rotazione\_visualizza\_lista\_per\_piscina TO 'addetto\_comunale'@'localhost';

GRANT EXECUTE ON PROCEDURE `Piscine-Roma-DB`.rotazione\_visualizza\_lista\_per\_insegnante TO 'addetto\_comunale'@'localhost';

FLUSH PRIVILEGES;

## Codice del Front-End

Program.c

------------------------------------------------------------------------------------------------------------------------------------

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <mysql/mysql.h>

#include <unistd.h>

#include "defines.h"

#define fflush(stdin) while ((getchar()) != '\n')

#define true 1

#define false 0

MYSQL \*conn;

MYSQL \*login;

char u[255];

char p[255];

char c;

int cmd1 = 0;

int cmd2 = 0;

int num\_fields;

MYSQL\_RES \*result;

MYSQL\_ROW row;

MYSQL\_FIELD \*field;

char CF\_I[17];

char NOME\_PISCINA\_ADDETTO[50];

char query[255];

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* INSEGNANTE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int is\_CF\_insegnante\_valid(char\* CodFisc\_I){

MYSQL\_RES\* res;

snprintf(query, 1000, "CALL is\_CF\_insegnante\_valid('%s');",CodFisc\_I);

res = get\_result\_from\_sql\_query(query);

row = mysql\_fetch\_row(res);

// verifico dal return di della procedure EXIST

// return 1: esiste il codice fiscale insegnante

// return 0: il codice fiscale non corrisponde ad un insegnante

printf("RETURN: %s\n",row[0]);

if (strcmp(row[0], "1") == 0){ // se esiste il codice fiscale...

return 1;

}

return 0;

}

void insegnante\_logged(){

printf("\n######################################\n");

printf("INFO: Accesso eseguito come insegnante.\n\n");

printf("> Inserisci il tuo codice fiscale per continuare: ");

scanf ("%s",CF\_I);

printf("\n");

if(is\_CF\_insegnante\_valid(CF\_I)){

printf("INFO: Accesso eseguito per l'insegnante %s\n",CF\_I);

while(true){

printf("\n##################################################\n");

printf(" Lista funzioni eseguibili dall'utente insegnante %s:\n",CF\_I);

printf("##################################################\n\n");

printf(" 1) Esegui report di lavoro settimanale (7 giorni dalla data inserita)\n");

printf(" 2) Esegui report di lavoro mensile (31 giorni dalla data inserita)\n");

printf(" 99) Termina\n\n");

printf("> Inserisci un Comando: ");

scanf ("%i",&cmd2);

char inizio[11];

if(cmd2 == 1){

printf("\n> Inserisci la data di inizio del report settimanale [AAAA-MM-GG]: ");

scanf ("%s",inizio);

snprintf(query, 1000, "call report\_insegnante('%s','%d','%s');",inizio,7,CF\_I);

print\_sql\_query(query);

}else if(cmd2 == 2){

printf("\n> Inserisci la data di inizio del report mensile [AAAA-MM-GG]: ");

scanf ("%s",inizio);

snprintf(query, 1000, "call report\_insegnante('%s','%d','%s');",inizio,31,CF\_I);

print\_sql\_query(query);

}else if(cmd2 == 99){

printf("\nINFO: Uscita...Bye\n");

break;

}

}

} else printf("ERROR: Codice fiscale non valido\n");

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ADDETTO SEGRETERIA \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void addetto\_segreteria\_logged(){

printf("\n######################################\n");

printf("INFO: Accesso eseguito come addetto segreteria.\n\n");

printf("> Inserisci il nome della piscina presso cui lavori: ");

scanf ("%s",NOME\_PISCINA\_ADDETTO);

printf("\n");

if(is\_piscina\_addetto\_valid(NOME\_PISCINA\_ADDETTO)){

printf("INFO: Accesso eseguito per l'addetto alla segreteria, presso %s\n",NOME\_PISCINA\_ADDETTO);

while(true){

printf("\n#########################################################\n");

printf(" Lista funzioni eseguibili dall'utente addetto segreteria :\n");

printf("##########################################################\n\n");

printf(" 1) Inserisci persona\n");

printf(" 2) Modifica persona\n");

printf(" 3) Elimina persona\n\n");

printf(" 4) Effettua ingresso libero presso %s\n\n",NOME\_PISCINA\_ADDETTO);

printf(" 5) Inserisci certificato medico per corsi presso %s\n",NOME\_PISCINA\_ADDETTO);

printf(" 6) Modifica certificato medico per corsi presso %s\n",NOME\_PISCINA\_ADDETTO);

printf(" 7) Elimina certificato medico per corsi presso %s\n\n",NOME\_PISCINA\_ADDETTO);

printf(" 8) Effettua iscrizione ad un corso presso %s\n",NOME\_PISCINA\_ADDETTO);

printf(" 9) Cancella iscrizione ad un corso presso %s\n\n",NOME\_PISCINA\_ADDETTO);

printf(" 10) Inserisci corso presso %s\n",NOME\_PISCINA\_ADDETTO);

printf(" 11) Modifica corso presso %s\n",NOME\_PISCINA\_ADDETTO);

printf(" 12) Elimina corso presso %s\n\n",NOME\_PISCINA\_ADDETTO);

