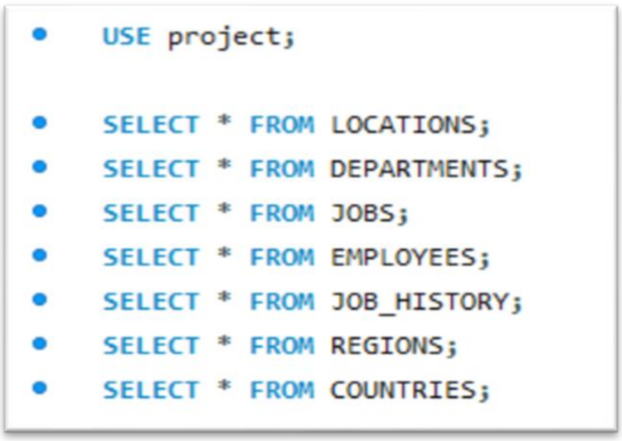
EDA = Exploratory Data Analysis

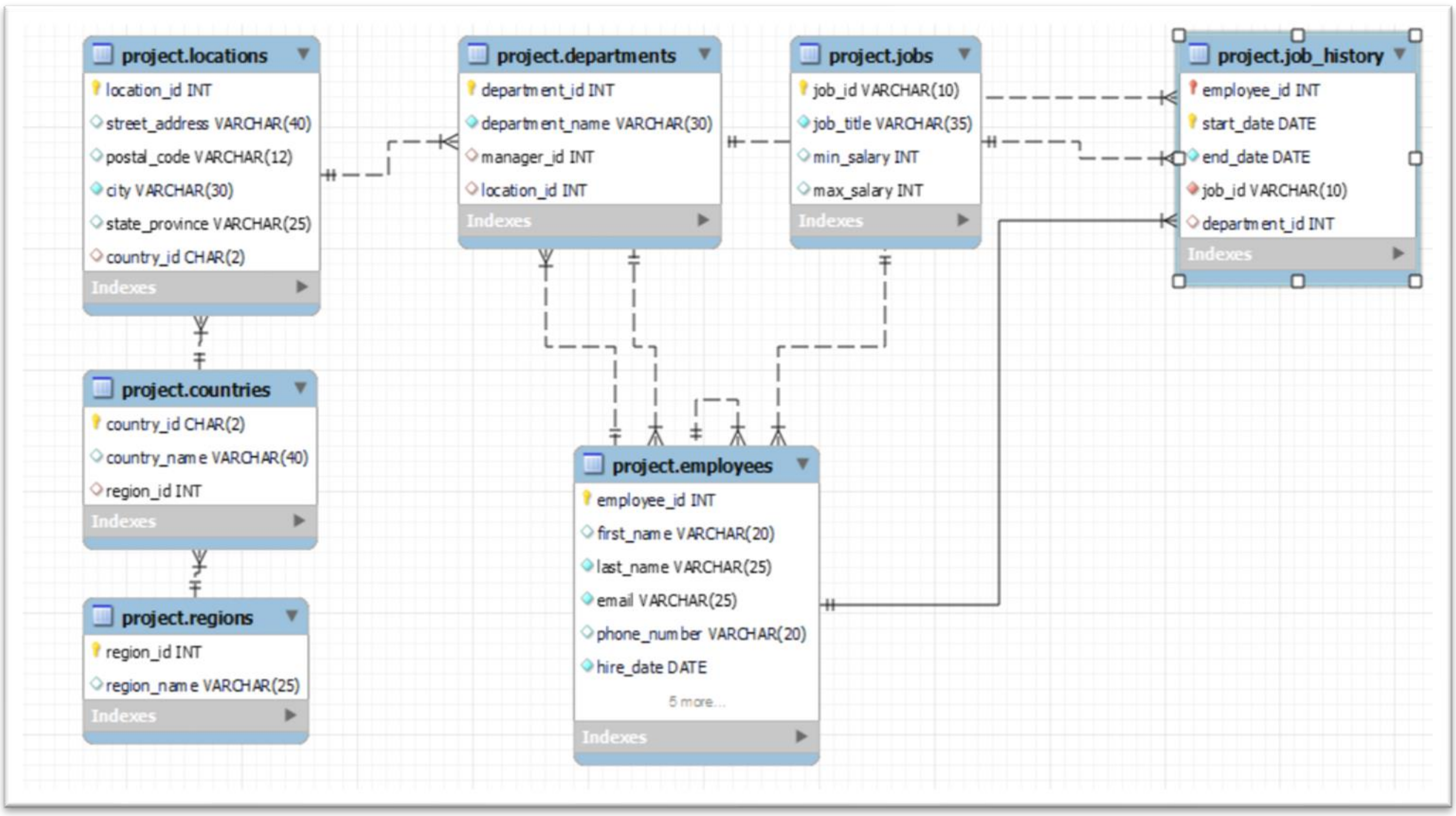
OBJECTIVE : To get the grasp on data sets and extract meaningful information from their historical patterns that further helps business team to perform required actions to meet business targets or enhance KPIs (Key Performance Indicators - to measure the performance of the tasks/targets).

Pre-processing : Importing dataset and viewing all tables.

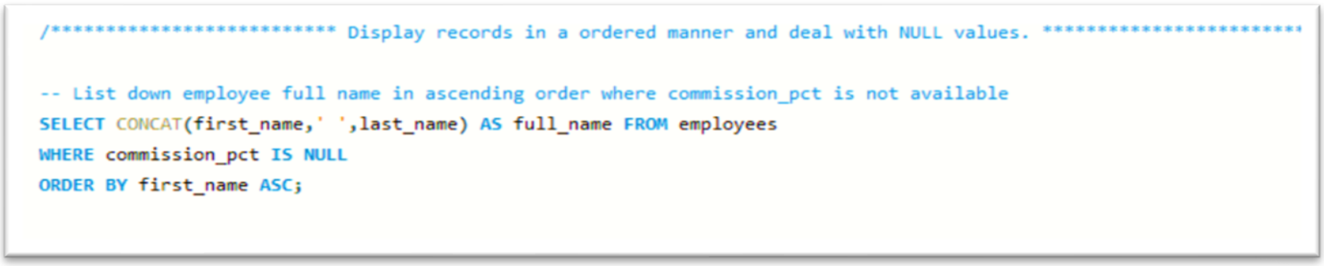


```
USE project;

SELECT * FROM LOCATIONS;
SELECT * FROM DEPARTMENTS;
SELECT * FROM JOBS;
SELECT * FROM EMPLOYEES;
SELECT * FROM JOB_HISTORY;
SELECT * FROM REGIONS;
SELECT * FROM COUNTRIES;
```

ER Diagram :

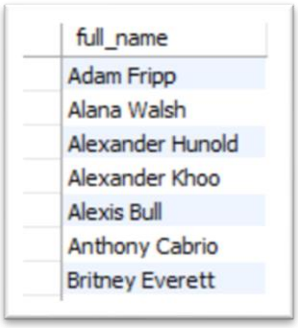


/* Display records in a ordered manner and deal with NULL values. ***/**



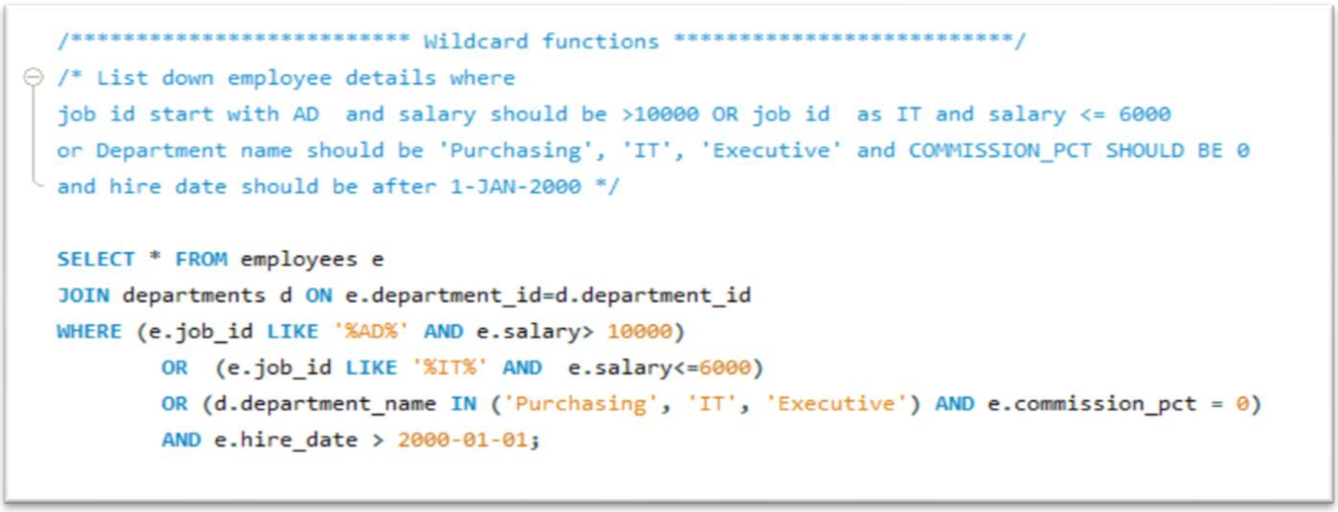
```sql
/************************** Display records in a ordered manner and deal with NULL values. ************************

-- List down employee full name in ascending order where commission_pct is not available
SELECT CONCAT(first_name,' ',last_name) AS full_name FROM employees
WHERE commission_pct IS NULL
ORDER BY first_name ASC;
```

