

Interrogazioni in linguaggio SQL Progettazione applicazione in linguaggio Python

Caso di studio scelto: Gestione centro sportivo polifunzionale.

In allegato alla presente:

- Files .sql delle query implementate, contenuti nel file:
QUERY- FRIDA SCHIAVONI bdd ET4.zip
- File *centro_sportivo_SQLAlchemy_esame_frida_schiavoni.py* (implementazione query)
link su GitHub: https://github.com/fschiavoni70/ETIVITY4_BDD_2022

Per la gestione della base dati e delle connessioni viene utilizzato l'IDE MySQL Workbench 8.0.29

La piattaforma di sviluppo utilizzata per la programmazione è l'IDE PyCharm Python v. 3.10

Il driver di connessione con il DB è Mysql_connector-python v. 8.0.30.

Per l'installazione dei pacchetti si è utilizzato Pip v. 22.2.2.

La libreria ORM utilizzata è SQLAlchemy v 1.4.40

Per il dialogo del codice Python con il Database si è adottato il paradigma **ORM**, Object Relational Mapper, in particolare la libreria di astrazione del database *SQLAlchemy*, estensione python che rappresenta la proposta ORM di Python, consentendo la mappatura tra gli oggetti Python e le loro identità di database, traducendo le classi Python in tabelle su database relazionali, con la conversione delle di funzione Python (es metodi select()) in istruzioni SQL
La mappa identità è una raccolta associata all'oggetto Session ORM

SQLAlchemy opera con l'implementazione DBAPI specifica per un database, sarebbe stato quindi necessario installare i driver DBAPI per il database con cui si vuole comunicare. Dato che i seguenti dialetti sono inclusi:

1. MySQL
2. Microsoft SQL Server
3. Oracle
4. PostgreSQL
5. SQLite
6. Firebird
7. Sybase

Nel nostro caso avendo un database implementato su MySQL, con la libreria SQLAlchemy non è stato necessario installare alcun altro driver, utilizzando *mysql-connector-python v.8.030*

Le principali funzionalità di SQLAlchemy includono un sistema di query orientato alle relazioni, che espone in modo esplicito l'intera gamma delle capacità di SQL, inclusi join, sottoquery.

Per implementare le query sotto descritte si è utilizzata la classe `select` del pacchetto `sqlalchemy.sql`

Prima di tutto occorre definire la connessione con il database attraverso la classe `create_engine` ed il suo costruttore `create_engine()` specificando

Il server *localhost*

il database *centro_sportivo*

l'utente precedentemente creato *frida* con la password *basedati_2022*

ed utilizzando poi il metodo `connect()` per ottenere un oggetto di connessione

come segue:

```
from sqlalchemy import create_engine

engine = create_engine('mysql+mysqlconnector://frida:basedati_2022@localhost/centro_sportivo')

conn = engine.connect()
```

Si istanzia un oggetto `MetaData` ovvero un CATALOGO di oggetti `Table` e loro costrutti che contiene definizioni di tabelle (`Table`) e oggetti associati come indici, viste, trigger, ecc

```
from sqlalchemy import MetaData

meta = MetaData()
```

Occorre importare gli oggetti per la definizione delle tabelle e degli indici all'interno del catalogo di metadati. I tipi generici ammessi in `SQLAlchemy` sono: `BigInteger` `SmallInteger` `Integer` `Numeric` `Float` `Boolean` `String` `Text` `Date` `DateTime` `Time`

```
from sqlalchemy import Table, Column, Integer, String, Date, Time, ForeignKey, Index, PrimaryKeyConstraint, ForeignKeyConstraint
```

Si mappano gli oggetti su identità DB:

```
Tabella = Table('nome', meta,
                Column('colonna1', String, primary_key=True )...
```

Al metodo `create_all()` dell'oggetto `meta` - classe `MetaData`, si passa l'oggetto `engine` per creare le tabelle e memorizzare tutte le informazioni in `metadata`

```
meta.create_all(engine)
```

Per realizzare le query al database si utilizza la Classe `select`

```
from sqlalchemy.sql import select
```

