

# Query con documentazione

In questo file sono state raggruppate tutte le query utilizzate dalla piattaforma, con una descrizione della loro funzione

## Query di costruzione delle tabelle

### Costruzione di “users”

```
CREATE TABLE users(  
  
    id VARCHAR(36) PRIMARY KEY,  
  
    tipo NUMERIC(1) NOT NULL,  
  
    name VARCHAR(50) NOT NULL,  
  
    surname VARCHAR(50) NOT NULL,  
  
    nickname VARCHAR(50) NOT NULL UNIQUE,  
  
    email VARCHAR(50) NOT NULL UNIQUE,  
  
    password VARCHAR NOT NULL);
```

### Costruzione di “matches”

```
CREATE TABLE matches(  
  
    id VARCHAR(36) PRIMARY KEY,  
  
    date VARCHAR(10) NOT NULL,  
  
    time VARCHAR(10) NOT NULL);
```

### Costruzione di “manches”

```
CREATE TABLE manches(  
  
    number NUMERIC(1) NOT NULL,  
  
    id VARCHAR(36) REFERENCES matches (id) NOT NULL,  
  
    phrase VARCHAR(60) REFERENCES phrases (phrase) NOT NULL,  
  
    PRIMARY KEY (id, number));
```

## Costruzione di “phrases”

```
CREATE TABLE phrases (  
  
    phrase VARCHAR(60) PRIMARY KEY,  
  
    theme VARCHAR(60) NOT NULL);
```

## Costruzione di “manchejoiners”

```
CREATE TABLE manchejoiners(  
  
    id VARCHAR(36),  
  
    number NUMERIC(1),  
  
    idplayer VARCHAR(36) REFERENCES users (id),  
  
    observer BOOLEAN NOT NULL,  
  
    FOREIGN KEY (id, number) REFERENCES manches,  
  
    PRIMARY KEY (id, number, idplayer));
```

## Costruzione di “manchewinner”

```
CREATE TABLE manchewinners(  
  
    id VARCHAR(36),  
  
    number NUMERIC(1),  
  
    idplayer VARCHAR(36) REFERENCES users (id),  
  
    amount NUMERIC(6) NOT NULL,  
  
    FOREIGN KEY (id, number) REFERENCES manches,  
  
    PRIMARY KEY (id, number, idplayer));
```

## Costruzione di “matchwinner”

```
CREATE TABLE matchwinners(  
  
    idmatch VARCHAR(36) REFERENCES matches (id) NOT NULL,  
  
    idplayer VARCHAR(36) REFERENCES users (id) NOT NULL,  
  
    amount NUMERIC(6) NOT NULL,  
  
    PRIMARY KEY (idmatch, idplayer));
```

## Costruzione di “moves”

```
CREATE TABLE moves (  
  
    moveid VARCHAR(36) PRIMARY KEY,  
  
    id VARCHAR(36) REFERENCES users (id) NOT NULL,  
  
    movetype VARCHAR(10) NOT NULL,  
  
    outcome NUMERIC(6) NOT NULL,  
  
    idmanche VARCHAR(36) NOT NULL,  
  
    number NUMERIC(1) NOT NULL,  
  
    FOREIGN KEY (idmanche, number) REFERENCES manches);
```

## Query di aggiunta

### Aggiunta di un utente

```
INSERT INTO users (id, tipo, name, surname, nickname, email, password)  
VALUES ('id dell'utente', 1 se e' un admin o 0 altrimenti,  
    'nome', 'cognome', 'nickname', 'indirizzo email', 'password');
```

## Aggiunta di una partita

```
INSERT INTO matches (id,date,time)

VALUES ('id del match', 'yyyy/mm/dd del giorno della creazione',

'hh:mm:ss dell'ora della creazione');
```

## Aggiunta di una manche

```
INSERT INTO manches (number,id,phrase)

VALUES (numero della manche compreso da 1 a 5, 'l'id della partita',

'frase misteriosa associata alla manche');
```