Output :



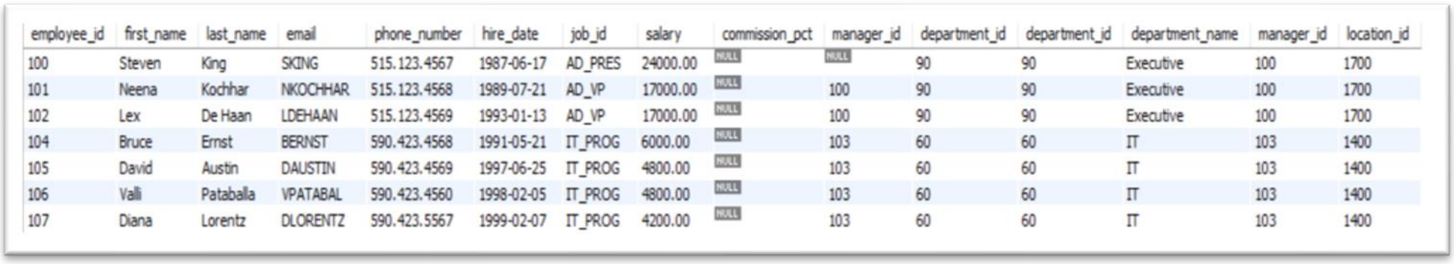
full_name
Adam Fripp
Alana Walsh
Alexander Hunold
Alexander Khoo
Alexis Bull
Anthony Cabrio
Britney Everett

/************************ Wildcard functions **************************/**



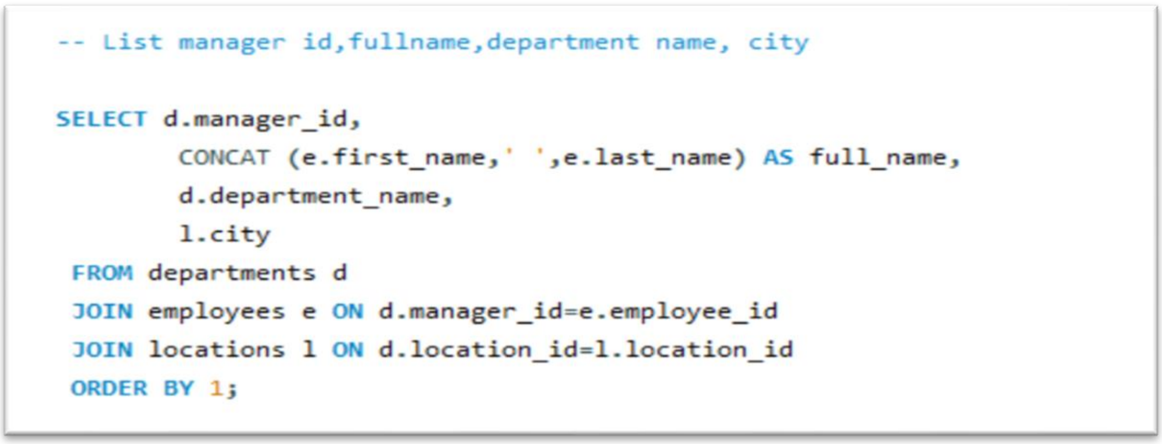
```sql
/************************** Wildcard functions **************************/
/* List down employee details where
job id start with AD  and salary should be >10000 OR job id  as IT and salary <= 6000
or Department name should be 'Purchasing', 'IT', 'Executive' and COMMISSION_PCT SHOULD BE 0
and hire date should be after 1-JAN-2000 */

SELECT * FROM employees e
JOIN departments d ON e.department_id=d.department_id
WHERE (e.job_id LIKE '%AD%' AND e.salary> 10000)
        OR  (e.job_id LIKE '%IT%' AND  e.salary<=6000)
        OR (d.department_name IN ('Purchasing', 'IT', 'Executive') AND e.commission_pct = 0)
        AND e.hire_date > 2000-01-01;
```

Output :



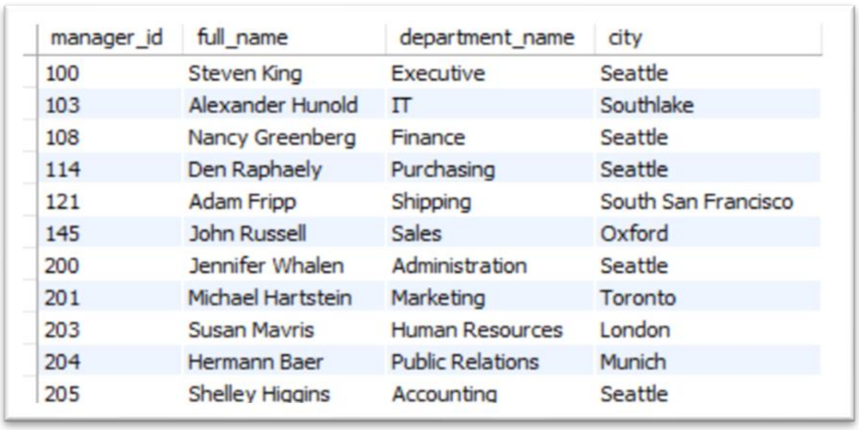
employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id	department_id	department_name	manager_id	location_id
100	Steven	King	SKING	515.123.4567	1987-06-17	AD_PRES	24000.00	NULL	NULL	90	90	Executive	100	1700
101	Neena	Kochhar	NKOCHHAR	515.123.4568	1989-07-21	AD_VP	17000.00	NULL	100	90	90	Executive	100	1700
102	Lex	De Haan	LDEHAAN	515.123.4569	1993-01-13	AD_VP	17000.00	NULL	100	90	90	Executive	100	1700
104	Bruce	Ernst	BERNST	590.423.4568	1991-05-21	IT_PROG	6000.00	NULL	103	60	60	IT	103	1400
105	David	Austin	DAUSTIN	590.423.4569	1997-06-25	IT_PROG	4800.00	NULL	103	60	60	IT	103	1400
106	Valli	Pataballa	VPATABAL	590.423.4560	1998-02-05	IT_PROG	4800.00	NULL	103	60	60	IT	103	1400
107	Diana	Lorentz	DLORENTZ	590.423.5567	1999-02-07	IT_PROG	4200.00	NULL	103	60	60	IT	103	1400

/************************ **Inner Join: display employee details.** *********************/



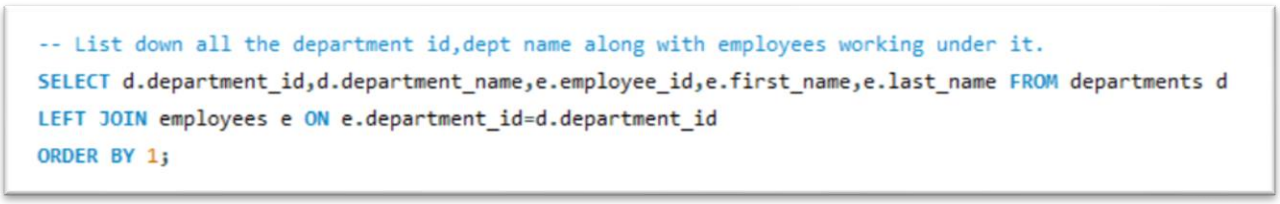
```sql
-- List manager id,fullname,department name, city

SELECT d.manager_id,
        CONCAT (e.first_name,' ',e.last_name) AS full_name,
        d.department_name,
        l.city
 FROM departments d
 JOIN employees e ON d.manager_id=e.employee_id
 JOIN locations l ON d.location_id=l.location_id
 ORDER BY 1;
```

Output :



