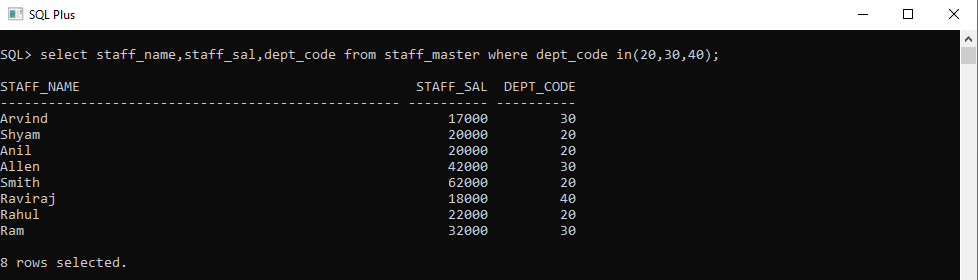
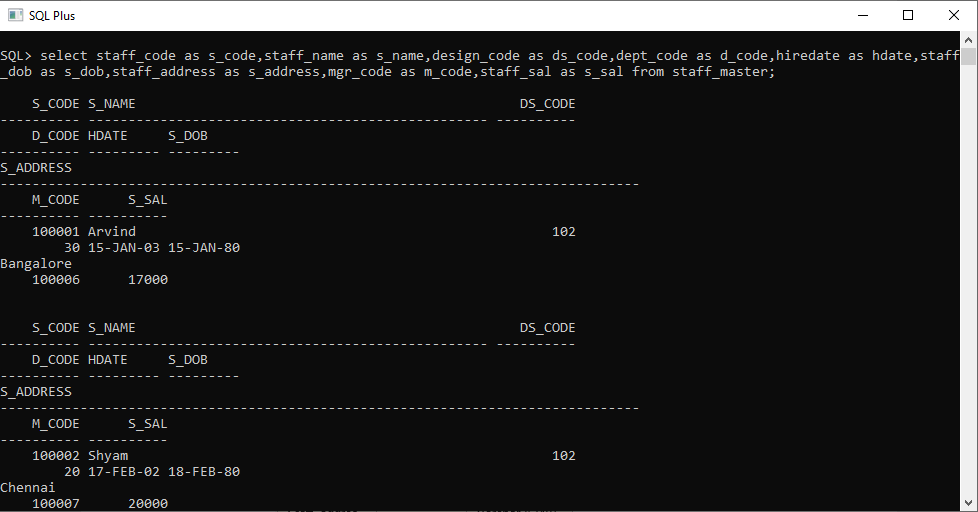
**Lab 1. Data Query Language**

* 1. **: Data Query Language**

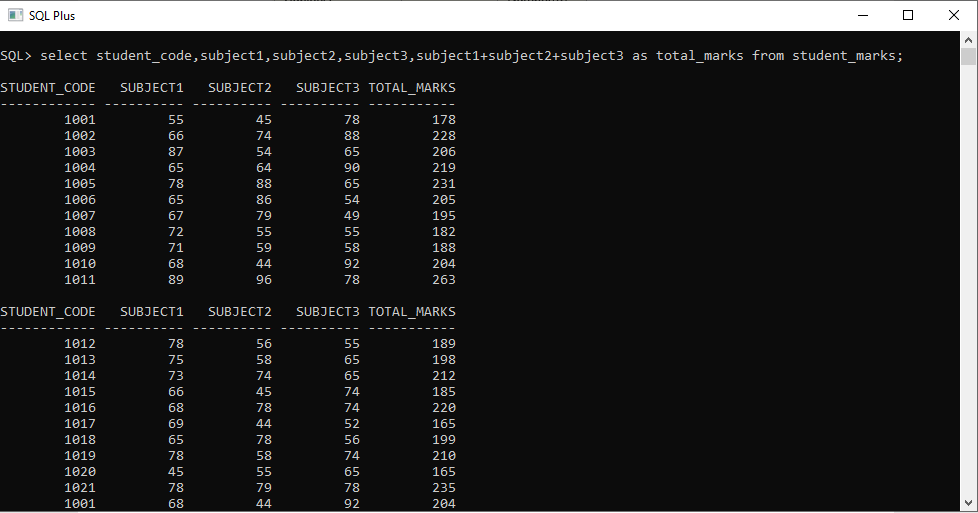
1. Retrieve the details (Name, Salary and dept code) of the employees who are working in department 20, 30 and 40.