printf(" 13) Inserisci lezione di un corso presso %s\n",NOME\_PISCINA\_ADDETTO);

printf(" 14) Modifica orario lezione di un corso presso %s\n",NOME\_PISCINA\_ADDETTO);

printf(" 15) Elimina lezione di un corso presso %s\n\n",NOME\_PISCINA\_ADDETTO);

printf(" 16) Visualizza lista corsi presso %s\n",NOME\_PISCINA\_ADDETTO);

printf(" 17) Visualizza lista lezioni di un corso presso %s\n",NOME\_PISCINA\_ADDETTO);

printf(" 18) Visualizza lista persone iscritte ad un corso presso %s\n\n",NOME\_PISCINA\_ADDETTO);

printf(" 99) Termina\n\n");

printf("> Inserisci un Comando: ");

scanf ("%i",&cmd2);

fflush(stdin);

switch(cmd2){

case 1:

persona\_inserisci();

break;

case 2:

persona\_modifica();

break;

case 3:

persona\_elimina();

break;

case 4:

persona\_effettua\_ingresso\_libero();

break;

case 5:

persona\_inserisci\_iscritta\_corsi();

break;

case 6:

persona\_modifica\_iscritta\_corsi();

break;

case 7:

persona\_elimina\_iscritta\_corsi();

break;

case 8:

persona\_effettua\_iscrizione\_corso();

break;

case 9:

persona\_cancella\_iscrizione\_corso();

break;

case 10:

corso\_inserisci();

break;

case 11:

corso\_modifica();

break;

case 12:

corso\_elimina();

break;

case 13:

corso\_inserisci\_lezione();

break;

case 14:

corso\_modifica\_lezione();

break;

case 15:

corso\_elimina\_lezione();

break;

case 16:

corso\_visualizza\_lista\_per\_piscina();

break;

case 17:

corso\_visualizza\_lista\_lezioni();

break;

case 18:

corso\_visualizza\_lista\_iscritti();

break;

case 99:

printf("\nINFO: Uscita...Bye\n");

exit(0);

}

}

}else printf("ERROR: Nome piscina non valido\n");

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ADDETTO COMUNALE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void addetto\_comunale\_logged(){

printf("\n######################################\n");

printf("INFO: Accesso eseguito come addetto comunale.\n\n");

while(true){

printf("\n#########################################################\n");

printf(" Lista funzioni eseguibili dall'utente addetto comunale :\n");

printf("##########################################################\n\n");

printf(" 1) Inserisci piscina\n");

printf(" 2) Modifica piscina\n");

printf(" 3) Elimina piscina\n\n");

printf(" 4) Inserisci insegnante\n");

printf(" 5) Modifica insegnante\n");

printf(" 6) Elimina insegnante\n\n");

printf(" 7) Inserisci qualifica\n");

printf(" 8) Modifica qualifica\n");

printf(" 9) Elimina qualifica\n\n");

printf(" 10) Assegna qualifica ad un insegnante\n");

printf(" 11) Cancella qualifica ad un insegnante\n\n");

printf(" 12) Inserisci rotazione insegnante\n");

printf(" 13) Modifica rotazione insegnante\n");

printf(" 14) Elimina rotazione insegnante\n\n");

printf(" 15) Visualizza piscine\n");

printf(" 16) Visualizza insegnanti\n");

printf(" 17) Visualizza rotazioni presso una piscina\n");

printf(" 18) Visualizza rotazioni assegnate ad un insegnante\n\n");

printf(" 99) Termina\n\n");

printf("> Inserisci un Comando: ");

scanf ("%i",&cmd2);

fflush(stdin);

switch(cmd2){

case 1:

piscina\_inserisci();

break;

case 2:

piscina\_modifica();

break;

case 3:

piscina\_elimina();

break;

case 4:

insegnante\_inserisci();

break;

case 5:

insegnante\_modifica();

break;

case 6:

insegnante\_elimina();

break;

case 7:

qualifica\_inserisci();

break;

case 8:

qualifica\_modifica();

break;

case 9:

qualifica\_elimina();

break;

case 10:

insegnante\_assegna\_qualifica();

break;

case 11:

insegnante\_cancella\_qualifica();

break;

case 12:

rotazione\_inserisci();

break;

case 13:

rotazione\_modifica();

break;

case 14:

rotazione\_elimina();

break;

case 15:

piscina\_visualizza\_lista();

break;

case 16:

insegnante\_visualizza\_lista();

break;

case 17:

rotazione\_visualizza\_lista\_per\_piscina();

break;

case 18:

rotazione\_visualizza\_lista\_per\_insegnante();

break;

case 99:

printf("\nINFO: Uscita...Bye\n");

exit(0);

}

}

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* LOGIN \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void login\_f () {

printf("\n########### Login ###########\n\n");

printf("> Inserisci l'username: ");

scanf("%s",u);

printf("> Inserisci la password: ");

scanf("%s",p);

printf("\n\n");

conn = mysql\_init (NULL);

login = mysql\_real\_connect(conn, "localhost",u,p, "Piscine-Roma-DB", 3306, NULL, 0);

if (login == NULL) {

fprintf(stderr, "%s\n", mysql\_error(conn));

mysql\_close(conn);

exit(1);

