SQL_03 - Joins

All

Name of the Problem	Type	Difficulty	Score	Status	Submissions	Asked In	Actions
Q1. Not working employee	SQL	Medium	50.0/50	Solved	1 submissions	-	Solve
Q2. Article Views I	SQL	Easy	50.0/50	Solved	1 submissions	-	Solve
Q3. Employee 107	SQL	Medium	50.0/50	Solved	1 submissions	-	Solve
Q4. Adam	SQL	Medium	50.0/50	Solved	1 submissions	-	Solve
Q5. No Job history	SQL	Medium	50.0/50	Solved	1 submissions	-	Solve
Q6. Department name	SQL	Hard	50.0/50	Solved	2 submissions	-	Solve

Q7. Seattle	SQL	Medium	50.0/50	Solved	1 submissions	-	Solve
Q8. Invalid Departments	SQL	Very Easy	50.0/50	Solved	1 submissions	-	Solve
Q9. Salary Bins	SQL	Hard	50.0/50	Solved	2 submissions	-	Solve

Q1. Not working employee

Question | Hints | Help Requests | Submissions | Discussions

Q1. Not working employee SQL Solved

Stuck somewhere?
Using hints is now penalty free
Check Now

Problem Description:

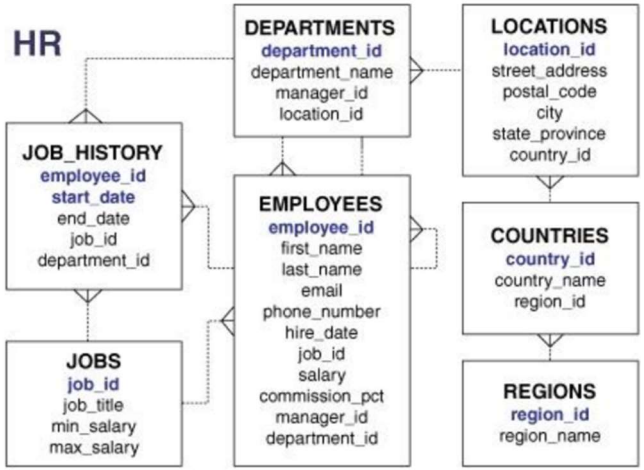
Find the details of employees who are not working in any department.

- Return the columns '**employee_id**', '**first_name**', '**last_name**', '**job_id**', and '**manager_id**'.
- Return the result ordered by **employee_id** in ascending order.

MySQL (MySQL 8.0)

```sql
SELECT
    employee_id,
    first_name,
    last_name,
    job_id,
    manager_id
FROM employees
WHERE department_id IS NULL
ORDER BY employee_id;
```

Dataset Description:



Sample Input:

