

# Università degli Studi di Padova



## **Relazione per la base di dati**

Active Pool

Per una piscina e centro benessere

### **Realizzata da:**

Gabriele Di Pietro, matricola 2010000

Alessandro Rossi, matricola 2045045

# ABSTRACT

Dall'azienda "Active Pool", una piscina di medie dimensioni, ci viene richiesto di sviluppare il loro data base per l'amministrazione di tutte le attività riguardanti la gestione degli iscritti, dei dipendenti e dei servizi proposti dal centro.

In particolare, ogni persona che vuole accedere alla piscina deve essere registrata, ma solo dopo aver effettuato il pagamento dell'iscrizione può usufruire dei servizi del centro (corsi sportivi e prestazioni biomediche). L'iscrizione si suddivide in varie tipologie di abbonamento (annuale, semestrale, trimestrale, mensile) oppure si possono acquistare pacchetti di singole entrate.

Il data base include anche la gestione dei dipendenti, ovvero prevede di tenere uno storico di tutte le persone che sono state assunte dall'azienda nel corso degli anni, specificando le loro generalità, la tipologia del loro contratto e lo stipendio previsto. Ogni dipendente attualmente stipendiato può ricoprire il ruolo di istruttore di un corso sportivo e quindi occuparsi dell'attività designata di competenza, oppure può essere un professionista (medico, nutrizionista, fisioterapista) e quindi svolgere attività ambulatoriale (rilasciare certificati medici, prescrizione di diete, terapie manuali) altrimenti è un lavoratore "generico" (addetto alle pulizie, impiegati). In particolare, gli impiegati, oltre ad avere un ruolo amministrativo, organizzano degli eventi al fine di pubblicizzare la piscina.

Per ogni evento organizzato da uno o più impiegati ci deve essere un responsabile dell'organizzazione (anch'esso un lavoratore della piscina), si deve conoscere il budget dell'evento e deve essere pubblicizzato da uno sponsor il quale finanzia l'evento in cambio di pubblicità.

## ANALISI DEI REQUISITI

### DESCRIZIONE TESTUALE

Nella base di dati sono presenti le generalità delle **persone** che sono state registrate per l'ingresso al centro:

- Codice fiscale
- Nome
- Cognome
- Età
- Data di nascita
- Sesso
- Telefono
- Residenza

Le persone che sono state registrate possono iscriversi alla palestra pagando una delle quote di iscrizione specificate nel **tariffario** della piscina:

- Prezzo
- Tipologia
- Descrizione

Una persona dopo aver pagato l'iscrizione diventa un **iscritto** di cui è noto:

- Tipologia abbonamento
- Data inizio
- Scadenza

L'azienda mantiene uno storico con tutti i **contratti** stipulati:

- Numero del contratto
- Cognome
- Nome
- Codice fiscale
- Data di nascita
- Et 
- Indirizzo
- Telefono
- E-mail
- Tipologia contratto
- Data assunzione
- Scadenza contratto
- Stipendio

I **dipendenti** sono i lavoratori attualmente assunti, i quali vengono distinti in istruttori, professionisti e lavoratori. In particolare, gli istruttori svolgono dei **corsi** sportivi, i professionisti svolgono delle **prestazioni**. I corsi e le prestazioni sono le attivit  che l'iscritto pu  svolgere all'interno del centro.

I lavoratori, invece, svolgono mansioni amministrative, tra cui l'organizzazione di **eventi** per pubblicizzare la piscina. Gli eventi sono finanziati da degli **sponsor**.

## GLOSSARIO DEI TERMINI

Termine	Descrizione	Collegamenti
Persone	Persone che sono state registrate nella piscina, ovvero sono le persone che sono state iscritte	Iscritto, tariffario
Iscritto	Persona che ha pagato l'iscrizione e il suo abbonamento � ancora attivo	Persone, tariffario, corsi, prestazioni
Tariffario	Elenco delle tariffe di iscrizione alla piscina	Iscritto, Persone
Tipologia	Tipo di abbonamento	Attributo dell'entit� tariffario
Corsi	Listino delle attivit� sportive tenuti dalla piscina	Iscritto, istruttori
Tipologia	Nome del corso	Attributo di dell'entit� corsi
Responsabile	Responsabile dell'attivit� che ne delinea la programmazione annuale	Attributo di dell'entit� corsi
Giorno	Giorno in cui si svolge un corso. Un corso si pu� svolgere in pi� giorni	Attributo di dell'entit� corsi
Orario	Fascia oraria in cui si svolge il corso. Ci possono essere pi� fasce orarie per un solo corso	Attributo di dell'entit� corsi
Edificio	Luogo in cui il corso viene svolto	Attributo di dell'entit� corsi
Prestazioni	Prestazioni svolte dai professionisti	Iscritto, professionisti
Contratti	Tutti i contratti stipulati dall'azienda	Dipendenti