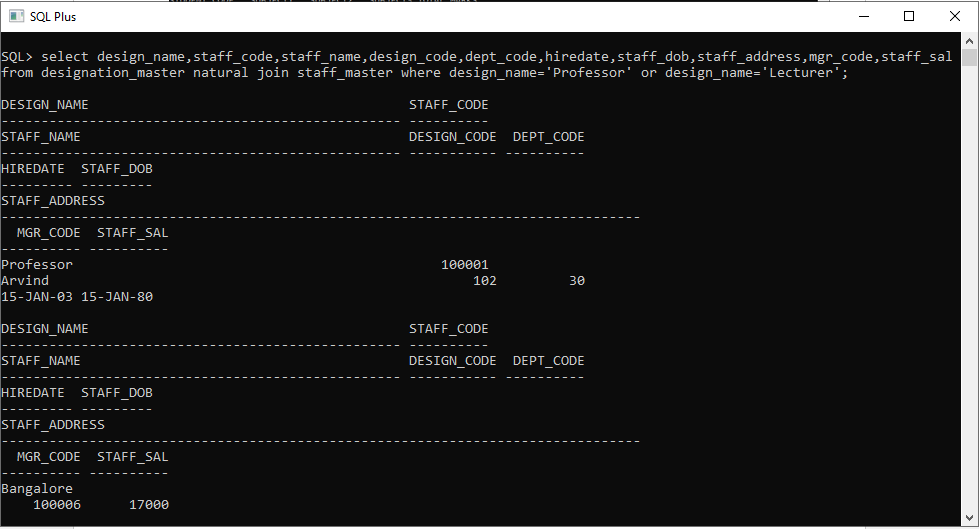
2. List the details of the employees with user defined Column headers.



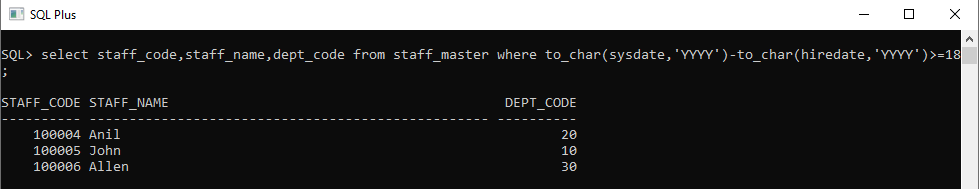
3. Display the code, subjects and total marks for every student. Total Marks will be calculated as Subject1+Subject2+Subject3. (Refer Student\_Marks table)



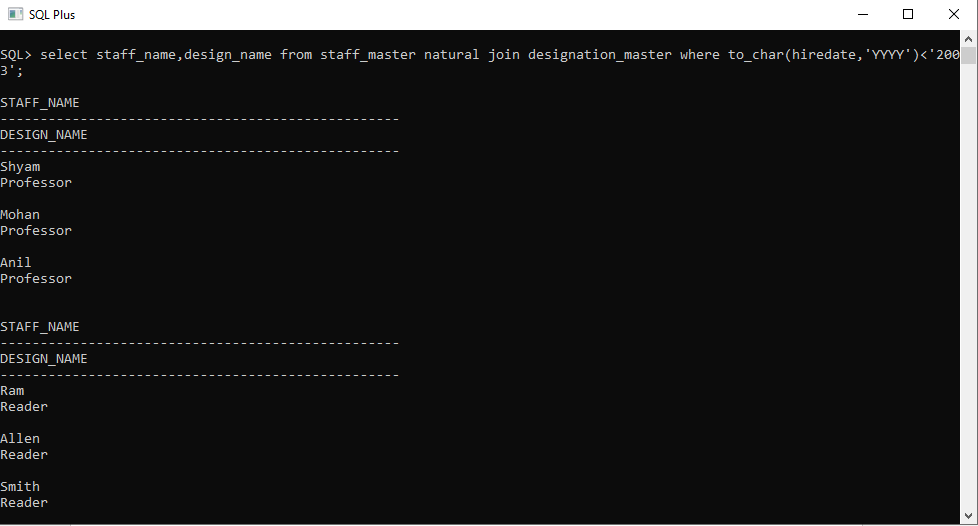
4. List the details of the staff whose designations are either PROFESSOR or LECTURER.