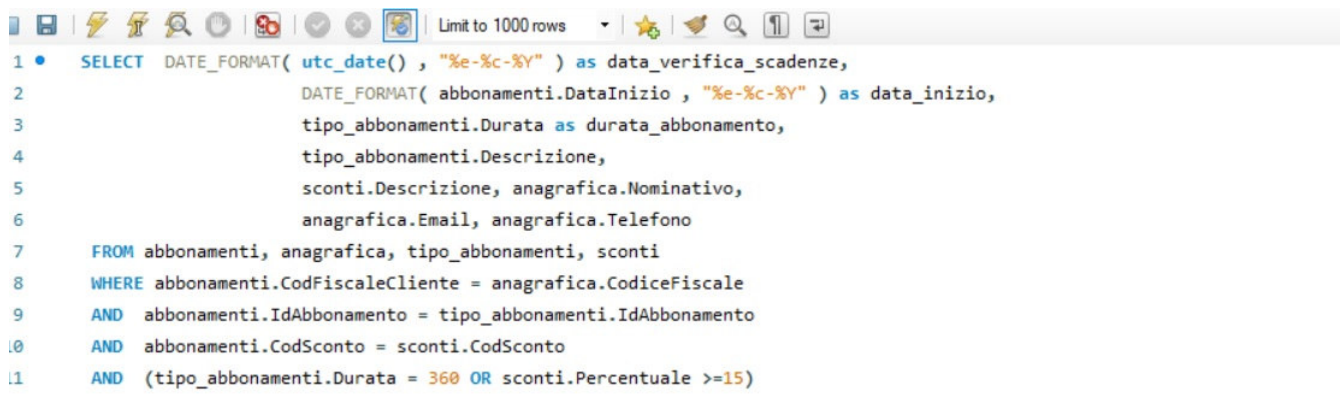
Sono istanziati oggetti di tipo `select` con il suo costruttore `select()` ed i suoi metodi per definire le query volute.

Tali oggetti sono passati al metodo `execute()` dell'oggetto connessione `conn` e si traducono le chiamate a funzioni in istruzioni SQL i cui risultati sono memorizzati nella stringa `result`

```
query = select( ).where( ).order_by( )
result = conn.execute(query)
```

Interrogazioni SQL sul database centro_sportivo implementate in codice python

1. Selezionare gli estremi degli abbonamenti stipulati che abbiano durata annuale o sconto superiore al 15%

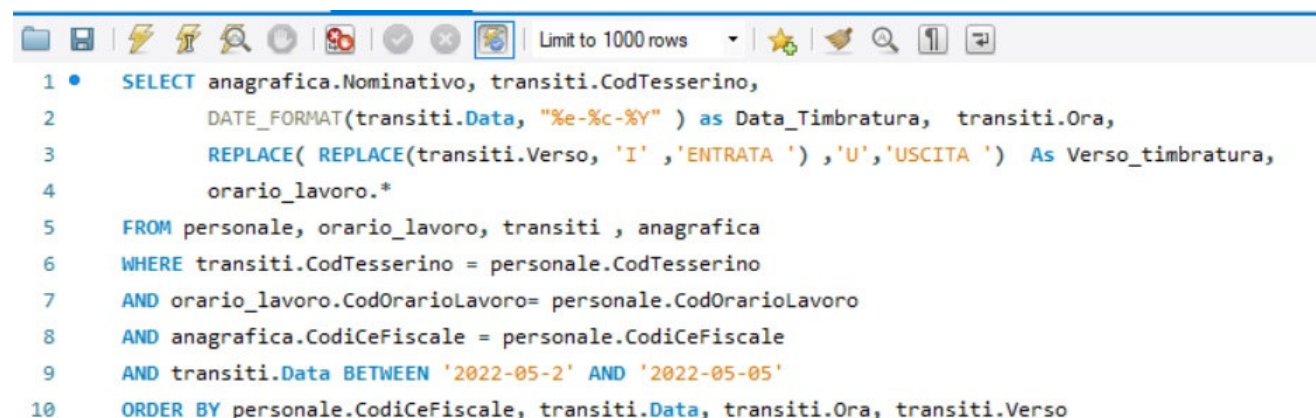


```
1 • SELECT DATE_FORMAT( utc_date() , "%e-%c-%Y" ) as data_verifica_scadenze,
2         DATE_FORMAT( abbonamenti.DataInizio , "%e-%c-%Y" ) as data_inizio,
3         tipo_abbonamenti.Durata as durata_abbonamento,
4         tipo_abbonamenti.Descrizione,
5         sconti.Descrizione, anagrafica.Nominativo,
6         anagrafica.Email, anagrafica.Telefono
7 FROM abbonamenti, anagrafica, tipo_abbonamenti, sconti
8 WHERE abbonamenti.CodFiscaleCliente = anagrafica.CodiceFiscale
9 AND abbonamenti.IdAbbonamento = tipo_abbonamenti.IdAbbonamento
10 AND abbonamenti.CodSconto = sconti.CodSconto
11 AND (tipo_abbonamenti.Durata = 360 OR sconti.Percentuale >=15)
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

data_verifica_scade	data_inizio	durata_abbonam	Descrizione	Descrizione	Nominativo	Email	Telefono
2-9-2022	3-11-2021	360	Annuale OMNICOMPRESIVO	sconto fidelity	MATTIA FRIGERIO	mattia.frigerio@virgilio.it	3365874125
2-9-2022	2-1-2022	360	1 anno acqua	sconto benvenuto	FRIDA SCHIAVONI	frida.schiavoni@vigilfuoco.it	3371210233

2. Visualizzare elenco timbrature dei dipendenti dal 2 maggio 2022 al 5 maggio 2022, ordinato per nominativo, data, ora e verso.