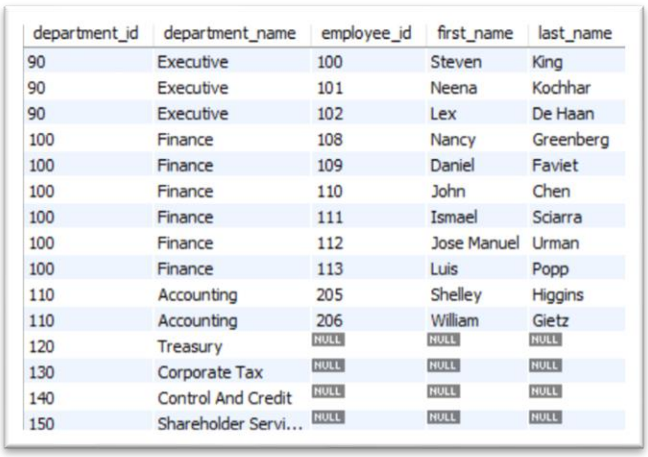
manager_id	full_name	department_name	city
100	Steven King	Executive	Seattle
103	Alexander Hunold	IT	Southlake
108	Nancy Greenberg	Finance	Seattle
114	Den Raphaely	Purchasing	Seattle
121	Adam Fripp	Shipping	South San Francisco
145	John Russell	Sales	Oxford
200	Jennifer Whalen	Administration	Seattle
201	Michael Hartstein	Marketing	Toronto
203	Susan Mavris	Human Resources	London
204	Hermann Baer	Public Relations	Munich
205	Shelley Higgins	Accounting	Seattle

/************************ **Left Join** ************************/



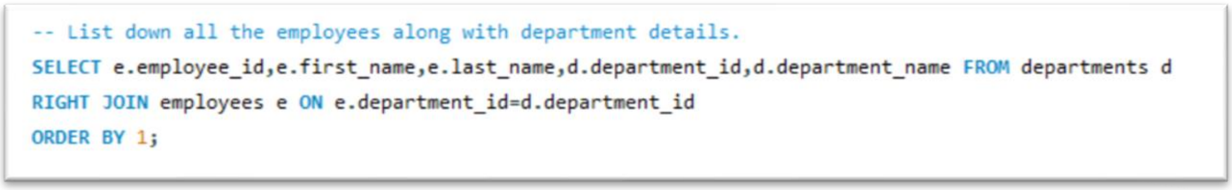
```sql
-- List down all the department id,dept name along with employees working under it.
SELECT d.department_id,d.department_name,e.employee_id,e.first_name,e.last_name FROM departments d
LEFT JOIN employees e ON e.department_id=d.department_id
ORDER BY 1;
```

Output:



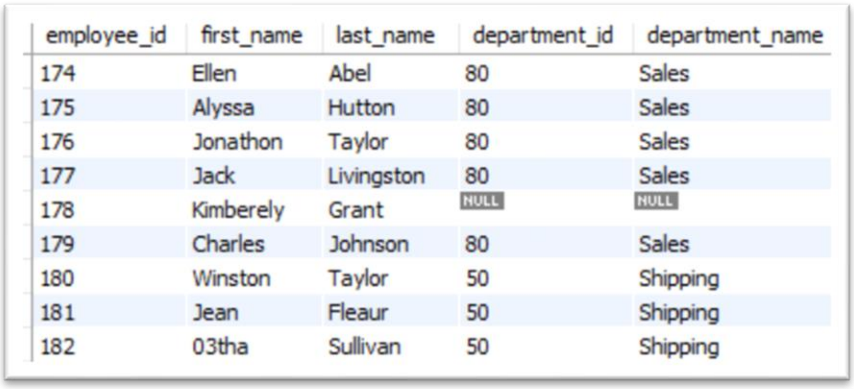
department_id	department_name	employee_id	first_name	last_name
90	Executive	100	Steven	King
90	Executive	101	Neena	Kochhar
90	Executive	102	Lex	De Haan
100	Finance	108	Nancy	Greenberg
100	Finance	109	Daniel	Faviet
100	Finance	110	John	Chen
100	Finance	111	Ismael	Sciarra
100	Finance	112	Jose Manuel	Urman
100	Finance	113	Luis	Popp
110	Accounting	205	Shelley	Higgins
110	Accounting	206	William	Gietz
120	Treasury	NULL	NULL	NULL
130	Corporate Tax	NULL	NULL	NULL
140	Control And Credit	NULL	NULL	NULL
150	Shareholder Servi...	NULL	NULL	NULL

/************************* Right Join *************************/



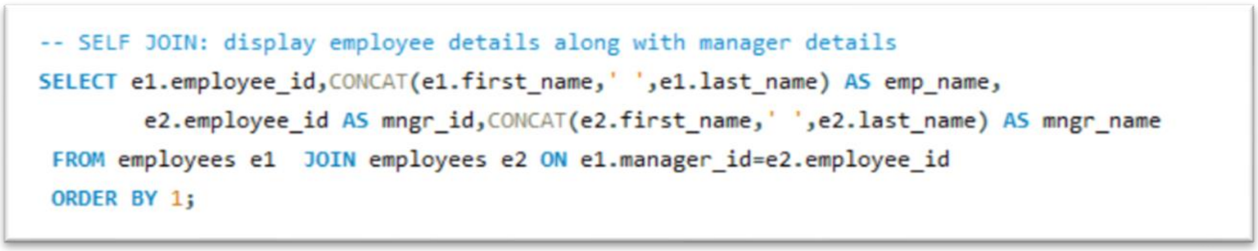
```
-- List down all the employees along with department details.
SELECT e.employee_id,e.first_name,e.last_name,d.department_id,d.department_name FROM departments d
RIGHT JOIN employees e ON e.department_id=d.department_id
ORDER BY 1;
```

Output :



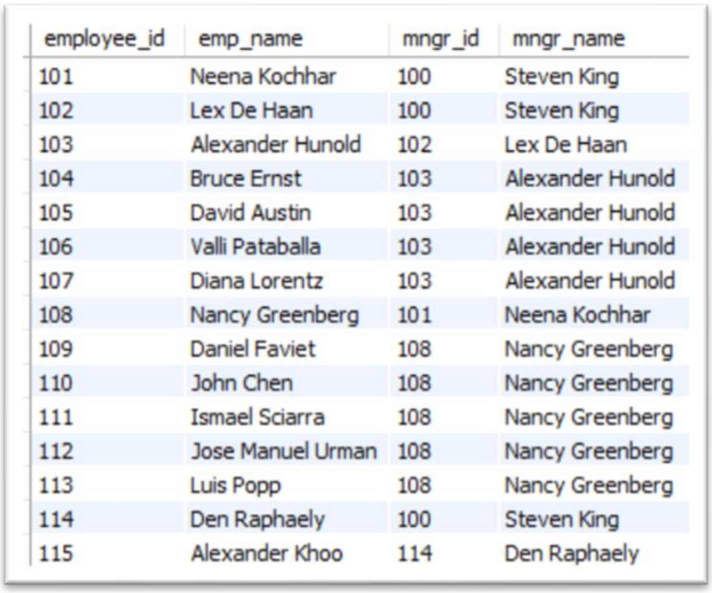
employee_id	first_name	last_name	department_id	department_name
174	Ellen	Abel	80	Sales
175	Alyssa	Hutton	80	Sales
176	Jonathon	Taylor	80	Sales
177	Jack	Livingston	80	Sales
178	Kimberely	Grant	NULL	NULL
179	Charles	Johnson	80	Sales
180	Winston	Taylor	50	Shipping
181	Jean	Fleaur	50	Shipping
182	03tha	Sullivan	50	Shipping

/************************* Self Join *************************/



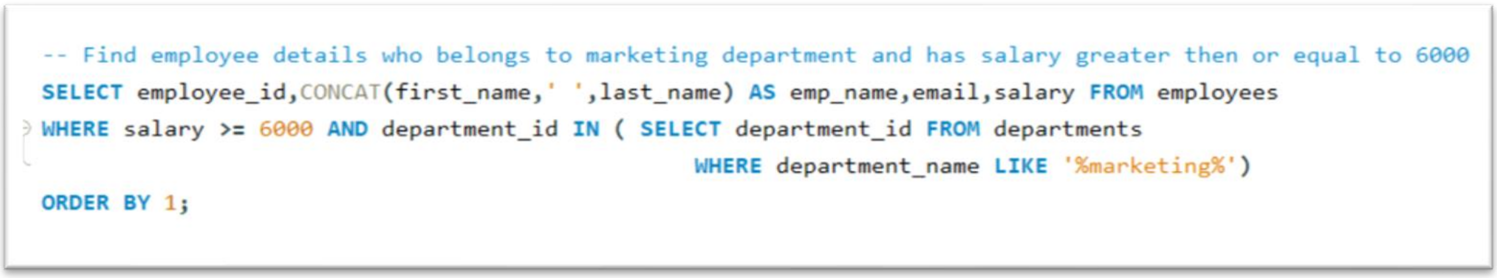
```
-- SELF JOIN: display employee details along with manager details
SELECT e1.employee_id,CONCAT(e1.first_name,' ',e1.last_name) AS emp_name,
        e2.employee_id AS mngr_id,CONCAT(e2.first_name,' ',e2.last_name) AS mngr_name
 FROM employees e1  JOIN employees e2 ON e1.manager_id=e2.employee_id
 ORDER BY 1;
```

Output :