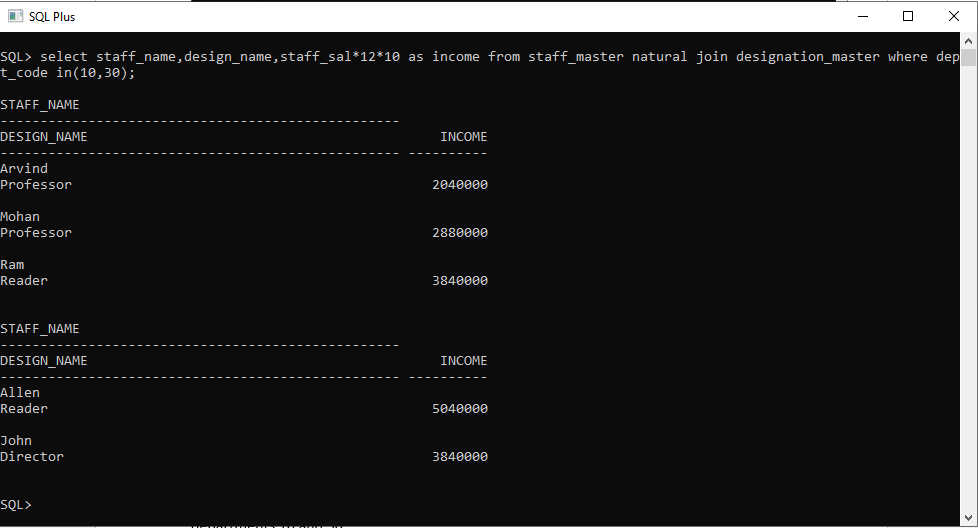
5. List the code, name, and department number of the employees who have experience of more than 18 years.



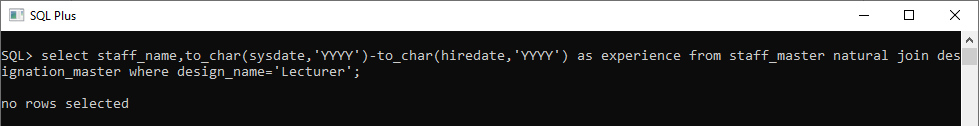
6. List the name and Designations of the staff who have joined before Jan 2003.



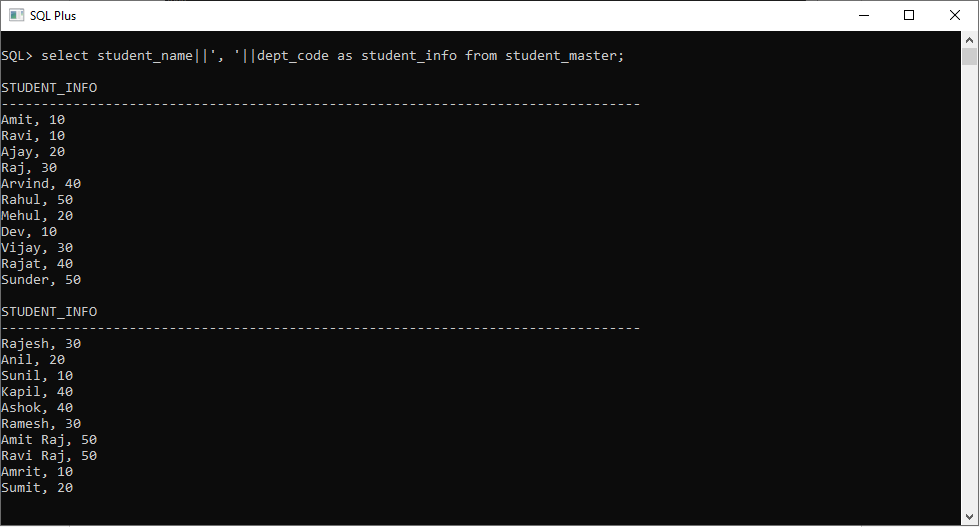
7. List the name, designation, and income for 10 years of the employees who are working in departments 10 and 30.