```
1 • SELECT anagrafica.Nominativo, transiti.CodTesserino,
2         DATE_FORMAT(transiti.Data, "%e-%c-%Y" ) as Data_Timbratura, transiti.Ora,
3         REPLACE( REPLACE(transiti.Verso, 'I', 'ENTRATA ' ) , 'U', 'USCITA ' ) As Verso_timbratura,
4         orario_lavoro.*
5 FROM personale, orario_lavoro, transiti , anagrafica
6 WHERE transiti.CodTesserino = personale.CodTesserino
7 AND orario_lavoro.CodOrarioLavoro= personale.CodOrarioLavoro
8 AND anagrafica.CodiCeFiscale = personale.CodiCeFiscale
9 AND transiti.Data BETWEEN '2022-05-2' AND '2022-05-05'
10 ORDER BY personale.CodiCeFiscale, transiti.Data, transiti.Ora, transiti.Verso
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Nominativo	CodTesserino	Data_Timbratura	Ora	Verso_timbratura	CodOrarioLavoro	OraInizio	OraFine
MATTIA FRIGERIO	CL00000004	2-5-2022	06:00:01	ENTRATA	TURN01	06:00:00	11:00:00
MATTIA FRIGERIO	CL00000004	2-5-2022	11:02:01	USCITA	TURN01	06:00:00	11:00:00
MATTIA FRIGERIO	CL00000004	3-5-2022	06:00:01	ENTRATA	TURN01	06:00:00	11:00:00
MATTIA FRIGERIO	CL00000004	3-5-2022	11:01:44	USCITA	TURN01	06:00:00	11:00:00
MARCO LOPEZ	CL00000005	2-5-2022	11:00:01	ENTRATA	TURN02	11:00:00	17:00:00
MARCO LOPEZ	CL00000005	2-5-2022	17:00:01	USCITA	TURN02	11:00:00	17:00:00
MARCO LOPEZ	CL00000005	4-5-2022	11:00:55	ENTRATA	TURN02	11:00:00	17:00:00
MARCO LOPEZ	CL00000005	4-5-2022	11:03:01	ENTRATA	TURN02	11:00:00	17:00:00
MARCO LOPEZ	CL00000005	4-5-2022	17:30:01	USCITA	TURN02	11:00:00	17:00:00
RICCARDO TRAMACCI	CL00000007	2-5-2022	08:01:00	ENTRATA	GIORNOBASE	08:00:00	17:00:00
RICCARDO TRAMACCI	CL00000007	2-5-2022	17:01:00	USCITA	GIORNOBASE	08:00:00	17:00:00
RICCARDO TRAMACCI	CL00000007	3-5-2022	08:03:00	ENTRATA	GIORNOBASE	08:00:00	17:00:00
RICCARDO TRAMACCI	CL00000007	3-5-2022	17:11:00	USCITA	GIORNOBASE	08:00:00	17:00:00

