



Progetto corso Basi di dati

Anno Accademico 2021/2022

# Sistema gestionale per impianti da bowling

Trillò Giorgia

Matricola 0287103

## Sommario

|                                      |    |
|--------------------------------------|----|
| Informazioni generali .....          | 5  |
| Analisi dei requisiti .....          | 6  |
| Descrizione .....                    | 6  |
| Glossario dei termini.....           | 7  |
| Schema logico.....                   | 8  |
| Schema fisico (normalizzato) .....   | 9  |
| Descrizione Entità e Relazioni ..... | 10 |
| Dizionario dei dati.....             | 11 |
| Impianto.....                        | 11 |
| Giocatore .....                      | 11 |
| Promozione .....                     | 11 |
| Dipendente .....                     | 12 |
| Inventario .....                     | 12 |
| Scarpe .....                         | 12 |
| Pista .....                          | 12 |
| Partita.....                         | 13 |
| Abbonamento.....                     | 13 |
| Regole aziendali.....                | 14 |
| Regole di vincolo.....               | 14 |
| Tabelle.....                         | 15 |
| Tabella impianto .....               | 16 |
| Creazione .....                      | 16 |
| Inserimenti.....                     | 16 |
| Sample.....                          | 16 |
| Tabella pista .....                  | 17 |
| Creazione .....                      | 17 |
| Inserimenti.....                     | 17 |
| Sample.....                          | 17 |
| Tabella giocatore .....              | 18 |
| Creazione .....                      | 18 |
| Inserimenti.....                     | 18 |
| Sample.....                          | 18 |
| Tabella partita.....                 | 19 |
| Creazione .....                      | 19 |
| Inserimenti.....                     | 19 |

|                                  |    |
|----------------------------------|----|
| Sample.....                      | 19 |
| Tabella inventario .....         | 20 |
| Creazione .....                  | 20 |
| Inserimenti.....                 | 20 |
| Sample.....                      | 20 |
| Tabella dipendente .....         | 21 |
| Creazione .....                  | 21 |
| Inserimenti.....                 | 21 |
| Sample.....                      | 21 |
| Tabella scarpe .....             | 22 |
| Creazione .....                  | 22 |
| Inserimenti.....                 | 22 |
| Sample.....                      | 22 |
| Tabella abbonamento.....         | 23 |
| Creazione .....                  | 23 |
| Inserimenti.....                 | 23 |
| Sample.....                      | 23 |
| Tabella promozione .....         | 24 |
| Creazione .....                  | 24 |
| Inserimenti.....                 | 24 |
| Sample.....                      | 24 |
| Tabella noleggio .....           | 25 |
| Creazione .....                  | 25 |
| Inserimenti.....                 | 25 |
| Sample.....                      | 25 |
| Tabella impianto_inventario..... | 26 |
| Creazione .....                  | 26 |
| Inserimenti.....                 | 26 |
| Sample.....                      | 26 |
| Query significative .....        | 27 |
| Abbonamenti .....                | 27 |
| Dipendenti di un impianto .....  | 28 |
| Media dei punteggi .....         | 29 |
| Promozioni attive.....           | 29 |
| Partite del giorno.....          | 30 |
| Le partite di Mario .....        | 30 |
| Abbonamenti attivi.....          | 31 |
| Classifica.....                  | 32 |

|  |    |
|--|----|
| Giocatori affezionati .....                        | 33 |
| Query 10 .....                                     | 33 |
| Addetti all'inventario .....                       | 34 |
| Scarpe 42 .....                                    | 34 |
| Ricerca di un articolo .....                       | 35 |
| Piste disponibili .....                            | 35 |
| Abbonamenti in scadenza .....                      | 36 |
| Partite in compagnia .....                         | 36 |
| Partite in contemporanea .....                     | 37 |
| Query algebra relazionale .....                    | 38 |
| Partire giocate da Mario .....                     | 38 |
| Scarpe utilizzate da un giocatore .....            | 38 |
| Stored Procedure .....                             | 39 |
| Controllo disponibilità scarpe .....               | 39 |
| Aggiunta di un nuovo giocatore .....               | 40 |
| Aggiunta di un nuovo dipendente .....              | 40 |
| Aggiunta di una nuova partita .....                | 41 |
| Piste disponibili in un determinato impianto ..... | 41 |
| Trigger .....                                      | 42 |
| Controllo nuovo abbonato .....                     | 42 |
| Controllo prenotazione .....                       | 43 |
| Viste .....  | 44 |
| Vista scarpe noleggiate .....                      | 44 |
| Vista dipendenti .....                             | 44 |
| Vista partite della settimana .....                | 44 |
| Ottimizzazione query .....                         | 45 |
| Considerazioni su InnoDB Buffer .....              | 45 |
| Tempi impiegati dalle query .....                  | 46 |
| Classifica .....                                   | 47 |
| Prima versione della query .....                   | 47 |
| Versione ottimizzata .....                         | 48 |
| Ottimizzazione Giocatori affezionati .....         | 49 |
| Prima versione della query .....                   | 49 |
| Versione ottimizzata .....                         | 50 |

## Informazioni generali

La versione MySQL utilizzata per gli esempi del progetto è 8.0.31 montato su un sistema operativo Windows 11 a 64 bit.

Le fasi di test e ottimizzazione sono state eseguite sul sistema operativo Windows 11 installato su un laptop Asus, con processore Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz, 1992 Mhz, 4 core, 8 processori logici, memoria RAM 8GB.

# Analisi dei requisiti

## Descrizione

Si vuole realizzare un database per la gestione di impianti da bowling per ottenere la gestione completa dei seguenti aspetti.

Ogni impianto ha un id univoco inoltre ha informazioni riguardanti il nome, l'indirizzo civico, un numero di telefono e un indirizzo email per contattare le diverse strutture.

I diversi impianti offrono delle offerte speciali (promozioni) ognuna delle quali è identificata da un id univoco quindi si intende memorizzare per ognuna di esse un nome, una descrizione, una data di inizio, una di fine e una percentuale di sconto.

Ogni impianto ha a disposizione diverse piste le quali sono identificate da un id univoco e da un *flag* che ne indica la disponibilità.

Si vuole gestire anche gli articoli a noleggio dei prodotti presenti all'interno di ogni impianto, tutto questo viene gestito all'interno di un inventario che memorizza gli id univoci di ogni articolo e una descrizione del prodotto. La quantità sarà gestita nella relazione tra impianto e inventario.

All'interno del database saranno memorizzati i dipendenti di ogni impianto ognuno identificato da un id univoco, da un nome, un cognome, una data di assunzione, un indirizzo civico, un recapito telefonico, dallo stipendio che percepisce e da un id dell'impianto specifico dove lavora.

Si vuole registrare il nome, il cognome, la data di nascita, un recapito telefonico e una email dei giocatori che intendono partecipare alle partite, ogni giocatore sarà identificato da un id univoco.

Le informazioni memorizzate delle partite saranno un orario di inizio e di fine, la data e l'id della pista in cui si svolgerà, un id del giocatore che partecipa alla partita, un punteggio ottenuto da ogni giocatore.

Inoltre ogni giocatore potrà noleggiare delle scarpe ognuna delle quali avrà un id univoco, una taglia, una marca e un prezzo di noleggio che varia dai 5 ai 6 euro a seconda della marca scelta/disponibile. Il tutto sarà tracciato in una tabella specifica (noleggio) che memorizza la data e l'ora di inizio, l'id dei giocatori che noleggiavano un paio di scarpe e un *flag* (consegnato) che indica se l'articolo è stato restituito.

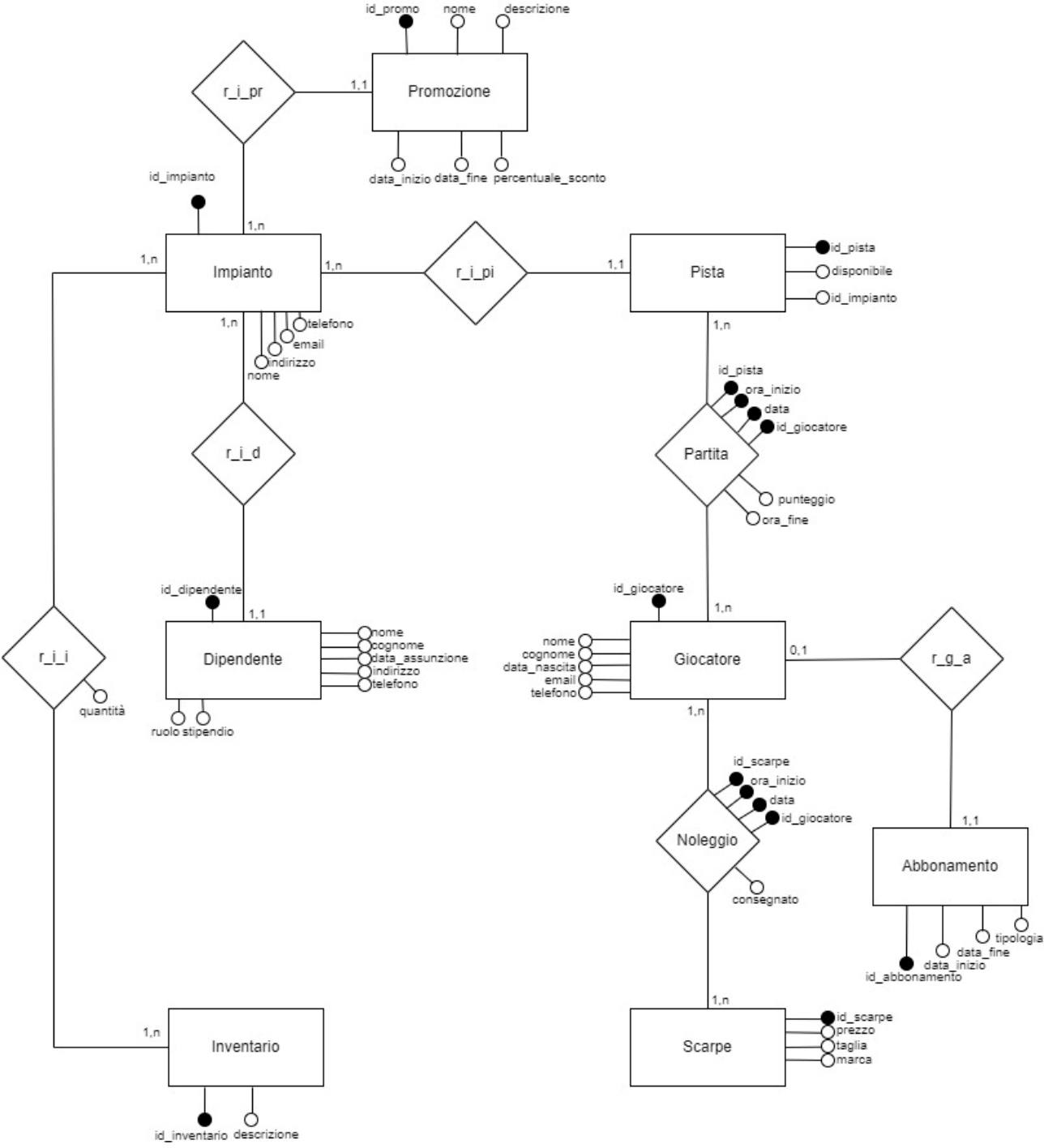
Saranno gestiti anche degli abbonamenti per permettere agli abbonati di usufruire di alcune offerte e servizi a loro dedicati. Si prevedono tre tipologie di abbonamento: mensile, trimestrale, annuale.