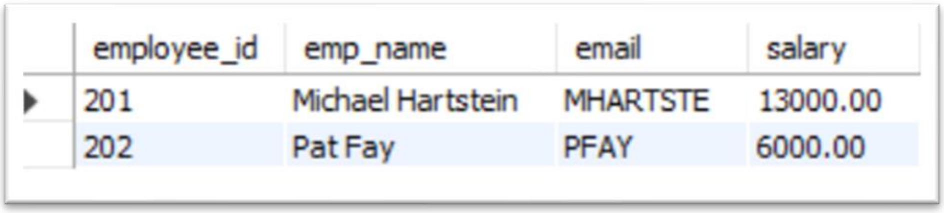
employee_id	emp_name	mngr_id	mngr_name
101	Neena Kochhar	100	Steven King
102	Lex De Haan	100	Steven King
103	Alexander Hunold	102	Lex De Haan
104	Bruce Ernst	103	Alexander Hunold
105	David Austin	103	Alexander Hunold
106	Valli Pataballa	103	Alexander Hunold
107	Diana Lorentz	103	Alexander Hunold
108	Nancy Greenberg	101	Neena Kochhar
109	Daniel Faviet	108	Nancy Greenberg
110	John Chen	108	Nancy Greenberg
111	Ismael Sciarra	108	Nancy Greenberg
112	Jose Manuel Urman	108	Nancy Greenberg
113	Luis Popp	108	Nancy Greenberg
114	Den Raphaely	100	Steven King
115	Alexander Khoo	114	Den Raphaely

/************************** Sub Query **************************/

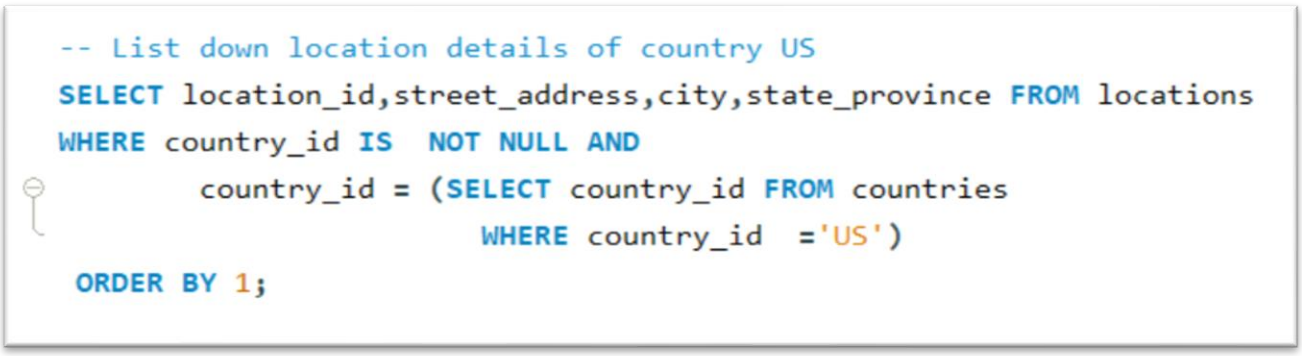


```sql
-- Find employee details who belongs to marketing department and has salary greater then or equal to 6000
SELECT employee_id,CONCAT(first_name,' ',last_name) AS emp_name,email,salary FROM employees
WHERE salary >= 6000 AND department_id IN ( SELECT department_id FROM departments
                                                WHERE department_name LIKE '%marketing%')
ORDER BY 1;
```

Output :

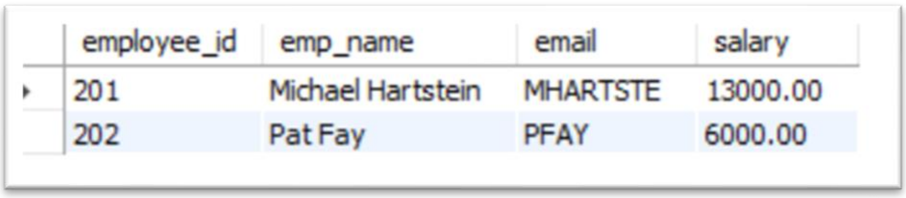


employee_id	emp_name	email	salary
201	Michael Hartstein	MHARTSTE	13000.00
202	Pat Fay	PFAY	6000.00



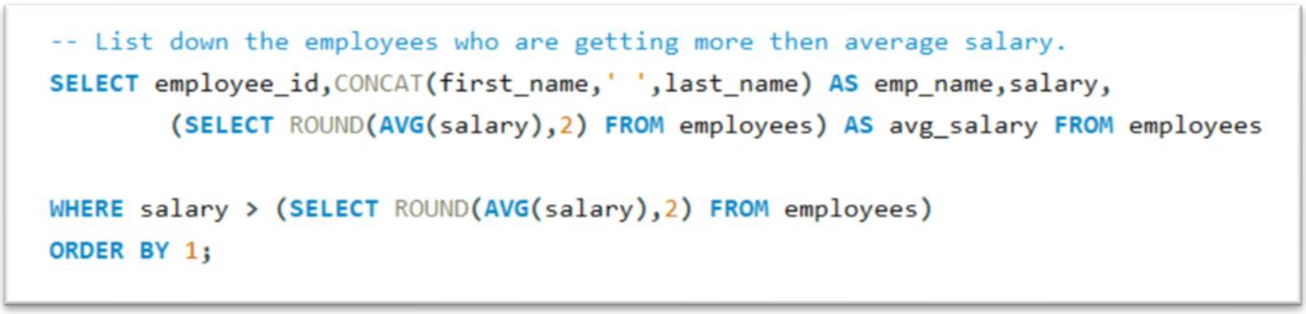
```sql
-- List down location details of country US
SELECT location_id,street_address,city,state_province FROM locations
WHERE country_id IS  NOT NULL AND
        country_id = (SELECT country_id FROM countries
                        WHERE country_id  ='US')
 ORDER BY 1;
```

Output :



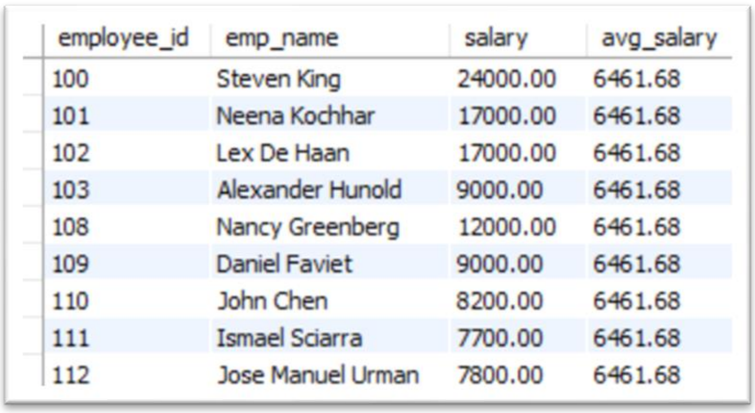
employee_id	emp_name	email	salary
201	Michael Hartstein	MHARTSTE	13000.00
202	Pat Fay	PFAY	6000.00

/************************* Aggregate Functions*************************/

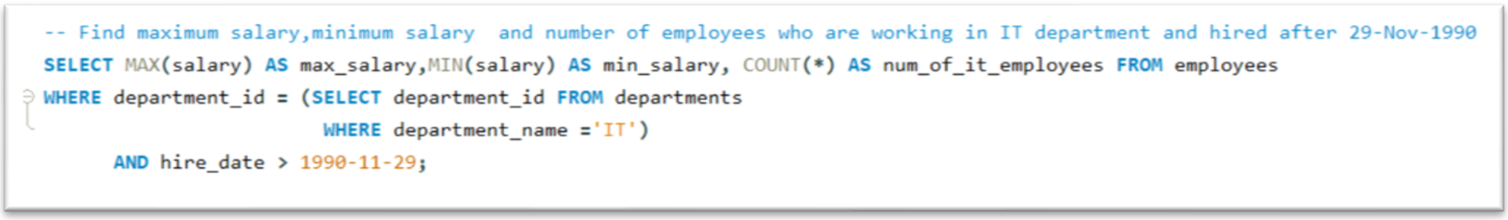


```
-- List down the employees who are getting more then average salary.
SELECT employee_id,CONCAT(first_name,' ',last_name) AS emp_name,salary,
       (SELECT ROUND(AVG(salary),2) FROM employees) AS avg_salary FROM employees

WHERE salary > (SELECT ROUND(AVG(salary),2) FROM employees)
ORDER BY 1;
```

Output :

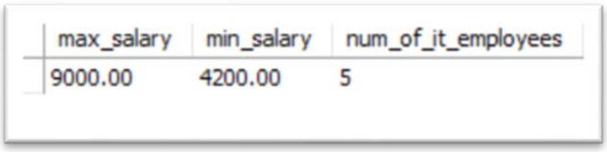


employee_id	emp_name	salary	avg_salary
100	Steven King	24000.00	6461.68
101	Neena Kochhar	17000.00	6461.68
102	Lex De Haan	17000.00	6461.68
103	Alexander Hunold	9000.00	6461.68
108	Nancy Greenberg	12000.00	6461.68
109	Daniel Faviet	9000.00	6461.68
110	John Chen	8200.00	6461.68
111	Ismael Sciarra	7700.00	6461.68
112	Jose Manuel Urman	7800.00	6461.68

