**DQL**

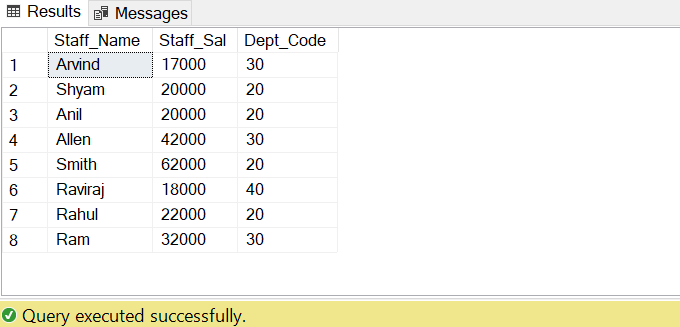
**1. Retrieve the details (Name, Salary and dept code) of the staff who are working in**

**department code 20, 30 and 40.**

SELECT Staff\_Name, Staff\_Sal, Dept\_Code

FROM Staff\_Masters

WHERE Dept\_Code IN ('20', '30', '40');



**2. Display the code and total marks for every student. Total Marks will be calculated as**

**subject1+subject2+subject3 .(Refer Student\_marks table )**

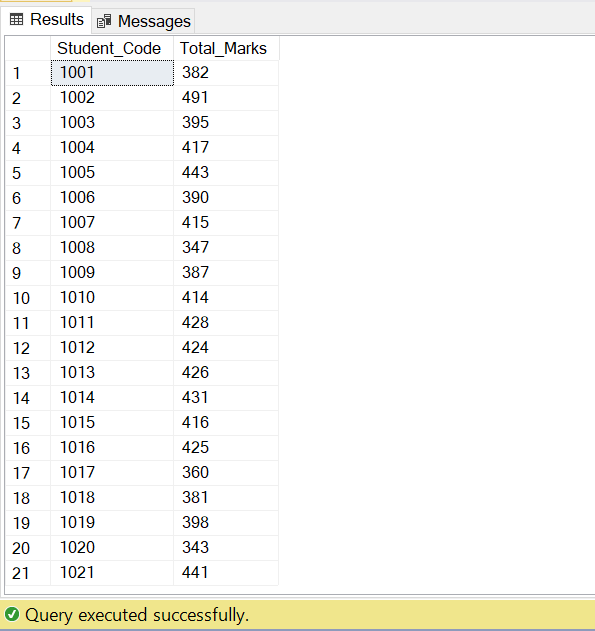
SELECT masters.Student\_Code,

SUM(marks.Subject1 + marks.Subject2 + marks.Subject3) AS Total\_Marks

FROM Student\_Masters masters

JOIN Student\_Marks marks ON masters.Student\_Code = marks.Student\_Code

GROUP BY masters.Student\_Code;



**3. List the Name and Designation code of the staff who have joined before Jan 2003 and whose salary range is between 12000 and 25000. Display the columns with user defined Column headers. Hint: Use As clause along with other operators**

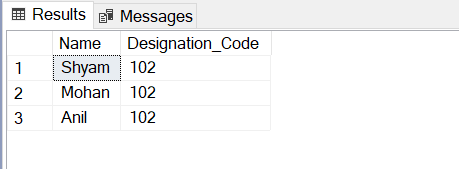
SELECT Staff\_Name AS Name, Design\_Code AS Designation\_Code

FROM Staff\_Masters

WHERE Hiredate < '2003-01-01'

AND

Staff\_Sal BETWEEN 12000 AND 25000;



**4. List the code, name, and department number of the staff who have experience of 18**

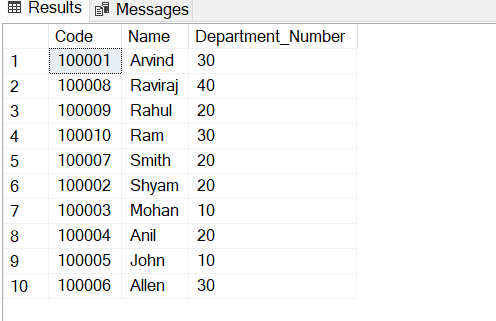
**or more years and sort them based on their experience.**