3. Selezionare, per ciascun dipendente, il massimo ritardo di ingresso e la data in cui si è verificato, e, per ciascun dipendente, la data e l'orario di uscita minimi, sempre in relazione al proprio orario di lavoro

```

1 • SELECT REPLACE(transiti.Verso, 'I', 'ENTRATA') As Verso_timbratura, anagrafica.Nominativo, orario_lavoro.CodOrarioLavoro,
2         orario_lavoro.OraInizio, DATE_FORMAT(transiti.Data, "%e-%c-%Y") as Data_Timbratura, MAX(transiti.Ora) AS CONTROLLO
3 FROM personale, orario_lavoro, transiti, anagrafica
4 WHERE transiti.CodTesserino = personale.CodTesserino
5 AND anagrafica.CodiCeFiscale = personale.CodiCeFiscale
6 AND orario_lavoro.CodOrarioLavoro= personale.CodOrarioLavoro
7 AND transiti.Verso = 'I'
8 GROUP BY anagrafica.Nominativo
9 UNION
10 SELECT REPLACE(transiti.Verso, 'U', 'USCITA') As Verso_timbratura, anagrafica.Nominativo, orario_lavoro.CodOrarioLavoro,
11         orario_lavoro.OraFine, DATE_FORMAT(transiti.Data, "%e-%c-%Y") as Data_Timbratura, MIN(transiti.Ora) AS CONTROLLO
12 FROM personale, orario_lavoro, transiti, anagrafica
13 WHERE transiti.CodTesserino = personale.CodTesserino
14 AND anagrafica.CodiCeFiscale = personale.CodiCeFiscale
15 AND orario_lavoro.CodOrarioLavoro= personale.CodOrarioLavoro
16 AND transiti.Verso = 'U'
17 GROUP BY anagrafica.Nominativo

```

Verso_timbratura	Nominativo	CodOrarioLavoro	OraInizio	Data_Timbratura	CONTROLLO
ENTRATA	MATTIA FRIGERIO	TURN01	06:00:00	2-5-2022	06:00:01
ENTRATA	MARCO LOPEZ	TURN02	11:00:00	2-5-2022	11:03:43
ENTRATA	RICCARDO TRAMACCI	GIORNOBASE	08:00:00	2-5-2022	08:03:00
ENTRATA	MANOLO PRATI	TURN02	11:00:00	7-5-2022	11:06:01
ENTRATA	MARTA VAROTTO	TURN02	11:00:00	7-5-2022	11:01:01
USCITA	MATTIA FRIGERIO	TURN01	11:00:00	2-5-2022	11:00:01
USCITA	MARCO LOPEZ	TURN02	17:00:00	2-5-2022	17:00:01
USCITA	RICCARDO TRAMACCI	GIORNOBASE	17:00:00	2-5-2022	17:01:00
USCITA	MARTA VAROTTO	TURN02	17:00:00	7-5-2022	17:02:20
USCITA	MANOLO PRATI	TURN02	17:00:00	8-5-2022	17:00:25

4. Calcolo massimali ore lavoro:

Selezionare il codice e la descrizione della tipologia di manutenzione su sale ed impianti raggruppate per numero di lavoro a partire dal giorno 19 aprile 2022, poi indicare anche il massimo monte ore raggiunto.

```

1 • SELECT max(massimali.ore_lavorate) as TotaleMaxOreLavorate FROM
2         (SELECT SUM(manutenzione_impianti.Durata) AS ore_lavorate,
3                 manutenzione_impianti.CodIntervento AS codice,
4                 tipo_manutenzione.Descrizione as lavoro_eseguito
5             FROM manutenzione_impianti, tipo_manutenzione
6             WHERE tipo_manutenzione.CodIntervento = manutenzione_impianti.CodIntervento
7             AND manutenzione_impianti.Data > "2020:04:18"
8             GROUP BY manutenzione_impianti.CodIntervento ORDER BY ore_lavorate ) AS massimali
9
10

```

ore_lavorate	codice	lavoro_eseguito
45	SICU1	Sistemi sicurezza
110	MANU3	Lavori edili
175	MANU4	Spostamenti-traslochi
415	PULI4	Pulizia piscine
425	MANU2	Lavori manut. non ordinaria
475	ELECT1	Impianti elettrici
478	OZONO	Ozonizzazione
530	MANU1	Lavori manutenz. ordinaria
545	VASCA	Manutenz. vasche acqua
992	PULI3	Pulizia Ambienti
2100	PULI1	Pulizia sale