} else {

printf ("\nINFO: Connessione riuscita\n");

if (strcmp(u, "insegnante") == 0) {

insegnante\_logged();

} else if (strcmp(u, "addetto\_segreteria") == 0) {

addetto\_segreteria\_logged();

} else if (strcmp(u, "addetto\_comunale") == 0) {

addetto\_comunale\_logged();

}

}

mysql\_close(conn);

exit(0);

}

int main (int argc, char \*argv[]) {

while (1) {

printf("################### Gestione piscine comunali di Roma ###################\n\n");

printf(" 1) Login\n");

printf(" 99) Termina\n\n");

printf("> Inserisci un Comando: ");

scanf ("%i",&cmd1);

fflush(stdin);

if (cmd1 == 1) {

login\_f();

} else if (cmd1 == 99) {

printf("\nINFO: Uscita...Bye\n");

exit(0);

}

}

}

------------------------------------------------------------------------------------------------------------------------------------

defines.h

------------------------------------------------------------------------------------------------------------------------------------

#include <mysql/mysql.h>

extern MYSQL \*conn;

extern MYSQL \*login;

extern char u[255];

extern char p[255];

extern char c;

extern int cmd1;

extern int cmd2;

extern int num\_fields;

extern MYSQL\_RES \*result;

extern MYSQL\_ROW row;

extern MYSQL\_FIELD \*field;

extern char CF\_I[17];

extern char NOME\_PISCINA\_ADDETTO[50];

extern char query[255];

// UTILITY

extern void finish\_with\_error();

extern void input\_wait();

extern MYSQL\_RES\* get\_result\_from\_sql\_query(char \*query);

extern void print\_sql\_query(char \*query);

extern void run\_sql\_query (char \*query);

// ADDETTO SEGRETERIA

extern void persona\_inserisci();

extern void persona\_modifica();

extern void persona\_elimina();

extern void persona\_effettua\_ingresso\_libero();

extern void persona\_inserisci\_iscritta\_corsi();

extern void persona\_modifica\_iscritta\_corsi();

extern void persona\_elimina\_iscritta\_corsi();

extern void persona\_effettua\_iscrizione\_corso();

extern void persona\_cancella\_iscrizione\_corso();

extern void corso\_inserisci();

extern void corso\_modifica();

extern void corso\_elimina();

extern void corso\_inserisci\_lezione();

extern void corso\_modifica\_lezione();

extern void corso\_elimina\_lezione();

extern int is\_piscina\_addetto\_valid(char\* piscina);

extern void corso\_visualizza\_lista\_per\_piscina();

extern void corso\_visualizza\_lista\_lezioni();

extern void corso\_visualizza\_lista\_iscritti();

// ADDETTO COMUNALE

extern void piscina\_inserisci();

extern void piscina\_modifica();

extern void piscina\_elimina();

extern void insegnante\_inserisci();

extern void insegnante\_modifica();

extern void insegnante\_elimina();

extern void qualifica\_inserisci();

extern void qualifica\_modifica();

extern void qualifica\_elimina();

extern void insegnante\_assegna\_qualifica();

extern void insegnante\_cancella\_qualifica();

extern void rotazione\_inserisci();

extern void rotazione\_modifica();

extern void rotazione\_elimina();

extern void piscina\_visualizza\_lista();

extern void insegnante\_visualizza\_lista();

extern void rotazione\_visualizza\_lista\_per\_piscina();

extern void rotazione\_visualizza\_lista\_per\_insegnante();

------------------------------------------------------------------------------------------------------------------------------------

utility.c

------------------------------------------------------------------------------------------------------------------------------------

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "defines.h"

#define fflush(stdin) while ((getchar()) != '\n')

void finish\_with\_error(MYSQL \*con, char \*err) {

fprintf(stderr, "%s error: %s\n", err, mysql\_error(con));

mysql\_close(con);

exit(1);

}

void input\_wait() {

char c;

printf("\nINFO: Premi invio per continuare: \n");

while (c = getchar() != '\n'){}

}

MYSQL\_RES\* get\_result\_from\_sql\_query(char \*query){

printf("QUERY: %s ",query);

mysql\_query (conn,query);

result = mysql\_store\_result(conn);

if (result == NULL){

finish\_with\_error(conn, "errore");

}

MYSQL\_RES \*res;

res = result;

mysql\_free\_result(result);

mysql\_next\_result(conn);

input\_wait();

return res;

}

void print\_sql\_query(char \*query){

printf("QUERY: %s \n",query);

mysql\_query (conn,query);

result = mysql\_store\_result(conn);

if (result == NULL){

finish\_with\_error(conn, "errore");

}

num\_fields = mysql\_num\_fields(result);

printf ("\n");

while ((row = mysql\_fetch\_row(result))){

for(int i = 0; i < num\_fields; i++) {

if (i == 0) {

while(field = mysql\_fetch\_field(result)){ //include il nome della colonna nella stampa

printf( "| %s ", field->name);

}

printf ("\n");

}

printf(" %s ", row[i] ? row[i] : "NULL");

}

}

printf("\n");

mysql\_free\_result(result);

mysql\_next\_result(conn);

input\_wait();

}

void run\_sql\_query (char \*query) {

printf("QUERY: %s \n",query);

if(mysql\_query(conn,query)) {

finish\_with\_error(conn, "Query");

