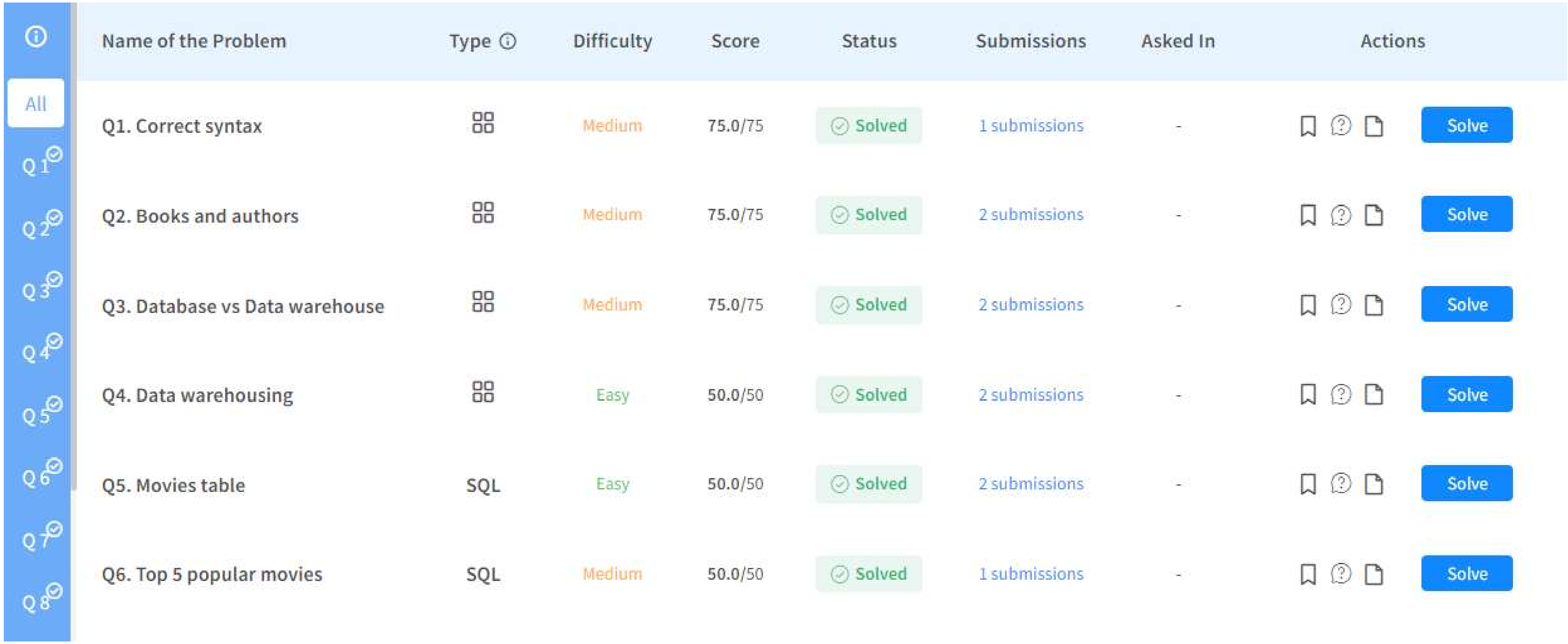
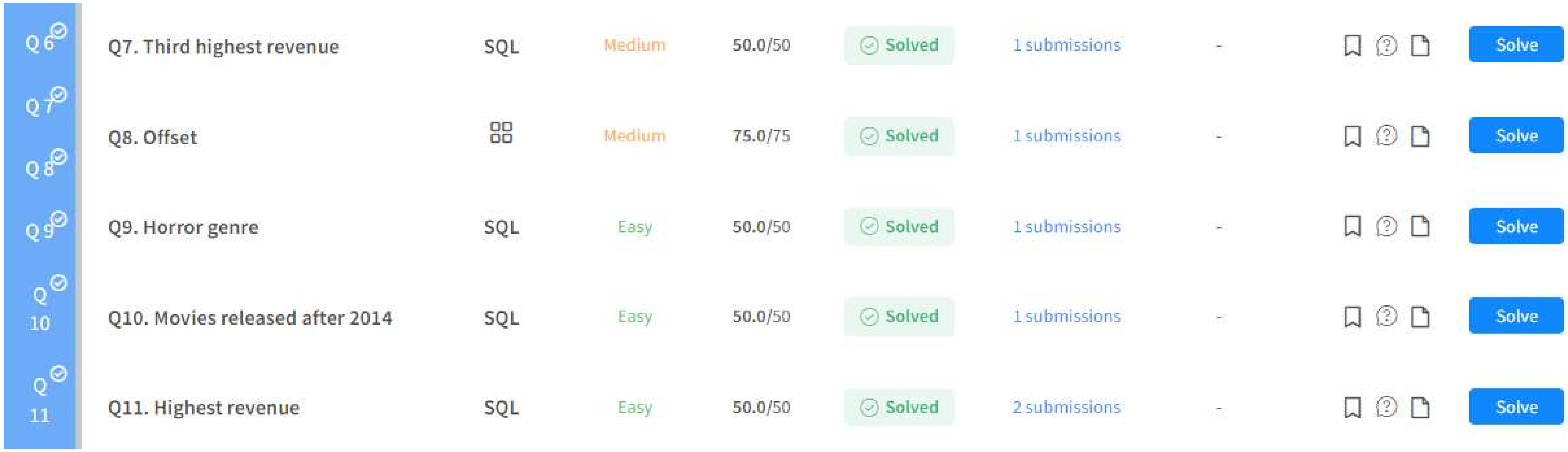
SQL_01 - Extracting data using SQL