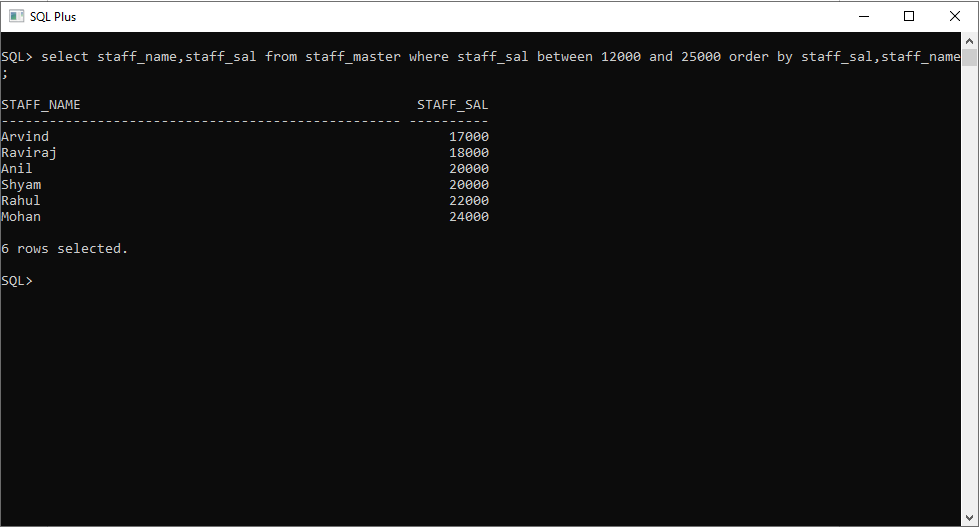
8. List the name and experience (in years) of employees who are working as LECTURER.



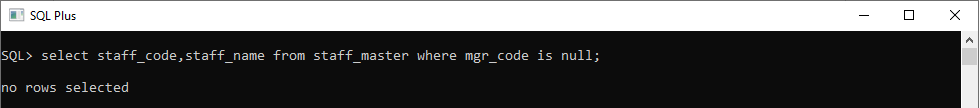
9. Display name concatenated with dept code separated by comma and space. Name the column as ‘Student Info’.



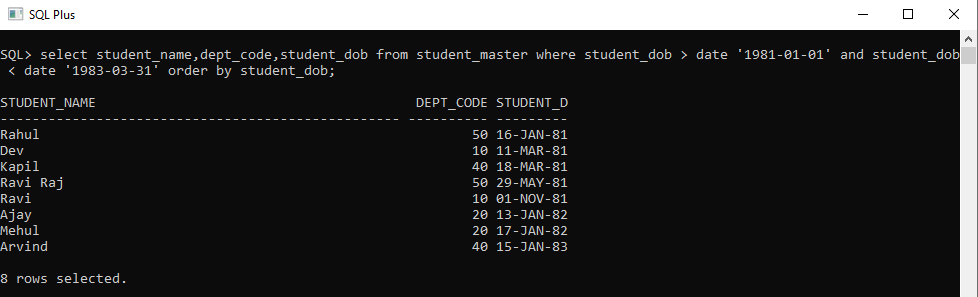
10. List the Name and Salary of the staff who are earning between 12000 and 25000. Sort them based on their salaries and name.



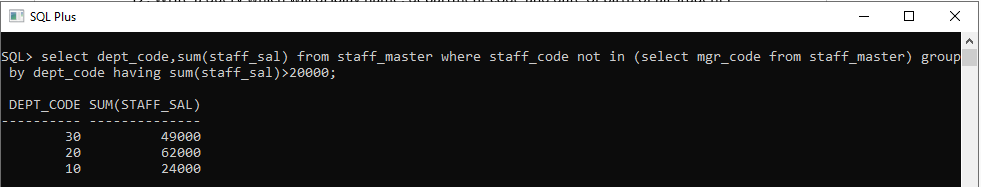
11. Display employees who do not have manager.



