

```
## repository `db-videogames-query`
```

```
## Dump. ![[mysql-db_videogames.sql]]
```

```
## Query
```

```
### SELECT
```

1- Selezionare tutte le software house americane (3)

```
SELECT *  
FROM software_houses sh  
WHERE country LIKE 'United States';
```

2- Selezionare tutti i giocatori della città di 'Rogahnland' (2)

```
SELECT *  
FROM players p  
WHERE city LIKE 'Rogahnland';
```

3- Selezionare tutti i giocatori il cui nome finisce per "a" (220)

```
SELECT *  
FROM players p  
WHERE name LIKE '%a';
```

4- Selezionare tutte le recensioni scritte dal giocatore con ID = 800 (11)

```
SELECT *  
FROM reviews r  
WHERE player_id = 800;
```

5- Contare quanti tornei ci sono stati nell'anno 2015 (9)

```
SELECT *  
FROM tournaments t  
WHERE `year` LIKE 2015;
```

6- Selezionare tutti i premi che contengono nella descrizione la parola 'facere' (2)

```
SELECT *  
FROM awards a  
WHERE description LIKE '%facere%';
```

7- Selezionare tutti i videogame che hanno la categoria 2 (FPS) o 6 (RPG), mostrandoli una sola volta (del videogioco vogliamo solo l'ID) (287)

```
SELECT DISTINCT videogame_id  
FROM category_videogame cv  
WHERE category_id = 2  
OR category_id = 6;
```

8- Selezionare tutte le recensioni con voto compreso tra 2 e 4 (2947)

```
SELECT *  
FROM reviews r  
WHERE rating > 1  
AND rating < 5;
```

9- Selezionare tutti i dati dei videogiochi rilasciati nell'anno 2020 (46)

```
SELECT *  
FROM videogames v  
WHERE YEAR(release_date) = 2020;
```

10- Selezionare gli id dei videogame che hanno ricevuto una recensione da 5 stelle, mostrandoli una sola volta (443)

```
SELECT DISTINCT videogame_id  
FROM reviews r  
WHERE rating = 5;
```

GROUP BY

1- Contare quante software house ci sono per ogni paese (3)

```
SELECT country, COUNT(*)  
FROM software_houses sh  
GROUP BY country ;
```

2- Contare quante recensioni ha ricevuto ogni videogioco (del videogioco vogliamo solo l'ID) (500)

```
SELECT videogame_id, COUNT(*)  
FROM reviews r  
GROUP BY videogame_id ;
```

3- Contare quanti videogiochi hanno ciascuna classificazione PEGI (della classificazione PEGI vogliamo solo l'ID) (13)

```
SELECT plv.pegi_label_id, COUNT(*)  
FROM pegi_label_videogame plv  
GROUP BY plv.pegi_label_id
```

4- Mostrare il numero di videogiochi rilasciati ogni anno (11)

```
SELECT YEAR(release_date), COUNT(*)  
FROM videogames v  
GROUP BY YEAR(release_date);
```

5- Contare quanti videogiochi sono disponibili per ciascun device (del device vogliamo solo l'ID) (7)

```
SELECT device_id, COUNT(*)  
FROM device_videogame dv  
GROUP BY device_id ;
```

6- Ordinare i videogame in base alla media delle recensioni (del videogioco vogliamo solo l'ID) (500)

```
SELECT videogame_id, AVG(rating)  
FROM reviews r  
GROUP BY videogame_id  
ORDER BY AVG(rating) DESC;
```

JOIN

1- Selezionare i dati di tutti i giocatori che hanno scritto almeno una recensione, mostrandoli una sola volta (996)

```
SELECT DISTINCT p.*  
FROM players p  
      JOIN reviews r  
      ON p.id = player_id;
```

2- Sezionare tutti i videogame dei tornei tenuti nel 2016, mostrandoli una sola volta (226)

```
SELECT DISTINCT v.*  
FROM videogames v  
      JOIN tournament_videogame tv  
      ON v.id = tv.videogame_id  
      JOIN tournaments t  
      ON t.id = tv.tournament_id  
WHERE t.`year` = 2016;
```

3- Mostrare le categorie di ogni videoggioco (1718)

```
SELECT c.name , v.name  
FROM category_videogame cv  
      JOIN videogames v  
      ON cv.videogame_id = v.id  
      JOIN categories c  
      ON cv.category_id = c.id ;
```

4- Selezionare i dati di tutte le software house che hanno rilasciato almeno un gioco dopo il 2020, mostrandoli una sola volta (6)

```
SELECT DISTINCT sh.*  
FROM software_houses sh  
      JOIN videogames v  
      ON sh.id = v.software_house_id  
WHERE YEAR(v.release_date) > 2020;
```

5- Selezionare i premi ricevuti da ogni software house per i videogiochi che ha prodotto (55)

```
SELECT a.name, av.videogame_id , sh.name
FROM awards a
      JOIN award_videogame av
            ON a.id = av.award_id
      JOIN videogames v
            ON av.videogame_id = v.id
      JOIN software_houses sh
            ON v.software_house_id = sh.id;
```

6- Selezionare categorie e classificazioni PEGI dei videogiochi che hanno ricevuto recensioni da 4 e 5 stelle, mostrandole una sola volta (3363)

```
SELECT DISTINCT c.name , pl.name, v.name
FROM reviews r
      JOIN videogames v
            ON r.videogame_id = v.id
      JOIN category_videogame cv
            ON v.id = cv.videogame_id
            JOIN categories c
                  ON cv.category_id = c.id
      JOIN pegi_label_videogame plv
            ON v.id = plv.videogame_id
            JOIN pegi_labels pl
                  ON plv.pegi_label_id = pl.id
WHERE r.rating > 3;
```

7- Selezionare quali giochi erano presenti nei tornei nei quali hanno partecipato i giocatori il cui nome inizia per 'S' (474)

```
SELECT DISTINCT v.id, v.name
FROM tournaments t
      JOIN tournament_videogame tv
            ON t.id = tv.tournament_id
            JOIN videogames v
                  ON tv.videogame_id = v.id
      JOIN player_tournament pt
            ON t.id = pt.tournament_id
            JOIN players p
                  ON pt.player_id = p.id
WHERE p.name LIKE "S%";
```

8- Selezionare le città in cui è stato giocato il gioco dell'anno del 2018 (36)

```
SELECT t.city
FROM tournaments t
      JOIN tournament_videogame tv
            ON t.id = tv.tournament_id
            JOIN videogames v
                  ON tv.videogame_id = v.id
      JOIN award_videogame av
            ON v.id = av.videogame_id
            JOIN awards a
                  ON av.award_id = a.id
WHERE a.name LIKE "Gioco dell'anno"
AND av.`year` = 2018;
```

9- Selezionare i giocatori che hanno giocato al gioco più atteso del 2018 in un torneo del 2019 (3306)

```
SELECT p.name
FROM players p
      JOIN player_tournament pt
            ON p.id = pt.player_id
            JOIN tournaments t
                  ON pt.tournament_id = t.id
      JOIN tournament_videogame tv
            ON t.id = tv.tournament_id
            JOIN videogames v
                  ON tv.videogame_id = v.id
      JOIN award_videogame av
            ON v.id = av.videogame_id
            JOIN awards a
                  ON av.award_id = a.id
WHERE t.`year` = 2019
AND av.`year` = 2018
AND a.name LIKE "Gioco più atteso";
```