**19PW22\_PROBLEM SHEET-3**

Create an employee table with attributes employee number, first name, last name, and

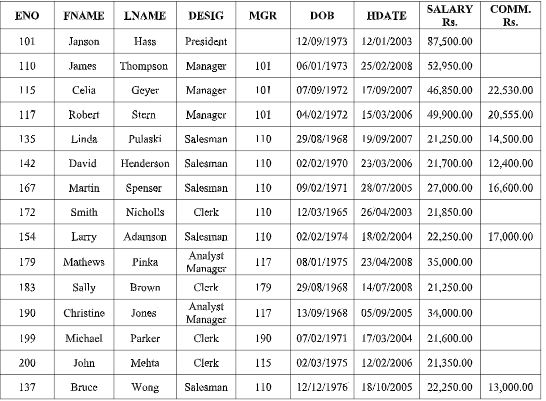
designation, manager ID as MGR, date of birth, hire date, salary and commission. Enforce the

following constraints on employee table:

Employee number is primary key, first and last names should not be null, Manager number will

be the foreign key which refers primary of this table; date of birth must be less than or equal to

1980; Salary is less than Rs.1, 00,000/- and greater than Rs.20, 000/-.



CREATE TABLE IF NOT EXISTS Employee

(

    ENO        BIGINT,

    FNAME      VARCHAR(50) NOT NULL,

    LNAME      VARCHAR(50) NOT NULL,

    DESIG      VARCHAR(50),

    MGR        BIGINT,

    DOB        DATE,

    HDATE      DATE,

    Salary     NUMERIC(8, 2),

    Commission NUMERIC(8, 2),

    CONSTRAINT pkey PRIMARY KEY (ENO),

    CONSTRAINT foreign\_key FOREIGN KEY (MGR) REFERENCES Employee (ENO),

    CONSTRAINT DOB\_Check CHECK ( YEAR(DOB) <= 1980 ),

    CONSTRAINT Salary\_Check CHECK ( Salary BETWEEN 20000 AND 100000 )

);

INSERTION OF VALUES

INSERT INTO Employee

VALUES (101, 'Janson', 'Hass', 'President', NULL, STR\_TO\_DATE('12/09/1973', '%d/%m/%Y'),

        STR\_TO\_DATE('12/01/2003', '%d/%m/%Y'), 87500.00, NULL),

       (110, 'James', 'Thompson', 'Manager', 101, STR\_TO\_DATE('06/01/1973', '%d/%m/%Y'),

        STR\_TO\_DATE('25/02/2008', '%d/%m/%Y'), 52950.00, NULL),

       (115, 'Celia', 'Geyer', 'Manager', 101, STR\_TO\_DATE('07/09/1972', '%d/%m/%Y'),

        STR\_TO\_DATE('17/09/2007', '%d/%m/%Y'), 46850.00, 22530.00),

       (117, 'Robert', 'Stern', 'Manager', 101, STR\_TO\_DATE('04/02/1972', '%d/%m/%Y'),

        STR\_TO\_DATE('15/03/2006', '%d/%m/%Y'), 49900.00, 20555.00),

       (135, 'Linda', 'Pulaski', 'Salesman', 110, STR\_TO\_DATE('29/08/1968', '%d/%m/%Y'),

        STR\_TO\_DATE('19/09/2007', '%d/%m/%Y'), 21250.00, 14500.00),

       (142, 'David', 'Henderson', 'Salesman', 110, STR\_TO\_DATE('02/02/1970', '%d/%m/%Y'),

        STR\_TO\_DATE('23/03/2006', '%d/%m/%Y'), 21700.00, 12400.00),

       (167, 'Martin', 'Spenser', 'Salesman', 110, STR\_TO\_DATE('09/02/1971', '%d/%m/%Y'),

        STR\_TO\_DATE('28/07/2005', '%d/%m/%Y'), 27000.00, 16600.00),

       (172, 'Smith', 'Nicholls', 'Clerk', 110, STR\_TO\_DATE('12/03/1965', '%d/%m/%Y'),

        STR\_TO\_DATE('26/04/2003', '%d/%m/%Y'), 21850.00, NULL),

       (154, 'Larry', 'Adamson', 'Salesman', 110, STR\_TO\_DATE('02/02/1974', '%d/%m/%Y'),

        STR\_TO\_DATE('18/02/2004', '%d/%m/%Y'), 22250.00, 17000.00),

       (179, 'Mathews', 'Pinka', 'Analyst Manager', 117, STR\_TO\_DATE('08/01/1975', '%d/%m/%Y'),