Limit to 1000 rows

```

1 • SELECT max(massimali.ore_lavorate) as TotaleMaxOreLavorate FROM
2   (SELECT SUM(manutenzione_impianti.Durata) AS ore_lavorate,
3        manutenzione_impianti.CodIntervento AS codice,
4        tipo_manutenzione.Descrizione as lavoro_eseguito
5   FROM manutenzione_impianti, tipo_manutenzione
6   WHERE tipo_manutenzione.CodIntervento = manutenzione_impianti.CodIntervento
7   AND manutenzione_impianti.Data > "2020:04:18"
8   GROUP BY manutenzione_impianti.CodIntervento ORDER BY ore_lavorate ) AS massimali
9
10

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: IA

TotaleMaxOreLavorate
2100

5. Controllare la data di ultima manutenzione delle vasche d'acqua, indicando l'ora e il nominativo del manutentore.

```

SELECT anagrafica.Nominativo, sale_impianti.Descrizione AS Impianto ,
tipo_manutenzione.Descrizione, manutenzione_impianti.OraInizio,
DATE_FORMAT(MAX(manutenzione_impianti.Data), "%e-%c-%Y" ) AS Ultima_data_manutenzione
FROM manutenzione_impianti, sale_impianti, tipo_manutenzione, anagrafica
WHERE manutenzione_impianti.CodIntervento = 'VASCA'
AND manutenzione_impianti.CodIntervento=tipo_manutenzione.CodIntervento
AND manutenzione_impianti.CFManutentore = anagrafica.CodiCeFiscale
AND manutenzione_impianti.IdSalaImpianto = sale_impianti.IdSalaImpianto
GROUP BY sale_impianti.IdSalaImpianto

```

irid Filter Rows: Export: Wrap Cell Content: IA				
Nominativo	Impianto	Descrizione	OraInizio	Ultima_data_manutenzione
CALLEGARI MARCO	vasca 25mt	Manutenz. vasche acqua	15:30:00	29-4-2020
CALLEGARI MARCO	vasca 50mt	Manutenz. vasche acqua	10:17:00	28-4-2020
MIRKO FIORE	piscina idro	Manutenz. vasche acqua	14:00:00	27-4-2020

6. Controllo delle prenotazioni attive e non disdette dal giorno 16 aprile 2022 al giorno 22 aprile 2022.

Limit to 1000 rows

```

1 • SELECT prenotazioni_corsi.Data AS DATA_PRENOTAZIONE, scaglionori.Descrizione AS ORARIO_PRENOTAZIONE,
2     tipologia_corsi.Descrizione AS CORSO_PRENOTATO, prenotazioni_corsi.CodFiscaleCliente,
3     anagrafica.Nominativo AS CLIENTE, orario_corsi_settimanali.CodCorso
4 FROM prenotazioni_corsi, anagrafica, orario_corsi_settimanali, tipologia_corsi, scaglionori
5 WHERE prenotazioni_corsi.CodFiscaleCliente = anagrafica.CodiceFiscale
6 AND prenotazioni_corsi.IdOrario = orario_corsi_settimanali.IdOrario
7 AND orario_corsi_settimanali.CodCorso = tipologia_corsi.IdCorso
8 AND orario_corsi_settimanali.CodScaglioneOrario = scaglionori.IdScaglioneOrario
9 AND prenotazioni_corsi.Data BETWEEN "2022-04-15" AND "2022-04-24"
10 AND prenotazioni_corsi.Stato = 'C'
11 ORDER BY prenotazioni_corsi.Data ASC, scaglionori.Descrizione DESC, anagrafica.Nominativo

