CREATE DATABASE Entri\_assignment;

use Entri\_assignment;

CREATE TABLE departments (

Department\_id INT PRIMARY KEY, Department\_name VARCHAR (30), Location\_id INT

);

## Insert into Departments table

INSERT INTO departments VALUES ( 20,'Marketing', 180);

INSERT INTO departments VALUES ( 30,'Purchasing', 1700);

INSERT INTO departments VALUES ( 40, 'Human Resources', 2400);

INSERT INTO departments VALUES ( 50, 'Shipping', 1500);

INSERT INTO departments VALUES ( 60 , 'IT', 1400);

INSERT INTO departments VALUES ( 70, 'Public Relations', 2700);

INSERT INTO departments VALUES ( 80 , 'Sales', 2500 );

INSERT INTO departments VALUES ( 90 , 'Executive', 1700);

INSERT INTO departments VALUES ( 100 , 'Finance', 1700);

INSERT INTO departments VALUES ( 110 , 'Accounting', 1700);

INSERT INTO departments VALUES ( 120 , 'Treasury' , 1700);

INSERT INTO departments VALUES ( 130 , 'Corporate Tax' , 1700 );

INSERT INTO departments VALUES ( 140, 'Control And Credit' , 1700);

INSERT INTO departments VALUES ( 150 , 'Shareholder Services', 1700);

INSERT INTO departments VALUES ( 160 , 'Benefits', 1700);

INSERT INTO departments VALUES ( 170 , 'Payroll' , 1700);

CREATE TABLE employees (

Employee\_id INT PRIMARY KEY,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

email VARCHAR(100),

phone\_number VARCHAR(20),

hire\_date DATE,

job\_id VARCHAR(10),

salary DECIMAL(10, 2),

commission\_pct DECIMAL(5, 2),

manager\_id INT,

department\_id INT,

FOREIGN KEY (department\_id) REFERENCES departments(department\_id) -- Assuming there's a departments table with department\_id as PK

);

INSERT INTO employees VALUES (100, 'Steven', 'King', 'SKING', '515.123.4567', '1987-06-17' , 'AD\_PRES', 24000 , NULL, NULL, 20);

Insert into employees VALUES (101, 'Neena' , 'Kochhar' , 'NKOCHHAR' , '515.123.4568' , '1989-11-21' , 'AD\_VP' , 17000 , NULL , 100 , 20);

INSERT INTO employees VALUES (102 , 'Lex' , 'De Haan' , 'LDEHAAN' , '515.123.4569' , '1993-09-12' , 'AD\_VP' , 17000 , NULL , 100 , 30);

INSERT INTO employees VALUES (104 , 'Bruce' , 'Ernst' , 'BERNST' , '590.423.4568' , '1991-05-21', 'IT\_PROG' , 6000 , NULL , 103 , 60);

INSERT INTO employees VALUES (105 , 'David' , 'Austin' , 'DAUSTIN' , '590.423.4569' , '1997-06-25', 'IT\_PROG' , 4800 , NULL , 103 , 60);

INSERT INTO employees VALUES (106 , 'Valli' , 'Pataballa' , 'VPATABAL' , '590.423.4560' , '1998-02-05', 'IT\_PROG' , 4800 , NULL , 103 , 40);

INSERT INTO employees VALUES (107 , 'Diana' , 'Lorentz' , 'DLORENTZ' , '590.423.5567' , '1999-02-09', 'IT\_PROG' , 4200 , NULL , 103 , 40);

INSERT INTO employees VALUES (108 , 'Nancy' , 'Greenberg' , 'NGREENBE' , '515.124.4569' , '1994-08-17', 'FI\_MGR' , 12000 , NULL , 101 , 100);

INSERT INTO employees VALUES (109 , 'Daniel' , 'Faviet' , 'DFAVIET' , '515.124.4169' , '1994-08-12', 'FI\_ACCOUNT' , 9000 , NULL , 108 , 170);