        STR\_TO\_DATE('23/04/2008', '%d/%m/%Y'), 35000.00, NULL),

       (183, 'Sally', 'Brown', 'Clerk', 179, STR\_TO\_DATE('29/08/1968', '%d/%m/%Y'),

        STR\_TO\_DATE('14/07/2008', '%d/%m/%Y'), 21250.00, NULL),

       (190, 'Christine', 'Jones', 'Analyst Manager', 117, STR\_TO\_DATE('13/09/1968', '%d/%m/%Y'),

        STR\_TO\_DATE('05/09/2005', '%d/%m/%Y'), 34000.00, NULL),

       (199, 'Michael', 'Parker', 'Clerk', 190, STR\_TO\_DATE('07/02/1971', '%d/%m/%Y'),

        STR\_TO\_DATE('17/03/2004', '%d/%m/%Y'), 21600.00, NULL),

       (200, 'John', 'Mehta', 'Clerk', 115, STR\_TO\_DATE('02/03/1975', '%d/%m/%Y'),

        STR\_TO\_DATE('12/02/2006', '%d/%m/%Y'), 21350.00, NULL),

       (137, 'Bruce', 'Wong', 'Salesman', 110, STR\_TO\_DATE('12/12/1976', '%d/%m/%Y'),

        STR\_TO\_DATE('18/10/2005', '%d/%m/%Y'), 22250.00, 13000.00);

1. List the details of all the employees who are younger than the employee with employee

number 167.

SELECT \*

FROM Employee

WHERE DOB <= ANY (SELECT DOB FROM Employee WHERE ENO = 167);

2. Display first and last names all the employees who are less than 30 years old.

SELECT FNAME, LNAME

FROM Employee

WHERE TIMESTAMPDIFF(YEAR, DOB, CURRENT\_TIMESTAMP) < 30;

3. Display the details of employees whose last name contains the consecutive letters ‘er’.

SELECT \*

FROM Employee

WHERE LNAME LIKE '%er%';

4. Display the names of employees whose first name starts with ‘M’ or ‘J’.

SELECT FNAME, LNAME

FROM Employee

WHERE FNAME REGEXP '^[MJ]\w\*';

5. Display first name, last name and salary of all the employees with the designation

“Analyst Manager” or “Manager”.

SELECT FNAME, LNAME, Salary

FROM Employee

WHERE DESIG LIKE '%Manager%';

6. Display the details of employee(s) who is (are) eldest in the employee table.

SELECT \*

FROM Employee

WHERE DOB IN (SELECT MIN(DOB) FROM Employee);

7. List out the employees who all are not manager and analyst.

SELECT \*

FROM Employee

WHERE DESIG NOT IN ('Manager', 'Analyst Manager');

8. Display the details of employees with a salary over Rs.35, 000.00 and born in between

1966 to 1976.

SELECT \*

FROM Employee

WHERE Salary > 35000

  AND (YEAR(DOB) BETWEEN 1966 AND 1976);

9. Display the employee number, first name, last name, designation and salary whose salary

is greater than or equal to Rs.35,000.00 and the job title is “Analyst” or “Manager”

SELECT FNAME, LNAME, DESIG, Salary

FROM Employee

WHERE Salary >= 35000

  AND DESIG IN ('Manager', 'Analyst Manager');

10. Display the employee number, first name, last name, designation and salary of employees

whose commission is greater than or equal to Rs.15,000.00 and designation is

‘Salesman’ or ‘Manager’.

SELECT ENO, FNAME, LNAME, DESIG, Salary

FROM Employee

WHERE Commission >= 15000

  AND DESIG IN ('Manager', 'Salesman');

11. Display the total number of employees in each job title.

SELECT DESIG, COUNT(DESIG)

FROM Employee

GROUP BY DESIG;

12. Display the employee number, last name, Salary of employees whose last name is in the

list, ‘Thompson’, ‘Spenser’, ‘Geyer’, and ’Brown’, ‘Jones’.

SELECT ENO, LNAME, Salary

FROM Employee

WHERE LNAME IN ('Thompson', 'Spenser', 'Geyer', 'Brown', 'Jones');

13. Find out and display the total number of employees joined in every year.

SELECT YEAR(HDATE), COUNT(\*)

FROM Employee

GROUP BY YEAR(HDATE)

ORDER BY YEAR(HDATE);

14. Display the employee ID, last name, salary and age of employees whose age is in

between 30 and 40.

SELECT ENO, LNAME, Salary, TIMESTAMPDIFF(YEAR, DOB, CURRENT\_TIMESTAMP)

FROM Employee

WHERE TIMESTAMPDIFF(YEAR, DOB, CURRENT\_TIMESTAMP) BETWEEN 30 AND 40;

15. Find out and display the first name employees who all are managed by ‘James’ or

‘Robert’ or ‘Celia’.