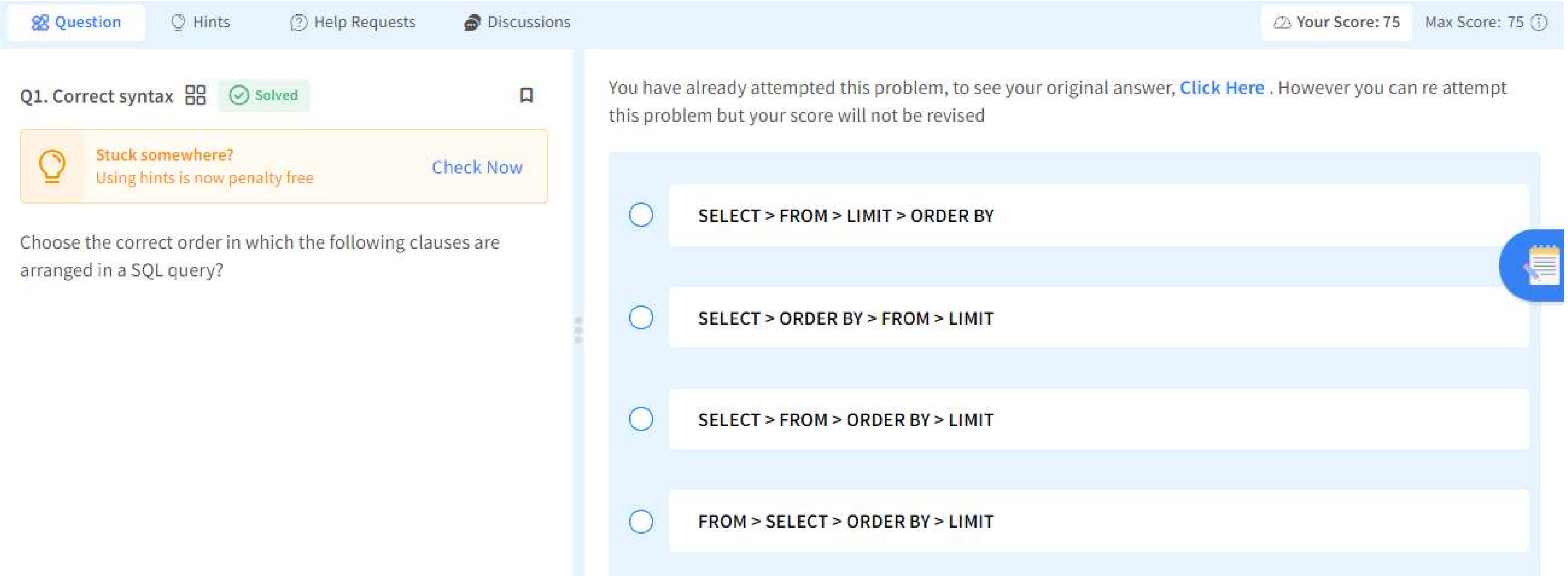


Name of the Problem	Type ⓘ	Difficulty	Score	Status	Submissions	Asked In	Actions
Q1. Correct syntax	⊞	Medium	75.0/75	✓ Solved	1 submissions	-	Solve
Q2. Books and authors	⊞	Medium	75.0/75	✓ Solved	2 submissions	-	Solve
Q3. Database vs Data warehouse	⊞	Medium	75.0/75	✓ Solved	2 submissions	-	Solve
Q4. Data warehousing	⊞	Easy	50.0/50	✓ Solved	2 submissions	-	Solve
Q5. Movies table	SQL	Easy	50.0/50	✓ Solved	2 submissions	-	Solve
Q6. Top 5 popular movies	SQL	Medium	50.0/50	✓ Solved	1 submissions	-	Solve



Q7. Third highest revenue	SQL	Medium	50.0/50	✓ Solved	1 submissions	-	Solve
Q8. Offset	⊞	Medium	75.0/75	✓ Solved	1 submissions	-	Solve
Q9. Horror genre	SQL	Easy	50.0/50	✓ Solved	1 submissions	-	Solve
Q10. Movies released after 2014	SQL	Easy	50.0/50	✓ Solved	1 submissions	-	Solve
Q11. Highest revenue	SQL	Easy	50.0/50	✓ Solved	2 submissions	-	Solve

Q1. Correct syntax



Your Score: 75 Max Score: 75

Q1. Correct syntax ✓ Solved

Stuck somewhere?
Using hints is now penalty free

Choose the correct order in which the following clauses are arranged in a SQL query?

You have already attempted this problem, to see your original answer, Click Here . However you can re attempt this problem but your score will not be revised