## Aggiunta di una frase

```
INSERT INTO phrases (theme,phrase)

VALUES ('tema della frase','frase misteriosa');
```

## Aggiunta di un partecipante ad una manche

```
INSERT INTO manchejoiners (id,number,idplayer,observer)

VALUES ('id della partita',numero della manche tra 1 e 5,'id del

giocatore',true se è osservatore altrimenti false);
```

## Aggiunta di un vincitore di una manche

```
INSERT INTO manchewinners (id,number,idplayer,amount)

VALUES ('id della partita',numero della manche tra 1 e 5, 'id del

vincitore', quantita' di punti vinta);
```

## Aggiunta di un vincitore di una partita

```
INSERT INTO matchewinners (idmatch,idplayer,amount)

VALUES ('id della partita','id del vincitore',quantita' di punti

vinta);
```

## Aggiunta di una mossa

(i tipi di mossa e i risultati possibili sono indicati nella documentazione dello schema ER concettuale)

```
INSERT INTO moves (id,movetype,outcome,idmanche,number,moveid)
VALUES ('id del giocatore', 'il tipo di mossa',il risultato,'id della
partita',numero della manche tra 1 e 5,'id della mossa');
```

## Query di modifica

### Aggiornare i dati di un utente

(non e' necessario che tutti i dati vengano effettivamente modificati)

```
UPDATE users SET name = 'nuovo nome', surname = 'nuovo
cognome',nickname = 'nuovo nickname', password = 'nuova password'
WHERE id = 'l'id dell'utente';
```

### Eliminazione di una partita dal database

```
DELETE FROM matches WHERE id = 'id della partita da cancellare';
```

## Query di ricerca

### Ottenere l'utente per email

```
SELECT * FROM users WHERE email = 'indirizzo email dell'utente';
```

### Ottenere l'utente per il nickname

```
SELECT * FROM users WHERE nickname = 'nickname dell'utente';
```

### Ottenere l'utente per l'id

```
SELECT * FROM users WHERE id = 'l'id dell'utente';
```

### Ottenere l'utente che ha ottenuto il miglior punteggio in una manche vinta

```
SELECT * FROM users U JOIN manchewinners MW ON U.id = MW.idPlayer WHERE  
amount = (SELECT MAX(amount) FROM manchewinners);
```

### Ottenere l'utente che ha ottenuto il miglior punteggio in una partita vinta

```
SELECT * FROM users U JOIN matchwinners MW ON U.id = MW.idPlayer WHERE  
Amount = (SELECT MAX(amount) FROM matchwinners);
```

### Ottenere l'utente che ha giocato più manche

```
SELECT nickname, COUNT(U.id) AS count FROM users U  
JOIN manchejoiners MJ ON U.id = MJ.idPlayer GROUP BY U.id ORDER BY  
count desc
```

### Ottenere l'utente con la media di punti per manche più alta

```
SELECT U.nickname, AVG(MW.amount) AS avg FROM users U JOIN  
manchewinners MW ON U.id = MW.idPlayer GROUP BY U.id ORDER BY avg desc;
```

### Ottenere l'utente che ha perso più volte il proprio turno in seguito ad errori

```

SELECT nickname, COUNT(U.id) AS count FROM users U                                JOIN
manchejoiners MW ON U.id = MW.idplayer                                          JOIN
manches MT ON MT.number = MW.number AND MT.id = MW.id                          JOIN
moves M ON M.number = MT.number AND MT.id = M.idmanche AND U.id = M.id          WHERE
M.outcome = 0 AND M.movetype <> 'perde' AND M.movetype <> 'passa' GROUP
BY U.id ORDER BY count desc;