Questi ultimi sono identificati da un id univoco, una data di inizio, una data di fine che indica la scadenza del contratto, una tipologia differente tra quelle evidenziate precedentemente e un id del giocatore che ne è possessore.

## Glossario dei termini

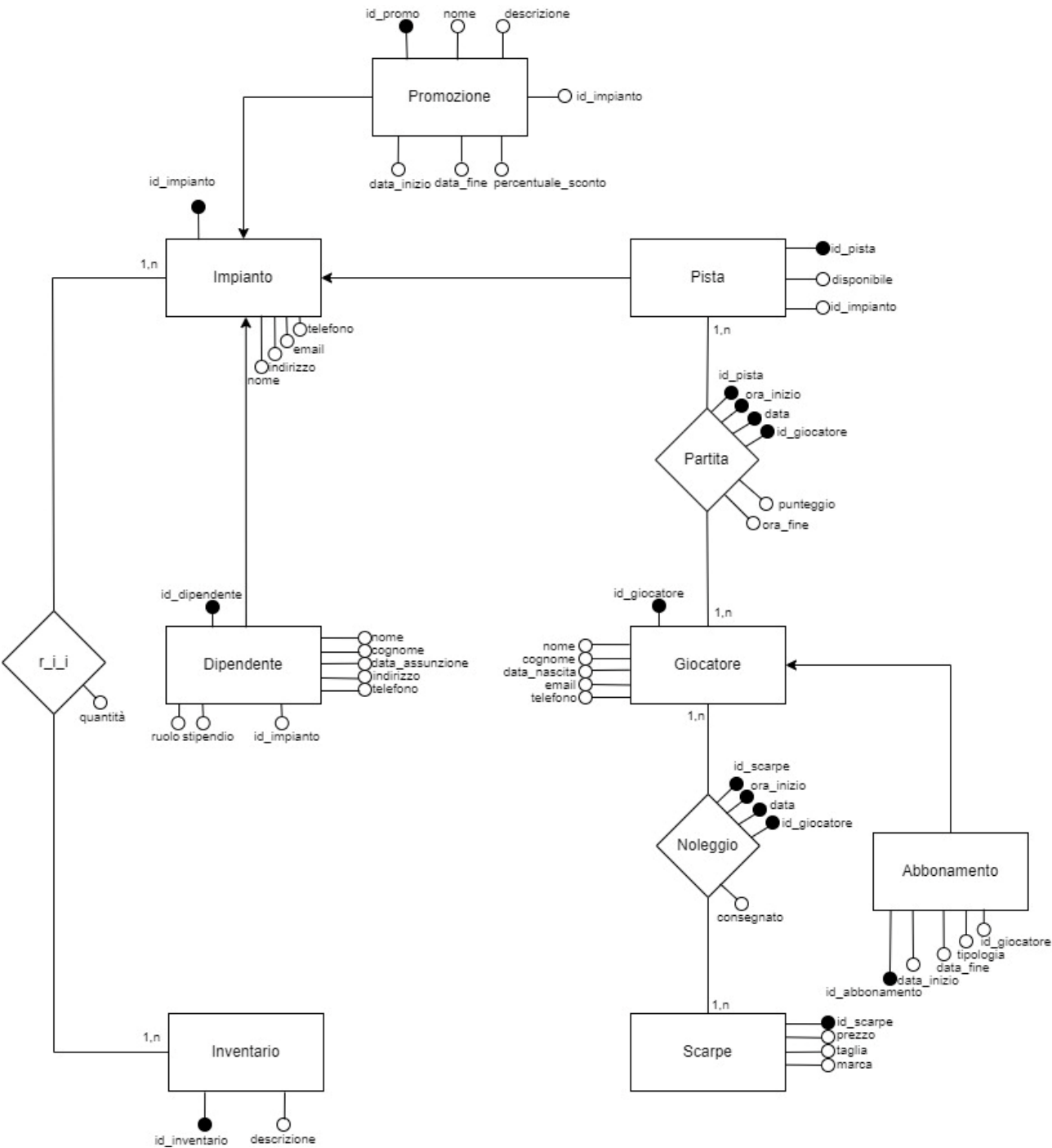
| Termine     | Descrizione   | Sinonimi           | Collegamenti                              |
|-------------|---|--------------------|---|
| Abbonamento | Contratto per cui il giocatore può usufruire di alcune offerte o servizi                          |                    | Giocatore                                 |
| Dipendente  | Persona che lavora all'interno di un impianto   |                    | Impianto                                  |
| Giocatore   | Persona che prende parte ad una partita   | Utente             | Abbonamento, partita, pista, scarpe       |
| Impianto    | Luogo opportunamente attrezzato e conformato per la pratica della disciplina sportiva del Bowling |                    | Dipendente, inventario, pista, promozione |
| Inventario  | Articoli presenti nell'impianto   | Articolo, prodotto | Impianto                                  |
| Noleggio    | Contratto relativo all'uso temporaneo delle scarpe  |                    | Scarpe                                    |
| Partita     | Competizione tra uno o più giocatori  |                    |   |
| Pista       | Luogo dove vengono giocate le partite   |                    | Impianto                                  |
| Promozione  | Offerta proposta durante un arco di tempo   | Offerta speciale   | Impianto                                  |
| Scarpe      | Articolo a noleggio   |                    | Noleggio                                  |

Schema logico





Schema fisico (normalizzato)



## Descrizione Entità e Relazioni

Promozione(id\_promo, nome, descrizione, data\_inizio, data\_fine, percentuale\_sconto, id\_impianto(FK))

Impianto(id\_impianto, nome, indirizzo, email, telefono)

Impianto\_inventario(id\_impianto(FK), id\_inventario(FK), quantità)

Inventario(id\_inventario, descrizione)

Dipendente(id\_dipendente, nome, cognome, data\_assunzione, indirizzo, telefono, ruolo, stipendio, id\_impianto(FK))

Pista (id\_pista, disponibile, numero\_partecipanti, id\_impianto(FK))

Partita (id\_pista, ora\_inizio, data, id\_giocatore, punteggio, ora\_fine)

Giocatore(id\_giocatore, nome, cognome, data\_nascita, email, telefono)

Noleggio(id\_scarpe, ora\_inizio, data, id\_giocatore, consegnato)

Scarpe(id\_scarpe, prezzo, taglia, marca)

Abbonamento(id\_abbonamento, data\_inizio, data\_fine, tipologia, id\_giocatore(FK))

## Dizionario dei dati

### Impianto

Entità che rappresenta le informazioni relative all'impianto. Contiene i dati relativi ad ogni impianto: nome, indirizzo, email e telefono.

| Attributi   | Descrizione                       |
|-------------|-----------------------------------|
| id_impianto | ID univoco dell'impianto          |
| nome        | Nome                              |
| indirizzo   | Indica l'ubicazione dell'impianto |
| email       | Email                             |
| telefono    | Numero di telefono                |

### Giocatore

Entità che rappresenta l'anagrafica di un cliente dell'impianto da bowling, indipendentemente dall'abbonamento.