Dipendenti	I lavoratori attualmente assunti dall'azienda	Generalizzazione
Istruttori	Dipendenti che insegnano un corso	Entità figlia di dipendenti, corsi
Professionisti	Lavoratori con titolo di studio che svolgono attività biomediche	Entità figlia di dipendenti, prestazioni
Lavoratori	Dipendenti con mansioni generiche	Entità figlia di dipendenti, eventi
Eventi	Gli eventi organizzati per pubblicizzare la piscina	Lavoratori, Sponsor
Sponsor	Aziende che finanziano gli eventi	Eventi

# PROGETTAZIONE CONCETTUALE

## LISTA DELLE ENTITÀ

### ➤ Persone

- ID\_persona: serial PRIMARY KEY
- CF: char(16) NOT NULL
- Nome: varchar(255) NOT NULL
- Cognome: varchar(255) NOT NULL
- Data\_di\_nascita date
- Eta: int
- Sesso: char(1)
- Telefono: varchar(16) NOT NULL UNIQUE
- Residenza: varchar(255) NOT NULL

### ➤ Tariffario

- Codice: int NOT NULL PRIMARY KEY
- Prezzo: int NOT NULL
- Tipologia: varchar(50) NOT NULL UNIQUE
- Descrizione: varchar(255)

### ➤ Iscritto

- ID\_iscritto: int PRIMARY KEY
- tip\_abbonamento: int
- Data\_inizio: date NOT NULL
- Scadenza: date NOT NULL

### ➤ Contratti

- Num\_contratto: serial PRIMARY KEY
- Cognome: varchar(255) NOT NULL
- Nome: varchar(255) NOT NULL
- CF: char(16) NOT NULL
- Data\_di\_nascita: date NOT NULL
- Eta: int
- Indirizzo: varchar(255) NOT NULL

- Telefono: varchar(16) NOT NULL UNIQUE
- Email: varchar(255)
- Tipologia\_contratto: varchar(255) NOT NULL
- Data\_assunzione: date NOT NULL
- Scadenza\_contratto: date
- Stipendio: int NOT NULL

➤ Dipendenti

- Matricola PRIMARY KEY

L'entità dipendenti si specializza in tre sottocategorie con una generalizzazione totale:

➤ Istruttori

- Mansione: varchar(255) NOT NULL
- Diploma: varchar(255) NOT NULL
- Certificazione: varchar(255)

➤ Professionisti

- Titolo\_di\_studio: varchar(255) NOT NULL
- Certificazione: varchar(255)
- Servizio: varchar(255) NOT NULL UNIQUE

➤ Lavoratori

- Mansione: varchar(255) NOT NULL
- Turno\_settimanale: varchar(255)

➤ Prestazioni

- ID\_prestazioni serial PRIMARY KEY
- Tipologia varchar(255)
- Costo int

➤ Corsi

- ID\_attivita: serial PRIMARY KEY
- Responsabile: int
- Tipologia: varchar(255) NOT NULL
- Giorni (1:N) : varchar(15)
- Orario (1:N) : time
- Edificio: varchar(50) NOT NULL

➤ Eventi

- ID\_evento: serial PRIMARY KEY
- Nome: varchar(255) NOT NULL
- Luogo: varchar(255)
- Indirizzo: varchar(255)
- Data: date NOT NULL
- Orario: time
- Budget: int NOT NULL