SELECT Staff\_Code AS Code, Staff\_Name AS Name, Dept\_Code AS Department\_Number

FROM Staff\_Masters

WHERE DATEDIFF(YEAR, Hiredate, GETDATE()) >= 18

ORDER BY DATEDIFF(YEAR, Hiredate, GETDATE());



**5. List the name, designation code, and salary for 10 years of the staff who are working**

**in departments 10 and 30.**

SELECT Staff\_Code AS Code, Staff\_Name AS Name, Dept\_Code AS Department\_Number

FROM Staff\_Masters

WHERE YEAR(GETDATE()) - YEAR(Hiredate) >= 18

ORDER BY Hiredate;

A screenshot of a computer

Description automatically generated

**6. Display name concatenated with dept code separated by comma and space. Name**

**the column as ‘Student Info’.**

SELECT CONCAT(Student\_Name, ', ', Dept\_Code) AS 'Student Info'

FROM Student\_Masters;

A screenshot of a computer

Description automatically generated

**7. Display the staff details who do not have manager. Hint: Use is null**

SELECT \*FROM Staff\_Masters

WHERE Mgr\_Code IS NULL;

A screenshot of a computer

Description automatically generated

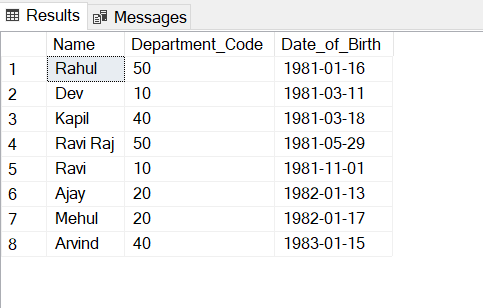
**8. Write a query which will display name, department code and date of birth of all students who were born between January 1, 1981 and March 31, 1983. Sort it based on date of birth (ascending).Hint: Use between operator**

SELECT Student\_Name AS Name,Dept\_Code AS Department\_Code,Student\_Dob AS Date\_of\_Birth

FROM Student\_Masters

WHERE Student\_Dob BETWEEN '1981-01-01' AND '1983-03-31'

ORDER BY Student\_Dob ASC;



**9. Display the Book details that were published during the period of 2001 to 2004. Also display book details with Book name having the character ‘&’ anywhere.**

SELECT \*FROM Book\_Masters

WHERE (Book\_pub\_year BETWEEN 2001 AND 2004) OR (Book\_name LIKE '%&%');

A screenshot of a computer

Description automatically generated

**10. Display the Book details where the records have the word “COMP” anywhere in the**

**Book name.**

SELECT \*FROM Book\_Masters

WHERE Book\_name LIKE '%COMP%';

A screenshot of a computer

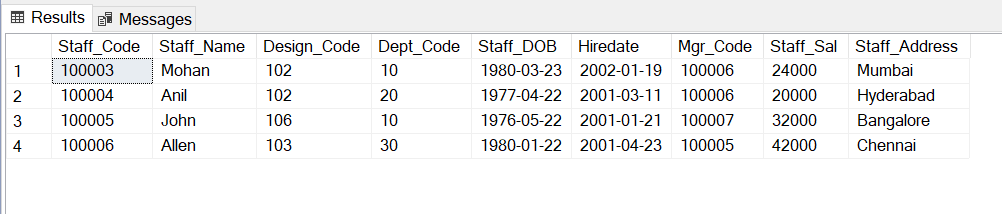
Description automatically generated

**11. List the details of the staff, whose names start with ‘A’ and end with ‘S’ or whose names contains N as the second or third character, and ending with either ‘N’ or ‘S’.**

