

Q1. Department (No emp)

Problem Statement:

Write a query to display the details of all those departments that **don't** have any working employees.

- Return the columns '**department_id**', and '**department_name**'.

Table: employees

Column Name	Type
employee_id	INT
employee_name	VARCHAR
Phone	VARCHAR
salary	INT
department_id	INT

Table: departments

Column Name	Type
department_id	INT
department_name	VARCHAR

SQL Schema:

Create table If Not Exists employees (employee_id int, employee_name varchar(255), phone varchar(20), salary int, department_id int);

Truncate table employees;

insert into employees (employee_id, employee_name, phone, salary, department_id) values ('1', 'Luis', '22995986404', '10023', '1');

insert into employees (employee_id, employee_name, phone, salary, department_id) values ('2', 'Den', '62795986600', '10903', '1');

insert into employees (employee_id, employee_name, phone, salary, department_id) values ('3', 'Alexander', '29475986399', '5346', '2');

insert into employees (employee_id, employee_name, phone, salary, department_id) values ('4', 'Shelli', '26445985238', '7606', '3');

create table if not Exists department (department_id int, department_name varchar(255));

Truncate table department;

insert into department (department_id, department_name) values (1, 'Sales');

insert into department (department_id, department_name) values (2, 'Marketing');

insert into department (department_id, department_name) values (3, 'HR');

insert into department (department_id, department_name) values (4, 'Finance');

insert into department (department_id, department_name) values (5, 'IT');

Sample Input:

Table: employees

employee_id	employee_name	Phone	salary	department_id
1	Luis	22995986404	10023	1
2	Den	6279878600	10903	1
3	Alexander	29473178399	5346	2
4	Shelli	26446335238	7606	3

Table: departments

department_id	department_name
1	Sales
2	Marketing
3	HR
4	Finance
5	IT

Sample Output:

department_id	department_name
4	Finance
5	IT

Problem Statement:

Write a query to report the **name** and **bonus** amount of each employee who gets a bonus of **less than 1000**.

- Return the bonus as '**NULL**' if there's an employee who doesn't get a bonus.
- Return the output ordered by the **bonus** in ascending order.

SQL Schema:

Create table If Not Exists Employee (empId int, name varchar(255), supervisor int, salary int);

Create table If Not Exists Bonus (empId int, bonus int);

Truncate table Employee;

insert into Employee (empId, name, supervisor, salary) values ('3', 'Brad', Null, '4000');

insert into Employee (empId, name, supervisor, salary) values ('1', 'John', '3', '1000');

insert into Employee (empId, name, supervisor, salary) values ('2', 'Dan', '3', '2000');

insert into Employee (empId, name, supervisor, salary) values ('4', 'Thomas', '3', '4000');

Truncate table Bonus;

insert into Bonus (empId, bonus) values ('2', '500');

insert into Bonus (empId, bonus) values ('4', '2000');

Sample Input:

Table: employee

empId	name	supervisor	salary
3	Brad	NULL	4000
1	John	3	1000
2	Dan	3	2000
4	Thomas	3	4000

Table: bonus

empId	bonus
2	500
4	2000

Sample Output:

name	bonus
Brad	NULL
John	NULL
Dan	500

Sample Explanation:

- Dan gets a bonus of 500 which is less than 1000.
- Thomas gets a bonus of 2000 which is greater than 1000.