} else {

input\_wait();

}

}

------------------------------------------------------------------------------------------------------------------------------------

addetto\_comunale.c

------------------------------------------------------------------------------------------------------------------------------------

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "defines.h"

#define fflush(stdin) while ((getchar()) != '\n')

void piscina\_inserisci(){

char nome[50],indirizzo[30],telefono[20],responsabile[30],da[11],a[11];

int tipo;

printf ("\nNome della piscina: ");

scanf ("%s",nome);

fflush(stdin);

printf ("\nIndirizzo della piscina : ");

scanf ("%[^\n]",indirizzo);

fflush(stdin);

printf ("\nTelefono della piscina : ");

scanf ("%[^\n]",telefono);

fflush(stdin);

printf ("\nResponsabile della piscina : ");

scanf ("%[^\n]",responsabile);

fflush(stdin);

printf ("\nLa piscina ha la vasca all'aperto? 1=si,0=no: ");

scanf ("%i",&tipo);

fflush(stdin);

if(tipo == 1){

printf ("\nMese inizio disponibilità vasca all'aperto : ");

scanf ("%[^\n]",da);

fflush(stdin);

printf ("\nMese fine disponibilità vasca all'aperto : ");

scanf ("%[^\n]",a);

fflush(stdin);

snprintf(query, 1000, "call piscina\_inserisci('%s','%s','%s','%s','%i','%s','%s')",

nome, indirizzo, telefono, responsabile, tipo, da, a);

run\_sql\_query(query);

} else if(tipo == 0){

snprintf(query, 1000, "call piscina\_inserisci('%s','%s','%s','%s','%i',NULL,NULL)",

nome, indirizzo, telefono, responsabile, tipo);

run\_sql\_query(query);

}

}

void piscina\_modifica(){

char nome[50],indirizzo[30],telefono[20],responsabile[30],da[11],a[11];

int tipo;

printf ("\nNome della piscina da modificare: ");

scanf ("%[^\n]",nome);

fflush(stdin);

printf ("\nNuovo indirizzo della piscina : ");

scanf ("%[^\n]",indirizzo);

fflush(stdin);

printf ("\nNuovo telefono della piscina : ");

scanf ("%[^\n]",telefono);

fflush(stdin);

printf ("\nNuovo responsabile della piscina : ");

scanf ("%[^\n]",responsabile);

fflush(stdin);

printf ("\nLa piscina ha la vasca all'aperto? 1=si,0=no: ");

scanf ("%i",&tipo);

fflush(stdin);

if(tipo == 1){

printf ("\nMese inizio disponibilità vasca all'aperto : ");

scanf ("%[^\n]",da);

fflush(stdin);

printf ("\nMese fine disponibilità vasca all'aperto : ");

scanf ("%[^\n]",a);

fflush(stdin);

snprintf(query, 1000, "call piscina\_modifica('%s','%s','%s','%s','%i','%s','%s')",

nome, indirizzo, telefono, responsabile, tipo, da, a);

run\_sql\_query(query);

} else if(tipo == 0){

snprintf(query, 1000, "call piscina\_modifica('%s','%s','%s','%s','%i',NULL,NULL)",

nome, indirizzo, telefono, responsabile, tipo);

run\_sql\_query(query);

}

}

void piscina\_elimina(){

char nome[50];

printf ("\nNome della piscina da eliminare: ");

scanf ("%s",nome);

fflush(stdin);

snprintf(query, 1000, "call piscina\_elimina('%s')",nome);

run\_sql\_query(query);

}

void piscina\_visualizza\_lista(){

snprintf(query, 1000, "call piscina\_visualizza\_lista()");

print\_sql\_query(query);

}

void insegnante\_inserisci(){

char CF\_Ins[17],nome[30],telefono[20],cellulare[20];

printf ("\nCodice fiscale dell'insegnante: ");

scanf ("%s",CF\_Ins);

fflush(stdin);

printf ("\nNome : ");

scanf ("%[^\n]",nome);

fflush(stdin);

printf ("\nTelefono : ");

scanf ("%[^\n]",telefono);

fflush(stdin);

printf ("\nCellulare : ");

scanf ("%[^\n]",cellulare);

fflush(stdin);

snprintf(query, 1000, "call insegnante\_inserisci('%s','%s','%s','%s')",CF\_Ins,nome,telefono,cellulare);

run\_sql\_query(query);

}

void insegnante\_modifica(){

char CF\_Ins[17],nome[30],telefono[20],cellulare[20];

printf ("\nCodice fiscale dell'insegnante da modificare: ");

scanf ("%s",CF\_Ins);

fflush(stdin);

printf ("\nNuovo nome : ");

scanf ("%[^\n]",nome);

fflush(stdin);

printf ("\nNuovo telefono : ");

scanf ("%[^\n]",telefono);

fflush(stdin);

printf ("\nNuovo cellulare : ");

scanf ("%[^\n]",cellulare);

fflush(stdin);

snprintf(query, 1000, "call insegnante\_modifica('%s','%s','%s','%s')",CF\_Ins,nome,telefono,cellulare);

run\_sql\_query(query);