INSERT INTO employees VALUES (110 , 'John' , 'Chen' , 'JCHEN' , '515.124.4269' , '1997-04-09', 'FI\_ACCOUNT' , 8200 , NULL , 108 , 170);

INSERT INTO employees VALUES (111 , 'Ismael' , 'Sciarra' , 'ISCIARRA' , '515.124.4369' , '1997-02-01', 'FI\_ACCOUNT' , 7700 , NULL , 108 , 160);

INSERT INTO employees VALUES (112 , 'Jose Manuel' , 'Urman' , 'JMURMAN' , '515.124.4469' , '1998-06-03', 'FI\_ACCOUNT' , 7800 , NULL, 8 , 150);

INSERT INTO employees VALUES (114 , 'Den' , 'Raphaely' , 'DRAPHEAL' , '515.127.4561' , '1994-11-08', 'PU\_MAN' , 11000 , NULL , 100 , 30);

INSERT INTO employees VALUES (115 , 'Alexander' , 'Khoo' , 'AKHOO' , '515.127.4562' , '1995-05-12', 'PU\_CLERK' , 3100 , NULL , 114 , 80);

INSERT INTO employees VALUES (116 , 'Shelli' , 'Baida' , 'SBAIDA' , '515.127.4563' ,'1997-12-13', 'PU\_CLERK' , 2900 , NULL , 114 , 70);

INSERT INTO employees VALUES (117 , 'Sigal' , 'Tobias' , 'STOBIAS' , '515.127.4564' , '1997-09-10', 'PU\_CLERK' , 2800 , NULL , 114 , 30);

INSERT INTO employees VALUES (118 , 'Guy' , 'Himuro' , 'GHIMURO' , '515.127.4565' , '1998-01-02', 'PU\_CLERK' , 2600 , NULL , 114 , 60);

INSERT INTO employees VALUES (119 , 'Karen' , 'Colmenares' , 'KCOLMENA' , '515.127.4566' , '1999-04-08', 'PU\_CLERK' , 2500 , NULL , 114 , 130);

INSERT INTO employees VALUES (120 , 'Matthew' , 'Weiss' , 'MWEISS' , '650.123.1234' ,'1996-07-18', 'ST\_MAN' , 8000 , NULL , 100 , 50);

INSERT INTO employees VALUES (122 , 'Payam' , 'Kaufling' , 'PKAUFLIN' , '650.123.3234' ,'1995-05-01', 'ST\_MAN' , 7900 , NULL , 100 , 40);

INSERT INTO employees VALUES (123 , 'Shanta' , 'Vollman' , 'SVOLLMAN' , '650.123.4234' , '1997-10-12', 'ST\_MAN' , 6500 , NULL , 100 , 50);

INSERT INTO employees VALUES (124, 'Kevin' , 'Mourgos' , 'KMOURGOS' , '650.123.5234' , '1999-11-12', 'ST\_MAN' , 5800 , NULL , 100 , 80);

INSERT INTO employees VALUES (125, 'Julia' , 'Nayer' , 'JNAYER' , '650.124.1214' , '1997-07-02', 'ST\_CLERK' , 3200 , NULL , 120 , 50);

INSERT INTO employees VALUES (126, 'Irene' , 'Mikkilineni' , 'IMIKKILI' , '650.124.1224' , '1998-11-12', 'ST\_CLERK' , 2700 , NULL , 120 , 50);

INSERT INTO employees VALUES (127, 'James' , 'Landry' , 'JLANDRY' , '650.124.1334' , '1999-01-02' , 'ST\_CLERK' , 2400 , NULL , 120 , 90);

INSERT INTO employees VALUES (128, 'Steven' , 'Markle' , 'SMARKLE' , '650.124.1434' , '2000-03-04' , 'ST\_CLERK' , 2200 , NULL , 120 , 50);

INSERT INTO employees VALUES (130, 'Mozhe' , 'Atkinson' , 'MATKINSO' , '650.124.6234' , '1997-10-12' , 'ST\_CLERK' , 2800 , NULL , 121 , 110);