Table: employees

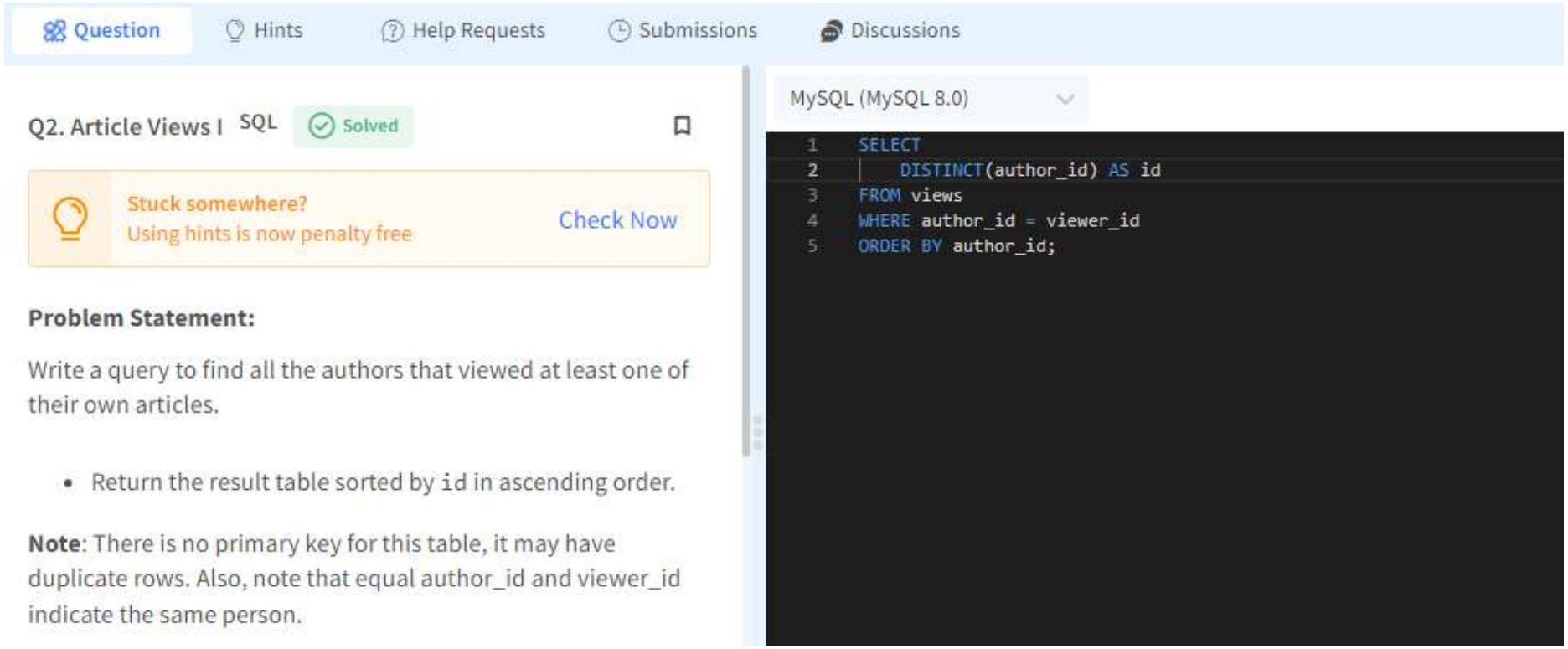
employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
175	Alyssa	Hutton	AHUTTON	011.44.1644.4...	1997-03-19	SA_REP	8800	0.25	149	80
176	Jonathon	Taylor	JTAYLOR	011.44.1644.4...	1998-03-24	SA_REP	8600	0.2	149	80
177	Jack	Livingston	JLIVINGS	011.44.1644.4...	1998-04-23	SA_REP	8400	0.2	149	80
178	Kimberely	Grant	KGRANT	011.44.1644.4...	1999-05-24	SA_REP	7000	0.15	149	NULL
179	Charles	Johnson	CJOHNSON	011.44.1644.4...	2000-01-04	SA_REP	6200	0.1	149	80
180	Winston	Taylor	WTAYLOR	650.507.9876	1998-01-24	SH_CL...	3200	NULL	120	50

- The missing value in the **department_id** column in the employee's table refers to not working in any department.

Sample Output:

employee_id	first_name	last_name	job_id	manager_id
178	Kimberely	Grant	SA_REP	149

Q2. Article Vews I



Sample Input:

Table: views

article_id	author_id	viewer_id	view_date
1	3	5	2019-08-01
1	3	6	2019-08-02
2	7	7	2019-08-01
2	7	6	2019-08-02
4	7	1	2019-07-22
3	4	4	2019-07-21
3	4	4	2019-07-21

Sample output:

id
4
7

Explanation:

- For article_id = 2 on view_date 2019-08-01, author and viewer is the same person with id = 7.

- For article_id = 3 on view_date 2019-07-21, author and viewer is the same person with id = 4.

Q3. Employee 107

Question | Hints | Help Requests | Submissions | Discussions

Q3. Employee 107 SQL Solved

Stuck somewhere?
Using hints is now penalty free
Check Now

Problem Statement:

Write a query to find the details of the other employees who work in the **same job** as the employee with **employee_id** as **107**.

Note:

- The column **manager_id** in the **employees** table represents the **employee_id** of the manager.

Note:

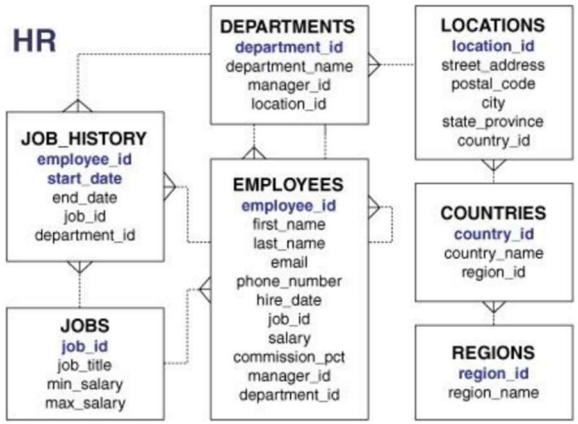
- Create a new column "**full_name**" by concatenating the **first_name** and **last_name** columns, separated by **space**.
- Return the columns '**full_name**', '**salary**', '**department_id**', and '**job_id**'.
- Return the output ordered by **full_name** in ascending order.