Contiene i dati personali del giocatore e informazioni di contatto. I dati personali sono obbligatori per poter riconoscere univocamente il giocatore, i dati di contatto sono opzionali.

| Attributi    | Descrizione                |
|--------------|----------------------------|
| id_giocatore | ID univoco di un giocatore |
| nome         | Nome                       |
| cognome      | Cognome                    |
| data_nascita | Data di nascita            |
| email        | Email                      |
| telefono     | Numero di telefono         |

### Promozione

Entità che rappresenta i dati relativi alle varie promozioni che vengono proposte dall'impianto.

| Attributi          | Descrizione                  |
|--------------------|------------------------------|
| id_promo           | ID univoco di una promozione |
| nome               | Nome                         |
| descrizione        | Cognome                      |
| data_inizio        | Data di nascita              |
| data_fine          | Email                        |
| percentuale_sconto | Numero di telefono           |

## Dipendente

L'entità dipendente contiene tutte le informazioni necessarie alla gestione dell'inventario dell'impianto.

| Attributi       | Descrizione  |
|-----------------|--|
| id_dipendente   | ID univoco di un dipendente                                      |
| nome            | Nome   |
| cognome         | Cognome  |
| data_assunzione | Data di assunzione presso l'impianto                             |
| ruolo           | Rappresenta la funzione del dipendente all'interno della società |
| telefono        | Indica il recapito telefonico                                    |
| indirizzo       | Contiene l'indirizzo   |
| stipendio       | Rappresenta lo stipendio   |

## Inventario

L'entità inventario rappresenta l'insieme degli articoli disponibili all'interno del magazzino dell'impianto. Può contenere la descrizione dell'articolo.

| Attributi     | Descrizione               |
|---------------|---------------------------|
| id_inventario | ID univoco dell'articolo  |
| descrizione   | Descrizione dell'articolo |

## Scarpe

L'entità scarpe rappresenta l'insieme delle scarpe disponibili al noleggio. Contiene informazioni relative alla taglia, alla marca.

| Attributi   | Descrizione                        |
|-------------|------------------------------------|
| id_scarpa   | ID univoco dell'articolo           |
| taglia      | Numero della scarpa                |
| marca       | Indica la marca della scarpa       |
| disponibile | Indica se una scarpa è disponibile |

## Pista

L'entità pista contiene informazioni relative alla disponibilità.

| Attributi   | Descrizione                       |
|-------------|-----------------------------------|
| id_pista    | ID univoco della pista            |
| disponibile | Indica se una pista è disponibile |

## Partita

La relazione partita contiene informazioni relative all'orario di inizio, l'orario di fine e i punteggi ottenuti dai giocatori.

| Attributi    | Descrizione              |
|--------------|--------------------------|
| id_partita   | ID univoco della partita |
| ora_inizio   | Orario inizio partita    |
| ora_fine     | Orario fine partita      |
| punteggio    | Punteggio della partita  |
| data         | Data della partita       |
| id_giocatore | ID univoco del giocatore |

## Abbonamento

L'entità abbonamento contiene informazioni relativo alla data inizio e la data di scadenza dell'abbonamento. Inoltre viene indicata la tipologia dell'abbonamento (mensile, trimestrale, annuale).

| Attributi      | Descrizione             |
|----------------|-------------------------|
| id_abbonamento | ID univoco abbonamento  |
| data_inizio    | Data inizio abbonamento |
| data_fine      | Data fine abbonamento   |
| tipologia      | Tipologia abbonamento   |

## Regole aziendali

### Regole di vincolo

|                       |  |
|-----------------------|--|
| <b>RegolaVincolo1</b> | Un utente non può abbonarsi se è sottoscritto già ad un altro abbonamento. |
| <b>RegolaVincolo2</b> | Un dipendente può lavorare solo per uno specifico impianto.                |
| <b>RegolaVincolo3</b> | Ogni giocatore può totalizzare un massimo di 300 punti a partita           |

## Tabelle

In questa sezione sono riportate le tabelle del progetto. Per ciascuna tabella è indicato il codice SQL per la creazione della tabella con relative chiavi e vincoli referenziali, un inserimento tipo e un sample dei dati ottenuti dal database finale.

## Tabella impianto

### Creazione

```
mysql> CREATE TABLE impianto (
```

- > id\_impianto INT PRIMARY KEY not null auto\_increment,
- > nome VARCHAR(255) not null,
- > indirizzo VARCHAR(255) not null,
- > email VARCHAR(255) not null,
- > telefono VARCHAR(255) not null);

### Inserimenti

```
INSERT INTO impianto (nome, indirizzo, email, telefono)
```

```
VALUES ('Spare 1', 'Via Roma 11', 'spare1@email.it', '3331234567');
```

### Sample

```
mysql> SELECT * FROM impianto;
```

| id_impianto | nome    | indirizzo      | email           | telefono   |
|-------------|---------|----------------|-----------------|------------|
| 101         | Spare 1 | Via Roma 11    | spare1@email.it | 3331234567 |
| 102         | Spare 2 | Via Latina 24  | spare2@email.it | 3332345678 |
| 103         | Spare 3 | Via Viterbo 32 | spare3@email.it | 3333456789 |

3 rows in set (0.00 sec)



## Tabella pista

### Creazione

```
mysql> CREATE TABLE pista (
```

- > id\_pista INT PRIMARY KEY not null auto\_increment,
- > disponibile BOOLEAN,
- > id\_impianto INT,
- > FOREIGN KEY (id\_impianto) REFERENCES impianto(id\_impianto)
- > on delete cascade on update cascade);

### Inserimenti

```
INSERT INTO pista (disponibile, id_impianto) VALUES (1, 101);
```

### Sample

| id_pista | disponibile | id_impianto |
|----------|-------------|-------------|
| 1        | 1           | 101         |
| 6        | 1           | 102         |
| 8        | 1           | 102         |
| 12       | 1           | 103         |
| 10       | 1           | 102         |
| 11       | 1           | 103         |
| 2        | 1           | 101         |
| 14       | 1           | 103         |
| 4        | 1           | 101         |
| 13       | 1           | 103         |

10 rows in set (0.00 sec)

## Tabella giocatore

### Creazione

```
mysql> CREATE TABLE giocatore (
```

- > id\_giocatore INT PRIMARY KEY not null auto\_increment,
- > nome VARCHAR(255) not null,
- > cognome VARCHAR(255) not null,
- > data\_nascita DATE not null,
- > email VARCHAR(255) not null,
- > telefono VARCHAR(255) not null);

### Inserimenti

```
INSERT INTO giocatore (nome, cognome, data_nascita, email, telefono)
```

```
VALUES ('Mario', 'Rossi', '1990-01-01', 'mario.rossi@email.com', '333-333-3333');
```

### Sample

```
mysql> SELECT * FROM giocatore ORDER BY RAND() LIMIT 10;
```

| id_giocatore | nome      | cognome    | data_nascita | email                         | telefono        |
|--------------|-----------|------------|--------------|-------------------------------|-----------------|
| 418403       | Ida       | Volterra   | 2007-07-09   | mtirabassi@example.com        | +39 0796648897  |
| 557762       | Puccio    | Zichichi   | 2016-01-21   | gaspare91@example.com         | 0208552284      |
| 567026       | Letizia   | Stefanelli | 2007-03-28   | lettaeraldo@example.net       | +39 3714491567  |
| 536023       | Saverio   | Comeriatto | 1998-11-09   | raffaellobattelli@example.com | 0187848027      |
| 407027       | Patrizio  | Micheletti | 1960-06-01   | lambertopometta@example.com   | +39 34765672642 |
| 407693       | Simone    | Calgari    | 1982-06-03   | annunziata67@example.org      | +39 3755456028  |
| 575585       | Alderano  | Jovinelli  | 1985-02-04   | ktomei@example.net            | +39 349270865   |
| 422711       | Alfredo   | Muti       | 1987-06-09   | arnulfo37@example.org         | 3300329943      |
| 520992       | Ansaldò   | Pagliaro   | 1965-08-15   | giampaolofaloppio@example.net | +39 351443774   |
| 542625       | Raffaello | Romano     | 1987-02-23   | elladiogradenigo@example.com  | 0543762045      |

```
10 rows in set (0.19 sec)
```

## Tabella partita

### Creazione

```
mysql> CREATE TABLE partita (
```

- > ora\_inizio TIME not null,
- > data DATE not null,
- > id\_giocatore INT,
- > punteggio INT not null,
- > ora\_fine TIME not null,
- > id\_pista INT,
- > primary key (id\_pista, ora\_inizio, data, id\_giocatore),
- > FOREIGN KEY (id\_pista) REFERENCES pista(id\_pista)
- > on delete cascade on update cascade,
- > FOREIGN KEY (id\_giocatore) REFERENCES giocatore(id\_giocatore)
- > on delete cascade on update cascade);

### Inserimenti

```
INSERT INTO partita (ora_inizio, data, id_giocatore, punteggio, ora_fine, id_pista)
```

```
VALUES ('10:30:00', '2022-07-21', 418060, 241, '17:51:00', 15);
```

### Sample

```
mysql> SELECT * FROM partita ORDER BY RAND() LIMIT 10;
```

| ora_inizio | data       | id_giocatore | punteggio | ora_fine | id_pista |
|------------|------------|--------------|-----------|----------|----------|
| 10:46:00   | 2022-09-03 | 595732       | 181       | 17:58:00 | 9        |
| 12:32:00   | 2021-09-18 | 598561       | 132       | 15:22:00 | 3        |
| 10:38:00   | 2020-07-23 | 540855       | 279       | 16:37:00 | 10       |
| 20:03:00   | 2022-04-05 | 502558       | 106       | 21:42:00 | 14       |
| 17:47:00   | 2022-01-12 | 498029       | 216       | 19:21:00 | 5        |
| 13:53:00   | 2021-07-06 | 538239       | 97        | 20:14:00 | 6        |
| 09:40:00   | 2021-05-17 | 492045       | 32        | 13:23:00 | 9        |
| 17:53:00   | 2020-05-01 | 468823       | 123       | 19:12:00 | 6        |
| 20:54:00   | 2022-03-08 | 452506       | 200       | 21:16:00 | 8        |
| 19:37:00   | 2021-05-14 | 579908       | 154       | 21:02:00 | 7        |

```
10 rows in set (0.12 sec)
```