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

DATA_PRENOTAZIONE	ORARIO_PRENOTAZIONE	CORSO_PRENOTATO	CodFiscaleCliente	CLIENTE	CodCorso
2022-04-15	h 12.30-13.30	Spinn acqua	DLGNTN65545V998K	ANTONELLO DALGANO	IDROBIKE
2022-04-16	h 20.30-21.30	Spinning Advanced	DLGNTN65545V998K	ANTONELLO DALGANO	SPINN2
2022-04-16	h 20.30-21.30	Spinning Advanced	SCHFRD70S52C770J	FRIDA SCHIAVONI	SPINN2
2022-04-16	h 20.30-21.30	Spinning Advanced	FRGMTT89R54V679L	MATTIA FRIGERIO	SPINN2
2022-04-16	h 16.30-17.30	Walking pedana Basic	FRGMTT89R54V679L	MATTIA FRIGERIO	WALK1
2022-04-16	h 06.30-07.30	CrossGym Basic	SCHFRD70S52C770J	FRIDA SCHIAVONI	CROSS1
2022-04-17	h 18.30-19.30	Interval Training	SCHFRD70S52C770J	FRIDA SCHIAVONI	HIIT
2022-04-17	h 18.30-19.30	Interval Training	FRGMTT89R54V679L	MATTIA FRIGERIO	HIIT
2022-04-17	h 14.30-15.30	Corso Nuoto base	SCHFRD70S52C770J	FRIDA SCHIAVONI	NUOTO1
2022-04-18	h 06.30-07.30	CrossGym Basic	SCHFRD70S52C770J	FRIDA SCHIAVONI	CROSS1
2022-04-19	h 06.30-07.30	CrossGym Basic	SCHFRD70S52C770J	FRIDA SCHIAVONI	CROSS1
2022-04-20	h 16.30-17.30	Walking pedana Basic	FRGMTT89R54V679L	MATTIA FRIGERIO	WALK1
2022-04-21	h 14.30-15.30	Gymn posturale	DLGNTN65545V998K	ANTONELLO DALGANO	POSTURALE
2022-04-22	h 16.30-17.30	Walking pedana Basic	DLGNTN65545V998K	ANTONELLO DALGANO	WALK1
2022-04-22	h 16.30-17.30	Walking pedana Basic	SCHFRD70S52C770J	FRIDA SCHIAVONI	WALK1
2022-04-23	h 16.30-17.30	Walking pedana Basic	DLGNTN65545V998K	ANTONELLO DALGANO	WALK1

7. Calcolare il numero di prenotazioni attive, distinte per tipologia di corso, sono state effettuate nella fascia oraria mattutina.

Note: per conoscere la fascia oraria si è ricorsi ad una query innestata. Nel codice python si è utilizzato il join con la tabella scaglionori.