MySQL (MySQL 8.0)

```sql
SELECT
    CONCAT(first_name, " ", last_name) AS full_name,
    salary,
    department_id,
    job_id
FROM employees
WHERE job_id IN (SELECT job_id      # inner query
                 FROM employees
                 WHERE employee_id = 107)
ORDER BY full_name;
```

Test Output Code's All Neat!

Dataset Description:



Sample Input:

Table: employees

employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
101	Neena	Kochhar	NKOCHHAR	515.123.4568	1989-09-21	AD_VP	17000	NULL	100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	1990-01-03	IT_PROG	9000	NULL	102	60
104	Bruce	Ernst	BERNST	590.423.4568	1991-05-21	IT_PROG	6000	NULL	103	60
105	David	Austin	DAUSTIN	590.423.4569	1997-06-25	IT_PROG	4800	NULL	103	60
107	Diana	Lorentz	DLORENTZ	590.423.5567	1999-02-07	IT_PROG	4200	NULL	103	60
115	Alexander	Khoo	AKHOO	515.127.4562	1995-05-18	PU_CLERK	3100	NULL	114	30
151	David	Bernstein	DBERNSTE	011.44.1344.345268	1997-03-24	SA_REP	9500	0.25	145	80
165	David	Lee	DLEE	011.44.1346.529268	2000-02-23	SA_REP	6800	0.1	147	80

Sample Output:

full_name	salary	department_id	job_id
Alexander Hunold	9000	60	IT_PROG
Bruce Ernst	6000	60	IT_PROG
David Austin	4800	60	IT_PROG
Diana Lorentz	4200	60	IT_PROG

Sample Explanation:

Here the employee with id 107 has the job_id as '**IT_PROG**'. In the sample data, we have 4 employees with the same job_id including employee 107. Hence, we return those 4 records with the same job_id as 107.

Q4. Adam

Question | Hints | Help Requests | Submissions | Discussions

Q4. Adam SQL Solved

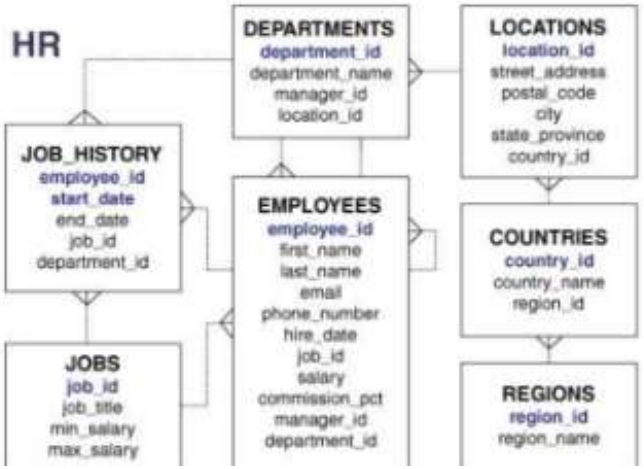
Stuck somewhere?
Using hints is now penalty free
Check Now

Problem Statement:

Write a query to display the employee details who report to **Adam**. The reporting structure can be inferred from the **manager_id** column in the employee's table.

- Return the columns '**employee_id**', '**full_name**'(first name and last name separated by space), and '**salary**'.
- Return the result ordered by **employee_id** in ascending order.