**# 1. Select employees first name, last name, job\_id and salary whose first name starts with alphabet S**

SELECT

first\_name

FROM

employees

WHERE first\_name LIKE "S%";

OUTPUT:

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

|  |  |
| --- | --- |
| first\_name | |
| Steven |  |
| Shelli |  |
| Sigal |  |
| Shanta |  |
| Steven |  |
|  |  |

**# 2. Write a query to select employee with the highest salary (using an inner query)**

select first\_name ,last\_name,salary FROM employees

where salary = (SELECT max(salary) FROM employees );

OUTPUT:

|  |  |  |  |
| --- | --- | --- | --- |
| first\_name | last\_name | salary |  |
| Steven | King | 24000 |  |
|  |  |  |  |

**# 3. Select employee with the second highest salary**

SELECT

first\_name, last\_name, salary

FROM

employees

ORDER BY salary DESC

LIMIT 5;

OUTPUT:

|  |  |  |  |
| --- | --- | --- | --- |
| first\_name | last\_name | salary |  |
| Steven | King | 24000 |  |
| Neena | Kochhar | 17000 |  |
| Lex | De Haan | 17000 |  |
| Nancy | Greenberg | 12000 |  |
| Den | Raphaely | 11000 |  |

**# 4. Write a query to select employees and their corresponding managers and their salaries**

**Select**

e1.first\_name as employee\_first\_name,

e1.last\_name as employee\_last\_name,

e2.first\_name as manager\_first\_name,

e2.last\_name AS manager\_last\_name,

e1.salary AS employee\_salary,

e2.salary AS manager\_salary

FROM

employees e1

LEFT JOIN

employees e2

ON

e1.manager\_id = e2.Employee\_id;

OUTPUT:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| employee\_first\_name | employee\_last\_name | manager\_first\_name | manager\_last\_name | employee\_salary | manager\_salary | |
| Steven | King | NULL | NULL | 24000 | NULL |  |
| Neena | Kochhar | Steven | King | 17000 | 24000 |  |
| Lex | De Haan | Steven | King | 17000 | 24000 |  |
| Bruce | Ernst | NULL | NULL | 6000 | NULL |  |
| David | Austin | NULL | NULL | 4800 | NULL |  |
| Valli | Pataballa | NULL | NULL | 4800 | NULL |  |
| Diana | Lorentz | NULL | NULL | 4200 | NULL |  |
| Nancy | Greenberg | Neena | Kochhar | 12000 | 17000 |  |
| Daniel | Faviet | Nancy | Greenberg | 9000 | 12000 |  |
| John | Chen | Nancy | Greenberg | 8200 | 12000 |  |
| Ismael | Sciarra | Nancy | Greenberg | 7700 | 12000 |  |
| Jose Manuel | Urman | NULL | NULL | 7800 | NULL |  |
| Den | Raphaely | Steven | King | 11000 | 24000 |  |
| Alexander | Khoo | Den | Raphaely | 3100 | 11000 |  |
| Shelli | Baida | Den | Raphaely | 2900 | 11000 |  |
| Sigal | Tobias | Den | Raphaely | 2800 | 11000 |  |
| Guy | Himuro | Den | Raphaely | 2600 | 11000 |  |
| Karen | Colmenares | Den | Raphaely | 2500 | 11000 |  |
| Matthew | Weiss | Steven | King | 8000 | 24000 |  |
| Payam | Kaufling | Steven | King | 7900 | 24000 |  |
| Shanta | Vollman | Steven | King | 6500 | 24000 |  |
| Kevin | Mourgos | Steven | King | 5800 | 24000 |  |
| Julia | Nayer | Matthew | Weiss | 3200 | 8000 |  |
| Irene | Mikkilineni | Matthew | Weiss | 2700 | 8000 |  |
| James | Landry | Matthew | Weiss | 2400 | 8000 |  |
| Steven | Markle | Matthew | Weiss | 2200 | 8000 |  |
| Mozhe | Atkinson | NULL | NULL | 2800 | NULL |  |