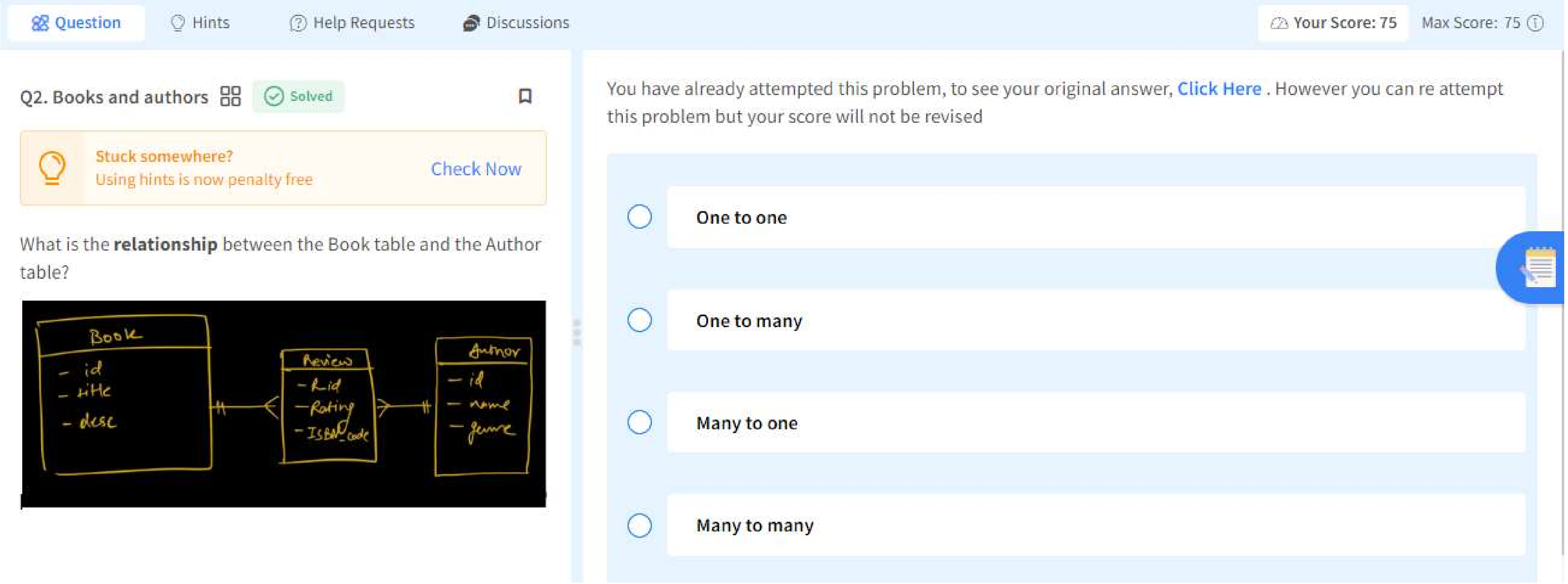
○ SELECT > FROM > LIMIT > ORDER BY

○ SELECT > ORDER BY > FROM > LIMIT

○ SELECT > FROM > ORDER BY > LIMIT

○ FROM > SELECT > ORDER BY > LIMIT

Q2. Books and authors



Q2. Books and authors

Stuck somewhere?
Using hints is now penalty free

What is the **relationship** between the Book table and the Author table?

You have already attempted this problem, to see your original answer, **Click Here** . However you can re attempt this problem but your score will not be revised

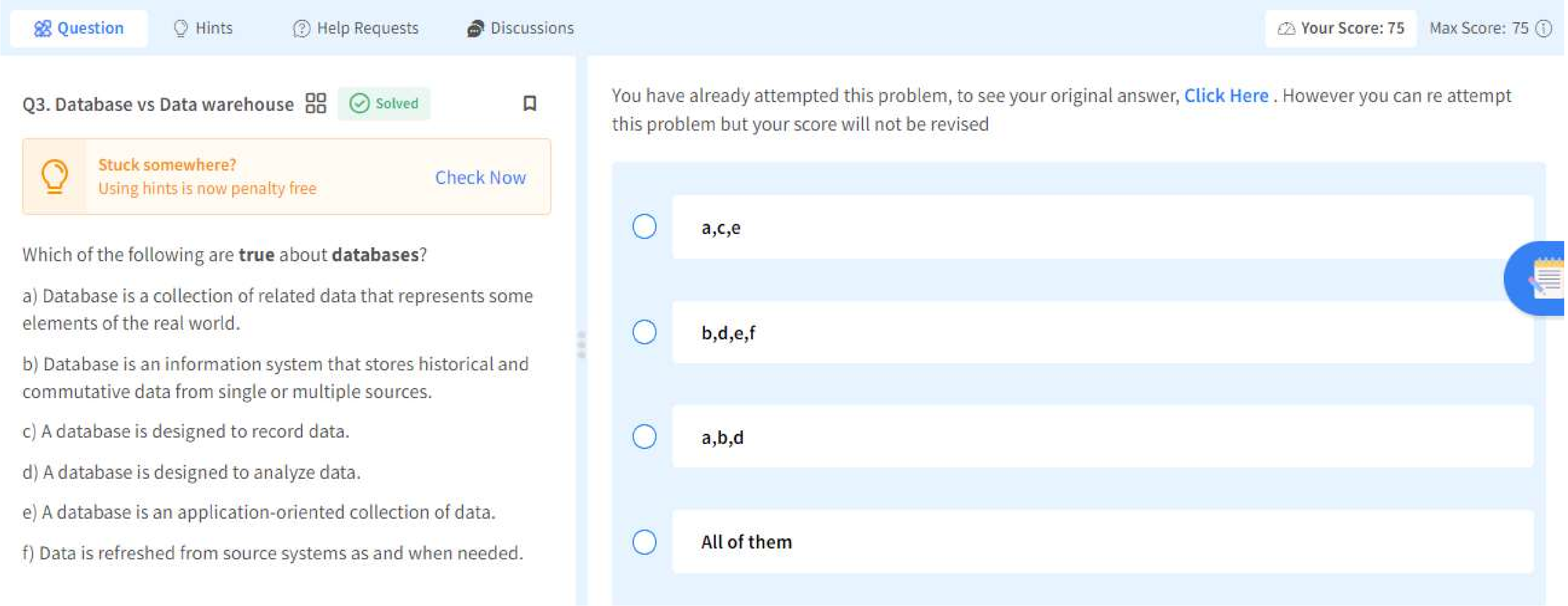
○ One to one

○ One to many

○ Many to one

○ Many to many

Q3. Database vs Data warehouse



Q3. Database vs Data warehouse

Which of the following are **true** about **databases**?

a) Database is a collection of related data that represents some elements of the real world.

b) Database is an information system that stores historical and commutative data from single or multiple sources.

c) A database is designed to record data.

d) A database is designed to analyze data.

e) A database is an application-oriented collection of data.

f) Data is refreshed from source systems as and when needed.