Dataset Description:



MySQL (MySQL 8.0)

```mysql
SELECT
    employee_id,
    CONCAT(first_name, " ", last_name) AS full_name,
    salary
FROM employees
WHERE manager_id IN (SELECT employee_id
                     FROM employees
                     WHERE first_name = "Adam")
ORDER BY employee_id;
```

Test Output

Test

Sample Input:

Table: employees

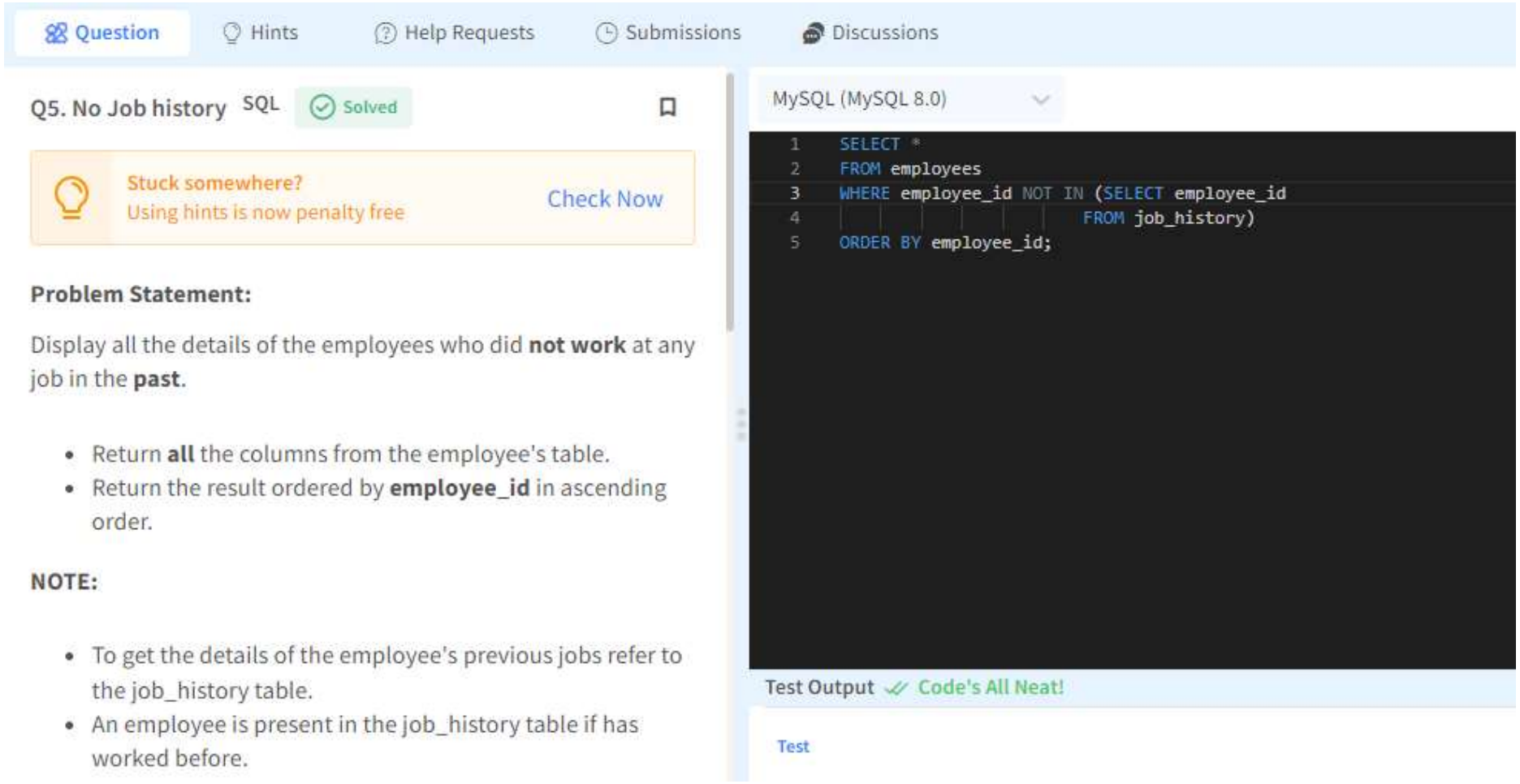
employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
121	Adam	Fripp	AFRIPP	650.123.2234	1997-04-10	ST_MAN	8200	NULL	100	50
132	TJ	Olson	TJOLSON	650.124.8234	1999-04-10	ST_CLERK	2100	NULL	121	50
184	Nandita	Sarchand	NSARCHAN	650.509.1876	1996-01-27	SH_CLERK	4200	NULL	121	50
185	Alexis	Bull	ABULL	650.509.2876	1997-02-20	SH_CLERK	4100	NULL	121	50
186	Julia	Dellinger	JDELLING	650.509.3876	1998-06-24	SH_CLERK	3400	NULL	121	50
187	Anthony	Cabrio	ACABRIO	650.509.4876	1999-02-07	SH_CLERK	3000	NULL	121	50
188	Kelly	Chung	KCHUNG	650.505.1876	1997-06-14	SH_CLERK	3800	NULL	122	50
189	Jennifer	Dilly	JDILLY	650.505.2876	1997-08-13	SH_CLERK	3600	NULL	122	50
190	Timothy	Gates	TGATES	650.505.3876	1998-07-11	SH_CLERK	2900	NULL	122	50

- The manager_id in the employees table is the **employee_id** of the manager.