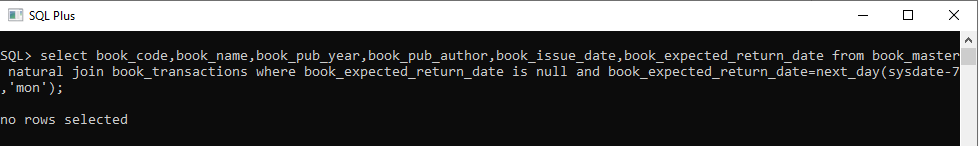
12. Write a query which will display name, department code and date of birth of all students who were born between January 1, 1981 and March 31, 1983. Sort it based on date of birth (ascending).



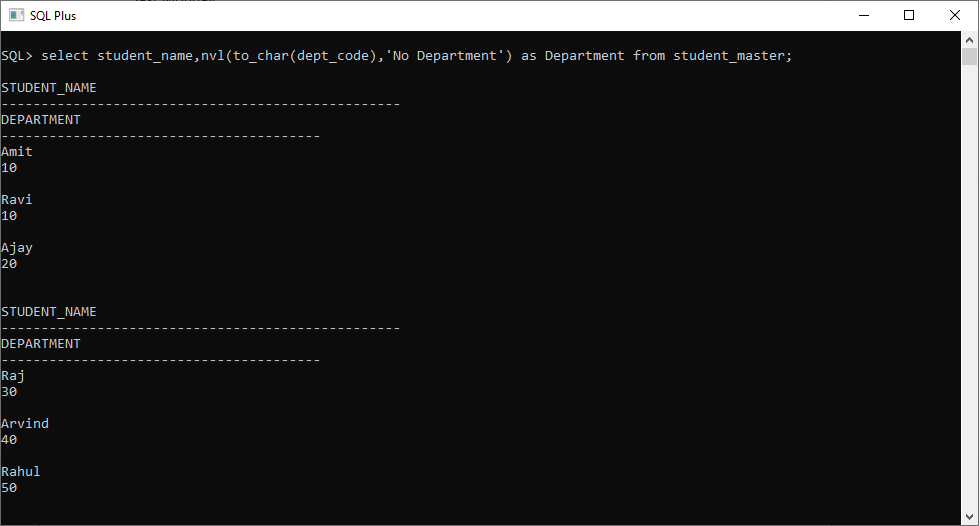
13. Get the Department number, and sum of Salary of all non managers where the sum is greater than 20000.



14. Display the details of books that have not been returned and expected return date was last Monday.



15. Display the name and department code of students. If student does not belong to any department, display “No Department”. Label the column as “Department”. (Hint: Use NVL function)

