

QUERY

*1- Selezionare tutte le software house americane (3)

```
SELECT *  
FROM software_houses  
WHERE country = 'united states'
```

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

```
SELECT *  
FROM players  
WHERE city = 'Rogahnland'
```

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

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

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

```
SELECT *  
FROM reviews  
WHERE player_id = '800'
```

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

```
SELECT COUNT(*) AS count_tornei_nel_2015
FROM tournaments
WHERE year = 2015
```

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

```
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)

```
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)**

```
SELECT *
FROM reviews
WHERE rating BETWEEN 2 AND 4
```

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

```
SELECT *
FROM videogames
```

```
WHERE YEAR(release_date) = 2020
```

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

-

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

*11- Selezionare il numero e la media delle recensioni per il videogioco con ID = 412 (review number = 12, avg_rating = 3)

-

```
SELECT COUNT(*) AS num_reviews, AVG(rating) AS media  
FROM reviews  
WHERE videogame_id = 412;
```

12- Selezionare il numero di videogame che la software house con ID = 1 ha rilasciato nel 2018 (13)

```
SELECT COUNT(*) AS videogames  
FROM videogames  
WHERE software_house_id = 1  
AND YEAR(release_date) = 2018;
```

***GROUP BY

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

```
SELECT country, COUNT(*) AS num_software_houses
FROM software_houses
GROUP BY country
```

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

```
SELECT videogame_id, COUNT(*) AS num_reviews
FROM reviews
GROUP BY videogame_id
```

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

```
SELECT pegi_label_id, COUNT(*) AS num_game
FROM pegi_label_videogame
GROUP BY pegi_label_id
```

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

```
SELECT YEAR(release_date) AS year, COUNT(*) AS count_game
FROM videogames
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(*) 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)

```
SELECT videogame_id, AVG(rating) AS media  
FROM reviews  
GROUP BY videogame_id  
ORDER BY AVG(rating)
```

***Query con join

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

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

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

```
SELECT DISTINCT videogames.id, videogames.name,  
videogames.release_date, tournaments.year  
FROM videogames  
JOIN tournament_videogame  
ON videogames.id = tournament_videogame.videogame_id  
JOIN tournaments  
ON tournament_videogame.tournament_id = tournaments.id  
WHERE tournaments.year = 2016;
```

*3- Mostrare le categorie di ogni videogioco (1718)

```
SELECT videogames.name, categories.name
FROM videogames
JOIN category_videogame
ON videogames.id = category_videogame.videogame_id
JOIN categories
ON category_videogame.category_id = categories.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 software_houses.name AS
software_house_name
FROM software_houses
JOIN videogames
ON software_houses.id = software_house_id
WHERE YEAR(release_date) > 2020
```

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

```
SELECT software_houses.name AS software_house_name,
videogames.name AS videogames_name, awards.name AS
awards_name
FROM awards
JOIN award_videogame
ON awards.id = award_videogame.award_id
JOIN videogames
ON award_videogame.videogame_id = videogames.id
```

```
JOIN software_houses
ON software_houses.id = software_house_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 categories.name AS categories_name,
videogames.name AS videogame_name, pegi_labels.name AS
PEGI_name
FROM categories
JOIN category_videogame
ON categories.id = category_videogame.category_id
JOIN videogames
ON category_videogame.videogame_id = videogames.id
JOIN pegi_label_videogame
ON videogames.id = pegi_label_videogame.videogame_id
JOIN pegi_labels
ON pegi_label_videogame.pegi_label_id = pegi_labels.id
JOIN reviews
ON videogames.id = reviews.videogame_id
WHERE rating BETWEEN 4 AND 5
```

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

```
SELECT DISTINCT videogames.name AS videogame_name
FROM videogames
JOIN tournament_videogame
ON videogames.id = tournament_videogame.videogame_id
JOIN tournaments
ON tournament_videogame.tournament_id = tournaments.id
JOIN player_tournament
ON tournaments.id = player_tournament.tournament_id
```

```
JOIN players
ON player_tournament.player_id = players.id
WHERE players.name LIKE 's%'
```

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

```
SELECT tournaments.city, videogames.name, awards.name,
award_videogame.year
FROM tournaments
JOIN tournament_videogame
ON tournaments.id = tournament_videogame.tournament_id
JOIN videogames
ON tournament_videogame.videogame_id = videogames.id
JOIN award_videogame
ON videogames.id = award_videogame.videogame_id
JOIN awards
ON award_videogame.award_id = awards.id
WHERE award_videogame.year = 2018 AND awards.name LIKE
'%anno'
```

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

```
SELECT players.name, players.lastname,
awards.name, award_videogame.year AS award_year,
tournaments.year AS tournament_year
FROM videogames
JOIN tournament_videogame
ON videogames.id = tournament_videogame.videogame_id
JOIN tournaments
ON tournament_videogame.tournament_id = tournaments.id
JOIN award_videogame
ON videogames.id = award_videogame.videogame_id
```



```

JOIN awards
ON award_videogame.award_id = awards.id
JOIN player_tournament
ON tournaments.id = player_tournament.tournament_id
JOIN players
ON player_tournament.player_id = players.id
WHERE award_videogame.year = 2018
AND awards.name LIKE '%atteso%'
AND tournaments.year = 2019

```

*10- Selezionare i dati della prima software house che ha rilasciato un gioco, assieme ai dati del gioco stesso (software house id : 5)

```

SELECT TOP 1 software_houses.id, software_houses.name AS
software_house_name, videogames.name AS videogame_name
FROM software_houses
JOIN videogames
ON software_houses.id = videogames.software_house_id
WHERE software_houses.id = 5
ORDER BY videogames.release_date;

```

*11- Selezionare i dati del videogame (id, name, release_date, totale recensioni) con più recensioni (videogame id : 398)

```

SELECT TOP 1 videogames.id, videogames.name,
videogames.release_date, COUNT(reviews.videogame_id) AS
reviews_count
FROM videogames
JOIN reviews
ON videogames.id = reviews.videogame_id
WHERE videogames.id = 398
GROUP BY videogames.id, videogames.name,

```

```
videogames.release_date  
ORDER BY COUNT(reviews.videogame_id) DESC;
```

*12- Selezionare la software house che ha vinto più premi tra il 2015 e il 2016 (software house id : 1)

```
SELECT TOP 1  software_houses.name AS software_house_name,  
COUNT(award_videogame.videogame_id) AS total_awards  
FROM software_houses  
JOIN videogames  
ON software_houses.id = videogames.software_house_id  
JOIN tournament_videogame  
ON videogames.id = tournament_videogame.videogame_id  
JOIN tournaments  
ON tournament_videogame.tournament_id = tournaments.id  
JOIN award_videogame  
ON videogames.id = award_videogame.videogame_id  
JOIN awards  
ON award_videogame.award_id = awards.id  
WHERE tournaments.year BETWEEN 2015 AND 2016  
GROUP BY software_houses.name  
ORDER BY COUNT(award_videogame.videogame_id) DESC
```

*13- Selezionare le categorie dei videogame i quali hanno una media recensioni inferiore a 2 (10)

```
SELECT DISTINCT categories.name AS category  
FROM videogames  
JOIN category_videogame  
ON videogames.id = category_videogame.videogame_id  
JOIN categories  
ON category_videogame.category_id = categories.id  
JOIN reviews  
ON videogames.id = reviews.videogame_id
```

```
GROUP BY categories.name, videogames.id  
HAVING AVG(reviews.rating)<2
```