Sample Output:

employee_id	full_name	salary
132	TJ Olson	2100
184	Nandita Sarchand	4200
185	Alexis Bull	4100
186	Julia Dellinger	3400
187	Anthony Cabrio	3000

Q5. No Job history



Sample Input:

Table: employees

employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
100	Steven	King	SKING	515.123.4567	1987-06-17	AD_PRES	25000	NULL	NULL	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	1989-09-21	AD_VP	17000	NULL	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	1993-01-13	AD_VP	17000	NULL	100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	1990-01-03	IT_PROG	9000	NULL	102	60
104	Bruce	Ernst	BERNST	590.423.4568	1991-05-21	IT_PROG	6000	NULL	103	60
105	David	Austin	DAUSTIN	590.423.4569	1997-06-25	IT_PROG	4800	NULL	103	60

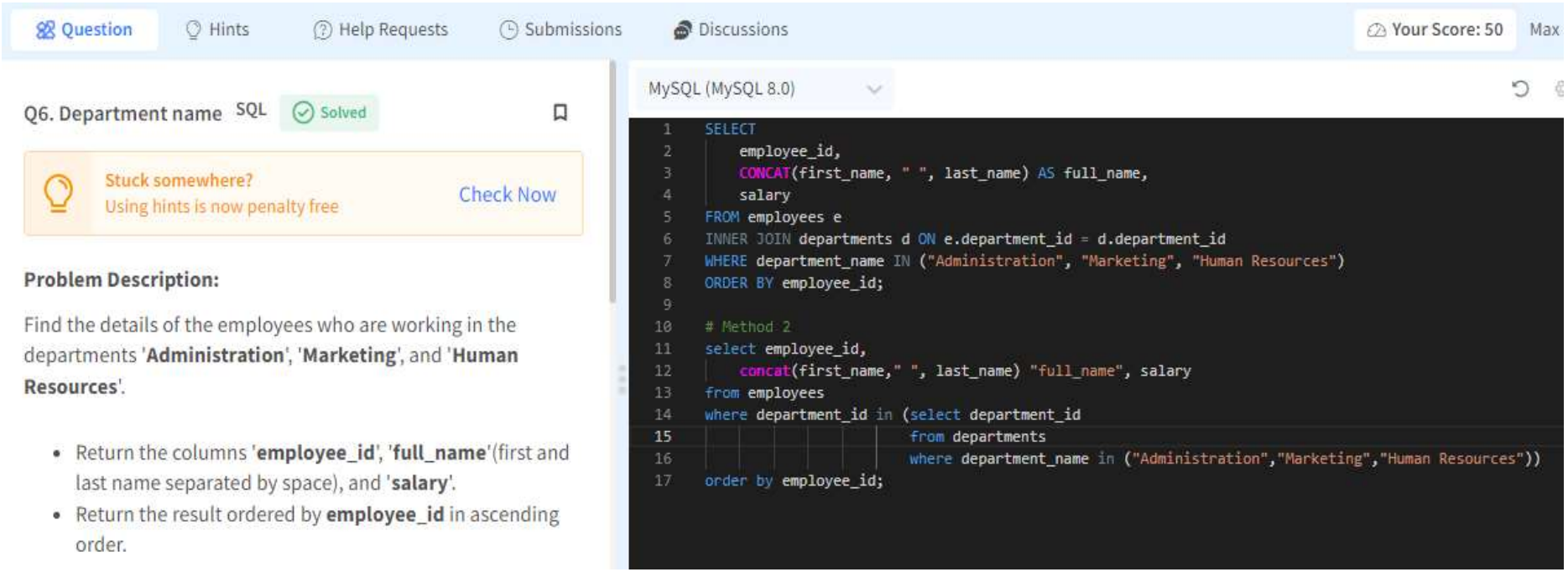
Table: job_history

employee_id	start_date	end_date	job_id	department_id
101	1989-09-21	1993-10-27	AC_ACCOUNT	110
101	1993-10-28	1997-03-15	AC_MGR	110
102	1993-01-13	1998-07-24	IT_PROG	60