**# 5. Write a query to select employees and their corresponding managers and their salaries (SELF Join)**

select

e1.first\_name as employee\_first\_name,

e1.last\_name as employee\_last\_name,

e2.first\_name as manager\_first\_name,

e2.last\_name AS manager\_last\_name,

e1.salary AS employee\_salary,

e2.salary AS manager\_salary

FROM

employees e1

INNER JOIN

employees e2

ON

e1.manager\_id = e2.Employee\_id;

OUTPUT:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| employee\_first\_name | employee\_last\_name | manager\_first\_name | manager\_last\_name | employee\_salary | manager\_salary | |
| Neena | Kochhar | Steven | King | 17000 | 24000 |  |
| Lex | De Haan | Steven | King | 17000 | 24000 |  |
| Nancy | Greenberg | Neena | Kochhar | 12000 | 17000 |  |
| Daniel | Faviet | Nancy | Greenberg | 9000 | 12000 |  |
| John | Chen | Nancy | Greenberg | 8200 | 12000 |  |
| Ismael | Sciarra | Nancy | Greenberg | 7700 | 12000 |  |
| Den | Raphaely | Steven | King | 11000 | 24000 |  |
| Alexander | Khoo | Den | Raphaely | 3100 | 11000 |  |
| Shelli | Baida | Den | Raphaely | 2900 | 11000 |  |
| Sigal | Tobias | Den | Raphaely | 2800 | 11000 |  |
| Guy | Himuro | Den | Raphaely | 2600 | 11000 |  |
| Karen | Colmenares | Den | Raphaely | 2500 | 11000 |  |
| Matthew | Weiss | Steven | King | 8000 | 24000 |  |
| Payam | Kaufling | Steven | King | 7900 | 24000 |  |
| Shanta | Vollman | Steven | King | 6500 | 24000 |  |
| Kevin | Mourgos | Steven | King | 5800 | 24000 |  |
| Julia | Nayer | Matthew | Weiss | 3200 | 8000 |  |
| Irene | Mikkilineni | Matthew | Weiss | 2700 | 8000 |  |
| James | Landry | Matthew | Weiss | 2400 | 8000 |  |
| Steven | Markle | Matthew | Weiss | 2200 | 8000 |  |
|  |  |  |  |  |  |  |

**# 6. Create a view for the above query**

CREATE VIEW hierarchy as

select

e1.first\_name as employee\_first\_name,

e1.last\_name as employee\_last\_name,

e2.first\_name as manager\_first\_name,

e2.last\_name AS manager\_last\_name,

e1.salary AS employee\_salary,

e2.salary AS manager\_salary

FROM

employees e1

INNER JOIN

employees e2

ON

e1.manager\_id = e2.Employee\_id;

show full tables;

OUTPUT:

|  |  |  |
| --- | --- | --- |
| Tables\_in\_entri\_assignment | Table\_type | |
| departments | BASE TABLE | |
| employees | BASE TABLE | |
| hierarchy | VIEW |  |
|  |  |  |

**#7. Write a query to show the count of employees under each manager in descending order (from view)**

SELECT

manager\_first\_name,

manager\_last\_name,

COUNT(employee\_first\_name) AS no\_of\_employee

FROM

hierarchy

GROUP BY manager\_first\_name , manager\_last\_name

ORDER BY no\_of\_employee;

OUTPUT:

|  |  |  |  |
| --- | --- | --- | --- |
| manager\_first\_name | manager\_last\_name | no\_of\_employee | |
| Neena | Kochhar | 1 |  |
| Nancy | Greenberg | 3 |  |
| Matthew | Weiss | 4 |  |
| Den | Raphaely | 5 |  |
| Steven | King | 7 |  |
|  |  |  |  |

**#8. Find the count of employees in each department**

select

d.department\_id,

d.department\_name,

count(e.first\_name) as counts

FROM

employees e

LEFT JOIN

departments d

ON

e.department\_id = d.department\_id

group by

d.department\_id,

d.department\_name ;