## Tabella inventario

### Creazione

```
mysql> CREATE TABLE inventario (
```

```
-> id_inventario INT PRIMARY KEY not null auto_increment,
```

```
-> descrizione VARCHAR(255) not null);
```

### Inserimenti

```
INSERT INTO inventario (descrizione) VALUES ('Palle da bowling');
```

### Sample

```
mysql> SELECT * FROM inventario ORDER BY RAND() LIMIT 10;
```

| id_inventario | descrizione                            |
|---------------|--|
| 14            | Sistemi di gestione delle prenotazioni |
| 1             | Palle da bowling                       |
| 2             | Scarpe da bowling                      |
| 11            | Tavoli per il servizio bar             |
| 3             | Birilli da bowling                     |
| 12            | Sgabelli da bar                        |
| 13            | Sistemi di videosorveglianza           |
| 9             | Sistemi di illuminazione               |
| 7             | Pettorine da bowling                   |
| 10            | Sistemi di sonorizzazione              |

```
10 rows in set (0.00 sec)
```

## Tabella dipendente

### Creazione

```
mysql> CREATE TABLE dipendente (  
id_dipendente INT PRIMARY KEY not null auto_increment,  
nome VARCHAR(255) not null,  
cognome VARCHAR(255) not null,  
ruolo VARCHAR(255) not null,  
stipendio INT not null,  
data_assunzione DATE not null,  
indirizzo VARCHAR(255) not null,  
telefono VARCHAR(255) not null,  
id_impianto INT,  
FOREIGN KEY (id_impianto) REFERENCES impianto(id_impianto)  
on delete cascade on update cascade);
```

### Inserimenti

```
INSERT INTO dipendente (nome, cognome, ruolo, stipendio, data_assunzione, indirizzo,  
telefono, id_impianto)
```

```
VALUES ('John', 'Doe', 'Manager', 50000, 101);
```

### Sample

```
mysql> SELECT * FROM dipendente ORDER BY RAND() LIMIT 10;
```

| id_dipendente | nome     | cognome     | ruolo                     | stipendio | id_impianto | data_assunzione | indirizzo       | telefono   |
|---------------|----------|-------------|---------------------------|-----------|-------------|-----------------|-----------------|------------|
| 215           | Licia    | Bottigliero | cassiera                  | 800       | 102         | 2022-05-01      | via Milano, 25  | 3331231234 |
| 222           | Marisa   | Bonino      | cassiera                  | 800       | 102         | 2022-11-01      | via Milano, 22  | 3331234567 |
| 218           | Ugo      | Fiorucci    | addetto alla manutenzione | 900       | 102         | 2022-08-01      | via Torino, 15  | 3334445555 |
| 216           | Renata   | Broggini    | amministratore            | 1200      | 103         | 2022-06-01      | via Napoli, 18  | 3333456789 |
| 213           | Simone   | Neri        | amministratore            | 1200      | 103         | 2022-03-01      | via Napoli, 15  | 3333333333 |
| 212           | Luca     | Bianchi     | addetto pulizie           | 800       | 102         | 2022-02-01      | via Milano, 10  | 3332222222 |
| 214           | Nico     | Pedrazzini  | addetto alla manutenzione | 900       | 101         | 2022-04-01      | via Roma, 23    | 3331112222 |
| 211           | Mario    | Rossi       | cassiere                  | 1000      | 101         | 2022-01-01      | via Roma, 12    | 3331234567 |
| 223           | Alberico | Faggiani    | addetto all'inventario    | 900       | 103         | 2022-12-01      | via Napoli, 21  | 3333333333 |
| 217           | Marisa   | Petrucchi   | addetto pulizie           | 800       | 101         | 2022-07-01      | via Firenze, 12 | 3332223333 |

```
10 rows in set (0.00 sec)
```

## Tabella scarpe

### Creazione

```
mysql> CREATE TABLE scarpe (
```

```
-> id_scarpe INT PRIMARY KEY not null auto_increment,
```

```
-> prezzo FLOAT not null,
```

```
-> taglia INT not null,
```

```
-> marca VARCHAR(255) not null,
```

```
-> id_giocatore INT);
```

### Inserimenti

```
INSERT INTO scarpe (prezzo, taglia, marca)
```

```
VALUES (5.00, 43, "Nike");
```

### Sample

```
mysql> SELECT * FROM scarpe ORDER BY RAND() LIMIT 10;
```

| id_scarpe | prezzo | taglia | marca  |
|-----------|--------|--------|--------|
| 8286      | 5.5    | 40     | Adidas |
| 8267      | 5.5    | 41     | Adidas |
| 8298      | 5.5    | 40     | Adidas |
| 8274      | 5.5    | 45     | Adidas |
| 8259      | 5      | 43     | Nike   |
| 8296      | 5      | 38     | Nike   |
| 8385      | 6      | 41     | Puma   |
| 8390      | 6      | 41     | Puma   |
| 8320      | 5.5    | 39     | Adidas |
| 8231      | 5      | 39     | Nike   |

```
10 rows in set (0.00 sec)
```

## Tabella abbonamento

### Creazione

```
mysql> CREATE TABLE abbonamento (
```

- > id\_abbonamento INT PRIMARY KEY not null auto\_increment,
- > data\_inizio DATE not null,
- > data\_fine DATE not null,
- > tipologia VARCHAR(255) not null,
- > id\_giocatore INT,
- > FOREIGN KEY (id\_giocatore) REFERENCES giocatore(id\_giocatore)
- > on delete cascade on update cascade );

### Inserimenti

```
INSERT INTO abbonamento (data_inizio, data_fine, tipologia, id_giocatore)
```

```
VALUES ('2021-12-23', '2022-12-23', 'annuale', 4824889);
```

### Sample

```
mysql> SELECT * FROM abbonamento ORDER BY RAND() LIMIT 10;
```

| id_abbonamento | data_inizio | data_fine  | tipologia   | id_giocatore |
|----------------|-------------|------------|-------------|--------------|
| 92309          | 2020-05-28  | 2020-06-27 | mensile     | 408246       |
| 21358          | 2019-09-05  | 2019-10-05 | mensile     | 501775       |
| 103186         | 2018-04-07  | 2018-05-07 | mensile     | 505056       |
| 32401          | 2021-04-03  | 2022-04-03 | annuale     | 534039       |
| 27727          | 2022-12-07  | 2023-12-07 | annuale     | 433952       |
| 16590          | 2019-10-22  | 2020-10-21 | annuale     | 445706       |
| 93625          | 2021-12-23  | 2022-12-23 | annuale     | 482489       |
| 30561          | 2020-09-29  | 2020-12-28 | trimestrale | 511669       |
| 24754          | 2022-07-30  | 2022-08-29 | mensile     | 561786       |
| 10904          | 2018-03-12  | 2019-03-12 | annuale     | 545539       |

```
10 rows in set (0.09 sec)
```

## Tabella promozione

### Creazione

```
mysql> CREATE TABLE promozione (
```

- > id\_promo INT PRIMARY KEY not null auto\_increment,
- > nome VARCHAR(255) not null,
- > data\_inizio DATE not null,
- > data\_fine DATE not null,
- > percentuale\_sconto FLOAT not null,
- > descrizione VARCHAR(255) not null,
- > id\_impianto INT,
- > FOREIGN KEY (id\_impianto) REFERENCES impianto(id\_impianto)
- > on delete cascade on update cascade);

### Inserimenti

```
INSERT INTO promozione (nome, data_inizio, data_fine, percentuale_sconto, descrizione, id_impianto)
```

```
VALUES ('Promozione 2x1', '2022-05-01', '2022-05-31', 50, '2 ore di gioco al prezzo di 1', 101);
```

### Sample

```
mysql> SELECT * FROM promozione ORDER BY RAND() LIMIT 5;
```

| id_promo | nome                  | data_inizio | data_fine  | percentuale_sconto | descrizione  | id_impianto |
|----------|-----------------------|-------------|------------|--------------------|--|-------------|
| 100008   | Promozione 2x1        | 2022-05-01  | 2022-05-31 | 50                 | 2 ore di gioco al prezzo di 1                        | 101         |
| 100011   | Promozione per gruppi | 2022-08-01  | 2022-08-31 | 25                 | Sconti del 25% per gruppi di almeno 10 persone       | 101         |
| 100005   | Promozione 4          | 2022-03-01  | 2022-03-31 | 0.1                | Sconto del 10% su tutti i pasti al bar               | 103         |
| 100009   | Sconti per studenti   | 2022-06-01  | 2022-06-30 | 20                 | Sconti del 20% per gli studenti                      | 102         |
| 100012   | Promozione festiva    | 2022-09-01  | 2022-09-30 | 40                 | Sconti del 40% per i giochi durante i giorni festivi | 102         |

```
5 rows in set (0.00 sec)
```



## Tabella noleggio

### Creazione

```
mysql> CREATE TABLE noleggio (  
-> id_giocatore INT,  
-> id_scarpe INT,  
-> ora_inizio TIME,  
-> data DATE,  
-> consegnato BOOLEAN,  
-> PRIMARY KEY (id_giocatore, id_scarpe, ora_inizio, data),  
-> FOREIGN KEY (id_giocatore) REFERENCES giocatore(id_giocatore)  
-> ON DELETE CASCADE,  
-> FOREIGN KEY (id_scarpe) REFERENCES scarpe(id_scarpe)  
-> ON DELETE CASCADE);
```