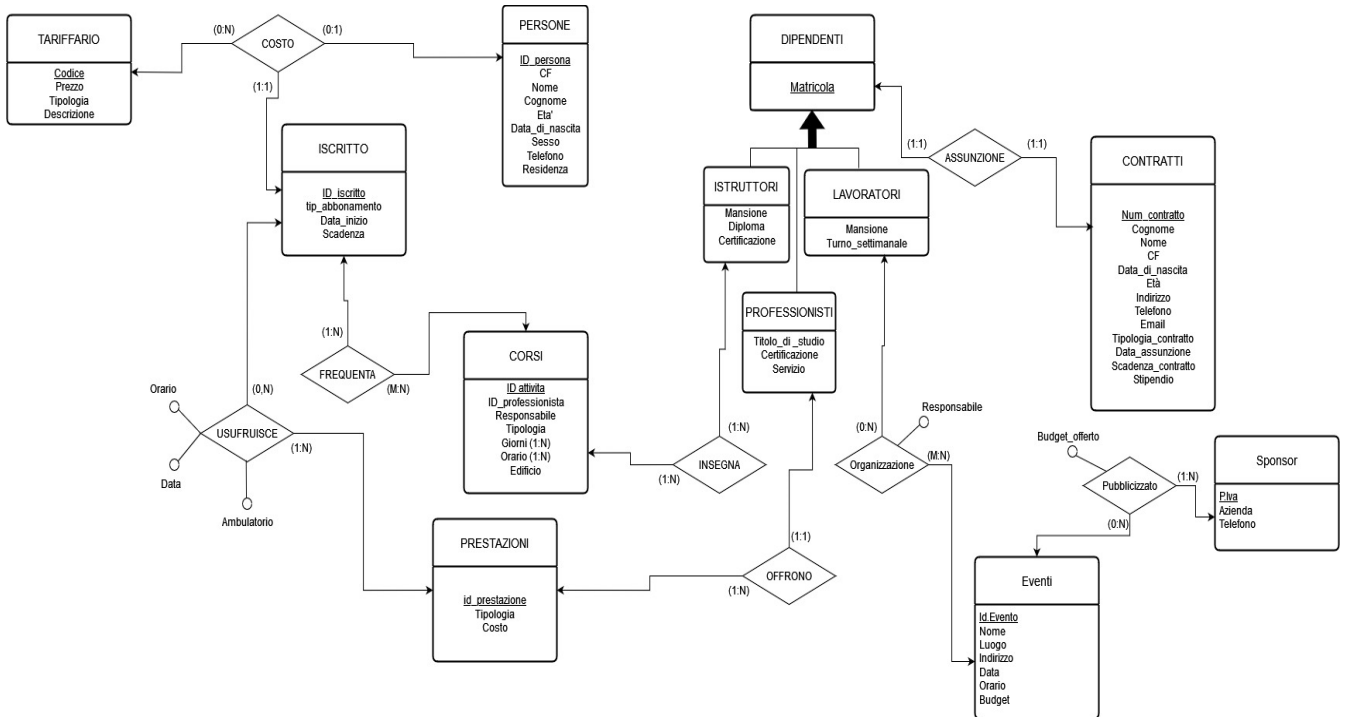
➤ Sponsor

- P\_iva: char(11) NOT NULL PRIMARY KEY
- Azienda: varchar(255) NOT NULL
- Telefono: varchar(16) NOT NULL UNIQUE

## LISTA DELLE RELAZIONI

Relazione	Entità coinvolte	Descrizione	Attributi
Costo	Tariffario(0:N) Iscritto(1:1) Persone(0:1)	Una persona può iscriversi alla piscina acquistando un abbonamento	
Frequenta	Iscritto(1:N) Corsi(M:N)	Un iscritto alla piscina frequenta uno o più corsi, un corso è seguito da più iscritti	
Usufruisce	Iscritto(0:N) Prestazioni(1:N)	Un iscritto può usufruire da zero o più prestazioni, una prestazione può esser svolta su più iscritti	Data, orario, ambulatorio
Insegna	Istruttori(1:N) Corsi(1:N)	Un istruttore può insegnare in più corsi differenti, un corso può essere svolto da più di un istruttore	
Offrono	Professionisti(1:1) Prestazioni(1:N)	Un professionista può svolgere una solo prestazione, le prestazioni sono svolte da uno o più professionisti	
Organizzazione	Lavoratori(0:N) Eventi(M:N)	Un lavoratore può organizzare zero o più eventi, un evento può essere organizzato da più lavoratori	Responsabile
Pubblicizzato	Sponsor(1:N) Eventi(0:N)	Ogni sponsor finanzia almeno un evento, gli eventi possono essere finanziati da zero o più sponsor	Budget offerto
Assunzione	Contratti(1:1) Dipendenti(1:1)	Per ogni dipendente viene stipulato un solo contratto, Un contratto stipulato corrisponde ad un solo dipendente	

