Q1. Actors and Directors

6 months - 1 year: Amazon(2)

Problem Statement:

Write a query for a report that provides the pairs (actor_id, director_id) where the actor has cooperated with the director at least three times.

- Return the result table ordered by `actor_id` in ascending manner.

SQL Schema:

```
Create table If Not Exists ActorDirector (actor_id int, director_id int, timestamp int);
Truncate table ActorDirector;
insert into ActorDirector (actor_id, director_id, timestamp) values ('1', '1', '0');
insert into ActorDirector (actor_id, director_id, timestamp) values ('1', '1', '1');
insert into ActorDirector (actor_id, director_id, timestamp) values ('1', '1', '2');
insert into ActorDirector (actor_id, director_id, timestamp) values ('1', '2', '3');
insert into ActorDirector (actor_id, director_id, timestamp) values ('1', '2', '4');
insert into ActorDirector (actor_id, director_id, timestamp) values ('2', '1', '5');
insert into ActorDirector (actor_id, director_id, timestamp) values ('2', '1', '6');
```

Sample Input:

Table: actordirector

actor_id	director_id	timestamp
1	1	0
1	1	1
1	1	2
1	2	3
1	2	4
2	1	5
2	1	6

Sample output:

actor_id	director_id
1	1

Explanation:

The only pair is (1, 1) where they cooperated exactly 3 times.

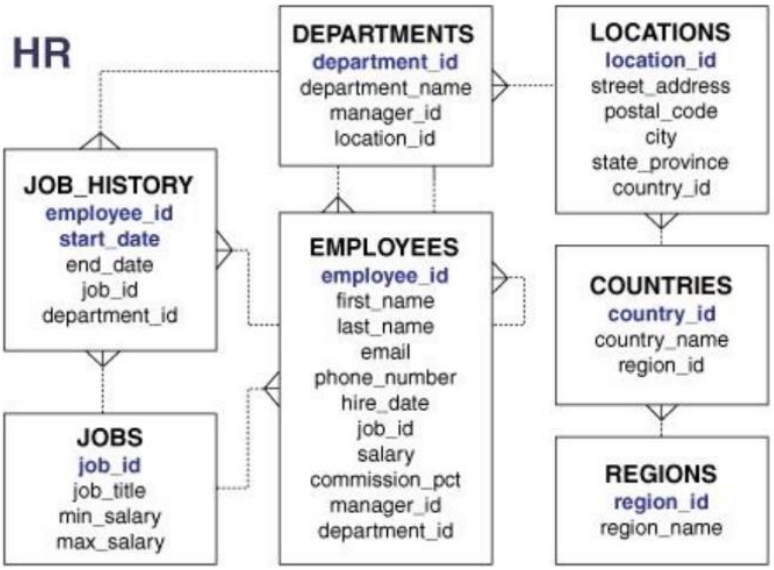
Q2. Average Salary

Problem Statement:

Write a query to find the **average salary** of the employees for **each** department.

- Save the new average salary as '**Average_salary**'.
- Return the columns '**department_id**', '**department_name**', and '**Average_salary**'.
- Return the result ordered by **department_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_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: 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

Sample Output:

department_id	department_name	Average_salary
60	IT	6600.0000
90	Executive	19666.6667

Q3. No. of employees

Problem Statement:

Write a query to display the count of employees as 'No_of_Employees' and, the total salary paid to employees as 'Total_Salary' for each department.

- Return the columns '**department_name**', '**No_of_Employees**', and '**Total_Salary**'.
- Return the output ordered by **department_name** in ascending order.

Note:

- If there are no employees in the department return the No_of_Employees as **0** and Total_Salary as **NULL**.

Dataset Description is the same as previous question.