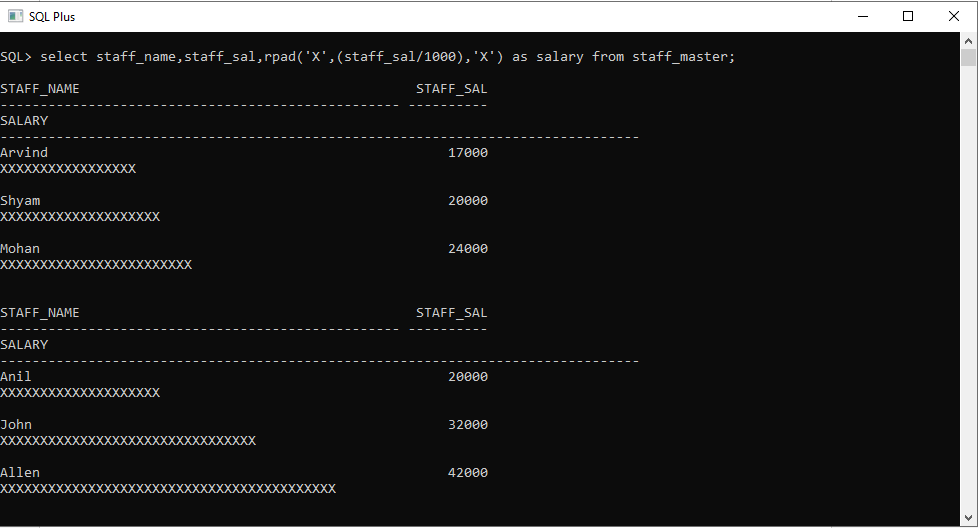


16. Display the name and salary of the staff. Salary should be represented as X. Each X represents a 1000 in salary.

Sample Output

JOHN 10000 XXXXXXXXXX

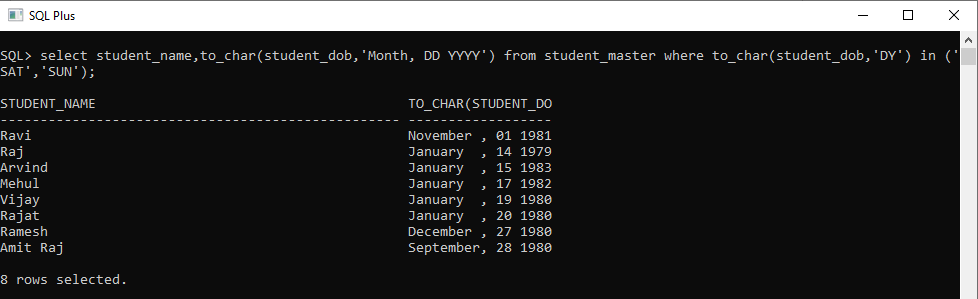
ALLEN 12000 XXXXXXXXXXXX



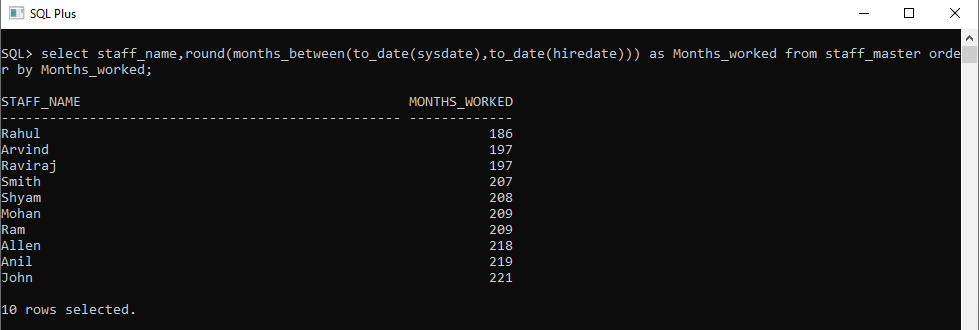
**Lab 2. Single Row Functions**

**2.1: Single Row Functions :**

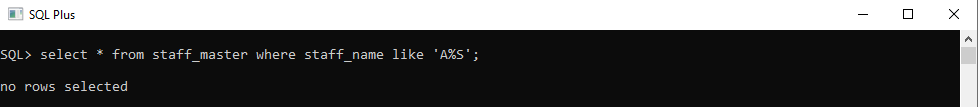
1. Display name and date of birth of students where date of birth must be displayed in the format similar to “January, 12 1981” for those who were born on Saturday or Sunday.



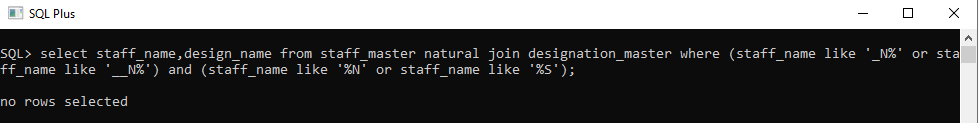
2. Display each staff name and number of months they worked for the organization. Label the column as ‘Months Worked’. Order your result by number of months employed. Round the number of months to closest whole number.



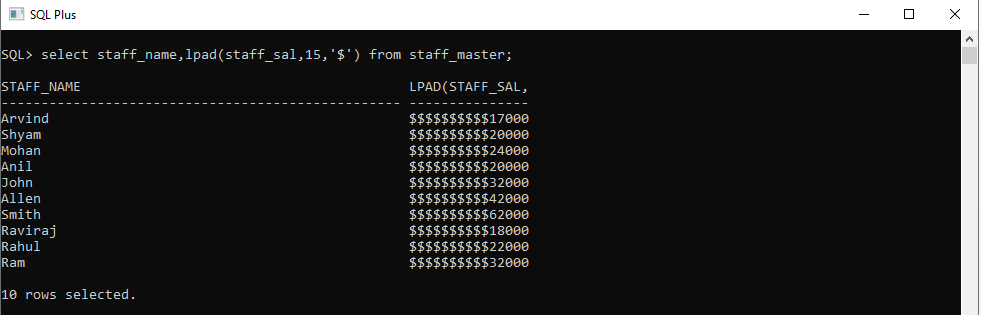
3. List the details of the employees, whose names start with ‘A’ and end with ‘S’.