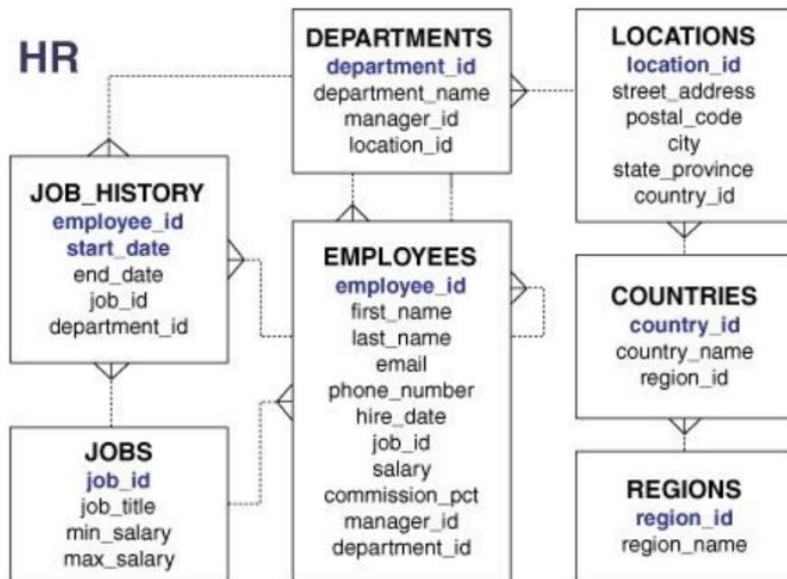
Q3. Joined first

Problem Statement:

Display the details of the employees who joined the company before their managers joined the company.

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

Dataset Description:



Sample Input:

Table: employees

employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_...	manager_id	department_id
193	Britney	Everett	BEVERETT	650.501.2876	1997-03-03	SH_CLERK	3900	NULL	123	50
195	Vance	Jones	VJONES	650.501.4876	1999-03-17	SH_CLERK	2800	NULL	123	50
196	Alana	Walsh	AWALSH	650.507.9811	1998-04-24	SH_CLERK	3100	NULL	124	50
197	Kevin	Feeney	KFEENEY	650.507.9822	1998-05-23	SH_CLERK	3000	NULL	124	50
198	Donald	OConnell	DOCONNEL	650.507.9833	1999-06-21	SH_CLERK	2600	NULL	124	50
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

Sample Output:

employee_id	first_name	last_name
193	Britney	Everett
196	Alana	Walsh
197	Kevin	Feeney
198	Donald	OConnell
200	Jennifer	Whalen

Problem Description:

Write a query to report all the sessions that **did not** get shown any ads.

Return the resultant table ordered by **session_id** in **ascending order**.

Table: **Playback**

Column Name	Type
session_id	INT
customer_id	INT
start_time	INT
end_time	INT

Table: **Ads**

Column Name	Type
ad_id	INT
customer_id	INT
timestamp	INT

SQL Schema:

```
Create table If Not Exists Playback (session_id int, customer_id int, start_time int, end_time int);
```

```
Truncate table Playback;
```

```
insert into Playback (session_id, customer_id, start_time, end_time) values ('1', '1', '1', '5');
```

```
insert into Playback (session_id, customer_id, start_time, end_time) values ('2', '1', '15', '23');
```

```
insert into Playback (session_id, customer_id, start_time, end_time) values ('3', '2', '10', '12');
```

```
insert into Playback (session_id, customer_id, start_time, end_time) values ('4', '2', '17', '28');
```

```
insert into Playback (session_id, customer_id, start_time, end_time) values ('5', '2', '2', '8');
```

```
Create table If Not Exists Ads (ad_id int, customer_id int, timestamp int);
```

```
Truncate table Ads;
```

```
insert into Ads (ad_id, customer_id, timestamp) values ('1', '1', '5');
```

```
insert into Ads (ad_id, customer_id, timestamp) values ('2', '2', '17');
```

```
insert into Ads (ad_id, customer_id, timestamp) values ('3', '2', '20');
```

Sample Input:

Table: **Playback**

session_id	customer_id	start_time	end_time
1	1	1	5
2	1	15	23
3	2	10	12
4	2	17	28
5	2	2	8

Table: **Ads**

ad_id	customer_id	timestamp
1	1	5
2	2	17
3	2	20

Sample Output:

session_id
2
3
5

Sample Explanation:

- The ad with ID 1 was shown to user 1 at time 5 while they were in session 1.
- The ad with ID 2 was shown to user 2 at time 17 while they were in session 4.
- The ad with ID 3 was shown to user 2 at time 20 while they were in session 4.
- We can see that sessions 1 and 4 had at least one ad.
- Sessions 2, 3, and 5 did not have any ads, so we return them.

Q5. Consecutive Seats

1 year - 2 years: Amazon (8)

Problem Statement:

Write a query to report **all** the consecutive available seats in the cinema.

- Return the result table ordered by `seat_id` in **ascending order**.

Note: In the table cinema, **1** means seats are available and **0** means that seats are not available.