## Schema E-R



# PROGETTAZIONE LOGICA

## RISTRUTTURAZIONE

## Eliminazione delle generalizzazioni

Generalizzazione	Risoluzione
Dipendenti ← Istruttori, professionisti, lavoratori	La generalizzazione è stata sostituita con le sue figlie, ovvero al posto della generalizzazione dipendenti si avranno come entità le sue tre entità figlie istruttori, professionisti e lavoratori. Quest'ultime erediteranno gli attributi dell'entità padre (ovvero "matricola")

## Analisi delle forme normali

Nell'analisi delle forme normali riscontriamo che gli attributi giorno e orario dell'entità corsi non rispettano la prima forma normale (1NF), poiché un corso può essere svolto in differenti fasce orarie nella medesima giornata e nelle stesse fasce orarie in giorni diversi.

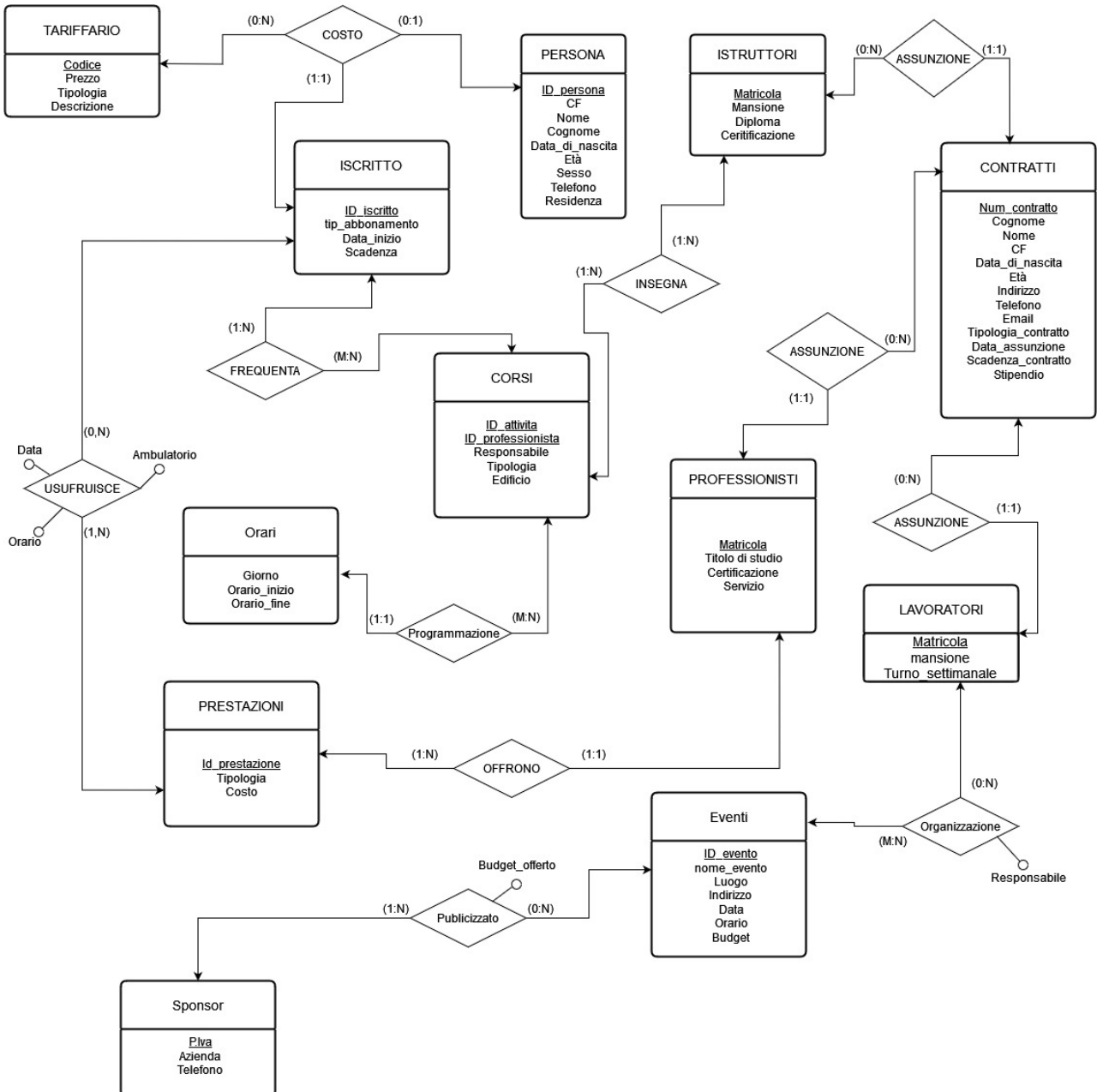
Quindi per risolvere il problema si introduce una nuova entità chiamata orari con i seguenti attributi:

- orari
  - ID\_attività: int PRIMARY KEY
  - Giorno: varchar(15) NOT NULL PRIMARY KEY
  - Orario\_inizio: time NOT NULL PRIMARY KEY

- Orario\_fine: time NOT NULL

L'entità orari si trova in terza forma normale (3NF), e quindi anche in seconda forma normale (2NF), poiché tutti gli attributi non chiave ovvero orario\_fine, dipendono interamente dalla chiave primaria.

## Schema E-R ristrutturato





## Creazione delle tabelle

**Tariffario**(Codice, Prezzo, Tipologia, Descrizione)

**Persone**(Id\_persona, CF, Nome, Cognome, Data\_di\_nascita, Età, Sesso, Telefono, Residenza)

**Iscritto**(ID\_iscritto, Tip\_abbonamento, Data\_inizio, Sadenza)

**Contratti**(Num\_contratto, Cognome, Nome, CF, Data\_di\_nascita, Età, Indirizzo, Telefono, Email, Tipologia\_contratto, Data\_assunzione, Scadenza\_contratto, Stipendio)

**Professionisti**(Matricola, Titolo\_di\_studio, Certificazione, Servizio)

**Istruttori**(Matricola, Mansione, Diploma, Certificazione)

**Lavoratori**(Matricola, Mansione, Turno settimanale)

**Corsi**(ID\_attività, Responsabile, Tipologia, Edificio)

**Orari**(ID\_attività, Giorno, Orario\_inizio, Orario\_fine)

**Frequenta**(ID\_iscritto, ID\_attività)

**Insegna**(ID\_attività, ID\_istruttori)

**Eventi**(ID\_evento, Nome, Luogo, Indirizzo, Data, Orario, Budget)

**Sponsor**(P\_iva, Azienda, Telefono)

**Organizzazione**(ID\_evento, Matricola, Responsabile)

**Pubblicizzato**(P\_iva, ID\_evento, Budget\_offerto)

**Prestazioni**(ID\_prestazioni, ID\_professionista, Tipologia, Costo)

**Usufruisce**(ID\_prestazione, ID\_iscritto, Data, Orario)

**Tariffario**.Codice → **Iscritto**.Tip\_abbonamento

**Persona**.ID\_persona → **Iscritto**.ID\_iscritto

**Contratti**.Num\_contratto → **Professionisti**.Matricola, **Istruttori**.Matricola, **Lavoratori**.Matricola

**Istruttori**.Matricola → **Corsi**.Responsabile, **Insegna**.ID\_istruttori

**Iscritto**.ID\_iscritto → **Frequenta**.ID\_iscritto, **Usufruisce**.ID\_iscritto

**Corsi**.ID\_attività → **Frequenta**.ID\_attività, **Insegna**.ID\_attività, **Orari**.ID\_attività

**Evento**.ID\_evento → **Organizzazione**.ID\_evento, **Pubblicizzato**.ID\_evento

**Lavoratori**.Matricola → **Organizzazione**.Matricola, **Organizzazione**.Responsabile

**Professionisti**.Matricola → **Prestazioni**.ID\_professionista

**Prestazioni**.ID\_prestazioni → **Usufruisce**.ID\_prestazione

**Sponsor**.P\_iva → **Pubblicizzato**.P\_iva

## QUERY

1. Mostrare i tre corsi con il maggior numero di iscritti.

```
SELECT tipologia, count(*) as num_iscritti
FROM iscritto,frequenta, corsi
WHERE iscritto.id_iscritto = frequenta.id_iscritto
    and frequenta.id_attività = corsi.id_attività
GROUP BY tipologia
ORDER BY num_iscritti DESC
LIMIT 3
```

```
+-----+-----+
| tipologia | num_iscritti |
+-----+-----+
| Tuffi     | 5             |
| Nuoto_libero | 5             |
| PallaNuoto | 5             |
+-----+-----+
```