OUTPUT:

|  |  |  |  |
| --- | --- | --- | --- |
| department\_id | department\_name | counts |  |
| 20 | Marketing | 2 |  |
| 30 | Purchasing | 3 |  |
| 60 | IT | 3 |  |
| 40 | Human Resources | 3 |  |
| 100 | Finance | 1 |  |
| 170 | Payroll | 2 |  |
| 160 | Benefits | 1 |  |
| 150 | Shareholder Services | 1 |  |
| 80 | Sales | 2 |  |
| 70 | Public Relations | 1 |  |
| 130 | Corporate Tax | 1 |  |
| 50 | Shipping | 5 |  |
| 90 | Executive | 1 |  |
| 110 | Accounting | 1 |  |
|  |  |  |  |

**# 9. Get the count of employees hired year wise**

SELECT

COUNT(\*) AS counts, YEAR(hire\_date) AS hire\_year

FROM

employees

GROUP BY YEAR(hire\_date)

ORDER BY hire\_year;

OUTPUT:

|  |  |  |
| --- | --- | --- |
| counts | hire\_year |  |
| 1 | 1987 |  |
| 1 | 1989 |  |
| 1 | 1991 |  |
| 1 | 1993 |  |
| 3 | 1994 |  |
| 2 | 1995 |  |
| 1 | 1996 |  |
| 8 | 1997 |  |
| 4 | 1998 |  |
| 4 | 1999 |  |
| 1 | 2000 |  |

**# 10 . create a stored procedure to get the “ Get the count of employees hired in the input year”(IN year , OUT count)**

DELIMITER $$

CREATE PROCEDURE COUNT\_EMPLOYEES(IN input\_year INT, OUT employee\_count INT)

BEGIN

SET employee\_count=(select count(\*) from employees

where year(hire\_date)=input\_year);

END $$

DELIMITER ;

call COUNT\_EMPLOYEES(1999,@count);

select @count;

OUTPUT:

|  |  |
| --- | --- |
| @count |  |
| 4 |  |

**# 11.Select the employees whose first\_name contains “an”**

SELECT

first\_name

FROM

employees

WHERE first\_name LIKE "%an%";

OUTPUT:

|  |  |
| --- | --- |
| first\_name | |
| Diana |  |
| Nancy |  |
| Daniel |  |
| Jose Manuel | |
| Alexander |  |
| Shanta |  |

**# 12. Select employee first name and the corresponding phone number in the format (\_ \_ \_)-(\_ \_ \_)-(\_ \_ \_ \_)**

SELECT

first\_name,

CONCAT('(',

SUBSTRING(phone\_number, 1, 3),

')-(',

SUBSTRING(phone\_number, 5, 3),

')-(',

SUBSTRING(phone\_number, 9, 4),

')') AS 'phone number'

FROM

employees;

OUPUT:

|  |  |  |  |
| --- | --- | --- | --- |
| first\_name | phone number | |  |
| Steven | (515)-(123)-(4567) | |  |
| Neena | (515)-(123)-(4568) | |  |
| Lex | (515)-(123)-(4569) | |  |
| Bruce | (590)-(423)-(4568) | |  |
| David | (590)-(423)-(4569) | |  |
| Valli | (590)-(423)-(4560) | |  |
| Diana | (590)-(423)-(5567) | |  |
| Nancy | (515)-(124)-(4569) | |  |
| Daniel | (515)-(124)-(4169) | |  |
| John | (515)-(124)-(4269) | |  |
| Ismael | (515)-(124)-(4369) | |  |
| Jose Manuel | (515)-(124)-(4469) | |  |
| Den | (515)-(127)-(4561) | |  |
| Alexander | (515)-(127)-(4562) | |  |
| Shelli | (515)-(127)-(4563) | |  |
| Sigal | (515)-(127)-(4564) | |  |
| Guy | (515)-(127)-(4565) | |  |
| Karen | (515)-(127)-(4566) | |  |
| Matthew | (650)-(123)-(1234) | |  |
| Payam | (650)-(123)-(3234) | |  |
| Shanta | (650)-(123)-(4234) | |  |
| Kevin | (650)-(123)-(5234) | |  |
| Julia | (650)-(124)-(1214) | |  |
| Irene | (650)-(124)-(1224) | |  |
| James | (650)-(124)-(1334) | |  |
| Steven | (650)-(124)-(1434) | |  |
| Mozhe | (650)-(124)-(6234) | |  |
|  |  |  |  |