You have already attempted this problem, to see your original answer, Click Here . However you can re attempt this problem but your score will not be revised

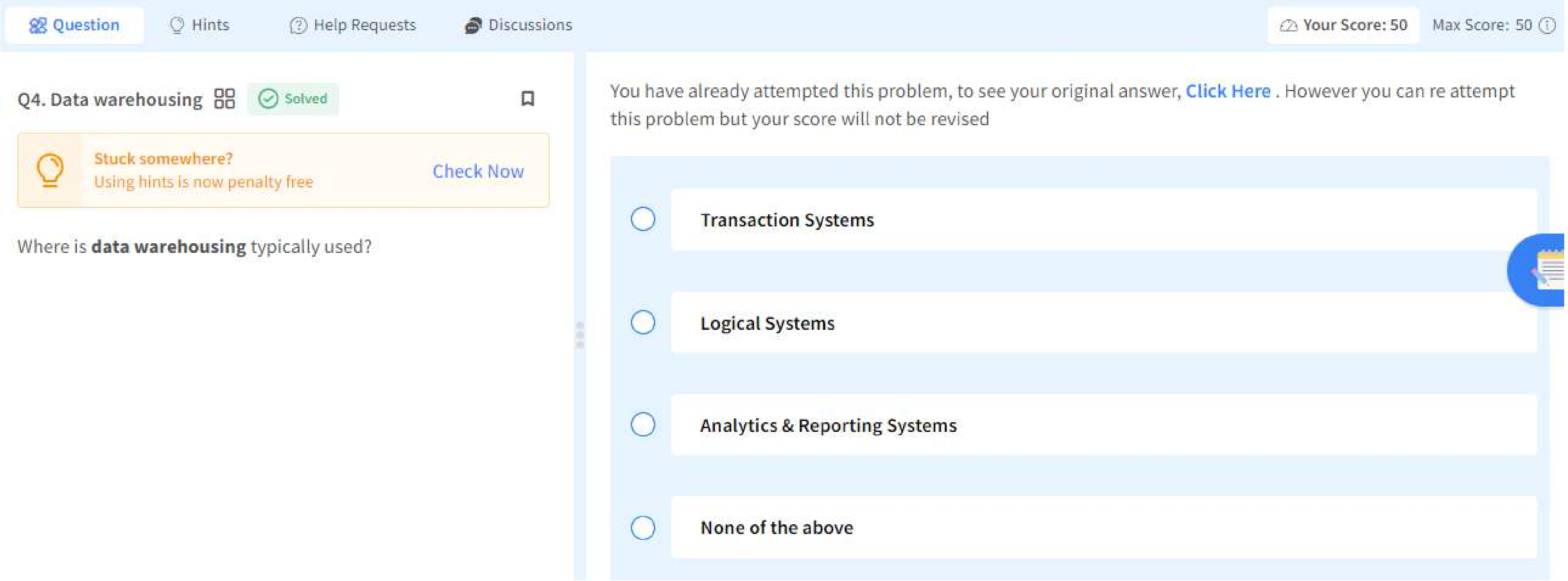
○ a,c,e

○ b,d,e,f

○ a,b,d

○ All of them

Q4. Data warehousing



Q4. Data warehousing

Stuck somewhere?
Using hints is now penalty free

Where is **data warehousing** typically used?

You have already attempted this problem, to see your original answer, Click Here . However you can re attempt this problem but your score will not be revised