### Inserimenti

```
INSERT INTO noleggio (id_giocatore, id_scarpe, ora_inizio, consegnato)  
VALUES (479054, 8294, 17:19, 1);
```

### Sample

| id_giocatore | id_scarpe | ora_inizio | data       | consegnato |
|--------------|-----------|------------|------------|------------|
| 548885       | 8256      | 15:11:00   | 2021-08-23 | 1          |
| 508685       | 8372      | 12:41:00   | 2022-06-30 | 1          |
| 419341       | 8312      | 11:22:00   | 2020-11-15 | 1          |
| 428273       | 8325      | 09:03:00   | 2020-11-24 | 0          |
| 524962       | 8273      | 13:01:00   | 2022-07-12 | 1          |
| 505373       | 8315      | 20:44:00   | 2022-06-24 | 1          |
| 421417       | 8406      | 12:38:00   | 2022-05-14 | 1          |
| 539824       | 8257      | 15:36:00   | 2022-09-18 | 1          |
| 599367       | 8416      | 14:26:00   | 2022-11-12 | 0          |
| 569892       | 8306      | 15:49:00   | 2020-12-01 | 0          |

## Tabella impianto\_inventario

### Creazione

```
CREATE TABLE impianto_inventario (  
    id_impianto INT,  
    id_inventario INT,  
    quantità INT,  
    PRIMARY KEY (id_impianto, id_inventario),  
    FOREIGN KEY (id_impianto) REFERENCES impianto(id_impianto)  
    ON DELETE cascade,  
    FOREIGN KEY (id_inventario) REFERENCES inventario(id_inventario)  
    ON DELETE CASCADE);
```

### Inserimenti

```
INSERT INTO impianto_inventario (id_impianto, id_inventario, quantità) VALUES (101, 1,  
10);
```

### Sample

| id_impianto | id_inventario | quantità |
|-------------|---------------|----------|
| 101         | 1             | 10       |
| 101         | 3             | 8        |
| 102         | 3             | 20       |
| 103         | 7             | 10       |
| 102         | 5             | 10       |
| 102         | 7             | 11       |
| 101         | 7             | 10       |
| 102         | 2             | 50       |
| 103         | 3             | 16       |
| 102         | 4             | 8        |

## Query significative

### Abbonamenti

Seleziona i nomi dei giocatori che hanno un abbonamento in corso.

```
SELECT nome, cognome
```

```
FROM giocatore
```

```
JOIN abbonamento ON giocatore.id_giocatore = abbonamento.id_giocatore
```

```
WHERE abbonamento.data_fine >= curdate();
```

Di seguito mostro una parte dei dati restituiti dalla query:

|          |              |  |
|----------|--------------|--|
| Ida      | Ludovisi     |  |
| Fausto   | Anichini     |  |
| Lisa     | Sommaruga    |  |
| Elmo     | Calgari      |  |
| Milena   | Jovinelli    |  |
| Ivan     | Vendetti     |  |
| Valeria  | Milanesi     |  |
| Ilaria   | Marangoni    |  |
| Dionigi  | Chigi        |  |
| Dionigi  | Guicciardini |  |
| Sylvia   | Avogadro     |  |
| Luisa    | Giannuzzi    |  |
| Morena   | Rismondo     |  |
| Paride   | Aporti       |  |
| Dionigi  | Capone       |  |
| Venancio | Chigi        |  |

+-----+-----+

8823 rows in set (0.05 sec)

## Dipendenti di un impianto

Mostra tutti i dipendenti di un impianto specifico

```
SELECT nome, cognome, ruolo FROM dipendente  
WHERE id_impianto = 101;
```

| nome       | cognome       | ruolo                     |
|------------|---------------|---------------------------|
| John       | Doe           | Manager                   |
| Mario      | Rossi         | cassiere                  |
| Nico       | Pedrazzini    | addetto alla manutenzione |
| Marisa     | Petrucchi     | addetto pulizie           |
| Renata     | Calbo         | amministratore            |
| Verdiana   | Bonatti       | addetto all'inventario    |
| Marco      | Caetani       | Addetto pulizie           |
| Sebastiano | Borromeo      | Manager                   |
| Arnulfo    | Onio          | Addetto alla manutenzione |
| Alfredo    | Gregori       | Addetto alla manutenzione |
| Pellegrino | Tebaldi       | Addetto alle vendite      |
| Ida        | Majorana      | Addetto all'inventario    |
| Panfilo    | Gualandi      | Addetto alla sicurezza    |
| Vincentio  | Murri         | Amministratore            |
| Morena     | Zaguri        | Amministratore            |
| Ettore     | Vezzali       | Addetto alle vendite      |
| Azeglio    | Cavanna       | Addetto alla sicurezza    |
| Luchino    | Luria         | Addetto all'inventario    |
| Paolo      | Pisani        | Manager                   |
| Mariano    | Ungaretti     | Manager                   |
| Eugenia    | Tarantini     | Addetto all'inventario    |
| Gioachino  | Fantozzi      | Amministratore            |
| Cirillo    | Brambilla     | Pubbliche relazioni       |
| Patrizio   | Magnani       | Pubbliche relazioni       |
| Pierangelo | Trombetta     | Addetto alla sicurezza    |
| Amanda     | Iannuzzi      | Manager                   |
| Stefano    | Boezio        | Addetto alla manutenzione |
| Giosue     | Campise       | Addetto alla sicurezza    |
| Barbara    | Soranzo       | Manager                   |
| Franco     | Balbi         | Addetto pulizie           |
| Alfredo    | Guarato       | Addetto pulizie           |
| Giuliano   | Rossini       | Manager                   |
| Eliana     | Morpurgo      | Amministratore            |
| Severino   | Boitani       | Addetto alla sicurezza    |
| Baccio     | Cristoforetti | Addetto alla manutenzione |
| Annunziata | Cutuli        | Addetto all'inventario    |
| Ruggero    | Comeriato     | Addetto pulizie           |
| Gabriella  | Gregorio      | Pubbliche relazioni       |
| Olga       | Chinnici      | Addetto alla sicurezza    |
| Romeo      | Foscari       | Addetto alle vendite      |
| Tiziano    | Fallaci       | Addetto alla manutenzione |
| Ugolino    | Pagliaro      | Addetto alla manutenzione |
| Venancio   | Sagese        | Addetto alla sicurezza    |
| Luciana    | Canevascini   | Pubbliche relazioni       |
| Fabrizia   | Gilardoni     | Addetto alle vendite      |

45 rows in set (0.02 sec)

### Media dei punteggi

Mostra la media dei punteggi ottenuti dai giocatori di un impianto specifico.

```
SELECT AVG(p.punteggio) as media_punteggio
FROM partita p
JOIN pista pi ON p.id_pista = pi.id_pista
JOIN impianto i ON pi.id_impianto = i.id_impianto
WHERE i.id_impianto = 101;
```

```
+-----+
| media_punteggio |
+-----+
|          149.9133 |
+-----+
1 row in set (0.11 sec)
```

### Promozioni attive

Mostra tutte le promozioni attive in un determinato periodo:

```
SELECT nome, percentuale_sconto FROM promozione
WHERE data_inizio >= '2022-01-01' AND data_fine <= '2022-04-30'
order by percentuale_sconto;
```

```
+-----+-----+
| nome          | percentuale_sconto |
+-----+-----+
| Promozione 4  | 0.1                |
| Promozione 2  | 0.15               |
| Promozione 3  | 0.2                |
| Promozione 5  | 0.3                |
+-----+-----+
4 rows in set (0.03 sec)
```

### Partite del giorno

Seleziona il nome dell'impianto, la data e l'ora di inizio di tutte le partite giocate in un determinato giorno.

```
SELECT DISTINCT impianto.nome, partita.data, partita.ora_inizio
FROM impianto
JOIN pista ON impianto.id_impianto = pista.id_impianto
JOIN partita ON pista.id_pista = partita.id_pista
WHERE partita.data = '2022-03-11';
```

|  |       |   |  |            |  |          |  |
|--|-------|---|--|------------|--|----------|--|
|  | Spare | 3 |  | 2022-03-11 |  | 10:12:00 |  |
|  | Spare | 3 |  | 2022-03-11 |  | 10:42:00 |  |
|  | Spare | 3 |  | 2022-03-11 |  | 10:47:00 |  |
|  | Spare | 3 |  | 2022-03-11 |  | 12:18:00 |  |
|  | Spare | 3 |  | 2022-03-11 |  | 13:50:00 |  |
|  | Spare | 3 |  | 2022-03-11 |  | 13:55:00 |  |
|  | Spare | 3 |  | 2022-03-11 |  | 16:26:00 |  |
|  | Spare | 3 |  | 2022-03-11 |  | 18:06:00 |  |
|  | Spare | 3 |  | 2022-03-11 |  | 18:34:00 |  |
|  | Spare | 3 |  | 2022-03-11 |  | 19:26:00 |  |
|  | Spare | 3 |  | 2022-03-11 |  | 19:38:00 |  |
|  | Spare | 3 |  | 2022-03-11 |  | 21:12:00 |  |
|  | Spare | 3 |  | 2022-03-11 |  | 21:34:00 |  |

+-----+-----+-----+

186 rows in set (0.07 sec)