SELECT \*FROM Staff\_Masters

WHERE (Staff\_Name LIKE 'A%S' OR Staff\_Name LIKE 'N\_%'

OR Staff\_Name LIKE '\_N%' OR Staff\_Name LIKE '%N' OR Staff\_Name LIKE '%S');



**12. List the names of the staff having ‘\_’ character in their name.**

SELECT Staff\_Name FROM Staff\_Masters

WHERE Staff\_Name LIKE '%\\_%' ESCAPE '\';

A screenshot of a computer

Description automatically generated

=====================================================================

**JOINS**

**1. Write a query which displays Staff Name, Department Code, Department Name, and**

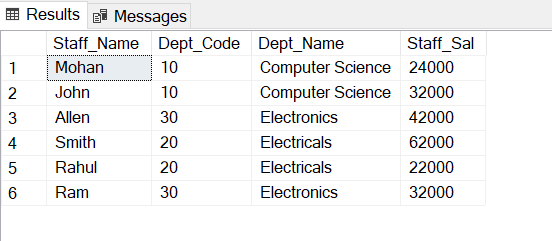
**Salary for all staff who earns more than 20000.**

SELECT sm.Staff\_Name,sm.Dept\_Code,dm.Dept\_Name,sm.Staff\_Sal

FROM Staff\_Masters sm

JOIN Department\_Masters dm ON sm.Dept\_Code = dm.Dept\_Code

WHERE sm.Staff\_Sal > 20000;



**2. Display Staff Code, Staff Name, Department Name, and his manager’s number and name. Label the columns Staff#, Staff, Mgr#, Manager.**

SELECT s1.Staff\_Code AS 'Staff#', s1.Staff\_Name AS 'Staff',

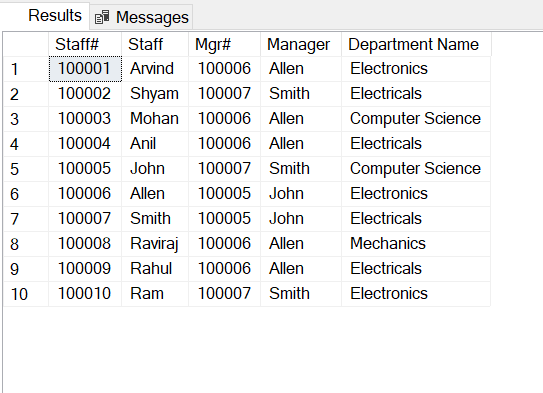
s1.Mgr\_Code AS 'Mgr#', s2.Staff\_Name AS 'Manager',

dm.Dept\_Name AS 'Department Name'

FROM Staff\_Masters s1

LEFT JOIN Staff\_Masters s2 ON s1.Mgr\_Code = s2.Staff\_Code

JOIN Department\_Masters dm ON s1.Dept\_Code = dm.Dept\_Code;



**3. Create a query that will display Student Code, Student Name, Department Name, Subject1, Subject2, and Subject3 for all students who are getting 60 and above in each subject from department 10 and 20.**

SELECT sm.Student\_Code, sm.Student\_Name, dm.Dept\_Name AS 'Department Name',

marks.Subject1, marks.Subject2, marks.Subject3

FROM Student\_Masters sm

JOIN Department\_Masters dm ON sm.Dept\_Code = dm.Dept\_Code

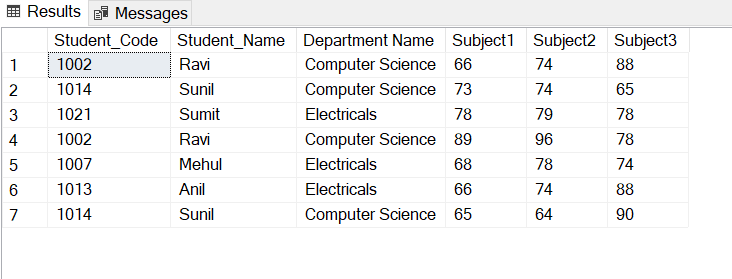
JOIN Student\_Marks marks ON sm.Student\_Code = marks.Student\_Code