SELECT FNAME

FROM Employee

WHERE MGR IN (SELECT ENO FROM Employee WHERE FNAME IN ('James', 'Robert', 'Celia'));

16. Display all the employees with their fist name, last name and designation as follows

“First name“, “last name”, “designation”.

SELECT CONCAT('"', FNAME, '","', LNAME, '","', DESIG, '"')

FROM Employee;

17. Find out the employee who is youngest earning commission.

SELECT \*

FROM Employee

WHERE Commission IS NOT NULL

ORDER BY DOB DESC

LIMIT 1;

18. Display all the employees with their date of birth and date of joining as follows.

Example: Dear ‘James ’, you were born on 06, Monday – September – 1972’.

SELECT CONCAT('Dear \'',FNAME,'\', you were born on ',DATE\_FORMAT(DOB, '%d, %D - %M - %Y'))

FROM Employee;

19. Display the employees who are all getting maximum salary in each job title.

SELECT \*

FROM Employee

WHERE (DESIG, Salary) IN (SELECT DESIG, MAX(Salary) FROM Employee GROUP BY DESIG);

20. Display the employee number, first name and salary of employees who were born in

either January or February or August.

SELECT \*

FROM Employee

WHERE MONTH(DOB) IN (1, 2, 8);

21. Display the name, salary, commission and total remuneration (salary + commission) of

all the employees.

SELECT CONCAT(FNAME, ' ', LNAME)        AS Name,

       Salary,

       Commission,

       Salary + COALESCE(Commission, 0) AS 'Total Remuneration'

FROM Employee;

22. Display the first name, salary, commission and total remuneration for all the employees

and fill the column with 0 for those who do not have any commission in the commission

column.

SELECT CONCAT(FNAME, ' ', LNAME)        AS Name,

       Salary,

       COALESCE(Commission, NOT ISNULL(Commission)),

       Salary + COALESCE(Commission, 0) AS 'Total Remuneration'

FROM Employee;

23. Display first name of all the employees having maximum experience in each designation.

SELECT FNAME

FROM Employee

WHERE (DESIG, HDATE) IN (SELECT DESIG, MIN(HDATE) FROM Employee GROUP BY DESIG);

24. Display the details of all the managers getting salary greater than Rs.40, 000.00 and born

before 1973.

SELECT \*

FROM Employee

WHERE DESIG = 'Manager'

  AND YEAR(DOB) < 1973

  AND Salary > 40000;

25. Display all the details of employees who are not getting any commission but getting

salary between Rs25,000.00 and Rs35,000.00.

SELECT \*

FROM Employee

WHERE Commission IS NULL

  AND Salary BETWEEN 25000 AND 35000;

26. Find and display the details of all the employees whose born and joined month are same.

SELECT \*

FROM Employee

WHERE MONTH(DOB) = MONTH(HDATE);

27. Display the employee number and first name of all the employees with hire date in the

format like February 22, 1991.

SELECT ENO, FNAME, DATE\_FORMAT(HDATE, '%M %d,%Y')

FROM Employee;

28. Display the employee number, name, designation, hire date, and experience of all the

managers.

SELECT ENO,

       CONCAT(FNAME, ' ', LNAME)                     AS Name,

       DESIG,

       HDATE,

       TIMESTAMPDIFF(YEAR, HDATE, CURRENT\_TIMESTAMP) AS Experience

FROM Employee

WHERE ENO IN (SELECT DISTINCT MGR FROM Employee WHERE MGR IS NOT NULL);

29. List the employee ID, name, salary of all the ‘Managers’ and ‘Analyst’ with more than 5

years of experience and not receiving any commission. Display the list in ascending order

the employee number.

SELECT ENO,

       CONCAT(FNAME, ' ', LNAME) AS Name,

       Salary

FROM Employee

WHERE TIMESTAMPDIFF(YEAR, HDATE, CURRENT\_TIMESTAMP) > 5

  AND DESIG IN ('Manager', 'Analyst Manager')

  AND Commission IS NULL

ORDER BY ENO;

30. List the employees who are senior to ‘Martin’.

SELECT \*

FROM Employee

WHERE HDATE > (SELECT HDATE FROM Employee WHERE FNAME = 'Martin');

31. Display the first names of all the employees whose salary is same as the salary of ‘Larry’.

SELECT FNAME

FROM Employee

WHERE Salary IN (SELECT Salary FROM Employee WHERE FNAME = 'Larry');

32. Display the names of employees whose designation is same as the designation of ‘James’

or salary is more than the salary of ‘Celia’.

SELECT FNAME

FROM Employee

WHERE DESIG IN (SELECT DESIG FROM Employee WHERE FNAME = 'James') # Should james be included are not?