### Le partite di Mario

Seleziona il nome e il punteggio di tutte le partite giocate da un giocatore con un determinato nome.

```
SELECT giocatore.nome, giocatore.cognome, partita.punteggio
FROM giocatore
JOIN partita ON giocatore.id_giocatore = partita.id_giocatore
WHERE giocatore.nome = 'Mario';
```

|  |       |  |            |  |     |  |
|--|-------|--|------------|--|-----|--|
|  | Mario |  | Malacarne  |  | 179 |  |
|  | Mario |  | Marsili    |  | 243 |  |
|  | Mario |  | Marsili    |  | 95  |  |
|  | Mario |  | Gozzi      |  | 270 |  |
|  | Mario |  | Gozzi      |  | 222 |  |
|  | Mario |  | Gozzi      |  | 117 |  |
|  | Mario |  | Luna       |  | 52  |  |
|  | Mario |  | Luna       |  | 87  |  |
|  | Mario |  | Galtarossa |  | 189 |  |

+-----+  
468 rows in set (0.07 sec)

## Abbonamenti attivi

Seleziona i giocatori che hanno un abbonamento attivo in un determinato periodo di tempo:

```
SELECT DISTINCT giocatore.nome, giocatore.cognome
```

```
FROM giocatore
```

```
JOIN abbonamento ON giocatore.id_giocatore = abbonamento.id_giocatore
```

```
WHERE abbonamento.data_inizio >= '2022-01-01' AND abbonamento.data_fine <= '2023-01-01';
```

|  |           |  |           |  |
|--|-----------|--|-----------|--|
|  | Serena    |  | Bottaro   |  |
|  | Giulia    |  | Tarantino |  |
|  | Berenice  |  | Boccaccio |  |
|  | Napoleone |  | Buonauro  |  |
|  | Martino   |  | Garrone   |  |
|  | Silvestro |  | Volterra  |  |
|  | Simonetta |  | Perozzo   |  |
|  | Gianmarco |  | Vespa     |  |
|  | Fredo     |  | Vivaldi   |  |
|  | Donna     |  | Pirelli   |  |
|  | Morena    |  | Gozzano   |  |
|  | Severino  |  | Scotto    |  |
|  | Lucio     |  | Ughi      |  |

+-----+  
10918 rows in set (0.06 sec)

## Classifica

Seleziona i primi 10 giocatori con il punteggio più alto tra tutti gli impianti.

```
SELECT giocatore.nome, giocatore.cognome, max(partita.punteggio) as punteggio FROM  
giocatore
```

```
JOIN partita ON giocatore.id_giocatore = partita.id_giocatore
```

```
GROUP BY giocatore.nome, giocatore.cognome
```

```
ORDER BY punteggio DESC LIMIT 10;
```

| nome       | cognome    | punteggio |
|------------|------------|-----------|
| Emilio     | Prada      | 300       |
| Orlando    | Gentili    | 300       |
| Ubaldo     | Pistoletto | 300       |
| Nicolo     | Leonardi   | 300       |
| Lamberto   | Viviani    | 300       |
| Gioachino  | Crispi     | 300       |
| Ferdinando | Rosmini    | 300       |
| Arturo     | Roth       | 300       |
| Hugo       | Giammusso  | 300       |
| Elmo       | Travaglio  | 300       |

10 rows in set (0.71 sec)



### Giocatori affezionati

Trova i nomi e i cognomi dei giocatori che hanno giocato più di 7 partite.

```
SELECT nome, cognome
```

```
FROM giocatore
```

```
JOIN partita ON partita.id_giocatore = giocatore.id_giocatore
```

```
GROUP BY giocatore.id_giocatore
```

```
HAVING COUNT(partita.id_giocatore) > 7;
```

```
+-----+-----+
| nome   | cognome |
+-----+-----+
| Giosue  | Chindamo |
| Romina  | Combi    |
| Daria   | Einaudi  |
| Orlando | Naccari  |
| Martino | Bocelli  |
| Pietro  | Pagliaro |
+-----+-----+
6 rows in set (0.96 sec)
```

### Query 10

Trova le scarpe utilizzate da un determinato giocatore.

```
SELECT scarpe.marca, scarpe.taglia
```

```
FROM scarpe
```

```
JOIN noleggio ON scarpe.id_scarpe = noleggio.id_scarpe
```

```
WHERE noleggio.id_giocatore = 547254;
```

```
+-----+-----+
| marca | taglia |
+-----+-----+
| Nike  | 36     |
+-----+-----+
1 row in set (0.00 sec)
```

### Addetti all'inventario

Trova tutti gli addetti all'inventario che lavorano in un determinato impianto:

```
SELECT dipendente.nome, dipendente.cognome
```

```
FROM dipendente
```

```
WHERE dipendente.ruolo = "addetto all'inventario" AND dipendente.id_impianto = 101;
```

| nome       | cognome   |
|------------|-----------|
| Verdiana   | Bonatti   |
| Ida        | Majorana  |
| Luchino    | Luria     |
| Eugenia    | Tarantini |
| Annunziata | Cutuli    |

5 rows in set (0.00 sec)

### Scarpe 42

Trova tutti i giocatori che hanno acquistato scarpe della taglia '42'

```
SELECT nome, cognome
```

```
FROM giocatore
```

```
JOIN noleggio ON giocatore.id_giocatore = noleggio.id_giocatore
```

```
JOIN scarpe ON noleggio.id_scarpe = scarpe.id_scarpe
```

```
WHERE scarpe.taglia = '42';
```

| nome     | cognome    |
|----------|------------|
| Tiziano  | Gussoni    |
| Agnolo   | Comboni    |
| Ezio     | Ottino     |
| Federica | Cavanna    |
| Licia    | Martinelli |
| Nina     | Carosone   |
| Ronaldo  | Leonetti   |
| Alfio    | Oscuro     |
| Ruggero  | Favata     |
| Eleanora | Argan      |

10 rows in set (0.00 sec)

### Ricerca di un articolo

Trova tutti gli impianti che hanno un determinato articolo in inventario;

```
SELECT impianto.nome, inventario.descrizione, quantità
```

```
FROM impianto
```

```
JOIN impianto_inventario ON impianto.id_impianto = impianto_inventario.id_impianto
```

```
JOIN inventario ON impianto_inventario.id_inventario = inventario.id_inventario
```

```
WHERE inventario.descrizione = 'Palle da bowling';
```

| nome    | descrizione      | quantità |
|---------|------------------|----------|
| Spare 1 | Palle da bowling | 10       |
| Spare 2 | Palle da bowling | 7        |
| Spare 3 | Palle da bowling | 12       |

3 rows in set (0.03 sec)

### Piste disponibili

Trova tutte le piste disponibili in un determinato impianto:

```
SELECT id_pista
```

```
FROM pista
```

```
WHERE id_impianto = 101 AND disponibile = 1;
```

| id_pista |
|----------|
| 1        |
| 2        |
| 3        |
| 4        |
| 5        |

5 rows in set (0.01 sec)

### Abbonamenti in scadenza

Trova tutti i giocatori con un abbonamento in scadenza:

```
SELECT giocatore.nome, giocatore.cognome, abbonamento.data_fine
FROM giocatore
JOIN abbonamento ON giocatore.id_giocatore = abbonamento.id_giocatore
WHERE abbonamento.data_fine >= CURDATE() AND abbonamento.data_fine <=
DATE_ADD(CURDATE(), INTERVAL 15 DAY);
```

|           |            |            |  |
|-----------|------------|------------|--|
| Graziella | Donatoni   | 2023-02-16 |  |
| Sabatino  | Gozzano    | 2023-02-19 |  |
| Ilaria    | Pigafetta  | 2023-02-15 |  |
| Nedda     | Pedrazzini | 2023-02-19 |  |
| Elisa     | Santorio   | 2023-02-04 |  |
| Mario     | Boccaccio  | 2023-02-06 |  |
| Melania   | Polani     | 2023-02-17 |  |
| Lazzaro   | Basso      | 2023-02-18 |  |
| Ansaldo   | Stefanelli | 2023-02-08 |  |
| Fausto    | Anichini   | 2023-02-11 |  |
| Ilaria    | Marangoni  | 2023-02-18 |  |
| Dionigi   | Chigi      | 2023-02-16 |  |

+-----+-----+-----+-----+

849 rows in set (0.03 sec)

### Partite in compagnia

Seleziona tutti i giocatori che hanno giocato le stesse partite di un certo giocatore.

```
SELECT x.id_giocatore
FROM partita x JOIN partita y on x.id_pista = y.id_pista and
x.ora_inizio = y.ora_inizio and x.ora_fine = y.ora_fine and x.data = y.data
WHERE y.id_giocatore = 424432
and x.id_giocatore != y.id_giocatore
GROUP BY x.id_giocatore;
```

|              |
|--------------|
| +-----+      |
| id_giocatore |
| +-----+      |
| 451718       |
| +-----+      |

1 row in set (0.01 sec)

## Partite in contemporanea

Seleziona tutte le partite che sono avvenute contemporaneamente a una partita.

```
SELECT x.id_pista, x.id_giocatore, x.ora_inizio, x.ora_fine, x.data
FROM partita x, partita y
WHERE x.id_pista = y.id_pista and x.ora_inizio = y.ora_inizio and x.ora_fine = y.ora_fine
and x.data = y.data
and x.id_giocatore != y.id_giocatore
GROUP by x.id_pista, x.id_giocatore, x.ora_inizio, x.ora_fine, x.data;
```

|  |    |  |        |  |          |  |          |  |            |  |
|--|----|--|--------|--|----------|--|----------|--|------------|--|
|  | 6  |  | 424432 |  | 21:54:00 |  | 21:59:00 |  | 2022-07-06 |  |
|  | 6  |  | 451718 |  | 21:54:00 |  | 21:59:00 |  | 2022-07-06 |  |
|  | 12 |  | 405580 |  | 21:56:00 |  | 21:59:00 |  | 2021-05-30 |  |
|  | 12 |  | 546028 |  | 21:56:00 |  | 21:59:00 |  | 2021-05-30 |  |
|  | 11 |  | 461319 |  | 21:58:00 |  | 21:59:00 |  | 2020-06-14 |  |
|  | 11 |  | 511126 |  | 21:58:00 |  | 21:59:00 |  | 2020-06-14 |  |
|  | 4  |  | 461348 |  | 21:58:00 |  | 21:58:00 |  | 2021-11-24 |  |
|  | 4  |  | 542452 |  | 21:58:00 |  | 21:58:00 |  | 2021-11-24 |  |
|  | 11 |  | 439792 |  | 21:59:00 |  | 21:59:00 |  | 2020-06-22 |  |
|  | 11 |  | 515340 |  | 21:59:00 |  | 21:59:00 |  | 2020-06-22 |  |
|  | 15 |  | 494402 |  | 21:59:00 |  | 21:59:00 |  | 2022-03-18 |  |
|  | 15 |  | 570944 |  | 21:59:00 |  | 21:59:00 |  | 2022-03-18 |  |

+-----+-----+-----+-----+-----+  
34 rows in set (1.07 sec)

## Query algebra relazionale

### Partire giocate da Mario

Seleziona il nome e il punteggio di tutte le partite giocate da un giocatore con un determinato nome.