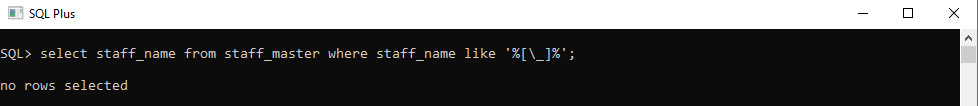
4. List the name and job of the employees whose names should contain N as the second or third character, and ending with either ‘N’ or ‘S’.



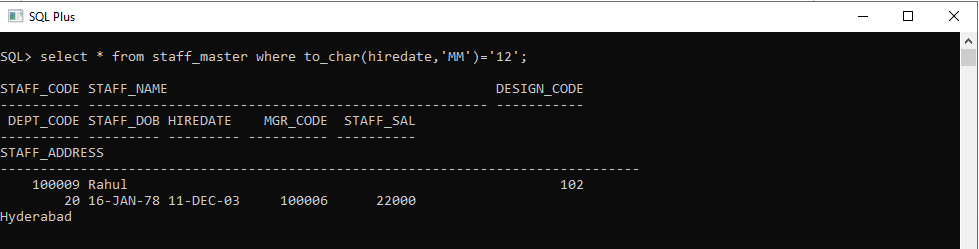
5. Create a query which will display Staff Name, Salary of each staff. Format the salary to be 15 character long and left padded with ‘$’.