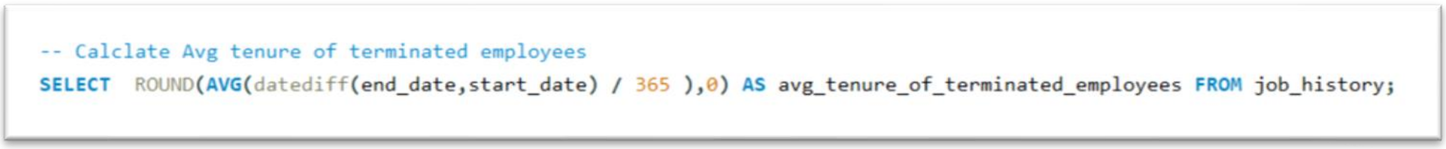


```
-- Find maximum salary,minimum salary  and number of employees who are working in IT department and hired after 29-Nov-1990
SELECT MAX(salary) AS max_salary,MIN(salary) AS min_salary, COUNT(*) AS num_of_it_employees FROM employees
WHERE department_id = (SELECT department_id FROM departments
                       WHERE department_name ='IT')
     AND hire_date > 1990-11-29;
```

Output :

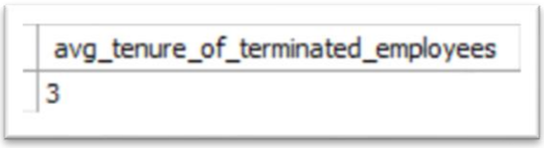


max_salary	min_salary	num_of_it_employees
9000.00	4200.00	5



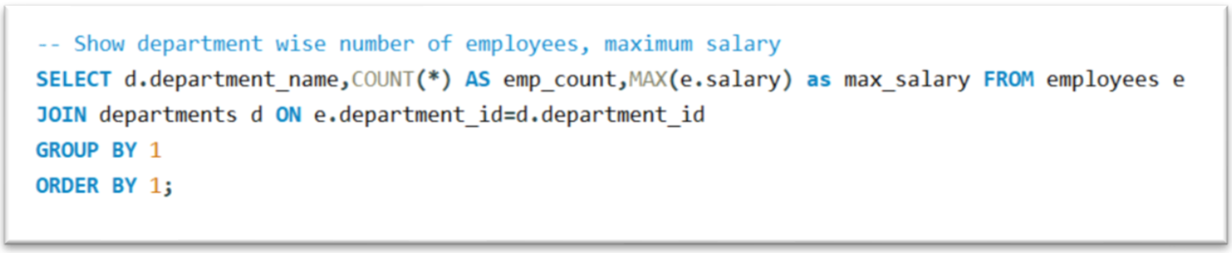
```
-- Calclate Avg tenure of terminated employees
SELECT  ROUND(AVG(datediff(end_date,start_date) / 365 ),0) AS avg_tenure_of_terminated_employees FROM job_history;
```

Output :



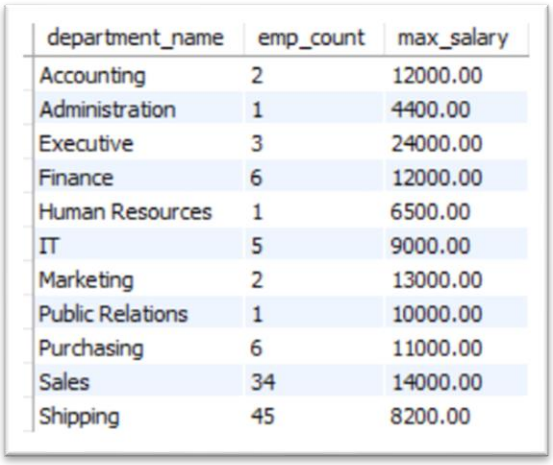
avg_tenure_of_terminated_employees
3

/************************* Group By *************************/

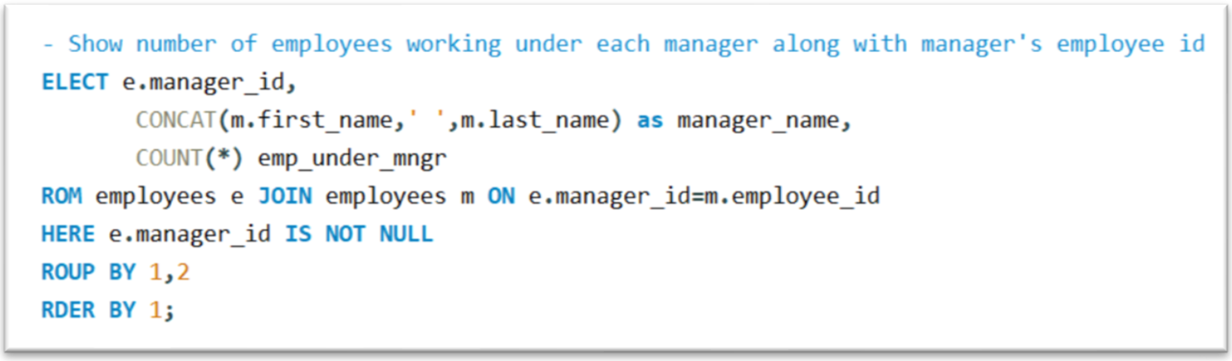


```
-- Show department wise number of employees, maximum salary
SELECT d.department_name,COUNT(*) AS emp_count,MAX(e.salary) as max_salary FROM employees e
JOIN departments d ON e.department_id=d.department_id
GROUP BY 1
ORDER BY 1;
```

Output :

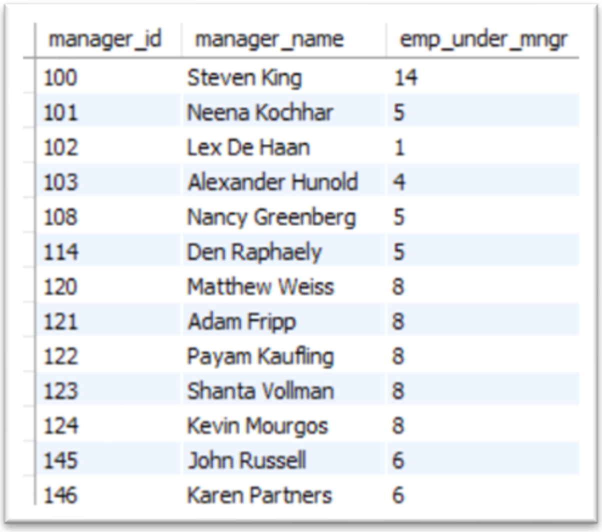


department_name	emp_count	max_salary
Accounting	2	12000.00
Administration	1	4400.00
Executive	3	24000.00
Finance	6	12000.00
Human Resources	1	6500.00
IT	5	9000.00
Marketing	2	13000.00
Public Relations	1	10000.00
Purchasing	6	11000.00
Sales	34	14000.00
Shipping	45	8200.00

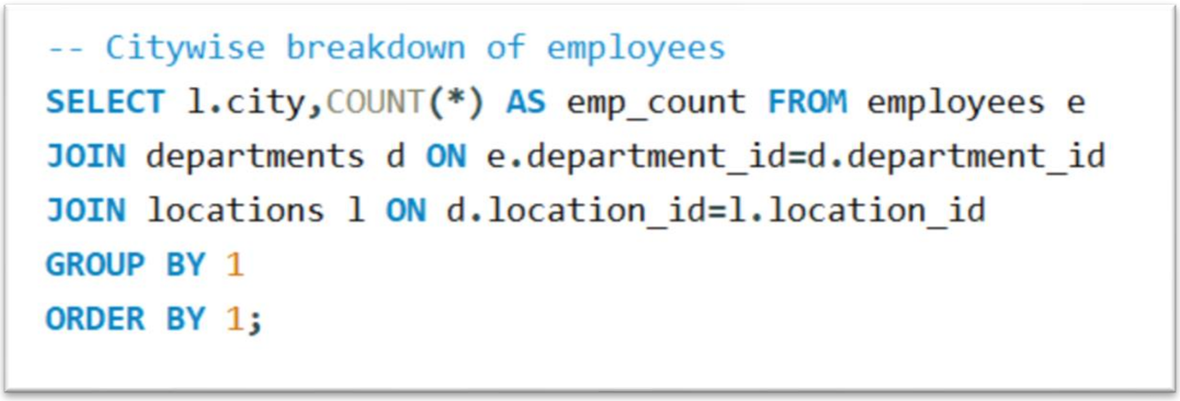


```
- Show number of employees working under each manager along with manager's employee id
ELECT e.manager_id,
       CONCAT(m.first_name,' ',m.last_name) as manager_name,
       COUNT(*) emp_under_mngr
ROM employees e JOIN employees m ON e.manager_id=m.employee_id
HERE e.manager_id IS NOT NULL
ROUP BY 1,2
RDER BY 1;
```