WHERE (dm.Dept\_Code = '10' OR dm.Dept\_Code = '20')

AND marks.Subject1 >= 60

AND marks.Subject2 >= 60

AND marks.Subject3 >= 60;



**4. Create a query that will display Student Code, Student Name, Book Code, and Book Name for all students whose expected book return date is today.**

SELECT sm.Student\_Code,sm.Student\_Name,bt.Book\_code,bm.Book\_name

FROM Student\_Masters sm

JOIN Book\_Transactions bt ON sm.Student\_Code = bt.Student\_Code

JOIN Book\_Masters bm ON bt.Book\_code = bm.Book\_code

WHERE CONVERT(DATE, bt.Book\_expected\_return\_date) = CONVERT(DATE, GETDATE());

A screenshot of a computer

Description automatically generated

**5. Create a query that will display Staff Code, Staff Name, Department Name, Designation name, Book Code, Book Name, and Issue Date. For only those staff who have taken any book in last 30 days.**

SELECT sm.Staff\_Code, sm.Staff\_Name, dm.Dept\_Name AS 'Department Name',

des.Design\_Name AS 'Designation Name', bt.Book\_code,bm.Book\_name,bt.Book\_issue\_Date

FROM Staff\_Masters sm

JOIN Department\_Masters dm ON sm.Dept\_Code = dm.Dept\_Code

JOIN Designation\_Masters des ON sm.Design\_Code = des.Design\_Code

JOIN Book\_Transactions bt ON sm.Staff\_Code = bt.Staff\_Code

JOIN Book\_Masters bm ON bt.Book\_code = bm.Book\_code

WHERE DATEDIFF(DAY, bt.Book\_issue\_Date, GETDATE()) <= 30;

A screenshot of a computer

Description automatically generated

=====================================================================

**DDL**

**1. Create the Customer table with the following columns.**

**CustomerId Number(5)**

**Cust\_Name varchar2(20)**

**Address1 Varchar2(30)**

**Address2  Varchar2(30)**

CREATE TABLE Customer (

CustomerId INT,

Cust\_Name VARCHAR(20),

Address1 VARCHAR(30),

Address2 VARCHAR(30)

);

**2. Modify the Customer table Cust\_Name column of datatype with Varchar2(30), rename the column to CustomerName and it should not accept Nulls.**

-- Add a new column with the desired datatype and constraints

ALTER TABLE Customer

ADD CustomerName VARCHAR(30) NOT NULL;

-- Drop the old column

ALTER TABLE Customer

DROP COLUMN Cust\_Name;

**3.**

**a) Add the following Columns to the Customer table.**

**Gender  Varchar2(1)**

**Age  Number(3)**

**PhoneNo Number(10)**

ALTER TABLE Customer

ADD Gender VARCHAR(1),

Age INT,

PhoneNo BIGINT;

**b) Rename the Customer table to Cust\_Table**

EXEC sp\_rename 'Customer', 'Cust\_Table';

==================================================================

**DML**

**Insert rows with the following data into the Customer table.**

**1000, Allen, #115 Chicago, #115 Chicago, M, 25, 7878776**

**1001, George, #116 France, #116 France, M, 25, 434524**

**1002, Becker, #114 New York, #114 New York, M, 45, 431525**

INSERT INTO Cust\_Table (CustomerId, CustomerName, Address1, Address2, Gender, Age, PhoneNo)

VALUES

(1000, 'Allen', '#115 Chicago', '#115 Chicago', 'M', 25, 7878776),

(1001, 'George', '#116 France', '#116 France', 'M', 25, 434524),

(1002, 'Becker', '#114 New York', '#114 New York', 'M', 45, 431525);