Sample Output:

employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
100	Steven	King	SKING	515.123.4567	1987-06-17	AD_PRES	25000	NULL	NULL	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	1990-01-03	IT_PROG	9000	NULL	102	60
104	Bruce	Ernst	BERNST	590.423.4568	1991-05-21	IT_PROG	6000	NULL	103	60
105	David	Austin	DAUSTIN	590.423.4569	1997-06-25	IT_PROG	4800	NULL	103	60

Q6. Department name



Sample Input:

Table: employees

employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_...	manager_id	department_id
200	Jennifer	Whalen	JWHALEN	515.123.4444	1987-09-17	AD_ASST	4400	NULL	101	10
201	Michael	Hartstein	MHARTSTE	515.123.5555	1996-02-17	MK_MAN	13000	NULL	100	20
202	Pat	Fay	PFAY	603.123.6666	1997-08-17	MK_REP	6000	NULL	201	20
203	Susan	Mavris	SMAVRIS	515.123.7777	1994-06-07	HR_REP	6500	NULL	101	40

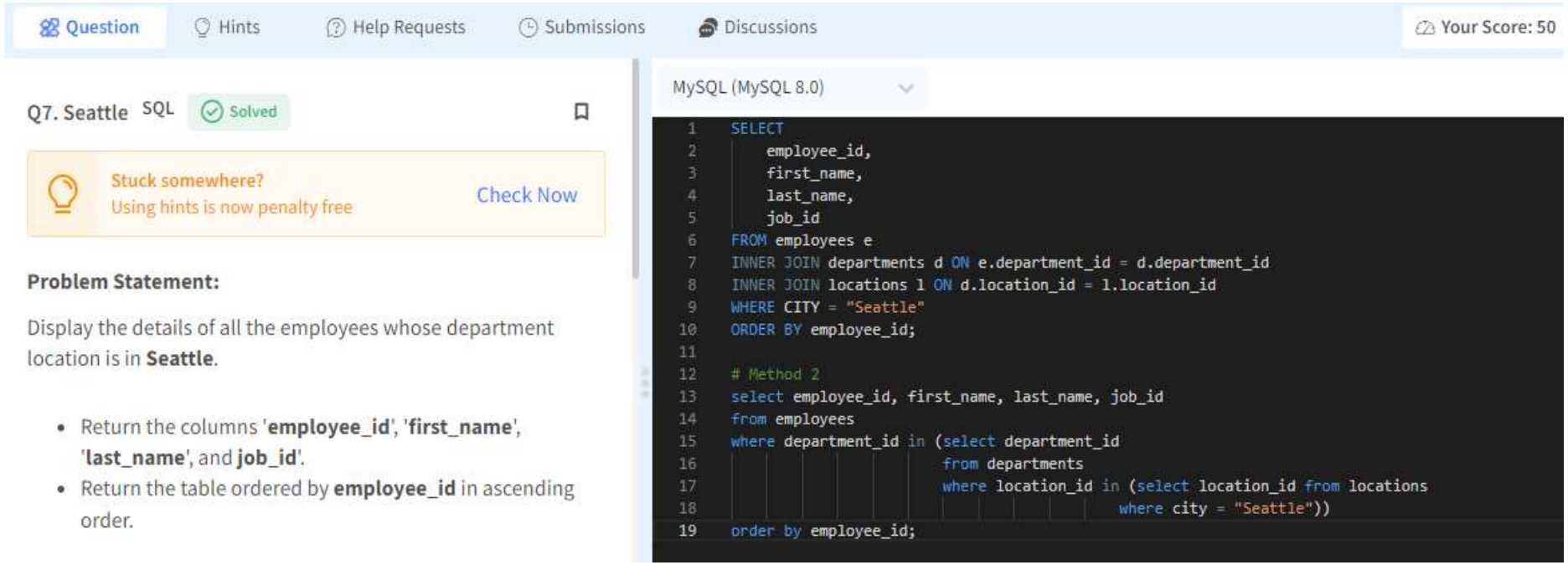
Table: departments

department_id	department_name	manager_id	location_id
10	Administration	200	1700
20	Marketing	201	1800
40	Human Resources	203	2400