2. Mostrare tutti gli iscritti che usufruiscono di almeno una prestazione.

```
SELECT nome, cognome, data, orario, tipologia, costo
FROM persone, iscritto, usufruisce, prestazioni
WHERE persone.id_persona = iscritto.id_iscritto and iscritto.id_iscritto = usufruisce.id_iscritto and
       prestazioni.id_prestazioni = usufruisce.id_prestazione
```

nome	cognome	data	orario	tipologia	costo
Dionisia	Palerma	2023-07-02	15:00:00	Visita medica agonistica	75
Adelinda	Colombo	2023-06-30	16:30:00	Piano alimentare	80
Maria	Cattaneo	2023-05-31	14:00:00	Seduta osteopaitca	50
Manuel	Ferri	2023-05-22	13:30:00	Visita medica	50
Manuel	Esposito	2023-05-07	14:30:00	Fisioterapia	60
Camelia	Conti	2023-05-07	15:10:00	Fisioterapia	60
Arianna	Castiglione	2023-05-06	10:30:00	Visita medica agonistica	75
Cora	Palerma	2023-02-18	09:00:00	Piano alimentare	80
Severino	Lombardi	2023-01-05	11:00:00	Piano alimentare	80

3. Mostrare tutti i contratti scaduti.

[illegible]

num_contratto	cognome	nome	cf	telefono	email	tipologia_contratto	data_assunzione	scadenza_contratto
4	Udinese	Benigna	DNSBGN61D65D810X	321 9963880	BenignaUdinese@dayrep.com	Tempo determinato	2017-01-01	2017-12-22
5	Padovesi	Alfonso	PDVLNS92R29D150Y	325 3217653	AlfonsoPadovesi@teleworm.us	Collaborazione sportiva	2012-04-05	2020-07-15
10	Li Fonti	Bruto	LFNBRT68AQ1G273C	371 1928051	BrutoLiFonti@jourrapide.com	Tempo determinato	2020-05-15	2021-05-15

4. Mostrare il guadagno e la spesa totale dell'azienda.

```

DROP VIEW IF EXISTS guadagno_tot;
DROP VIEW IF EXISTS spesa_tot;
CREATE VIEW guadagno_tot as
((SELECT sum(costo) as guadagno
FROM usufruisce, prestazioni
WHERE usufruisce.id_prestazione = prestazioni.id_prestazioni)
UNION
(SELECT sum(prezzo) as guadagno
FROM iscritto, tariffario
WHERE iscritto.tip_abbonamento = tariffario.codice));

CREATE VIEW spesa_tot AS
(SELECT sum(stipendio) as stipendio
FROM contratti
WHERE num_contratto IN ((select matricola
                           from istruttori)
                           UNION
                           (select matricola
                           from professionisti)
                           UNION
                           (select matricola
                           from LAVORATORI)));

SELECT sum(guadagno_tot.guadagno) as guadagno_totale, sum(spesa_tot.stipendio) as spesa_totale
FROM guadagno_tot, spesa_tot

```

guadagno_totale	spesa_totale
6225	36400

5. Mostrare il nome e il cognome degli iscritti che frequentano più di un corso.

```

SELECT nome, cognome
FROM iscritto, persone
WHERE persone.id_persona = iscritto.id_iscritto and id_iscritto IN (SELECT iscritto.id_iscritto
FROM iscritto, frequenta, corsi
WHERE iscritto.id_iscritto = frequenta.id_iscritto
and frequenta.id_attività = corsi.id_attività
GROUP BY iscritto.id_iscritto
HAVING count(iscritto.id_iscritto) > 1)

```

nome	cognome
Giacomo	Marchesi
Dionisia	Palerma
Susanna	Manfrin
Manuel	Ferri
Gianleone	Pugliesi
Francesca	Trevisan
Learco	Lucciano

6. Dato un evento mostrare l'azienda che vi ha partecipato, il responsabile e il budget. Nella query si è usato l'evento "Memorial\_Mario\_Zanchi".

```
SELECT responsabile as matricola_responsabile, cognome, contratti.nome, azienda, budget
FROM CONTRATTI, organizzazione, eventi, sponsor, publicizzato
WHERE organizzazione.responsabile = contratti.num_contratto
and organizzazione.id_evento = eventi.id_evento
and eventi.id_evento = publicizzato.id_evento
and publicizzato.p_iva = sponsor.p_iva and eventi.nome='Memorial_Mario_Zanchi'
```

matricola_responsabile	cognome	nome	azienda	budget
14	Mazzi	Anastasio	Alternativa Impianti	200