Sample Input:

Table: employees

employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
200	Jennifer	Whalen	JWHALEN	515.123.4444	1987-09-17	AD_ASST	4400	NULL	101	10
203	Susan	Mavris	SMAVRIS	515.123.7777	1994-06-07	HR_REP	6500	NULL	101	40
205	Shelley	Higgins	SHIGGINS	515.123.8080	1994-06-07	AC_MGR	12000	NULL	101	110
206	William	Gietz	WGIETZ	51hr5.123.8181	1994-06-07	AC_ACCOUNT	8300	NULL	205	110

Table: departments

department_id	department_name	manager_id	location_id
10	Administration	200	1700
40	Human Resources	203	2400
110	Accounting	205	1700
160	Benefits	NULL	1700
180	Construction	NULL	1700

Sample Output:

department_name	No_of_Employees	Total_Salary
Accounting	2	20300
Administration	1	4400
Benefits	0	NULL
Construction	0	NULL
Human Resources	1	6500

Q4. Human Resources

Problem Statement:

Write a query to find details of the employees who work in the '**Human Resources**' department.

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

Dataset Description is the same as previous question

Sample Input:

Table: employees

employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	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
204	Hermann	Baer	HBAER	515.123.8888	1994-06-07	PR_REP	10000	NULL	101	70

Table: departments

department_id	department_name	manager_id	location_id
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700
40	Human Resources	203	2400
50	Shipping	121	1500
70	Public Relations	204	2700

Sample Output:

employee_id	department_id	first_name	last_name	job_id	department_name
203	40	Susan	Mavris	HR_REP	Human Resources

Q5. Product Sales I

1 year - 2 years -Amazon

SQL Schema:

```
Create table If Not Exists Sales (sale_id int, product_id int, year int, quantity int, price int);
Create table If Not Exists Product (product_id int, product_name varchar(10));
Truncate table Sales;
insert into Sales (sale_id, product_id, year, quantity, price) values ('1', '100', '2008', '10', '5000');
insert into Sales (sale_id, product_id, year, quantity, price) values ('2', '100', '2009', '12', '5000');
insert into Sales (sale_id, product_id, year, quantity, price) values ('7', '200', '2011', '15', '9000');
Truncate table Product;
insert into Product (product_id, product_name) values ('100', 'Nokia');
insert into Product (product_id, product_name) values ('200', 'Apple');
insert into Product (product_id, product_name) values ('300', 'Samsung');
```

Problem Statement:

Write a SQL query that reports the **product_name**, **year**, and **price** for each **sale_id** in the `sales` table.

- Return the result table ordered by **year** and **product_name** in ascending order.

Sample Input:

Table: sales

sale_id	product_id	year	quantity	price
1	100	2008	10	5000
2	100	2009	12	5000
7	200	2011	15	9000

Table: product

product_id	product_name
100	Nokia
200	Apple
300	Samsung

Sample Output:

product_name	year	price
Nokia	2008	5000
Nokia	2009	5000
Apple	2011	9000

Explanation:

- From sale_id = 1, we can conclude that Nokia was sold for 5000 in the year 2008.
- From sale_id = 2, we can conclude that Nokia was sold for 5000 in the year 2009.
- From sale_id = 7, we can conclude that Apple was sold for 9000 in the year 2011.

Q6. Products Recommendation

SQL Schema:

```
create database product_recommendation;
use product_recommendation;
Create table If Not Exists customers (customer_id int, customer_name varchar(255));
Truncate table customers;
insert into customers (customer_id, customer_name) values ('1', 'Andrew');
insert into customers (customer_id, customer_name) values ('2', 'Erin');
insert into customers (customer_id, customer_name) values ('3', 'Stanley');
Create table If Not Exists orders (order_id int, customer_id int, product_name
varchar(255));
Truncate table orders;
insert into orders (order_id, customer_id, product_name) values ('10', '1', 'Bread');
insert into orders (order_id, customer_id, product_name) values ('20', '1', 'Milk');
insert into orders (order_id, customer_id, product_name) values ('30', '1', 'Butter');
insert into orders (order_id, customer_id, product_name) values ('40', '1', 'Eggs');
insert into orders (order_id, customer_id, product_name) values ('50', '2', 'Bread');
insert into orders (order_id, customer_id, product_name) values ('60', '2', 'Milk');
insert into orders (order_id, customer_id, product_name) values ('70', '3', 'Butter');
```

Problem Description:

Write a query to find the **customer_id** and **customer_name** of customers who bought products "**Bread**" and "**Milk**" but did not buy the product "**Eggs**".