}

void insegnante\_elimina(){

char CF\_Ins[17];

printf ("\nCodice fiscale dell'insegnante da modificare: ");

scanf ("%s",CF\_Ins);

fflush(stdin);

snprintf(query, 1000, "call insegnante\_elimina('%s')",CF\_Ins);

run\_sql\_query(query);

}

void insegnante\_visualizza\_lista(){

snprintf(query, 1000, "call insegnante\_visualizza\_lista()");

print\_sql\_query(query);

}

void qualifica\_inserisci(){

char nome[30];

printf ("\nNome della qualifica: ");

scanf ("%[^\n]",nome);

fflush(stdin);

snprintf(query, 1000, "call qualifica\_inserisci('%s')",nome);

run\_sql\_query(query);

}

void qualifica\_modifica(){

char old\_nome[30],new\_nome[30];

printf ("\nNome della qualifica da modificare: ");

scanf ("%[^\n]",old\_nome);

fflush(stdin);

printf ("\nNuovo nome della qualifica: ");

scanf ("%[^\n]",new\_nome);

fflush(stdin);

snprintf(query, 1000, "call qualifica\_modifica('%s','%s')",old\_nome,new\_nome);

run\_sql\_query(query);

}

void qualifica\_elimina(){

char nome[30];

printf ("\nNome della qualifica da eliminare: ");

scanf ("%[^\n]",nome);

fflush(stdin);

snprintf(query, 1000, "call qualifica\_elimina('%s')",nome);

run\_sql\_query(query);

}

void insegnante\_assegna\_qualifica(){

char CodFisc\_I[17],nome\_qual[30];

printf ("\nCodice fiscale dell'insegnante: ");

scanf ("%s",CodFisc\_I);

fflush(stdin);

printf ("\nNome della qualifica: ");

scanf ("%[^\n]",nome\_qual);

fflush(stdin);

snprintf(query, 1000, "call insegnante\_assegna\_qualifica('%s','%s')",CodFisc\_I,nome\_qual);

run\_sql\_query(query);

}

void insegnante\_cancella\_qualifica(){

char CodFisc\_I[17],nome\_qual[30];

printf ("\nCodice fiscale dell'insegnante: ");

scanf ("%s",CodFisc\_I);

fflush(stdin);

printf ("\nNome della qualifica da cancellare: ");

scanf ("%[^\n]",nome\_qual);

fflush(stdin);

snprintf(query, 1000, "call insegnante\_cancella\_qualifica('%s','%s')",CodFisc\_I,nome\_qual);

run\_sql\_query(query);

}

void rotazione\_inserisci(){

char CodFisc\_I[17],piscina[50],inizio[11],fine[11];

printf ("\nCodice fiscale dell'insegnante: ");

scanf ("%s",CodFisc\_I);

fflush(stdin);

printf ("\nNome della piscina : ");

scanf ("%[^\n]",piscina);

fflush(stdin);

printf ("\nData inizio rotazione [AAAA-MM-GG]: ");

scanf ("%s",inizio);

fflush(stdin);

printf ("\nData fine rotazione [AAAA-MM-GG]: ");

scanf ("%s",fine);

fflush(stdin);

snprintf(query, 1000, "call rotazione\_inserisci('%s','%s','%s','%s')",inizio,fine,CodFisc\_I,piscina);

run\_sql\_query(query);

}

void rotazione\_modifica(){

char CodFisc\_I[17],piscina[50],old\_inizio[11],old\_fine[11],new\_inizio[11],new\_fine[11];

printf ("\nCodice fiscale dell'insegnante: ");

scanf ("%s",CodFisc\_I);

fflush(stdin);

printf ("\nNome della piscina : ");

scanf ("%[^\n]",piscina);

fflush(stdin);

printf ("\nData inizio rotazione da modificare [AAAA-MM-GG]: ");

scanf ("%s",old\_inizio);

fflush(stdin);

printf ("\nData fine rotazione da modificare [AAAA-MM-GG]: ");

scanf ("%s",old\_fine);

fflush(stdin);

printf ("\nNuova data inizio rotazione [AAAA-MM-GG]: ");

scanf ("%s",new\_inizio);

fflush(stdin);

printf ("\nNuova data fine rotazione [AAAA-MM-GG]: ");

scanf ("%s",new\_fine);

fflush(stdin);

snprintf(query, 1000, "call rotazione\_modifica('%s','%s','%s','%s','%s','%s')",old\_inizio,old\_fine,CodFisc\_I,piscina,new\_inizio,new\_fine);

run\_sql\_query(query);

}

void rotazione\_elimina(){

char CodFisc\_I[17],piscina[50],inizio[11],fine[11];

printf ("\nCodice fiscale dell'insegnante: ");

scanf ("%s",CodFisc\_I);

fflush(stdin);

printf ("\nNome della piscina : ");

scanf ("%[^\n]",piscina);

fflush(stdin);

printf ("\nData inizio rotazione da eliminare [AAAA-MM-GG]: ");

scanf ("%s",inizio);

fflush(stdin);

printf ("\nData fine rotazione da eliminare [AAAA-MM-GG]: ");

scanf ("%s",fine);

fflush(stdin);

snprintf(query, 1000, "call rotazione\_elimina('%s','%s','%s','%s')",inizio,fine,CodFisc\_I,piscina);

run\_sql\_query(query);