7. Dato nome e cognome di un iscritto mostra la tipologia di abbonamento acquistato e i corsi che frequenta l'iscritto. Nella query si è usato come nome "Gennaro" e come cognome "Bucchio".

```
SELECT tariffario.tipologia, iscritto.data_inizio, iscritto.scadenza, corsi.tipologia
FROM iscritto, frequenta, corsi, tariffario
WHERE tariffario.codice = iscritto.tip_abbonamento and iscritto.id_iscritto = frequenta.id_iscritto and frequenta.id_attività = corsi.id_attività
and iscritto.id_iscritto in (select id_persona
from persone
where nome = 'Gennaro' and cognome = 'Bucchio')
```

tipologia	data_inizio	scadenza	tipologia
Annuale	2022-12-15	2023-12-05	Acquagym

8. Data l'azienda mostrare il budget che essa ha offerto per ogni evento alla quale essa ha partecipato. Nella query si è usato l'azienda "I.C.E.\_Srl".

```
SELECT nome, budget as budget_evento, budget_offerto
FROM eventi, sponsor, publicizzato
WHERE azienda = 'I.C.E._Srl' and sponsor.p_iva = publicizzato.p_iva and eventi.id_evento = publicizzato.id_evento
```

nome	budget_evento	budget_offerto
Campionato regionale apnea	1000	600
IRONMAN_70.3	2000	250

9. Mostrare la media dei budget offerti dalle aziende superiori ad una certa soglia. Nella query come valore soglia è stato utilizzato "400".

```
SELECT azienda, avg(publicizzato.budget_offerto)::numeric(10,2) as massimo_budget_offerto
FROM publicizzato,sponsor
WHERE sponsor.p_iva = publicizzato.p_iva
GROUP BY publicizzato.p_iva, azienda
HAVING avg(publicizzato.budget_offerto) > '400'
ORDER BY azienda
```

azienda	massimo_budget_offerto
Gioielleria_Duca_Srl	600.00
I.C.E._Srl	425.00

10. Dato il professionista mostra il numero dei suoi clienti e il guadagno totale delle prestazioni.  
Come professionista è stato utilizzato "Nino Toscano".

```
SELECT sum(prestazioni.costo) as Totale_guadagno_del_professionista, count(usufruisce.id_iscritto) as numero_clienti, prestazioni.tipologia
FROM contratti, prestazioni, professionisti, usufruisce
WHERE contratti.nome = 'Nino' and contratti.cognome = 'Toscano' and contratti.num_contratto = professionisti.matricola
and professionisti.matricola = prestazioni.id_professionista and prestazioni.id_prestazioni = usufruisce.id_prestazione
GROUP BY prestazioni.tipologia
```

totale_guadagno_del_professionista	numero_clienti	tipologia
120	2	Fisioterapia
50	1	Seduta osteopaitca

## Indici

Questo database è progettato per una piscina da medie a grandi dimensioni di conseguenza sulla tabella persone verranno svolte numerose operazioni di lettura richiedendo una certa velocità.

Quindi si è deciso di indicizzare la tabella "Persone" sugli attributi Nome e Cognome.

```
CREATE INDEX indice_persone ON Persone(Nome , Cognome);
```

I tempi di accesso a confronto sono:

eta	sezzo	telefono	residenza
integer	character	character varying (16)	character varying (255)
35	M	335 9180284	Stradone Antonio Provolo, ..
26	M	398 7542801	Via Longhena, 138

✓ Successfully run. Total query runtime: 146 msec. 35 rows affected. ✕

Ln 1, Col 22

Tempo senza indici

43	M	332 0011903	Piazza Bovio, 91
56	F	349 0545477	Via Rosmini, 122
32	F	355 4724570	Via del Pontiere, 7
24	M	370 5010522	Via Antonio Cecchi, 99

✓ Successfully run. Total query runtime: 117 msec. 35 rows affected. ✕

Ln 1, Col 22

Tempo con indici

# Codice C++

## Descrizione dell'utilizzo del codice

Il codice C++ per l'esecuzione delle query consiste in un unico file `.cpp` che va compilato attraverso il comando:

```
g++ -ansi -static-libstdc++ PROGETTO.cpp funzioni_progetto.cpp -o progetto -L <N_cartella> -lpq
```