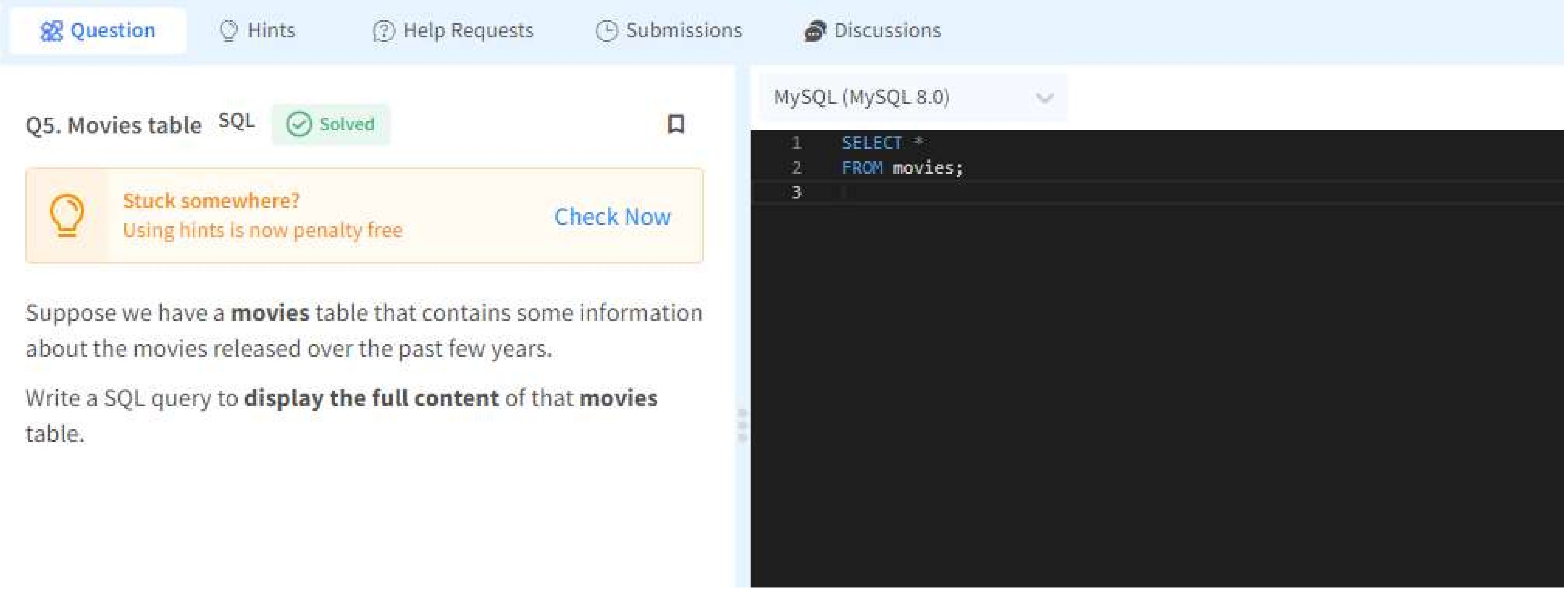
○ Transaction Systems

○ Logical Systems

○ Analytics & Reporting Systems

○ None of the above

Q5. Movies table



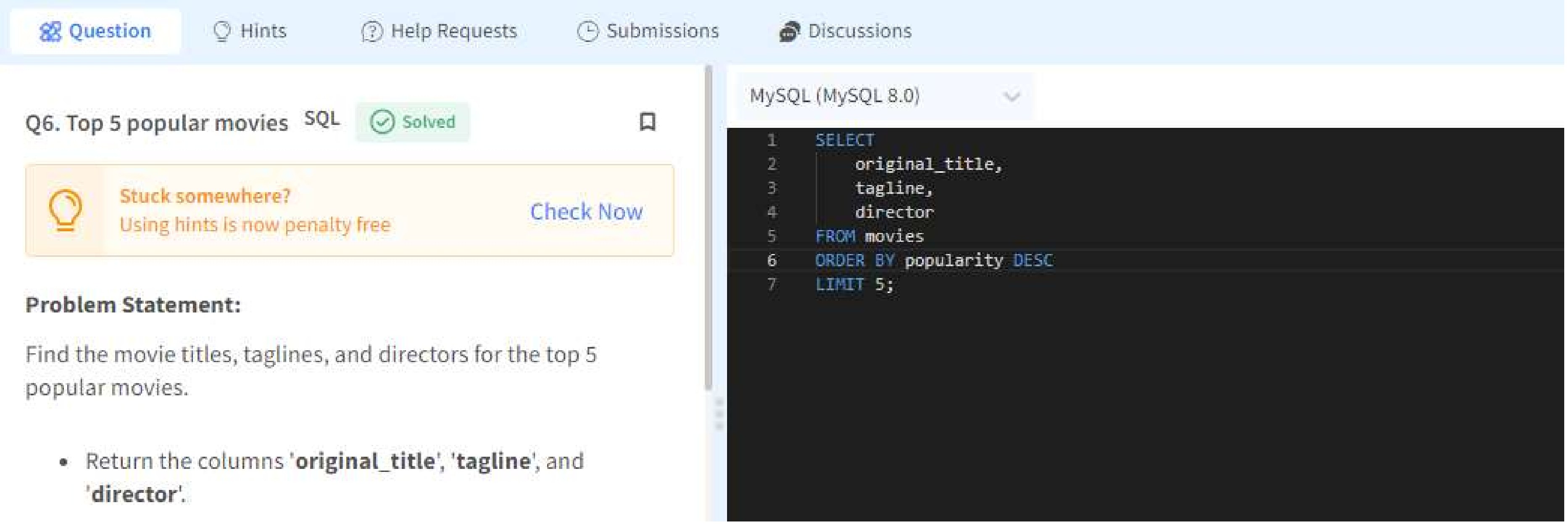
Q5. Movies table

Suppose we have a **movies** table that contains some information about the movies released over the past few years.

Write a SQL query to **display the full content** of that **movies** table.

```sql
SELECT *
FROM movies;
```

Q6. Top 5 popular movies



Q6. Top 5 popular movies

Problem Statement:

Find the movie titles, taglines, and directors for the top 5 popular movies.