- Return the output ordered by **customer_name** in ascending order

Sample Input:

Table: customers

customer_id	customer_name
1	Andrew
2	Erin
3	Stanley

Table: orders

order_id	customer_id	product_name
10	1	Bread
20	1	Milk
30	1	Butter
40	1	Eggs
50	2	Bread
60	2	Milk
70	3	Butter

Sample Output:

customer_id	customer_name
2	Erin

Sample Explanation:

Here, only the customer_id with id 2 bought Bread and Milk products but did not buy the Eggs.

Q7. Europe

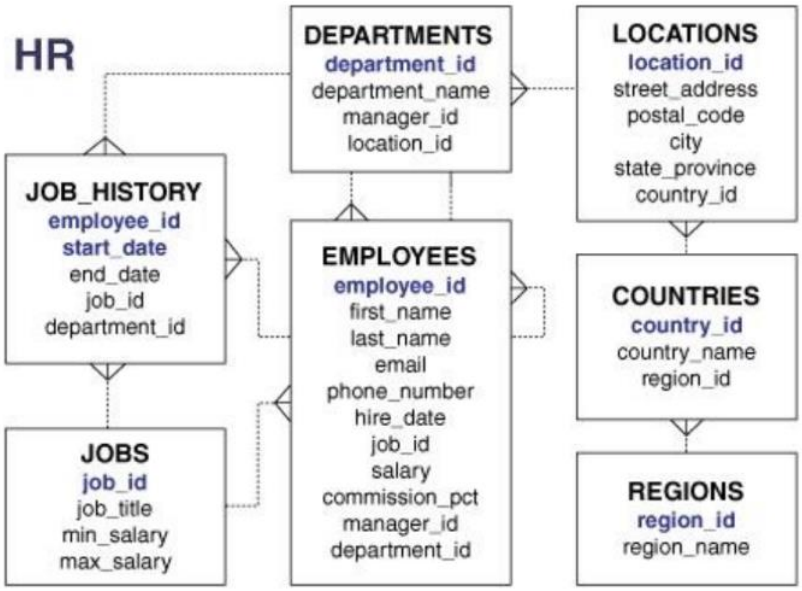
Problem Statement:

Write a query to display the details of the employees who belong to the **'Europe'** region.

Note:

- Return the columns '**employee_id**', '**full_name**' (first_name and last_name separated by space), '**salary**', '**phone_number**', '**department_id**', '**department_name**', '**street_address**', '**city**', '**country_name**', '**region_id**', '**region_name**'.
- Return the result ordered by **salary** in descending order and 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
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
204	Hermann	Baer	HBAER	515.123.8888	1994-06-07	PR_REP	10000	NULL	101	70
205	Shelley	Higgins	SHIGGINS	515.123.8080	1994-06-07	AC_MGR	12000	NULL	101	110

Table: departments

department_id	department_name	manager_id	location_id
40	Human Resources	203	2400
50	Shipping	121	1500
70	Public Relations	204	2700

Table: locations

location_id	street_address	postal_code	city	state_province	country_id
2100	1298 Vileparle (E)	490231	Bombay	Maharashtra	IN
2200	12-98 Victoria Street	2901	Sydney	New South Wales	AU
2300	198 Clementi North	540198	Singapore	NULL	SG
2400	8204 Arthur St	NULL	London	NULL	UK

Table: countries

country_id	country_name	region_id
DE	Germany	1
IN	India	3
UK	United Kingdom	1

Table: regions

region_id	region_name
1	Europe
2	Americas
3	Asia
4	Middle East and Africa

Sample Output:

employee_id	full_name	salary	phone_number	department_id	department_name	street_address	city	country_name	region_id	region_name
204	Hermann Baer	10000	515.123.8888	70	Public Relations	Schwanthalerstr. 7031	Munich	Germany	1	Europe
203	Susan Mavris	6500	515.123.7777	40	Human Resources	8204 Arthur St	London	United Kingdom	1	Europe