Prima di poter compilare il codice è necessario inserire i file `libpq.dll`, `libpq.lib`, `libpq-fe.h`, `pg_config_ext.h`, `postgres_ext.h` nella cartella designata e modificare il comando precedente cambiando `<N_cartella>` con la cartella dove sono stati inseriti i file. Eseguire il codice tramite `progetto.exe`. Inoltre, è necessario modificare le costanti all'inizio del file `PROGETTO.cpp` in base alla propria configurazione del database: `PG_HOST`, `PG_USER`, `PG_DB`, `PG_PASS`, `PG_PORT`.

```
PS C:\Users\gabriele\OneDrive\Desktop\Basi\Progetto\dependencies> .\progetto.exe
Connessione avvenuta correttamente
Benvenuti nel gestionale Active Pool
Inserire il numero della query desiderata [1-10], [0] o un carattere chiudono il programma

1) QUERY_1: I tre corsi con il maggior numero di iscritti
2) QUERY_2: Tutti gli iscritti che usufruiscono di almeno una prestazione
3) QUERY_3: Trova i contratti scaduti
4) QUERY_4: Trova Guadagno e Spesa totale dell'azienda
5) QUERY_5: Trova Nome e Cognome degli iscritti che frequentano piu' di un corso
6) QUERY_6: Dato un evento trova L'azienda che vi ha partecipato, il responsabile e il budget
7) QUERY_7: Dato Nome e Cognome di un iscritto trova la tipologia di abbonamento pagato e i corsi che frequenta
8) QUERY_8: Data l'azienda trova che budget essa ha offerto per ogni evento alla quale ha partecipato
9) QUERY_9: Trova la media dei budget offerti dalle aziende superiori a una certa soglia inserita dall'utente
10) QUERY_10: Dato il professionista trova il numero_clienti e il guadagno totale delle prestazioni

Quale query si vuole visualizzare:
```

Sullo schermo apparirà il seguente Menù con la lista delle query, identificate da 1 a 10, mentre lo 0 o un qualsiasi carattere chiuderanno il programma. Se si inserisce un numero non compreso nel Menù l'utente potrà riprovare inserendo un numero corretto.

Le query dalla 6 alla 10 richiedono l'inserimento di parametri da parte dell'utente. In particolare:

- La query 6 richiede l'inserimento del nome di un evento.
- La query 7 richiede l'inserimento del nome e del cognome di una persona.
- La query 8 richiede l'inserimento del nome di un'azienda.

- La query 9 richiede l'inserimento di un numero intero rappresentante il valore soglia desiderato.
- La query 10 richiede l'inserimento del nome del professionista e successivamente il suo cognome

## Documentazione del codice

Principali funzioni utilizzate dal codice sono:

```
PGconn *conn = PQconnectdb(conninfo);
```

Ritorna una connessione al database, utilizzando i parametri di accesso.

```
PGresult * PQexec(PGconn, const char * command);
```

Fornisce un comando al server e aspetta il risultato.

```
PGresult * PQprepare(PGconn * conn, const char *stmtName, const char * query, int nParams, const Oid *paramTypes);
PGresult * PQexecPrepared(PGconn * conn, const char *stmtName, int nParams, const char * const *paramValues,
                           const int *paramLengths, const int *paramFormats, int resultFormat);
```

PQprepare, fornisce una richiesta per preparare un'istruzione con dei parametri dati e aspetta il completamento mentre PQexecPrepared invia una richiesta di esecuzione a un'istruzione preparata con parametri dati e aspetta il risultato.

```
void PQclear(PGresult *res);
```

Libera la memoria associata ad un PGresult.

```
void printQuery(PGresult* res);
```

La seguente istruzione stampa il risultato res di una query come tabella a schermo, e la funzione si adatta automaticamente alla dimensione delle colonne.

## AMBIENTE DI SVILUPPO

Gabriele Di Pietro	Alessandro Rossi
SO: Windows 11 Pro 10.0.22621	SO: Windows 11 Home 10.0.22621
C++	C++
Editor: Visual Studio Code	Editor: Visual Studio Code
Compilatore: g++ 11.2.0	Compilatore: g++ 11.2.0
SQL	SQL
Editor: PostgreSQL	Editor: PostgreSQL
Versione: 15.3	Versione: 15.2