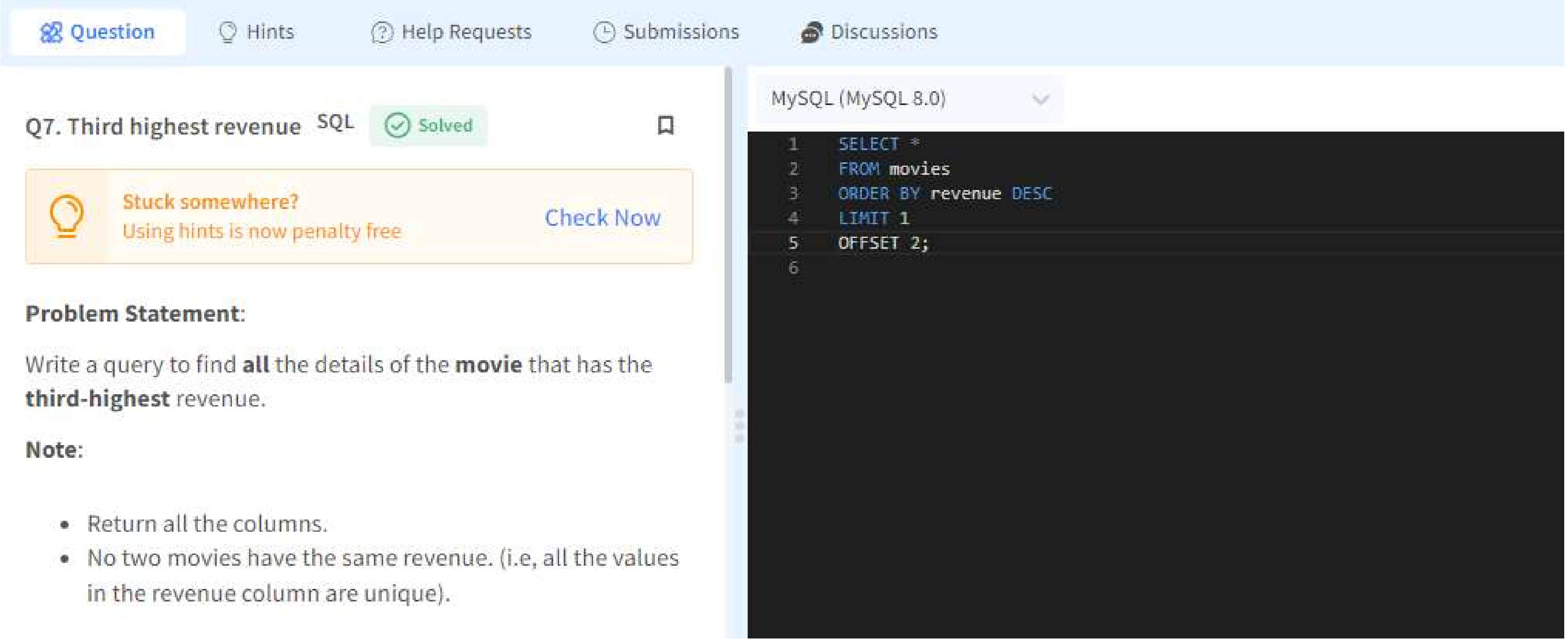
- Return the columns '**original_title**', '**tagline**', and '**director**'.

```sql
SELECT
    original_title,
    tagline,
    director
FROM movies
ORDER BY popularity DESC
LIMIT 5;
```

Dataset description for movies table:

1) id - tmdb movie id
2) imdb_id - imdb movie id
3) popularity -A numeric quantity specifying the movie's popularity.
4) budget -The budget in which the movie was made.
5) revenue - The worldwide revenue generated by the movie.
6) original_title- The title of the movie
7) cast - The name of the lead and supporting actors.
8) homepage - A link to the homepage of the movie.
9) director - The name of the director of the movie
10) tagline - Movie's tagline.
11) keywords -The keywords or tags related to the movie.
12) overview -A brief description of the movie.
13) runtime -The running time of the movie in minutes.
14) genres -The genres of the movies
15) production_companies-The production house of the movie.
16) release_date -the date on which it was released.
17) vote_count -the count of votes received.
18) vote_average - average ratings the movie received.
19) release_year - the year on which it was released

Q7. Third highest revenue



Q7. Third highest revenue SQL

Stuck somewhere?
Using hints is now penalty free

Problem Statement:

Write a query to find **all** the details of the **movie** that has the **third-highest** revenue.

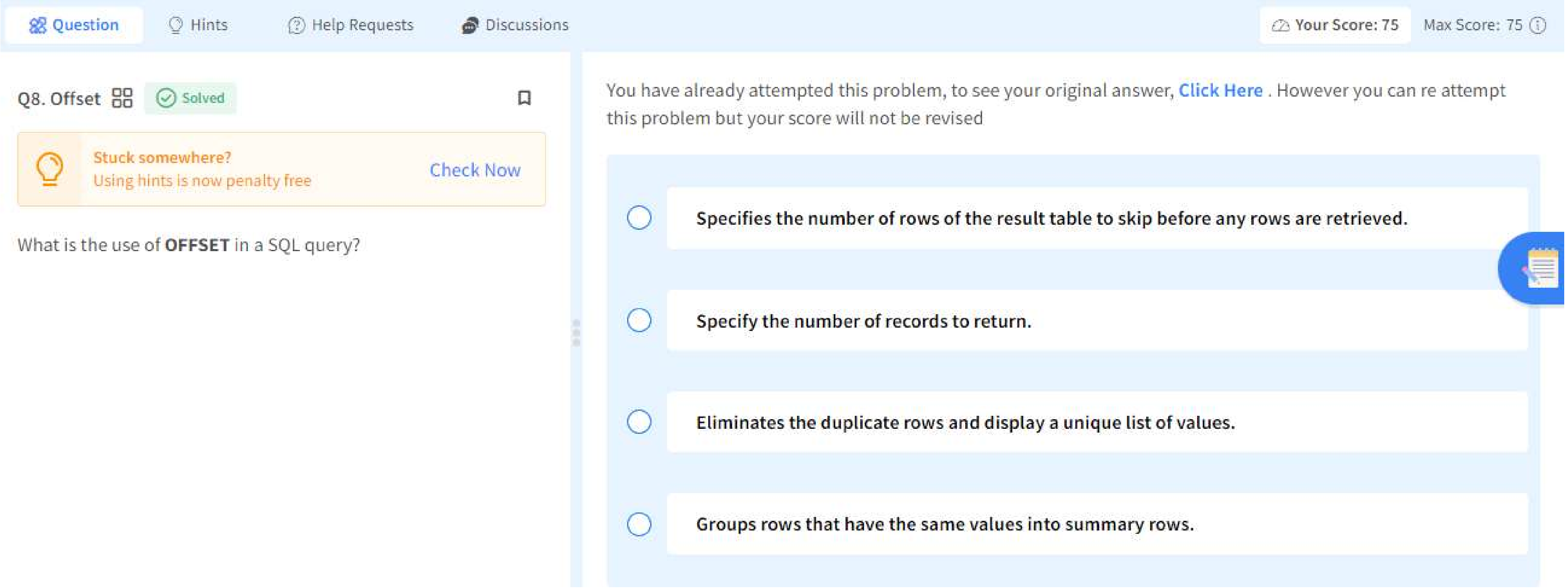
Note:

- Return all the columns.
- No two movies have the same revenue. (i.e, all the values in the revenue column are unique).