Output :

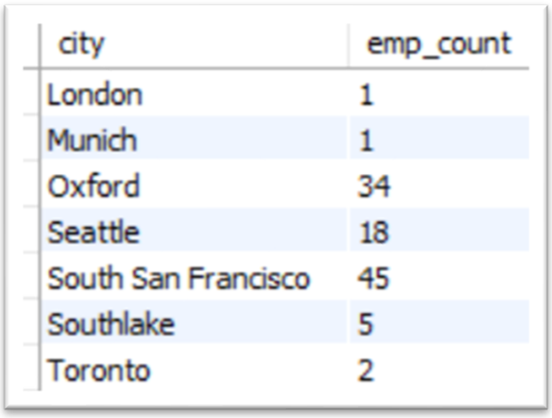


manager_id	manager_name	emp_under_mngr
100	Steven King	14
101	Neena Kochhar	5
102	Lex De Haan	1
103	Alexander Hunold	4
108	Nancy Greenberg	5
114	Den Raphaely	5
120	Matthew Weiss	8
121	Adam Fripp	8
122	Payam Kaufling	8
123	Shanta Vollman	8
124	Kevin Mourgos	8
145	John Russell	6
146	Karen Partners	6

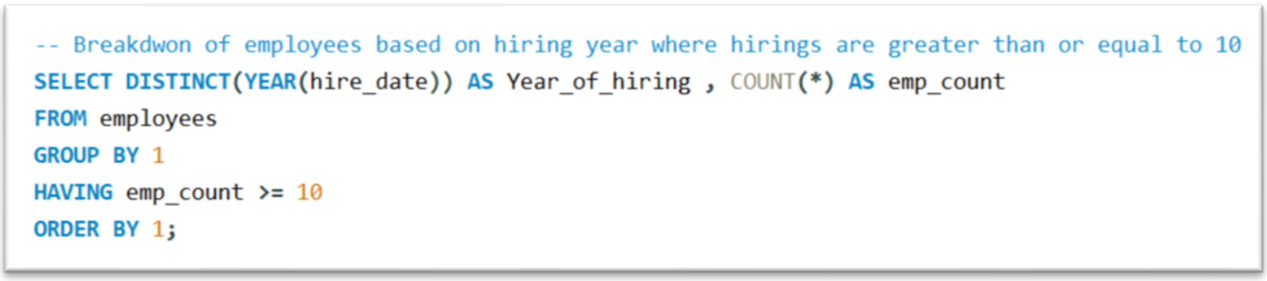


```sql
-- Citywise breakdown of employees
SELECT l.city,COUNT(*) AS emp_count FROM employees e
JOIN departments d ON e.department_id=d.department_id
JOIN locations l ON d.location_id=l.location_id
GROUP BY 1
ORDER BY 1;
```

Output :

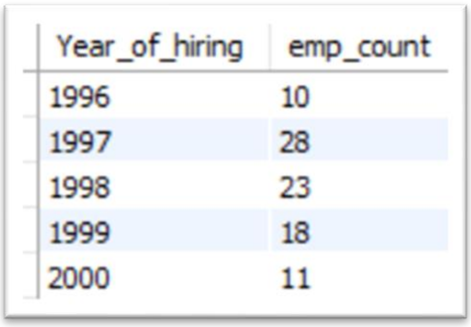


city	emp_count
London	1
Munich	1
Oxford	34
Seattle	18
South San Francisco	45
Southlake	5
Toronto	2



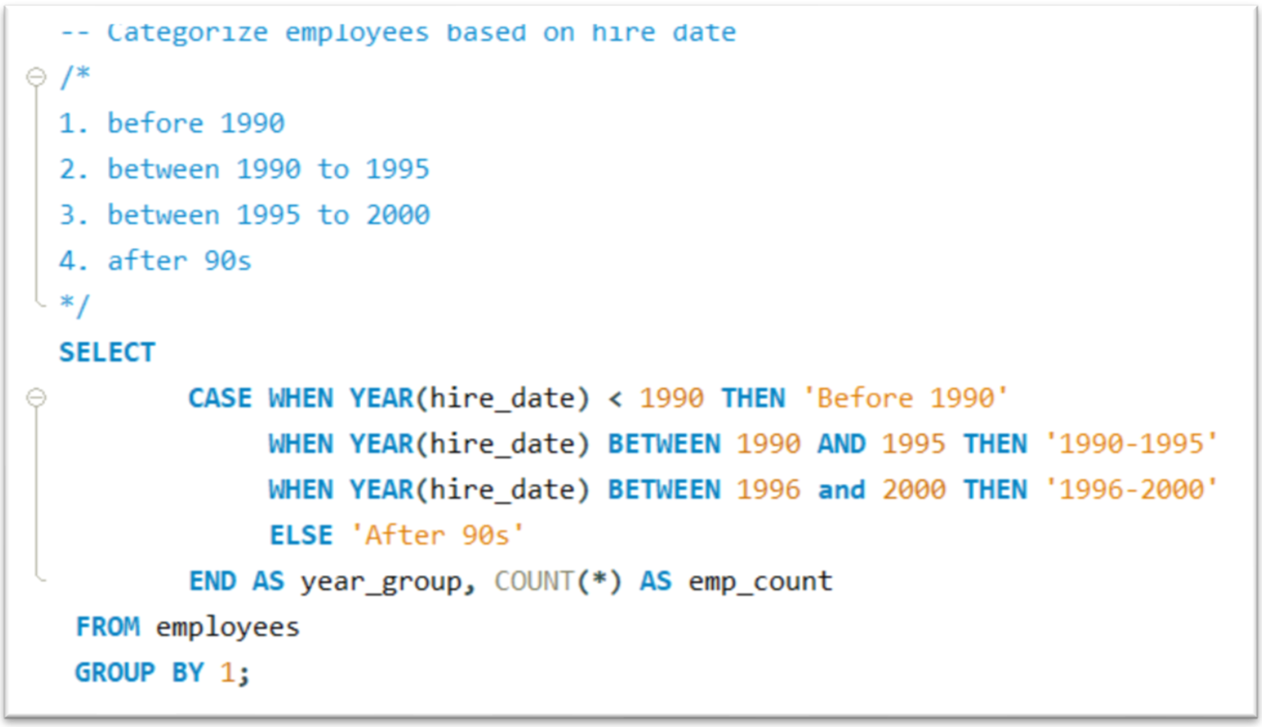
```sql
-- Breakdwon of employees based on hiring year where hirings are greater than or equal to 10
SELECT DISTINCT(YEAR(hire_date)) AS Year_of_hiring , COUNT(*) AS emp_count
FROM employees
GROUP BY 1
HAVING emp_count >= 10
ORDER BY 1;
```

Output :



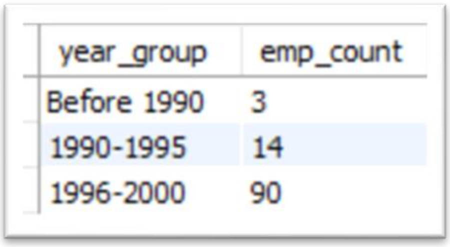
Year_of_hiring	emp_count
1996	10
1997	28
1998	23
1999	18
2000	11

/************************* CASE Statement *************************/

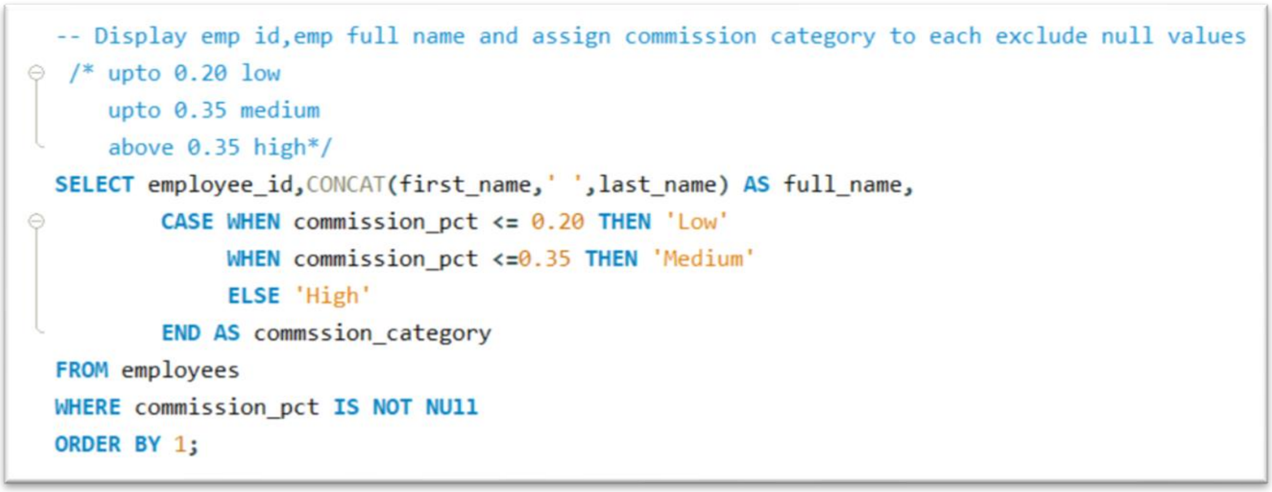


```sql
-- Categorize employees based on hire date
/*
1. before 1990
2. between 1990 to 1995
3. between 1995 to 2000
4. after 90s
*/
SELECT
        CASE WHEN YEAR(hire_date) < 1990 THEN 'Before 1990'
             WHEN YEAR(hire_date) BETWEEN 1990 AND 1995 THEN '1990-1995'
             WHEN YEAR(hire_date) BETWEEN 1996 and 2000 THEN '1996-2000'
             ELSE 'After 90s'
        END AS year_group, COUNT(*) AS emp_count
 FROM employees
 GROUP BY 1;
```

