

SELECT

1- Selezionare tutte le software house americane (3)

1-SELECT name FROM `software_houses` WHERE country = "United States";

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

2-SELECT * FROM `players` WHERE city = "Rogahnland";

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

3-SELECT * FROM `players` WHERE name like '%a';

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

4-SELECT * FROM `reviews` WHERE player_id = 800;

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

5-SELECT COUNT(*) FROM `tournaments` WHERE year = 2015;

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

6-SELECT * FROM `awards` 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)

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

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

8-SELECT * FROM `reviews` WHERE rating>=2 AND rating<=4;

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

9-SELECT * FROM `videogames` WHERE release_date LIKE "%2020%";

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

10-SELECT DISTINCT videogame_id FROM `reviews` WHERE rating = 5;

GROUP BY

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

1-SELECT COUNT(id) FROM `software_houses` GROUP BY country;

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

2-SELECT COUNT(videogame_id) FROM `reviews` GROUP BY videogame_id;

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

```
3-SELECT COUNT(videogame_id) FROM `pegi_label_videogame` GROUP BY  
pegi_label_id;
```

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

```
4-SELECT YEAR (release_date), COUNT(*) from videogames GROUP BY  
YEAR(release_date);
```

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

```
5-SELECT device_id as device, COUNT(*) as num_games  
FROM device_videogame  
GROUP BY device_id;
```

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

```
6-SELECT videogame_id FROM reviews 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)

```
1-SELECT DISTINCT p.* from players p JOIN reviews r ON p.id = r.player_id;
```

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

```
2-SELECT DISTINCT (v.name) FROM tournaments t  
JOIN tournament_videogame tv  
JOIN videogames v  
WHERE t.id = tv.tournament_id AND tv.videogame_id = v.id AND t.`year` = 2016;
```

3- Mostrare le categorie di ogni videogioco (1718)

```
3-SELECT DISTINCT cv.videogame_id, c.name FROM categories c  
JOIN category_videogame cv  
JOIN videogames v  
WHERE 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)

```
4-SELECT DISTINCT so.name  
FROM software_houses so  
JOIN videogames v  
WHERE YEAR(v.release_date) >= 2020 AND so.id =v.software_house_id ;
```

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

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

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

8- Selezionare le città in cui è stato giocato il gioco dell'anno del 2018 (36)
8-SELECT tournaments.city, awards.name, tournaments.year
FROM tournaments
INNER JOIN tournament_videogame ON tournament_videogame.tournament_id =
tournaments.id
INNER JOIN videogames ON tournament_videogame.videogame_id = videogames.id
INNER JOIN award_videogame ON videogames.id = award_videogame.videogame_id
INNER JOIN awards on award_videogame.award_id = awards.id WHERE awards.id = 1
AND award_videogame.year = 2018;

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