**# 13. Find the employees who joined in August, 1994.**

SELECT

first\_name, last\_name, hire\_date

FROM

employees

WHERE

hire\_date LIKE '1994-08-%'

OUTPUT:

|  |  |  |  |
| --- | --- | --- | --- |
| first\_name | last\_name | hire\_date |  |
| Nancy | Greenberg | 8/17/1994 |  |
| Daniel | Faviet | 8/12/1994 |  |

**# 14. Find the maximum salary from each department.**

SELECT

e.department\_id AS 'Department ID',

d.Department\_name AS 'Department',

MAX(salary) AS 'Maximum salary'

FROM

employees e

JOIN

departments d ON e.department\_id = d.Department\_id

GROUP BY e.department\_id;

OUTPUT:

|  |  |  |  |
| --- | --- | --- | --- |
| Department ID | Department | Maximum salary | |
| 20 | Marketing | 24000 |  |
| 30 | Purchasing | 17000 |  |
| 60 | IT | 6000 |  |
| 40 | Human Resources | 7900 |  |
| 100 | Finance | 12000 |  |
| 170 | Payroll | 9000 |  |
| 160 | Benefits | 7700 |  |
| 150 | Shareholder Services | 7800 |  |
| 80 | Sales | 5800 |  |
| 70 | Public Relations | 2900 |  |
| 130 | Corporate Tax | 2500 |  |
| 50 | Shipping | 8000 |  |
| 90 | Executive | 2400 |  |
| 110 | Accounting | 2800 |  |
|  |  |  |  |

# 15.Write a SQL query to display the 5 least earning employees

SELECT

first\_name, salary

FROM

employees

ORDER BY salary

LIMIT 5;

OUTPUT:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | first\_name | salary |  |  | | Steven | 2200 |  |  | | James | 2400 |  |  | | Karen | 2500 |  |  | | Guy | 2600 |  |  | | Irene | 2700 |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  |   **# 16. Find the employees hired in the 80s**  SELECT  first\_name, last\_name, YEAR(hire\_date)  FROM  employees  WHERE  YEAR(hire\_date) LIKE '198%'    OUTPUT:   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | first\_name | last\_name | YEAR(hire\_date) | |  |  | | Steven | King | 1987 |  |  |  | | Neena | Kochhar | 1989 |  |  |  | |  |  |  |  |  |  | |  |  |  |  |  |  | |  |  |  |  |  |  | |  |  |  |  |  |  | |  |  |  |  |  |  | |  |  |  |  |  |  | |  |  |  |  |  |  |   **# 17. Find the employees who joined the company after 15th of the month**    SELECT  \*  FROM  employees  WHERE  DAY(hire\_date) > 15;  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 | ######## | AD\_PRES | 24000 | NULL | NULL | 20 |  |  |  | | 101 | Neena | Kochhar | NKOCHHAR | 515.123.4568 | ######## | AD\_VP | 17000 | NULL | 100 | 20 |  |  |  | | 104 | Bruce | Ernst | BERNST | 590.423.4568 | ######## | IT\_PROG | 6000 | NULL | 103 | 60 |  |  |  | | 105 | David | Austin | DAUSTIN | 590.423.4569 | ######## | IT\_PROG | 4800 | NULL | 103 | 60 |  |  |  | | 108 | Nancy | Greenberg | NGREENBE | 515.124.4569 | ######## | FI\_MGR | 12000 | NULL | 101 | 100 |  |  |  | | 120 | Matthew | Weiss | MWEISS | 650.123.1234 | ######## | ST\_MAN | 8000 | NULL | 100 | 50 |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |
|  |
|  |
|  |
|  |
|  |
|  |
|  |