SQL Schema:

Create table If Not Exists Cinema (seat_id int, free int);

Truncate table Cinema;

insert into Cinema (seat_id, free) values ('1', '1');

insert into Cinema (seat_id, free) values ('2', '0');

insert into Cinema (seat_id, free) values ('3', '1');

insert into Cinema (seat_id, free) values ('4', '1');

insert into Cinema (seat_id, free) values ('5', '1');

Sample Input:

Table: cinema

seat_id	free
1	1
2	0
3	1
4	1
5	1

Sample output:

seat_id
3
4
5

Explanation:

seats with seat_id 1, 3, 4, and 5 are available but only seat_id 3,4, and 5 have consecutive available seats in the cinema.

Q6. Manager left

Problem Description:

Write a query to find the **employee_id** of the employees whose salary is strictly **less than** \$15000 and whose manager left the company.

When a manager leaves the company, their information is deleted from the employees table, but the reports still have their manager_id set to the manager that left.

Note: Return the result ordered by **employee_id** in ascending order.

Table: employess

Column Name	Type
employee_id	INT
name	VARCHAR
manager_id	INT
salary	INT

SQL Schema:

Create table If Not Exists Employees (employee_id int, name varchar(20), manager_id int, salary int);

Truncate table Employees;

insert into Employees (employee_id, name, manager_id, salary) values ('1', 'Luis', '2', '14384');

insert into Employees (employee_id, name, manager_id, salary) values ('2', 'Den', 'None', '12208');

insert into Employees (employee_id, name, manager_id, salary) values ('3', 'Alexander', 'None', '9970');

insert into Employees (employee_id, name, manager_id, salary) values ('4', 'Shelli', '3', '6943');

insert into Employees (employee_id, name, manager_id, salary) values ('5', 'Sigal', '6', '5564');

insert into Employees (employee_id, name, manager_id, salary) values ('7', 'Guy', '6', '8120');

Sample Input:

Table: employees

employee_id	name	manager_id	salary
1	Luis	2	14384
2	Den	NULL	12208
3	Alexander	NULL	9970
4	Shelli	3	6943
5	Sigal	6	5564
7	Guy	6	8120

Sample Output:

employee_id
5
7

Sample Explanation:

- The employees with a salary less than \$15000 are 1, 2, 3, 4, 5, and 7.
- 1 manager is employee 2, who is still in the company.
- 2, 3 employees have manager_id as null as they are in the highest position in the company and they have no managers.
- 4 manager is employee 3, who is still in the company.
- 5, 7 manager is employee 6, who left the company because there is no row for employee 6 as it was deleted.
- So, 5,7 employees manager left and their salary is less than \$15000. Hence, that employee's **employee_id** is returned.

Q7. Department in US

Problem Statement:

Display the details of those employees who have a **manager** working in the department that is **US** (i.e., country_id) based.

Note:

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

Dataset Description is as question 3.

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

- manager_id** column represents the employee_id of the manager.

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
1100	93091 Calle della Testa	10934	Venice	NULL	IT
1200	2017 Shinjuku-ku	1689	Tokyo	Tokyo Prefect...	JP
1300	9450 Kamiya-cho	6823	Hiroshima	NULL	JP
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

Sample Output:

employee_id	first_name	last_name
101	Neena	Kochhar
102	Lex	De Haan
103	Alexander	Hunold
104	Bruce	Ernst
105	David	Austin

Q8. Shortest distance

Problem Description:

Write a query to find the shortest distance between any two points from the **points** table. Round the distance to two decimal points

Note:

- The distance between two points p1(x1, y1) and p2(x2, y2) is calculated using euclidean distance formula $\sqrt{(x2 - x1)^2 + (y2 - y1)^2}$.
- Save the new column as '**shortest**'.

SQL Schema:

Create table If Not Exists points (x int, y int);

Truncate table points;

insert into points (x, y) values (-1, -1);

insert into points (x, y) values (0, 0);

insert into points (x, y) values (-1, -2);

Sample Input:

Table: points

x	y
-1	-1
0	0
-1	-2

Sample Output:

shortest
1

Sample Explanation: The shortest distance is **1** from point (-1, -1) to (-1, 2).