$\Pi(\text{nome}, \text{cognome}, \text{punteggio}(\text{giocatore} \mid X \mid \text{id\_giocatore} = \text{id\_giocatore} \mid X \mid \text{partita} (\sigma_{\text{nome}='Mario'}(\text{giocatore}))))$

### Scarpe utilizzate da un giocatore

Trova le scarpe utilizzate da un determinato giocatore.

$\Pi(\text{marca}, \text{taglia}(\text{scarpe} \mid X \mid \text{id\_scarpe} = \text{id\_scarpe} \mid X \mid \text{noleggior} (\sigma_{\text{id\_giocatore}=810}(\text{noleggior}))))$

## Stored Procedure

A seguire saranno presentate le Stored Procedure d'esempio create.

### Controllo disponibilità scarpe

Verifica che un numero di scarpe sia disponibile per il noleggio, nel caso in cui non siano disponibili viene mandato un messaggio di errore

```
CREATE PROCEDURE noleggia_scarpe (IN id_giocatore INT, IN id_scarpe INT, IN ora_inizio  
TIME, IN data DATE)  
  
BEGIN  
  
DECLARE scarpe_disponibili INT;  
  
  
SELECT COUNT(*) INTO scarpe_disponibili  
  
FROM noleggio  
  
WHERE id_giocatore = id_giocatore AND id_scarpe = id_scarpe AND ora_inizio =  
ora_inizio AND data = data AND consegnato = FALSE;  
  
  
IF scarpe_disponibili = 0 THEN  
  
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Numero di scarpe non disponibile per il  
noleggio.';  
  
ELSE  
  
INSERT INTO noleggio (id_giocatore, id_scarpe, ora_inizio, data, consegnato)  
VALUES (id_giocatore, id_scarpe, ora_inizio, data, FALSE);  
  
END IF;  
  
END
```

### Aggiunta di un nuovo giocatore

La seguente stored procedure viene utilizzata per facilitare l'aggiunta di un nuovo giocatore:

```
CREATE PROCEDURE aggiungiGiocatore (  
    IN nome VARCHAR(255),  
    IN cognome VARCHAR(255),  
    IN data_nascita DATE,  
    IN email VARCHAR(255),  
    IN telefono VARCHAR(255))  
BEGIN  
    INSERT INTO giocatore (nome, cognome, data_nascita, email, telefono)  
    VALUES (nome, cognome, data_nascita, email, telefono);  
END;
```

### Aggiunta di un nuovo dipendente

La seguente Stored Procedure viene utilizzata per facilitare l'aggiunta di un nuovo dipendente.

```
CREATE PROCEDURE aggiungiDipendente (  
    IN nome VARCHAR(255),  
    IN cognome VARCHAR(255),  
    IN ruolo VARCHAR(255),  
    IN stipendio INT,  
    IN data_assunzione DATE,  
    IN indirizzo VARCHAR(255),  
    IN telefono VARCHAR(255),  
    IN id_impianto INT)  
BEGIN  
    INSERT INTO dipendente (nome, cognome, ruolo, stipendio, data_assunzione,  
    indirizzo, telefono, id_impianto)  
    VALUES (nome, cognome, ruolo, stipendio, data_assunzione, indirizzo, telefono,  
    id_impianto);  
END;
```



### Aggiunta di una nuova partita

La seguente Stored Procedure viene utilizzata per facilitare l'aggiunta di una nuova partita.

```
CREATE PROCEDURE aggiungiPartita (  
    IN ora_inizio TIME,  
    IN data DATE,  
    IN id_giocatore INT,  
    IN punteggio INT,  
    IN ora_fine TIME,  
    IN id_pista INT  
)  
BEGIN  
    INSERT INTO partita (ora_inizio, data, id_giocatore, punteggio, ora_fine, id_pista)  
    VALUES (ora_inizio, data, id_giocatore, punteggio, ora_fine, id_pista);  
END;
```

### Piste disponibili in un determinato impianto

La seguente Stored Procedure viene utilizzata per verificare quali sono le piste disponibili in un determinato impianto.

```
PROCEDURE `GetAvailableLanes` (IN impiantoid INT)  
BEGIN  
    SELECT id_pista FROM pista  
    WHERE id_impianto = impiantoid AND disponibile = 1;  
END
```

## Trigger

Al fine di preservare la consistenza della base di dati ed automatizzare alcune operazioni, sono stati inseriti i seguenti trigger.

### Controllo nuovo abbonato

Questo trigger viene attivato prima dell'inserimento di un nuovo record nella tabella abbonamento. L'obiettivo è quello di vincolare un giocatore a non fare più di un abbonamento fino a che quest'ultimo non è scaduto;

```
mysql> CREATE TRIGGER controllo_abbonato
-> BEFORE INSERT abbonamento
-> FOR EACH ROW
-> BEGIN
->   DECLARE numero_iscrizioni INT;
->   SELECT COUNT(*) INTO numero_iscrizioni
->   FROM abbonamento
->   WHERE id_giocatore = NEW.id_giocatore
->   AND data_fine >= NEW.data_inizio;
->   IF numero_iscrizioni >= 1 THEN
->     SIGNAL SQLSTATE '45000'
->     SET MESSAGE_TEXT = 'Utente già abbonato';
->   END IF;
-> END;
```

Verifichiamo il funzionamento del trigger nel caso tentiamo di effettuare un inserimento nella tabella abbonamento per un giocatore già abbonato.

```
mysql> INSERT INTO abbonamento (data_inizio, data_fine, tipologia, id_giocatore)
-> VALUES ('2023-01-10', '2023-02-09', 'mensile', 566615);
ERROR 1644 (45000): Utente già abbonato
```

## Controllo prenotazione

Il seguente trigger evita di aggiungere una nuova partita se quest'ultima inizia alla stessa ora di un'altra partita già in corso, ovvero, evita di avere delle sovrapposizioni generando un messaggio di errore.

```
CREATE TRIGGER controllo_prenotazione
AFTER INSERT ON partita
FOR EACH ROW
BEGIN
IF EXISTS (SELECT * FROM partita
WHERE id_pista = NEW.id_pista
AND ora_inizio <= NEW.ora_inizio
AND ora_fine > NEW.ora_inizio
AND data = NEW.data) THEN
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = ' Prenotazione non disponibile, la pista è
occupata per l'orario indicato';
END IF;
END;
```

```
mysql> INSERT INTO partita (ora_inizio, data, id_giocatore, punteggio, ora_fine, id_pista)
-> VALUES ('12:30:00', '2020-07-25', 418060, 241, '17:51:00', 12);
ERROR 1644 (45000): Prenotazione non disponibile, la pista è occupata per l'orario indicato
```

## Viste

### Vista scarpe noleggiate

Restituisce le scarpe noleggiate dai giocatori e indica se sono state riconsegnate.

```
CREATE VIEW vista_scarpe_noleggiate AS
```

```
SELECT scarpe.id_scarpe, prezzo, taglia, marca, noleggio.id_giocatore, ora_inizio, data,  
consegnato
```

```
FROM scarpe
```

```
JOIN noleggio
```

```
ON scarpe.id_scarpe = noleggio.id_scarpe;
```

### Vista dipendenti

Mostra i dipendenti di un impianto specifico.

```
CREATE VIEW vista_dipendenti AS
```

```
SELECT id_dipendente, nome, cognome, ruolo, stipendio, data_assunzione, indirizzo,  
telefono, id_impianto
```

```
FROM dipendente
```

```
WHERE id_impianto = 101;
```

### Vista partite della settimana

Mostra le partite dell'ultima settimana.

```
CREATE VIEW vista_partita_giocatore AS
```

```
SELECT giocatore.id_giocatore, nome, cognome, data_nascita, email, telefono,  
ora_inizio, data
```

```
FROM giocatore
```

```
JOIN partita
```

```
ON giocatore.id_giocatore = partita.id_giocatore
```

```
WHERE data >= DATE_SUB(CURDATE(), INTERVAL 7 DAY);
```

## Ottimizzazione query

Tutte le query svolte sono già performanti in quanto i tempi di esecuzione, dopo l'inserimento del numero di dati riportati nella tabella sottostante, sono notevolmente bassi. Fatta la premessa andiamo a vedere e analizzare le query meno veloci.