MySQL (MySQL 8.0)

```sql
SELECT *
FROM movies
ORDER BY revenue DESC
LIMIT 1
OFFSET 2;
```

Q8. Offset



Q8. Offset Solved

Stuck somewhere?
Using hints is now penalty free
Check Now

What is the use of **OFFSET** in a SQL query?

You have already attempted this problem, to see your original answer, Click Here . However you can re attempt this problem but your score will not be revised

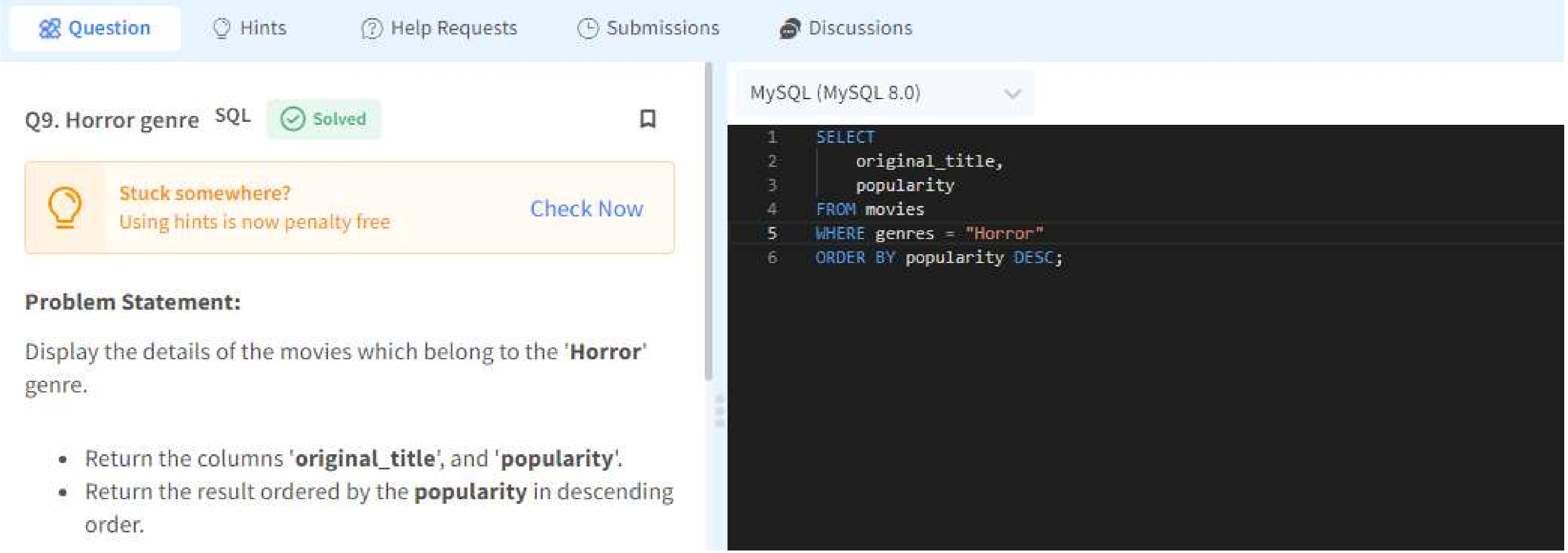
○ Specifies the number of rows of the result table to skip before any rows are retrieved.

○ Specify the number of records to return.

○ Eliminates the duplicate rows and display a unique list of values.

○ Groups rows that have the same values into summary rows.