}

void rotazione\_visualizza\_lista\_per\_piscina(){

char piscina[50];

printf ("\nNome della piscina : ");

scanf ("%[^\n]",piscina);

fflush(stdin);

snprintf(query, 1000, "call rotazione\_visualizza\_lista\_per\_piscina('%s')",piscina);

print\_sql\_query(query);

}

void rotazione\_visualizza\_lista\_per\_insegnante(){

char CodFisc\_I[17];

printf ("\nCodice fiscale dell'insegnante: ");

scanf ("%s",CodFisc\_I);

fflush(stdin);

snprintf(query, 1000, "call rotazione\_visualizza\_lista\_per\_insegnante('%s')",CodFisc\_I);

print\_sql\_query(query);

}

------------------------------------------------------------------------------------------------------------------------------------

addetto\_segreteria.c

------------------------------------------------------------------------------------------------------------------------------------

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "defines.h"

#define fflush(stdin) while ((getchar()) != '\n')

void persona\_inserisci(){

char CF\_P[17],indirizzo[50],nome[30];

printf ("\nCodice fiscale della persona: ");

scanf ("%s",CF\_P);

fflush(stdin);

printf ("\nIndirizzo: ");

scanf ("%[^\n]",indirizzo);

fflush(stdin);

printf ("\nNome: ");

scanf ("%[^\n]",nome);

fflush(stdin);

snprintf(query, 1000, "call persona\_inserisci('%s','%s','%s')", CF\_P, indirizzo, nome);

run\_sql\_query(query);

printf("> Vuoi inserire i contatti (mail, cellulare, telefono)? s/n: ");

char choice;

scanf ("%c",&choice);

fflush(stdin);

if (choice == 's') {

char mail[30],cellulare[20],telefono[20];

printf ("\nMail: ");

scanf ("%[^\n]",mail);

fflush(stdin);

printf ("\nCellulare: ");

scanf ("%[^\n]",cellulare);

fflush(stdin);

printf ("\nTelefono: ");

scanf ("%[^\n]",telefono);

fflush(stdin);

snprintf(query, 1000, "call persona\_inserisci\_contatti('%s','%s','%s','%s')",

CF\_P, mail, cellulare, telefono);

run\_sql\_query(query);

}

}

void persona\_modifica(){

char CF\_P[17],indirizzo[50],nome[30];

printf ("\nCodice fiscale della persona da modificare: ");

scanf ("%s",CF\_P);

fflush(stdin);

printf ("\nNuovo indirizzo: ");

scanf ("%[^\n]",indirizzo);

fflush(stdin);

printf ("\nNuovo nome: ");

scanf ("%[^\n]",nome);

fflush(stdin);

snprintf(query, 1000, "call persona\_modifica('%s','%s','%s')", CF\_P, indirizzo, nome);

run\_sql\_query(query);

printf("> Vuoi modificare i contatti (mail, cellulare, telefono)? s/n: ");

char choice;

scanf ("%c",&choice);

fflush(stdin);

if (choice == 's') {

char mail[30],cellulare[20],telefono[20];

printf ("\nMail: ");

scanf ("%[^\n]",mail);

fflush(stdin);

printf ("\nCellulare: ");

scanf ("%[^\n]",cellulare);

fflush(stdin);

printf ("\nTelefono: ");

scanf ("%[^\n]",telefono);

fflush(stdin);

snprintf(query, 1000, "call persona\_modifica\_contatti('%s','%s','%s','%s')",

CF\_P, mail, cellulare, telefono);

run\_sql\_query(query);

}

}

void persona\_elimina(){

//TODO ON DELETE CASCADE?

char CF\_P[17];

printf ("\nCodice fiscale della persona da eliminare: ");

scanf ("%s",CF\_P);

fflush(stdin);

snprintf(query, 1000, "call persona\_elimina('%s')", CF\_P);

run\_sql\_query(query);

}

void persona\_effettua\_ingresso\_libero(){

char CF\_P[17],data\_ingresso[11],nome\_piscina[50];

printf ("\nCodice fiscale della persona: ");

scanf ("%s",CF\_P);

fflush(stdin);

printf ("\nData ingresso : ");

scanf ("%[^\n]",data\_ingresso);

fflush(stdin);

snprintf(query, 1000, "call persona\_effettua\_ingresso\_libero('%s','%s','%s')",

CF\_P, data\_ingresso, NOME\_PISCINA\_ADDETTO);

run\_sql\_query(query);

}

void persona\_inserisci\_iscritta\_corsi(){

char CF\_P[17],medico[30],data\_certificato[11],data\_nascita[11];

printf ("\nCodice fiscale della persona: ");

scanf ("%s",CF\_P);

fflush(stdin);

printf ("\nMedico che ha redatto il certificato: ");

scanf ("%[^\n]",medico);

fflush(stdin);

printf ("\nData certificato medico: ");

scanf ("%[^\n]",data\_certificato);

fflush(stdin);

printf ("\nData di nascita: ");

scanf ("%[^\n]",data\_nascita);

fflush(stdin);

snprintf(query, 1000, "call persona\_inserisci\_iscritta\_corsi('%s','%s','%s','%s')",

CF\_P, medico, data\_certificato, data\_nascita);

run\_sql\_query(query);