  AND Salary > (SELECT Salary FROM Employee WHERE FNAME = 'Celia');

33. Find and display the details of the highest paid employee(s).

SELECT \*

FROM Employee

WHERE (Salary + COALESCE(Commission, 0)) =

      (SELECT MAX(Salary + COALESCE(Commission, 0)) FROM Employee);

34. List the details of employees who have been hired most recently as ‘salesman’.

SELECT \*

FROM Employee

WHERE DESIG = 'Salesman'

  AND HDATE = (SELECT MAX(HDATE) FROM Employee WHERE DESIG = 'Salesman');

35. List the employees who are senior to the most recently hired employees working under

employee id 110.

SELECT \*

FROM Employee

WHERE MGR = 110

  AND HDATE < (SELECT MAX(HDATE) FROM Employee WHERE MGR = 110);

36. Find and display the details of the senior employees as on year 1991.

SELECT \*

FROM Employee

WHERE HDATE < (SELECT MAX(HDATE) FROM Employee WHERE YEAR(HDATE) >= 1991);

37. List the employees’ names whose salary is equal or more to the average of maximum and

minimum salary.

SELECT CONCAT(FNAME, ' ', LNAME) AS Name

FROM Employee

WHERE Salary >= (SELECT (MIN(Salary) + MAX(Salary)) / 2 FROM Employee);

38. Write a query to find and display the names of the least 5 earners of the company.

SELECT CONCAT(FNAME, ' ', LNAME) AS Name

FROM Employee

ORDER BY (Salary + COALESCE(Commission, 0))

LIMIT 5;

39. List the name, salary, commission and netpay for those employees whose netpay (salary

+ commission) is more than any other employee.

SELECT CONCAT(FNAME, ' ', LNAME)        AS Name,

       Salary,

       COALESCE(Commission, NOT ISNULL(Commission)) AS 'Commission',

       Salary + COALESCE(Commission, 0) AS 'Netpay'

FROM Employee

WHERE (Salary + COALESCE(Commission, 0)) = (SELECT MAX(Salary + COALESCE(Commission, 0)) FROM Employee);

40. Display the names of employees who joined the company on the same date.

SELECT COUNT(\*)

FROM Employee

WHERE MGR IS NULL;

41. Display the employee numbers of employees who are not working under any managers.

SELECT COUNT(\*)

FROM Employee

WHERE MGR IS NULL;

42. List the names of the managers who are having maximum number of employees working

under him.

SELECT CONCAT(FNAME, ' ', LNAME) AS Name

FROM Employee

WHERE ENO IN (SELECT MGR

              FROM Employee

              GROUP BY MGR

              HAVING COUNT(\*) = (SELECT COUNT(\*) FROM Employee GROUP BY MGR ORDER BY COUNT(\*) DESC LIMIT 1));

43. Find and display the details of the most recently hired employees in each department and

order by hire\_date.

SELECT \*

FROM Employee

WHERE (DESIG, HDATE) IN (SELECT DESIG, MAX(HDATE) FROM Employee GROUP BY DESIG)

ORDER BY HDATE;

44. Find the name and designation of the employees who earn a commission and salary is the

maximum.

SELECT CONCAT(FNAME, ' ', LNAME) AS Name, DESIG

FROM Employee

WHERE Commission IS NOT NULL

  AND Salary IN (SELECT MAX(Salary) FROM Employee WHERE Commission IS NOT NULL);

45. List the details of employees who are getting same salary.

SELECT \*

FROM Employee

WHERE Salary IN (SELECT Salary FROM Employee GROUP BY Salary HAVING COUNT(\*) > 1);

46. Find and display the maximum of the average salary drawn for each designation except

for President.

SELECT DESIG, AVG(Salary)

FROM Employee

WHERE DESIG != 'President'

GROUP BY DESIG

ORDER BY AVG(Salary) DESC

LIMIT 1;

47. Display the employee numbers of employees who are not managing other employees.

SELECT ENO

FROM Employee

WHERE ENO NOT IN (SELECT DISTINCT MGR FROM Employee WHERE MGR IS NOT NULL);

48. Display the total remuneration (salary + commission) of all employees working under the

manager number 110.

SELECT SUM(Salary + COALESCE(Commission, 0)) AS 'Total Remuneration'

FROM Employee

WHERE MGR = 110;

49. Update the commission column with 10% increase in salary for the employees whose

birthday falls in the current month.

UPDATE Employee

SET Commission = Commission + (Salary \* 0.1)

WHERE MONTH(DOB) = MONTH(CURRENT\_DATE);

50. Remove all the employees who are earning more than Rs. 40,000/- as their salary except

‘president’ and ‘manager’.

DELETE

FROM Employee

WHERE Salary > 40000

  AND DESIG NOT IN ('President', 'Manager');