Q9. Horror genre



Q9. Horror genre SQL ✓ Solved

Stuck somewhere?
Using hints is now penalty free

Check Now

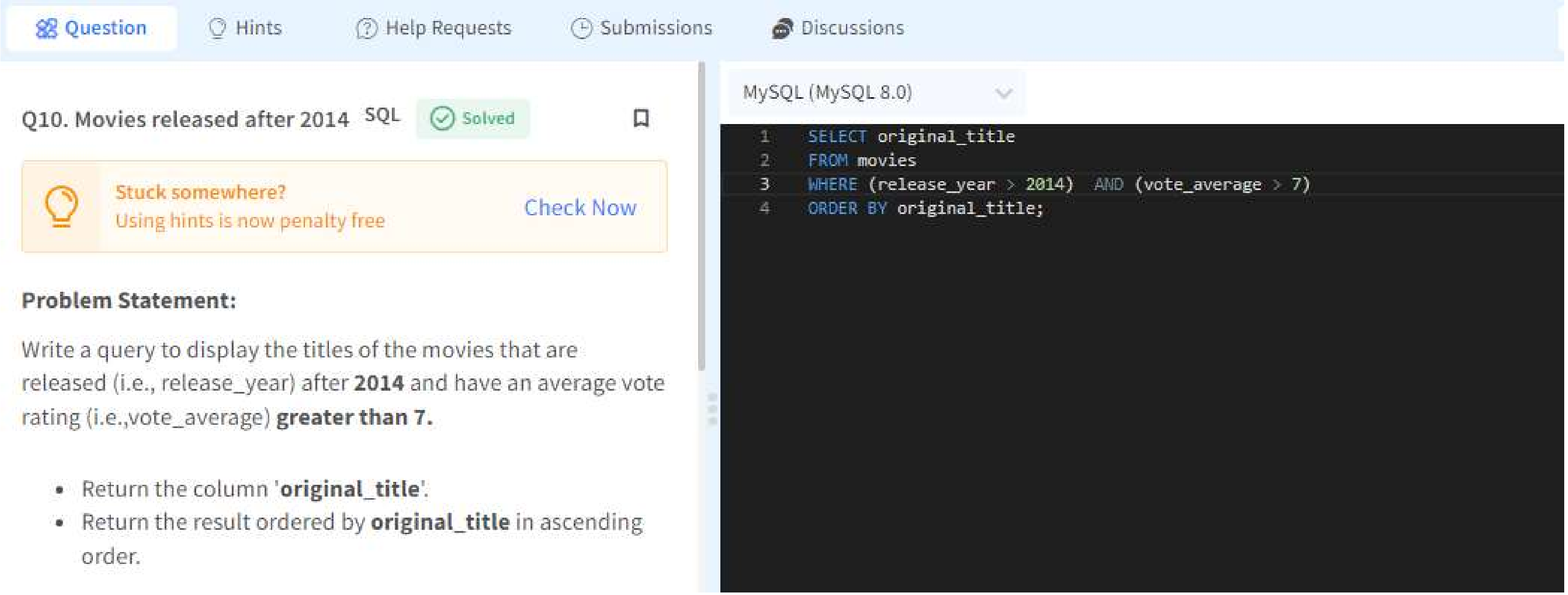
Problem Statement:

Display the details of the movies which belong to the '**Horror**' genre.

- Return the columns '**original_title**', and '**popularity**'.
- Return the result ordered by the **popularity** in descending order.

```sql
SELECT
    original_title,
    popularity
FROM movies
WHERE genres = "Horror"
ORDER BY popularity DESC;
```

Q10. Movies released after 2014



Q10. Movies released after 2014

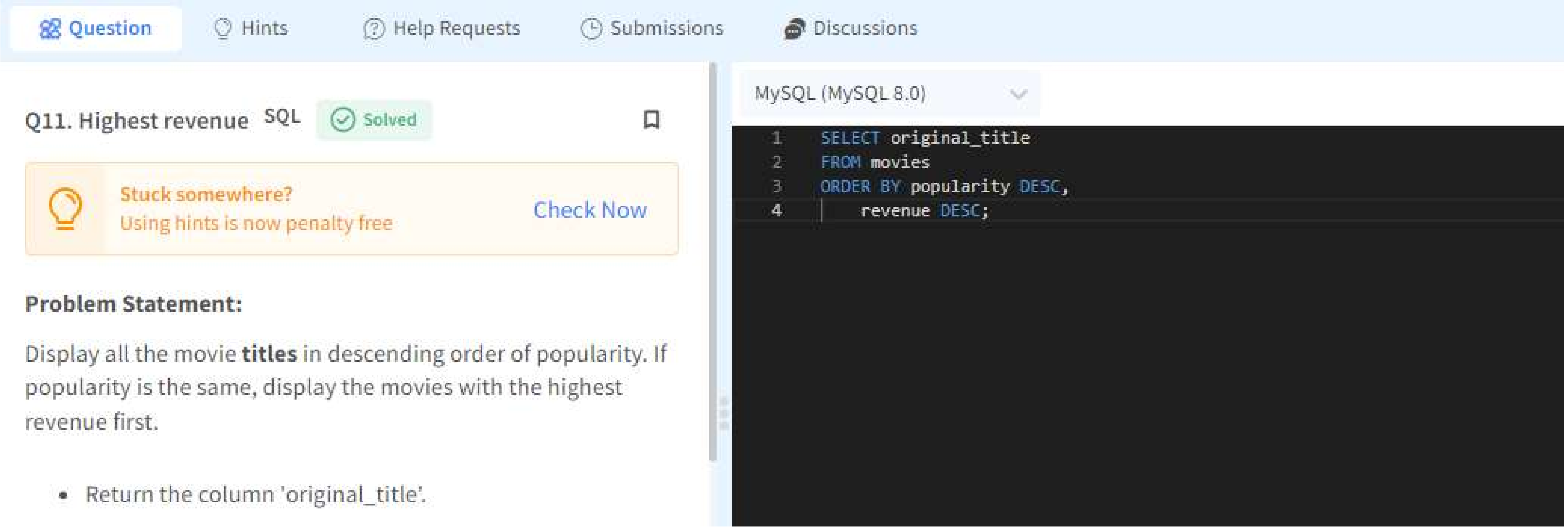
Problem Statement:

Write a query to display the titles of the movies that are released (i.e., release_year) after **2014** and have an average vote rating (i.e.,vote_average) **greater than 7.**

- Return the column '**original_title**'.
- Return the result ordered by **original_title** in ascending order.

```sql
SELECT original_title
FROM movies
WHERE (release_year > 2014)  AND (vote_average > 7)
ORDER BY original_title;
```

Q11. Highest revenue



Q11. Highest revenue SQL

Problem Statement:

Display all the movie **titles** in descending order of popularity. If popularity is the same, display the movies with the highest revenue first.

- Return the column 'original_title'.

```
SELECT original_title
FROM movies
ORDER BY popularity DESC,
    revenue DESC;
```