}

void persona\_modifica\_iscritta\_corsi(){

char CF\_P[17],medico[30],data\_certificato[11],data\_nascita[11];

printf ("\nCodice fiscale della persona iscritta ai corsi da modificare: ");

scanf ("%s",CF\_P);

fflush(stdin);

printf ("\nNuovo medico che ha redatto il certificato : ");

scanf ("%[^\n]",medico);

fflush(stdin);

printf ("\nNuova data certificato medico: ");

scanf ("%[^\n]",data\_certificato);

fflush(stdin);

printf ("\nNuova data di nascita: ");

scanf ("%[^\n]",data\_nascita);

fflush(stdin);

snprintf(query, 1000, "call persona\_modifica\_iscritta\_corsi('%s','%s','%s','%s')",

CF\_P, medico, data\_certificato, data\_nascita);

run\_sql\_query(query);

}

void persona\_elimina\_iscritta\_corsi(){

//TODO ON DELETE CASCADE?

char CF\_P[17];

printf ("\nCodice fiscale della persona iscritta ai corsi da eliminare: ");

scanf ("%s",CF\_P);

fflush(stdin);

snprintf(query, 1000, "call persona\_elimina('%s')", CF\_P);

run\_sql\_query(query);

}

void persona\_effettua\_iscrizione\_corso(){

char CF\_P[17],corso[30];

printf ("\nCodice fiscale della persona: ");

scanf ("%s",CF\_P);

fflush(stdin);

printf ("\nCorso esistente presso %s: ",NOME\_PISCINA\_ADDETTO);

scanf ("%[^\n]",corso);

fflush(stdin);

snprintf(query, 1000, "call persona\_effettua\_iscrizione\_corso('%s','%s','%s')",

CF\_P, corso, NOME\_PISCINA\_ADDETTO);

run\_sql\_query(query);

}

void persona\_cancella\_iscrizione\_corso(){

char CF\_P[17],corso[30];

printf ("\nCodice fiscale della persona cui si vuole cancellare l'iscrizione: ");

scanf ("%s",CF\_P);

fflush(stdin);

printf ("\nCorso esistente per la quale si vuole cancellare l'iscrizione presso %s per la persona %s : ",NOME\_PISCINA\_ADDETTO, CF\_P);

scanf ("%[^\n]",corso);

fflush(stdin);

snprintf(query, 1000, "call persona\_cancella\_iscrizione\_corso('%s','%s','%s')", CF\_P, corso, NOME\_PISCINA\_ADDETTO);

run\_sql\_query(query);

}

void corso\_visualizza\_lista\_iscritti(){

char corso[30];

printf ("\nCorso esistente presso %s : ",NOME\_PISCINA\_ADDETTO);

scanf ("%[^\n]",corso);

fflush(stdin);

snprintf(query, 1000, "call corso\_visualizza\_lista\_iscritti('%s','%s')", corso, NOME\_PISCINA\_ADDETTO);

print\_sql\_query(query);

}

void corso\_inserisci(){

char Nome\_Corso[30];

float costo;

int massimo,minimo;

printf ("\nNome del corso: ");

scanf ("%[^\n]",Nome\_Corso);

fflush(stdin);

printf ("\nCosto del corso: ");

scanf ("%f",&costo);

fflush(stdin);

printf ("\nNumero massimo di partecipanti: ");

scanf ("%i",&massimo);

fflush(stdin);

printf ("\nNumero minimo di partecipanti: ");

scanf ("%i",&minimo);

fflush(stdin);

snprintf(query, 1000, "call corso\_inserisci('%s','%s','%f','%i','%i')",

Nome\_Corso, NOME\_PISCINA\_ADDETTO, costo, massimo, minimo);

run\_sql\_query(query);

}

void corso\_modifica(){

char Nome\_Corso[30];

float costo;

int massimo,minimo;

printf ("\nNome del corso da modificare presso %s: ",NOME\_PISCINA\_ADDETTO);

scanf ("%[^\n]",Nome\_Corso);

fflush(stdin);

printf ("\nNuovo costo del corso: ");

scanf ("%f",&costo);

fflush(stdin);

printf ("\nNuovo numero massimo di partecipanti: ");

scanf ("%i",&massimo);

fflush(stdin);

printf ("\nNuovo numero minimo di partecipanti: ");

scanf ("%i",&minimo);

fflush(stdin);

snprintf(query, 1000, "call corso\_modifica('%s',%s','%f','%i','%i')",

Nome\_Corso, NOME\_PISCINA\_ADDETTO, costo, massimo, minimo);

run\_sql\_query(query);

}

void corso\_elimina(){

char Nome\_Corso[30];

printf ("\nNome del corso da eliminare presso %s: ",NOME\_PISCINA\_ADDETTO);

scanf ("%[^\n]",Nome\_Corso);

fflush(stdin);

snprintf(query, 1000, "call corso\_elimina('%s',%s')",

Nome\_Corso, NOME\_PISCINA\_ADDETTO );

run\_sql\_query(query);

}

void corso\_visualizza\_lista\_per\_piscina(){

snprintf(query, 1000, "call corso\_visualizza\_lista\_per\_piscina('%s')",

NOME\_PISCINA\_ADDETTO );

print\_sql\_query(query);