Limit to 1000 rows

```

1 • SELECT prenotazioni_corsi.IdOrario, count(prenotazioni_corsi.CodFiscaleCliente) AS NUMERO_PRENOTAZIONI,
2     orario_corsi_settimanali.GiornoSettimana, orario_corsi_settimanali.CodScaglioneOrario AS SCAGLIONE,
3     tipologia_corsi.Descrizione
4 FROM prenotazioni_corsi, orario_corsi_settimanali, tipologia_corsi
5 WHERE prenotazioni_corsi.IdOrario = orario_corsi_settimanali.IdOrario
6 AND orario_corsi_settimanali.CodCorso = tipologia_corsi.IdCorso
7 AND prenotazioni_corsi.Stato = 'C'
8 GROUP BY (prenotazioni_corsi.IdOrario)
9 HAVING orario_corsi_settimanali.CodScaglioneOrario IN
10 ( SELECT scaglionori.IdScaglioneOrario FROM scaglionori WHERE scaglionori.FasciaOrariaAppartenenza = "FASCIA1")
11
12

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

IdOrario	NUMERO_PRENOTAZIONI	GiornoSettimana	SCAGLIONE	Descrizione
ACQF4MLU	7	Lunedì	SCA04M	Fit acqua
CROS1MLU	8	Lunedì	SCA01M	CrossGym Basic
CROS7MME	1	Mercoledì	SCA07M	CrossGym Advanced
IDRO7MVE	1	Venerdì	SCA07M	Spinn acqua

8. Selezionare i nominativi dei clienti che prenotano corsi di lunedì ma che nello stesso giorno non hanno mai prenotato sale o impianti.

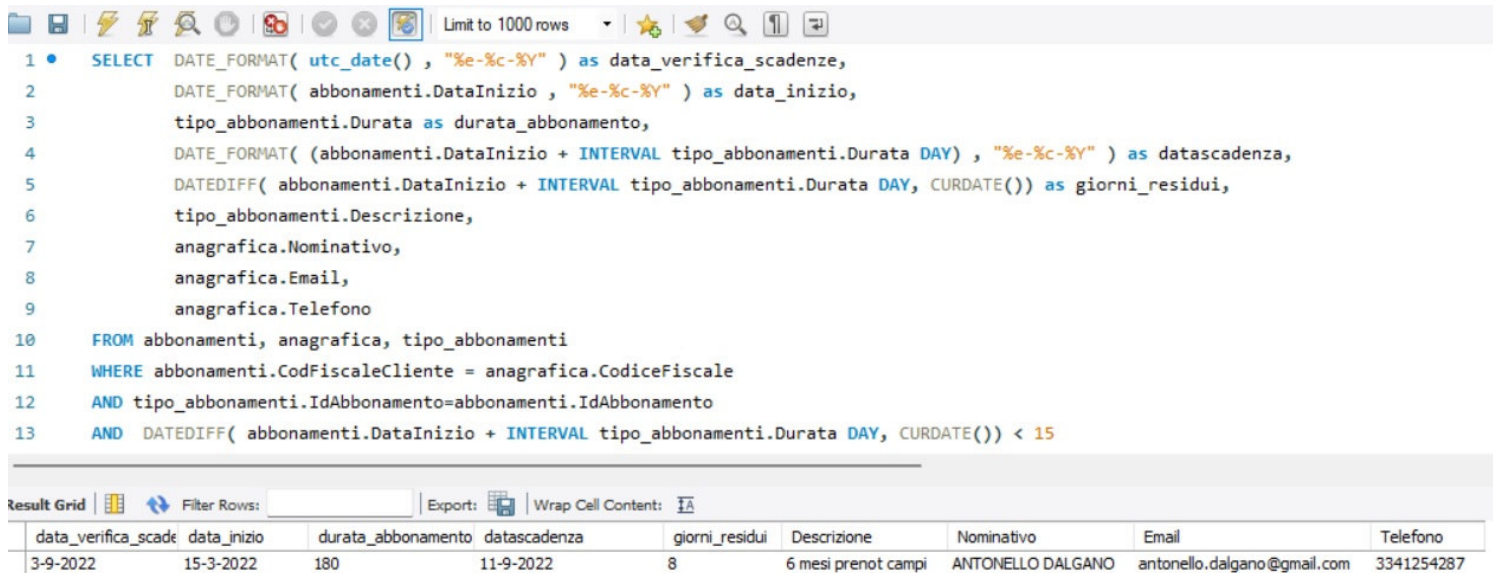
```
2 FROM prenotazioni_corsi, anagrafica, orario_corsi_settimanali
3 WHERE prenotazioni_corsi.CodFiscaleCliente = anagrafica.CodiCeFiscale
4 AND prenotazioni_corsi.IdOrario = orario_corsi_settimanali.IdOrario
5 AND orario_corsi_settimanali.GiornoSettimana = "Lunedì"
6 AND anagrafica.Nominativo <> ANY
7 (SELECT DISTINCT anagrafica.Nominativo
8 FROM prenotazioni_sale_impianti, anagrafica, orario_sale_impianti
9 WHERE prenotazioni_sale_impianti.CodFiscaleCliente= anagrafica.CodiCeFiscale
0 AND prenotazioni_sale_impianti.IdOrario = orario_sale_impianti.IdOrario
1 AND orario_sale_impianti.GiornoSettimana = "Lunedì")
2
```

ult Grid			Filter Rows:	<input type="text"/>	Export:		Wrap Cell Content:	
Nominativo								
ANTONELLO DALGANO								
MATTIA FRIGERIO								

ALTRE QUERY DI APPROFONDIMENTO SU DATABASE centro_sportivo

9. La segreteria necessita di mandare notifiche di scadenza al fine di invitare al rinnovo dell'abbonamento. Selezionare i nominativi, email e numero telefonico dei clienti del centro sportivo che hanno abbonamenti in scadenza (termine abbonamento a meno di 15 giorni)

Nota: la data scadenza non è un dato noto ma è da calcolare in base alla data di stipula dell'abbonamento, tenendo conto della durata dello stesso e si utilizza la funzione DATEDIFF per confrontare la data di scadenza con quella di controllo