Output :

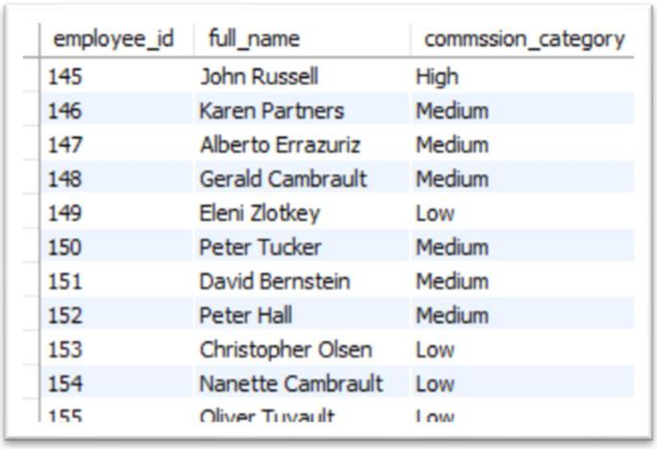


year_group	emp_count
Before 1990	3
1990-1995	14
1996-2000	90



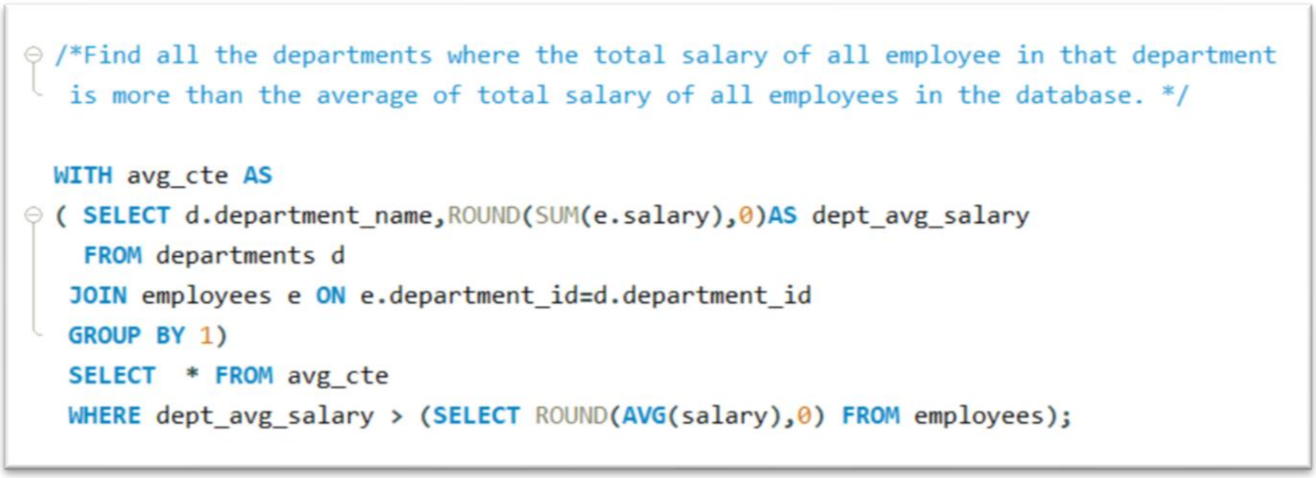
```
-- Display emp id,emp full name and assign commission category to each exclude null values
 /* upto 0.20 low
    upto 0.35 medium
    above 0.35 high*/
SELECT employee_id,CONCAT(first_name,' ',last_name) AS full_name,
        CASE WHEN commission_pct <= 0.20 THEN 'Low'
             WHEN commission_pct <=0.35 THEN 'Medium'
             ELSE 'High'
        END AS commssion_category
FROM employees
WHERE commission_pct IS NOT NUll
ORDER BY 1;
```

Output :



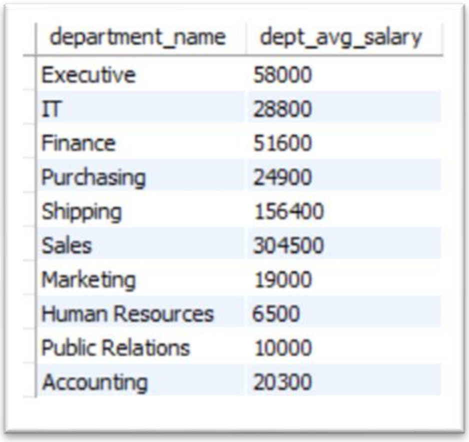
employee_id	full_name	commssion_category
145	John Russell	High
146	Karen Partners	Medium
147	Alberto Errazuriz	Medium
148	Gerald Cambrault	Medium
149	Eleni Zlotkey	Low
150	Peter Tucker	Medium
151	David Bernstein	Medium
152	Peter Hall	Medium
153	Christopher Olsen	Low
154	Nanette Cambrault	Low
155	Oliver Tuvault	Low

/*********************** CTE *************************/**

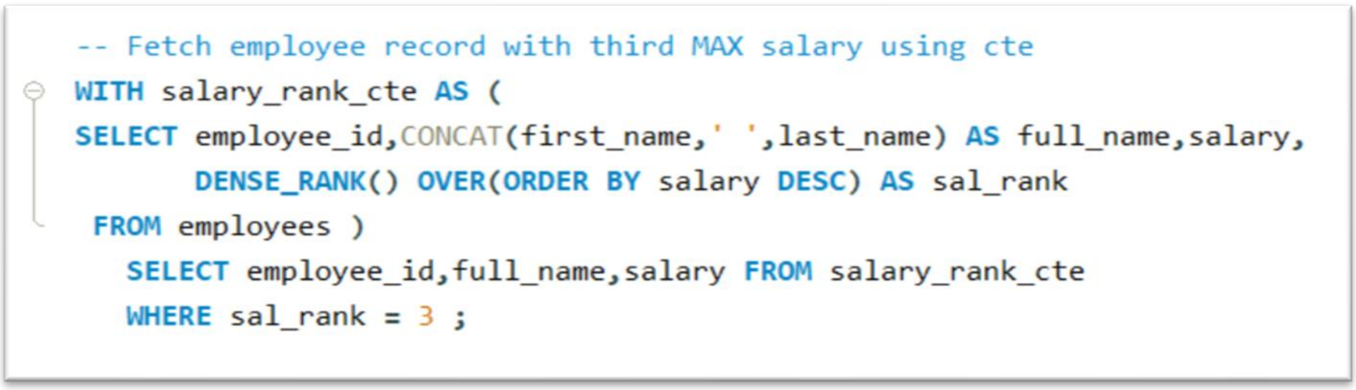


```
/*Find all the departments where the total salary of all employee in that department
 is more than the average of total salary of all employees in the database. */

WITH avg_cte AS
( SELECT d.department_name,ROUND(SUM(e.salary),0)AS dept_avg_salary
  FROM departments d
 JOIN employees e ON e.department_id=d.department_id
 GROUP BY 1)
 SELECT  * FROM avg_cte
 WHERE dept_avg_salary > (SELECT ROUND(AVG(salary),0) FROM employees);
```

Output :

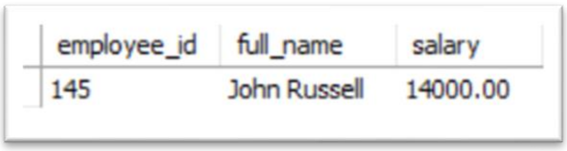


department_name	dept_avg_salary
Executive	58000
IT	28800
Finance	51600
Purchasing	24900
Shipping	156400
Sales	304500
Marketing	19000
Human Resources	6500
Public Relations	10000
Accounting	20300