Sample Output:

employee_id	full_name	salary
200	Jennifer Whalen	4400
201	Michael Hartstein	13000
202	Pat Fay	6000
203	Susan Mavris	6500

Q7. Seattle


Question
Hints
Help Requests
Submissions
Discussions
Your Score: 50
Q7. Seattle SQL Solved
Stuck somewhere?
Using hints is now penalty free
Check Now
Problem Statement:
Display the details of all the employees whose department location is in Seattle.
Return the columns 'employee_id', 'first_name', 'last_name', and job_id'.
Return the table ordered by employee_id in ascending order.
MySQL (MySQL 8.0)
SELECT
employee_id,
first_name,
last_name,
job_id
FROM employees e
INNER JOIN departments d ON e.department_id = d.department_id
INNER JOIN locations l ON d.location_id = l.location_id
WHERE CITY = "Seattle"
ORDER BY employee_id;
Method 2
select employee_id, first_name, last_name, job_id
from employees
where department_id in (select department_id
from departments
where location_id in (select location_id from locations
where city = "Seattle"))
order by employee_id;

Sample Input:

Table: employees

employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
100	Steven	King	SKING	515.123.4567	1987-06-17	AD_PRES	25000	NULL	NULL	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	1989-09-21	AD_VP	17000	NULL	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	1993-01-13	AD_VP	17000	NULL	100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	1990-01-03	IT_PROG	9000	NULL	102	60
104	Bruce	Ernst	BERNST	590.423.4568	1991-05-21	IT_PROG	6000	NULL	103	60
105	David	Austin	DAUSTIN	590.423.4569	1997-06-25	IT_PROG	4800	NULL	103	60
106	Valli	Pataballa	VPATABAL	590.423.4560	1998-02-05	IT_PROG	4800	NULL	103	60

Table: departments

department_id	department_name	manager_id	location_id
50	Shipping	121	1500
60	IT	103	1400
70	Public Relations	204	2700
80	Sales	145	2500
90	Executive	100	1700