Le ottimizzazioni apportate al database sono state effettuate sul seguente numero di record per tabella.

| Tabella             | Numero record |
|---------------------|---------------|
| Abbonamento         | 100.000       |
| Dipendente          | 100.118       |
| Giocatore           | 200.001       |
| Impianto            | 3             |
| Impianto_Inventario | 18            |
| Inventario          | 13            |
| Noleggio            | 200           |
| Partita             | 227.646       |
| Pista               | 15            |
| Promozione          | 13            |
| Scarpe              | 200           |

## Considerazioni su InnoDB Buffer

Eseguendo le stesse query ripetutamente ho riscontrato tempi di accesso differenti tra le prime esecuzioni e le successive. Questo grazie a dei meccanismi di cache che possono alterare i tempi di esecuzione di una query.

InnoDB mantiene un'area in memoria chiamata buffer pool per il salvataggio di dati ed indici a scopo di cache. La cache consente di limitare il numero di accesso al disco velocizzando l'accesso alle informazioni.

### Tempi impiegati dalle query

Nella seguente tabella sono riportati i tempi a confronto tra le query eseguite la prima volta e le stesse eseguite la seconda volta per mostrare l'uso del buffer.

| Query                     | Tempo 1 ° esecuzione | Tempo 2 ° esecuzione |
|---------------------------|----------------------|----------------------|
| Abbonamenti               | 0.05                 | 0.04                 |
| Dipendenti di un impianto | 0.39                 | 0.06                 |
| Media dei punteggi        | 0.15                 | 0.10                 |
| Promozioni attive         | 0.01                 | 0.00                 |
| Partite del giorno        | 0.07                 | 0.06                 |
| Le partite di Mario       | 0.07                 | 0.06                 |
| Abbonamenti attivi        | 0.06                 | 0.05                 |
| Classifica                | 0.90                 | 0.71                 |
| Giocatori affezionati     | 1.47                 | 1.21                 |
| Query 10                  | 0.02                 | 0.00                 |
| Addetti all'inventario    | 0.07                 | 0.06                 |
| Scarpe 42                 | 0.01                 | 0.00                 |
| Ricerca di un articolo    | 0.02                 | 0.00                 |
| Piste disponibili         | 0.01                 | 0.00                 |
| Abbonamenti in scadenza   | 0.03                 | 0.02                 |
| Partite in compagnia      | 0.02                 | 0.00                 |
| Partite in contemporanea  | 1.14                 | 1.10                 |

N.B. In giallo sono evidenziate le query che saranno ottimizzate

## Classifica

### Prima versione della query

```
SELECT giocatore.nome, giocatore.cognome, max(partita.punteggio) as punteggio FROM  
giocatore
```

```
JOIN partita ON giocatore.id_giocatore = partita.id_giocatore
```

```
GROUP BY giocatore.nome, giocatore.cognome
```

```
ORDER BY punteggio DESC LIMIT 10;
```

| nome       | cognome    | punteggio |
|------------|------------|-----------|
| Emilio     | Prada      | 300       |
| Orlando    | Gentili    | 300       |
| Ubaldo     | Pistoletto | 300       |
| Nicolo     | Leonardi   | 300       |
| Lamberto   | Viviani    | 300       |
| Gioachino  | Crispi     | 300       |
| Ferdinando | Rosmini    | 300       |
| Arturo     | Roth       | 300       |
| Hugo       | Giammusso  | 300       |
| Elmo       | Travaglio  | 300       |

10 rows in set (0.90 sec)

La query non dà risultati particolarmente efficienti.

Come è possibile osservare dall'*explain* la dicitura *using temporary* indica che sta creando una tabella temporanea per gestire i risultati intermedi della query. Questo perché ho incluso l'istruzione *group by*.

La dicitura *using filesort* indica che sta utilizzando un ordinamento temporaneo.

Questo può significare che la query non sta utilizzando un indice adeguato per ottenere i risultati in modo efficiente.

| id | select_type | table     | partitions | type   | possible_keys         | key     | key_len | ref                          | rows   | filtered | Extra                           |
|----|-------------|-----------|------------|--------|-----------------------|---------|---------|------------------------------|--------|----------|---------------------------------|
| 1  | SIMPLE      | partita   | NULL       | ALL    | idx_giocatore_partita | NULL    | NULL    | NULL                         | 202105 | 100.00   | Using temporary; Using filesort |
| 1  | SIMPLE      | giocatore | NULL       | eq_ref | PRIMARY               | PRIMARY | 4       | bowling.partita.id_giocatore | 1      | 100.00   | NULL                            |

Quindi ho optato per una creazione di un indice sui campi id\_giocatore e punteggio della tabella partita forzandone l'uso, in particolare è stato aggiunto il seguente indice:

```
CREATE INDEX idx_giocatore_partita_punteggio ON partita (id_giocatore, punteggio);
```

#### Versione ottimizzata

```
SELECT giocatore.nome, giocatore.cognome, max(partita.punteggio) as punteggio
```

```
FROM giocatore
```

```
JOIN partita USE INDEX (idx_giocatore_partita_punteggio) ON giocatore.id_giocatore =  
partita.id_giocatore
```

```
GROUP BY giocatore.nome, giocatore.cognome
```

```
ORDER BY punteggio DESC
```

```
LIMIT 10;
```

Come si può notare l'utilizzo di un nuovo indice ha aumentato le prestazioni riducendo i tempi di esecuzione della query.

```
+-----+-----+-----+  
| nome      | cognome    | punteggio |  
+-----+-----+-----+  
| Lairetta  | Mantegna   | 300       |  
| Roman     | Corradi    | 300       |  
| Susanna   | Venditti   | 300       |  
| Gionata   | Pedrazzini | 300       |  
| Aria      | Gagliano   | 300       |  
| Claudia   | Palladio   | 300       |  
| Rosario   | Tonnaseo   | 300       |  
| Salvi     | Calgari    | 300       |  
| Elmo      | Pedrazzini | 300       |  
| Alessia   | Pigafetta  | 300       |  
+-----+-----+-----+  
10 rows in set (0.48 sec)
```

Nella tabella sottostante riporto i tempi ottenuti prima e dopo l'ottimizzazione.

| Versione            | Tempi |
|---------------------|-------|
| Senza uso di indici | 0.90  |
| Con uso di indici   | 0.48  |



## Ottimizzazione Giocatori affezionati

## Prima versione della query

```
SELECT nome, cognome
```

FROM giocatore

JOIN partita ON partita.id\_giocatore = giocatore.id\_giocatore

GROUP BY giocatore.id\_giocatore

HAVING COUNT(partita.id\_giocatore) > 7;

```

+-----+-----+
| nome   | cognome |
+-----+-----+
| Giosue | Chindamo |
| Romina | Combi    |
| Daria  | Einaudi  |
| Orlando | Naccari  |
| Martino | Bocelli  |
| Pietro | Pagliaro |
+-----+-----+
6 rows in set (1.47 sec)

```

La query non dà risultati particolarmente efficienti.

Come è possibile osservare dall'explain la dicitura `using index` indica che il database sta utilizzando un indice per eseguire la query. Questo significa che il database non deve accedere a tutte le righe richieste.

La dicitura `using temporary` indica che sta creando una tabella temporanea per gestire i risultati intermedi della query. Questo perché ho incluso l'istruzione `group by`.

In generale l'utilizzo di un indice e la creazione di una tabella temporanea possono migliorare le prestazioni. Tuttavia possono anche aumentare la complessità della query e richiedere più memoria.

| id                           | select_type | table     | partitions | type   | possible_keys         | key            | key_len | ref                          | rows   | filtered |
|------------------------------|-------------|-----------|------------|--------|-----------------------|----------------|---------|------------------------------|--------|----------|
| Extra                        |             |           |            |        |                       |                |         |                              |        |          |
| 1                            | SIMPLE      | partita   | NULL       | index  | idx_giocatore_partita | partita_ibfk_1 | 4       | NULL                         | 202185 | 100.00   |
| Using index; Using temporary |             |           |            |        |                       |                |         |                              |        |          |
| 1                            | SIMPLE      | giocatore | NULL       | eq_ref | PRIMARY               | PRIMARY        | 4       | bowling.partita.id_giocatore | 1      | 100.00   |
| NULL                         |             |           |            |        |                       |                |         |                              |        |          |

2 rows in set, 1 warning (0.00 sec)

Dunque ho optato per una creazione di un indice sul campo id\_giocatore della tabella partita forzandone l'uso, in particolare è stato aggiunto il seguente indice:

```
CREATE INDEX idx_giocatore_partita ON partita (id_giocatore);
```

Inoltre ho pensato di includere l'istruzione ORDER BY per abbassarne ulteriormente la complessità temporale.

### Versione ottimizzata

```
SELECT nome, cognome
```

```
FROM giocatore
```

```
JOIN partita USE INDEX (idx_giocatore_partita) ON partita.id_giocatore =  
giocatore.id_giocatore
```

```
GROUP BY giocatore.id_giocatore
```

```
HAVING COUNT(partita.id_giocatore) > 7
```

```
ORDER BY cognome;
```

Come si può notare l'utilizzo di un nuovo indice ha ridotto i tempi.

```
+-----+-----+  
| nome   | cognome |  
+-----+-----+  
| Martino | Bocelli |  
| Giosue  | Chindamo |  
| Romina  | Combi   |  
| Daria   | Einaudi |  
| Orlando | Naccari |  
| Pietro  | Pagliaro |  
+-----+-----+  
6 rows in set (0.68 sec)
```

Nella tabella sottostante riporto i tempi ottenuti prima e dopo l'ottimizzazione.

| Versione            | Tempi |
|---------------------|-------|
| Senza uso di indici | 1.47  |
| Con uso di indici   | 0.68  |