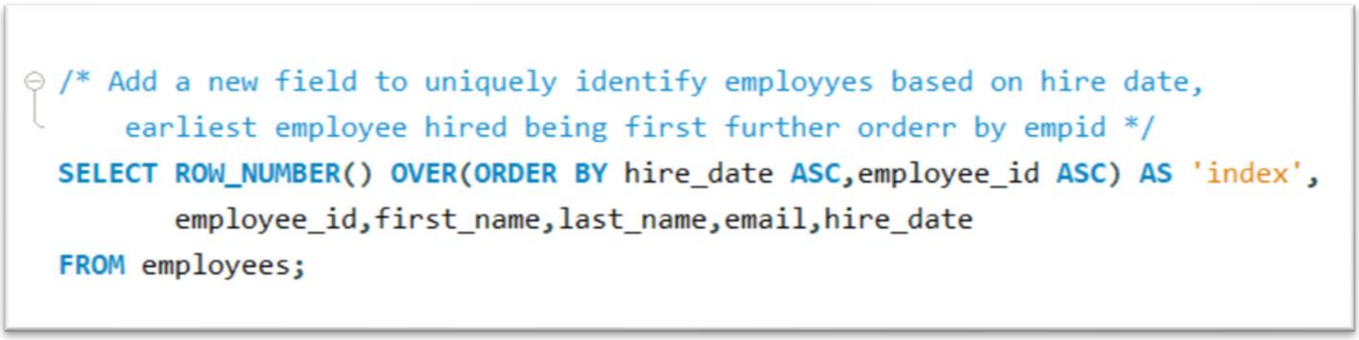
```
-- Fetch employee record with third MAX salary using cte
WITH salary_rank_cte AS (
SELECT employee_id,CONCAT(first_name,' ',last_name) AS full_name,salary,
        DENSE_RANK() OVER(ORDER BY salary DESC) AS sal_rank
 FROM employees )
   SELECT employee_id,full_name,salary FROM salary_rank_cte
   WHERE sal_rank = 3 ;
```

Output :



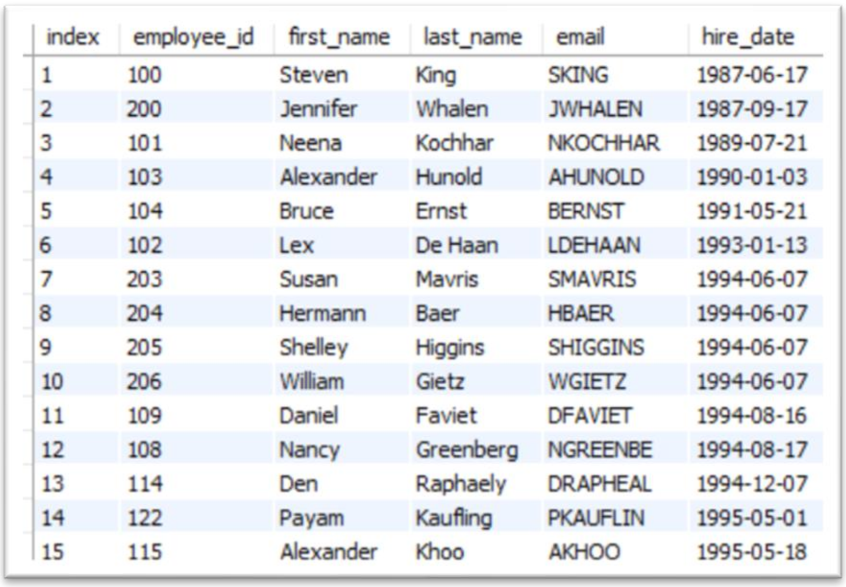
employee_id	full_name	salary
145	John Russell	14000.00

/************************* Window Functions *************************/

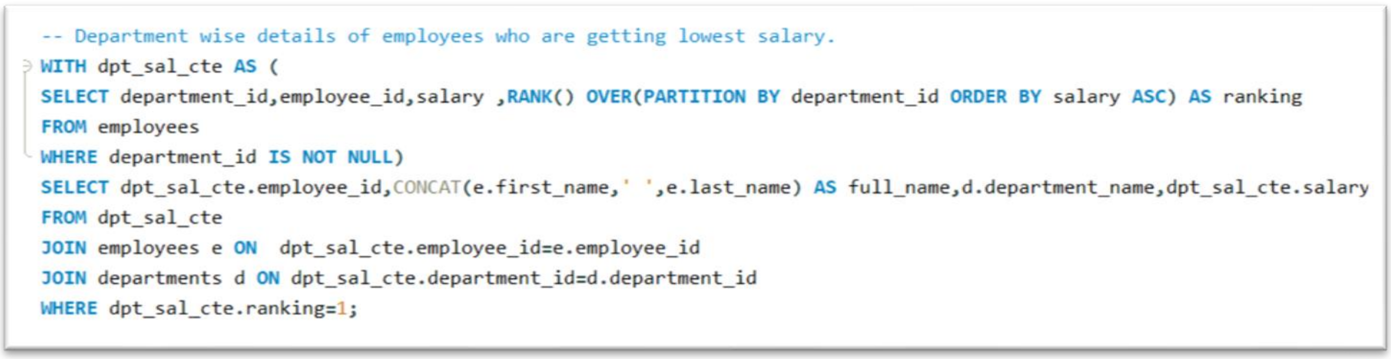


```sql
/* Add a new field to uniquely identify employyes based on hire date,
   earliest employee hired being first further orderr by empid */
SELECT ROW_NUMBER() OVER(ORDER BY hire_date ASC,employee_id ASC) AS 'index',
       employee_id,first_name,last_name,email,hire_date
FROM employees;
```

Output :

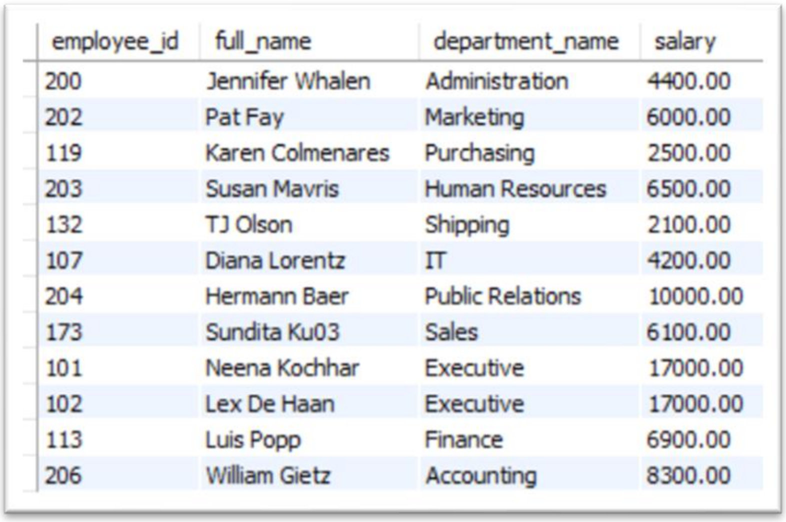


index	employee_id	first_name	last_name	email	hire_date
1	100	Steven	King	SKING	1987-06-17
2	200	Jennifer	Whalen	JWHALEN	1987-09-17
3	101	Neena	Kochhar	NKOCHHAR	1989-07-21
4	103	Alexander	Hunold	AHUNOLD	1990-01-03
5	104	Bruce	Ernst	BERNST	1991-05-21
6	102	Lex	De Haan	LDEHAAN	1993-01-13
7	203	Susan	Mavris	SMAVRIS	1994-06-07
8	204	Hermann	Baer	HBAER	1994-06-07
9	205	Shelley	Higgins	SHIGGINS	1994-06-07
10	206	William	Gietz	WGIETZ	1994-06-07
11	109	Daniel	Faviet	DFAVIET	1994-08-16
12	108	Nancy	Greenberg	NGREENBE	1994-08-17
13	114	Den	Raphaely	DRAPHEAL	1994-12-07
14	122	Payam	Kaufling	PKAUFLIN	1995-05-01
15	115	Alexander	Khoo	AKHOO	1995-05-18



```sql
-- Department wise details of employees who are getting lowest salary.
WITH dpt_sal_cte AS (
SELECT department_id,employee_id,salary ,RANK() OVER(PARTITION BY department_id ORDER BY salary ASC) AS ranking
FROM employees
WHERE department_id IS NOT NULL)
SELECT dpt_sal_cte.employee_id,CONCAT(e.first_name,' ',e.last_name) AS full_name,d.department_name,dpt_sal_cte.salary
FROM dpt_sal_cte
JOIN employees e ON  dpt_sal_cte.employee_id=e.employee_id
JOIN departments d ON dpt_sal_cte.department_id=d.department_id
WHERE dpt_sal_cte.ranking=1;
```

Output :



employee_id	full_name	department_name	salary
200	Jennifer Whalen	Administration	4400.00
202	Pat Fay	Marketing	6000.00
119	Karen Colmenares	Purchasing	2500.00
203	Susan Mavris	Human Resources	6500.00
132	TJ Olson	Shipping	2100.00
107	Diana Lorentz	IT	4200.00
204	Hermann Baer	Public Relations	10000.00
173	Sundita Ku03	Sales	6100.00
101	Neena Kochhar	Executive	17000.00
102	Lex De Haan	Executive	17000.00
113	Luis Popp	Finance	6900.00
206	William Gietz	Accounting	8300.00