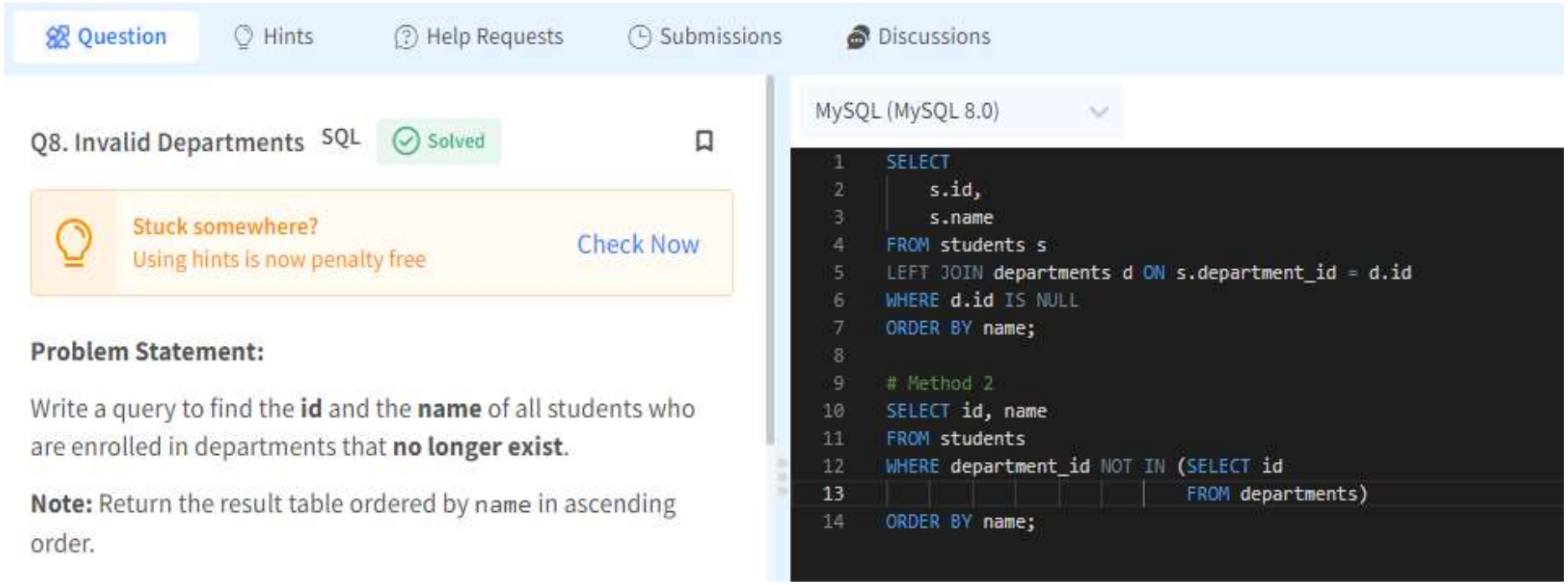
Table: locations

location_id	street_address	postal_code	city	state_province	country_id
1400	2014 Jabberwocky Rd	26192	Southlake	Texas	US
1500	2011 Interiors Blvd	99236	South San Francisco	California	US
1600	2007 Zagora St	50090	South Brunswick	New Jersey	US
1700	2004 Charade Rd	98199	Seattle	Washington	US
2400	8204 Arthur St	NULL	London	NULL	UK
2700	Schwanthalerstr. 7031	80925	Munich	Bavaria	DE

Sample Output:

employee_id	first_name	last_name	job_id
100	Steven	King	AD_PRES
101	Neena	Kochhar	AD_VP
102	Lex	De Haan	AD_VP

Q8. Invalid Departments



Sample Input:

Table: departments

id	name
1	Electrical Engineering
7	Computer Engineering
13	Bussiness Administration

Table: students

id	name	department_id
23	Alice	1
1	Bob	7
5	Jennifer	13
2	John	14
4	Jasmine	77
3	Steve	74
6	Luis	1
8	Jonathan	7
7	Daiana	33
11	Madelynn	1

Sample Output:

id	name
7	Daiana
4	Jasmine
2	John
3	Steve

Explanation:

- John, Daiana, Steve, and Jasmine are enrolled in departments 14, 33, 74, and 77 respectively.
- Department 14, 33, 74, and 77 do not exist anymore in the departments table.

Q9. Salary Bins

Question | Hints | Help Requests | Submissions | Discussions

Q9. Salary Bins SQL Solved

Stuck somewhere?
Using hints is now penalty free
Check Now

Problem Statement:

Based on the employee's salary, divide the employees into three different classes.

1. Salary **greater than** 20,000 (i.e, excluding 20,000) as '**Class A**'
2. Salary **between** 10,000 to 20,000 (i.e, including both 10,000 and 20,000) as '**Class B**'
3. Salary **less than** 10,000 (i.e, excluding 10,000) as '**Class C**'. Return the new column as 'Salary_bin'.

- Return the columns '**employee_id**', '**salary**', and '**Salary_bin**'.
- Return the result ordered by **employee_id** in ascending order.

MySQL (MySQL 8.0)

```mysql
SELECT
    employee_id,
    salary,
    CASE
        WHEN salary > 20000
        THEN "Class A"
        WHEN salary BETWEEN 10000 AND 20000
        THEN "Class B"
        WHEN salary < 10000
        THEN "Class C"
    END AS Salary_bin
FROM employees
ORDER BY employee_id;

# Method 2
select employee_id, salary,
    case
        when salary > 20000 then "Class A"
        when salary <=20000 and salary >=10000 then "Class B"
        when salary < 10000 then "Class C"
    end as Salary_bin from employees
order by employee_id;
```

Test Output Code's All Neat!

Test

Sample Input:

Table: employees

employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
100	Steven	King	SKING	515.123.4567	1987-06-17	AD_PRES	25000	NULL	NULL	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	1989-09-21	AD_VP	17000	NULL	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	1993-01-13	AD_VP	17000	NULL	100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	1990-01-03	IT_PROG	9000	NULL	102	60
104	Bruce	Ernst	BERNST	590.423.4568	1991-05-21	IT_PROG	6000	NULL	103	60
105	David	Austin	DAUSTIN	590.423.4569	1997-06-25	IT_PROG	4800	NULL	103	60

Sample Output:

employee_id	salary	Salary_bin
100	25000	Class A
101	17000	Class B
102	17000	Class B
103	9000	Class C
104	6000	Class C
105	4800	Class C