```

## Ottenere l'utente che ha ottenuto dalla ruota 'PERDE' più volte

```

SELECT nickname, COUNT(U.id) AS count FROM users U
JOIN manchejoiners MW ON U.id = MW.idplayer JOIN
manches MT ON MT.number = MW.number AND MT.id = MW.id JOIN
moves M ON M.number = MT.number AND MT.id = M.idmanche AND U.id = M.id
WHERE
M.movetype = 'perde' GROUP BY U.id ORDER BY count desc;

```

## Ottenere le frasi non viste dai tre giocatori

```

SELECT * FROM phrases WHERE phrase NOT IN
(SELECT phrase FROM Manches M JOIN MancheJoiners MJ ON M.id=MJ.id AND
M.number = MJ.number
WHERE idPlayer = 'id del primo giocatore' OR idPlayer = 'id del secondo
giocatore' OR idPlayer = 'id del terzo giocatore' GROUP BY
phrase);

```

## Ottenere il numero di manche giocate da un utente

```

SELECT COUNT(*) AS count FROM manchejoiners WHERE
idplayer = 'id del giocatore' AND observer = false;

```

## Ottenere il numero di partite giocato da un utente

```

SELECT COUNT(*) AS count FROM
(SELECT id FROM manchejoiners WHERE idplayer = 'id del giocatore' AND
observer = false GROUP BY id) as matches;

```

## Ottenere il numero di manche osservate da un utente

```

SELECT COUNT(*) AS count FROM manchejoiners WHERE
idplayer = 'id dell'utente' AND observer = true;

```

## Ottenere il numero di partite osservate da un utente

```

SELECT COUNT(*) AS count FROM
(SELECT id FROM manchejoiners WHERE idplayer = 'id del giocatore' AND
observer = true GROUP BY id) as matches;

```

## Ottenere il numero di manche vinte (quindi concluse)

```
SELECT COUNT(*) AS count FROM manchewinners
```

## Ottenere il numero di manche vinte da un utente

```
SELECT COUNT(*) AS count FROM manchewinners  
WHERE idplayer = 'id del giocatore';
```

## Ottenere il numero di partite vinte da un giocatore

```
SELECT COUNT(*) AS count FROM matchwinners  
WHERE  
idplayer = 'id del giocatore';
```

## Ottenere tutte le partite vinte da un utente

```
SELECT * FROM matchwinners WHERE idplayer = 'id del giocatore';
```

## Ottenere la mossa che permesso di guadagnare la maggiore quantità di punti, insieme al giocatore che ha effettuato la mossa e la frase misteriosa associata

```
SELECT * FROM users M JOIN moves U ON M.id = U.id  
manches MT ON U.idmanche = MT.id AND U.number = MT.number  
phrases PT ON MT.phrase = PT.phrase  
WHERE  
U.outcome = (SELECT MAX(outcome) FROM moves);
```

## Ottenere il numero totale di mosse eseguite in manches in cui e' stata data la soluzione

(da usare insieme al numero di manche vinte per ottenere il numero medio di mosse necessarie ad indovinare la frase misteriosa)

```
SELECT COUNT(*) AS count FROM moves MT  
  
JOIN manches MAN ON MT.idmanche = MAN.id AND MT.number = MAN.number  
  
JOIN manchewinners MW ON MAN.id = MW.id AND MAN.number = MW.number;
```

## Ottenere il numero di turni persi da un utente

(da usare insieme al numero di manche giocate da un utente per ottenere il numero medio di turni persi per manche oppure insieme al numero di partite giocate per il numero medio per partita)

```
SELECT COUNT(*) AS count FROM moves
```



```
WHERE id = 'id del giocatore' AND outcome = 0;
```

## Ottenere il numero di volte che un utente ha ottenuto 'PERDE' dal giro di ruota

(da usare insieme al numero di manche giocare per ottenere il valore medio per manche oppure insieme al numero di match giocati per il valore medio per partita)

```
SELECT COUNT(*) AS count FROM moves
```

```
WHERE id = 'id del giocatore' AND movetype = 'perde';
```