}

void corso\_inserisci\_lezione(){

char Nome\_Corso[30],dalle[9],alle[9];

int giorno;

printf ("\nNome del corso esistente presso %s: ",NOME\_PISCINA\_ADDETTO);

scanf ("%[^\n]",Nome\_Corso);

fflush(stdin);

printf ("\nGiorno della settimana [1-7] (1:Domenica, 2:Lunedi, 3:Martedi, 4:Mercoledi, 5:Giovedi, 6:Venerdi, 7:Sabato) ");

scanf ("%i",&giorno);

fflush(stdin);

printf ("\nOrario di inizio [HH:MM] : ");

scanf ("%[^\n]",dalle);

fflush(stdin);

printf ("\nOrario di fine [HH:MM] : ");

scanf ("%[^\n]",alle);

fflush(stdin);

snprintf(query, 1000, "call corso\_inserisci\_lezione('%i','%s','%s','%s','%s')",

giorno, dalle, alle, Nome\_Corso, NOME\_PISCINA\_ADDETTO);

run\_sql\_query(query);

}

void corso\_modifica\_lezione(){

char Nome\_Corso[30],old\_dalle[9],old\_alle[9],new\_dalle[9],new\_alle[9];

int giorno;

printf ("\nNome del corso presso %s da modificare: ",NOME\_PISCINA\_ADDETTO);

scanf ("%[^\n]",Nome\_Corso);

fflush(stdin);

printf ("\nGiorno della settimana da modificare [1-7] (1:Domenica, 2:Lunedi, 3:Martedi, 4:Mercoledi, 5:Giovedi, 6:Venerdi, 7:Sabato) ");

scanf ("%i",&giorno);

fflush(stdin);

printf ("\nOrario di inizio da modificare[HH:MM] : ");

scanf ("%[^\n]",old\_dalle);

fflush(stdin);

printf ("\nOrario di fine da modificare[HH:MM] : ");

scanf ("%[^\n]",old\_alle);

fflush(stdin);

printf ("\nNuovo orario di inizio[HH:MM] : ");

scanf ("%[^\n]",new\_dalle);

fflush(stdin);

printf ("\nNuovo orario di fine[HH:MM] : ");

scanf ("%[^\n]",new\_alle);

fflush(stdin);

snprintf(query, 1000, "call corso\_modifica\_lezione('%i','%s','%s','%s','%s','%s','%s')",

giorno, old\_dalle, old\_alle, Nome\_Corso, NOME\_PISCINA\_ADDETTO, new\_dalle, new\_alle);

run\_sql\_query(query);

}

void corso\_elimina\_lezione(){

char Nome\_Corso[30],dalle[9],alle[9];

int giorno;

printf ("\nNome del corso esistente da eliminare presso %s: ",NOME\_PISCINA\_ADDETTO);

scanf ("%[^\n]",Nome\_Corso);

fflush(stdin);

printf ("\nGiorno della settimana del corso da eliminare [1-7] (1:Domenica, 2:Lunedi, 3:Martedi, 4:Mercoledi, 5:Giovedi, 6:Venerdi, 7:Sabato) ");

scanf ("%i",&giorno);

fflush(stdin);

printf ("\nOrario di inizio del corso da eliminare [HH:MM] : ");

scanf ("%[^\n]",dalle);

fflush(stdin);

printf ("\nOrario di fine del corso da eliminare [HH:MM] : ");

scanf ("%[^\n]",alle);

fflush(stdin);

snprintf(query, 1000, "call corso\_elimina\_lezione('%i','%s','%s','%s','%s')",

giorno, dalle, alle, Nome\_Corso, NOME\_PISCINA\_ADDETTO);

run\_sql\_query(query);

}

void corso\_visualizza\_lista\_lezioni(){

char Nome\_Corso[30];

printf ("\nNome del corso esistente presso %s: ",NOME\_PISCINA\_ADDETTO);

scanf ("%[^\n]",Nome\_Corso);

fflush(stdin);

snprintf(query, 1000, "call corso\_visualizza\_lista\_lezioni('%s','%s')",

Nome\_Corso,NOME\_PISCINA\_ADDETTO );

print\_sql\_query(query);

}

int is\_piscina\_addetto\_valid(char\* piscina){

MYSQL\_RES\* res;

snprintf(query, 1000, "CALL is\_piscina\_addetto\_valid('%s');",piscina);

res = get\_result\_from\_sql\_query(query);

row = mysql\_fetch\_row(res);

// verifico dal return di della procedure EXIST

// return 1: esiste la piscina

printf("RETURN: %s\n",row[0]);

if (strcmp(row[0], "1") == 0){ // se esiste la piscina...

return 1;

}

return 0;

}

------------------------------------------------------------------------------------------------------------------------------------

Makefile

------------------------------------------------------------------------------------------------------------------------------------

all:

gcc -g \*.c -o client `mysql\_config --cflags --libs`

clean:

-rm client

1. Indicare con E le entità, con R le relazioni [↑](#footnote-ref-2)
2. PK = primary key, NN = not null, UQ = unique, UN = unsigned, AI = auto increment. È ovviamente possibile specificare più di un attributo per ciascuna colonna. [↑](#footnote-ref-3)
3. IDX = index, UQ = unique, FT = full text, PR = primary. [↑](#footnote-ref-4)