```
1 • SELECT DATE_FORMAT( utc_date() , "%e-%c-%Y" ) as data_verifica_scadenze,
2         DATE_FORMAT( abbonamenti.DataInizio , "%e-%c-%Y" ) as data_inizio,
3         tipo_abbonamenti.Durata as durata_abbonamento,
4         DATE_FORMAT( (abbonamenti.DataInizio + INTERVAL tipo_abbonamenti.Durata DAY) , "%e-%c-%Y" ) as datascadenza,
5         DATEDIFF( abbonamenti.DataInizio + INTERVAL tipo_abbonamenti.Durata DAY, CURDATE() ) as giorni_residui,
6         tipo_abbonamenti.Descrizione,
7         anagrafica.Nominativo,
8         anagrafica.Email,
9         anagrafica.Telefono
10 FROM abbonamenti, anagrafica, tipo_abbonamenti
11 WHERE abbonamenti.CodFiscaleCliente = anagrafica.CodiceFiscale
12 AND tipo_abbonamenti.IdAbbonamento=abbonamenti.IdAbbonamento
13 AND DATEDIFF( abbonamenti.DataInizio + INTERVAL tipo_abbonamenti.Durata DAY, CURDATE() ) < 15
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

data_verifica_scade	data_inizio	durata_abbonamento	datascadenza	giorni_residui	Descrizione	Nominativo	Email	Telefono
3-9-2022	15-3-2022	180	11-9-2022	8	6 mesi prenot campi	ANTONELLO DALGANO	antonello.dalgano@gmail.com	3341254287

10. Scarto tollerato negli ingressi:

selezionare le timbrature dei dipendenti che in ingresso superano la soglia di tollerabilità dei 5 minuti e le timbrature dei dipendenti da cui risulta uno straordinario maggiore ai 30 minuti

```
• SELECT anagrafica.Nominativo, transiti.CodTesserino,  
    DATE_FORMAT(transiti.Data, "%e-%c-%Y" ) as Data_Timbratura, transiti.Ora,  
    REPLACE( REPLACE(transiti.Verso, 'I', 'ENTRATA ') , 'U', 'USCITA ') As Verso_timbratura,  
    orario_lavoro.* , TIMEDIFF(transiti.Ora, orario_lavoro.OraInizio) as scarto,  
    MAKETIME(00, 05,00) as Scarto_Tollerato  
    FROM personale, orario_lavoro, transiti , anagrafica  
WHERE transiti.CodTesserino = personale.CodTesserino  
AND transiti.Verso = 'I'  
AND orario_lavoro.CodOrarioLavoro= personale.CodOrarioLavoro  
AND anagrafica.CodiCeFiscale = personale.CodiCeFiscale  
AND TIMEDIFF(transiti.Ora, orario_lavoro.OraInizio) > MAKETIME(00 , 05, 00)  
UNION  
SELECT anagrafica.Nominativo, transiti.CodTesserino,  
    DATE_FORMAT(transiti.Data, "%e-%c-%Y" ) as Data_Timbratura, transiti.Ora,  
    REPLACE( REPLACE(transiti.Verso, 'I', 'ENTRATA ') , 'U', 'USCITA ') As Verso_timbratura,  
    orario_lavoro.* , TIMEDIFF(transiti.Ora, orario_lavoro.OraFine) as scarto,  
    MAKETIME(00, 30,00) as scarto_tollerato  
    FROM personale, orario_lavoro, transiti , anagrafica  
WHERE transiti.CodTesserino = personale.CodTesserino  
AND transiti.Verso = 'U'  
AND orario_lavoro.CodOrarioLavoro= personale.CodOrarioLavoro  
AND anagrafica.CodiCeFiscale = personale.CodiCeFiscale  
AND TIMEDIFF(transiti.Ora, orario_lavoro.OraFine) > MAKETIME(00, 30, 00)
```

Result Grid   Filter Rows: <input type="text"/> Export:  Wrap Cell Content: 										
	Nominativo	CodTesserino	Data_Timbratura	Ora	Verso_timbratura	CodOrarioLavoro	OraInizio	OraFine	scarto	Scarto_Tollerato
▶	MANOLO PRATI	CL00000006	7-5-2022	11:06:01	ENTRATA	TURN02	11:00:00	17:00:00	00:06:01	00:05:00
	MARCO LOPEZ	CL00000005	4-5-2022	17:30:01	USCITA	TURN02	11:00:00	17:00:00	00:30:01	00:30:00
	RICCARDO TRAMACCI	CL00000007	10-5-2022	17:41:00	USCITA	GIORNOBASE	08:00:00	17:00:00	00:41:00	00:30:00
	MARTA VAROTTO	CL00000008	13-5-2022	17:45:20	USCITA	TURN02	11:00:00	17:00:00	00:45:20	00:30:00