6. List the names of the Employees having ‘\_’ character in their name.

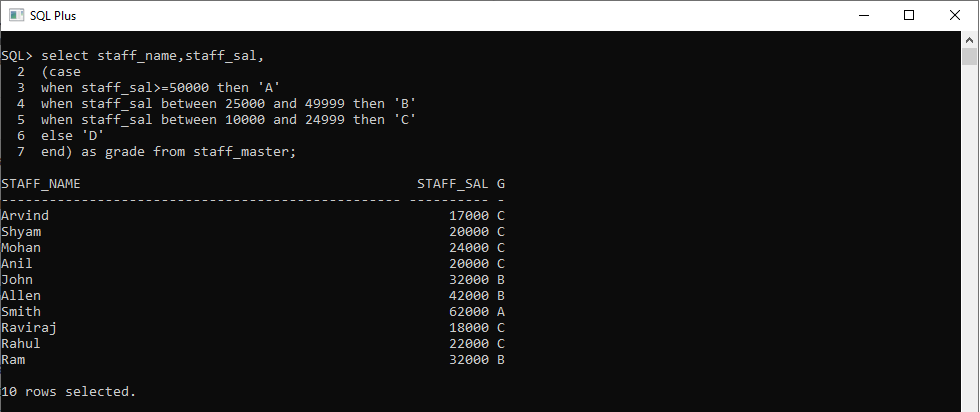


7. List the details of the employees who have joined in December (irrespective of the year).

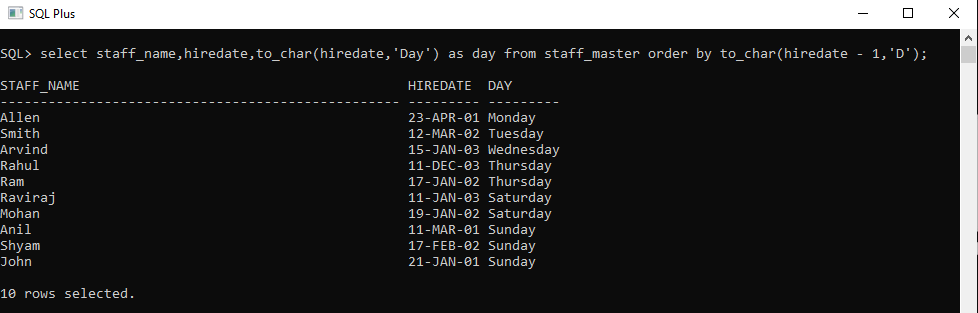


8. Write a query that displays Staff Name, Salary, and Grade of all staff. Grade depends on the following table.

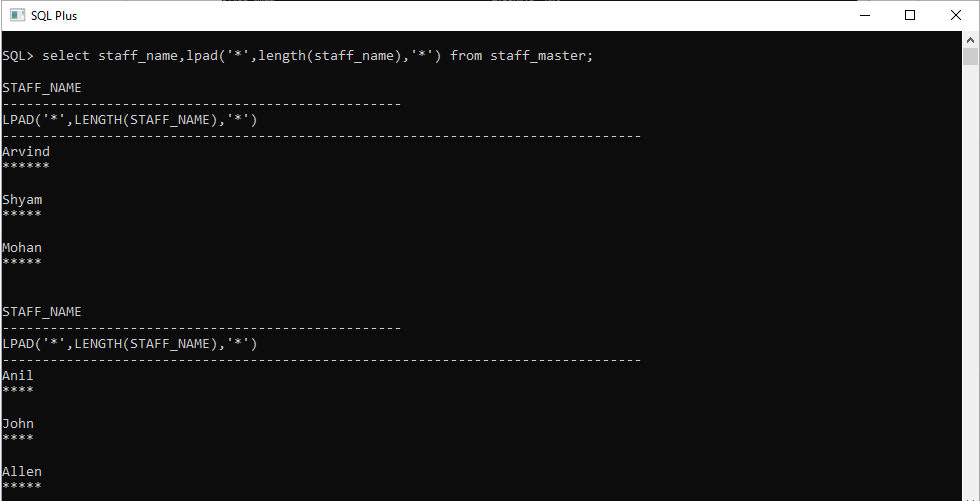
|  |  |
| --- | --- |
| Salary | Grade |
| Salary >=50000 | A |
| Salary >= 25000 < 50000 | B |
| Salary>=10000 < 25000 | C |
| OTHERS | D |



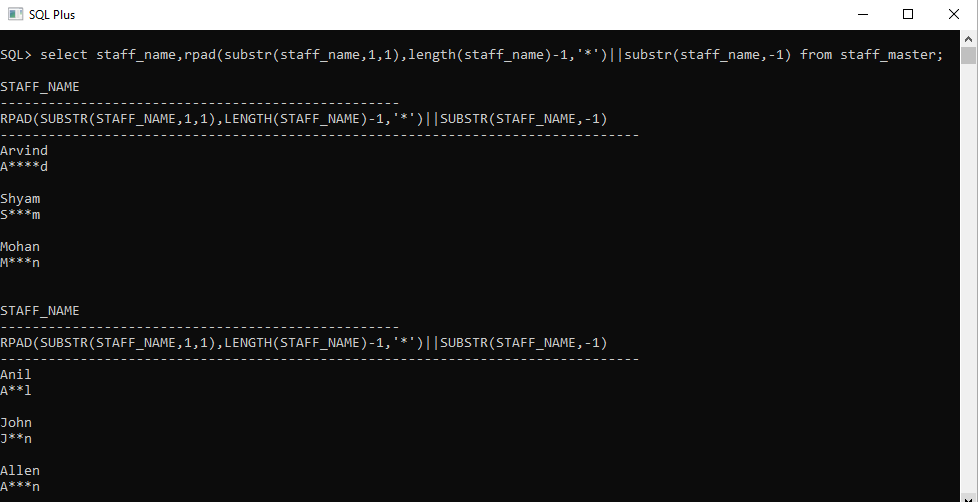
9. Display the Staff Name, Hire date and day of the week on which staff was hired. Label the column as DAY. Order the result by the day of the week starting with Monday.



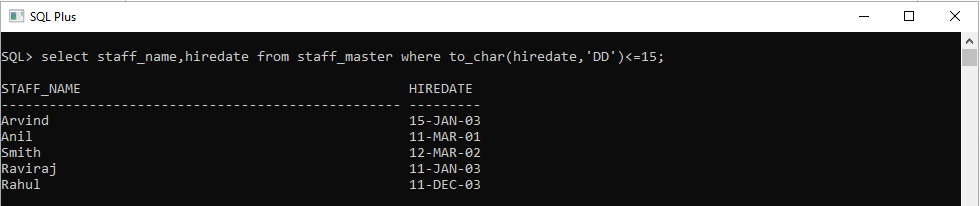
10. Show staff names with the respective numbers of asterisk from Staff table.



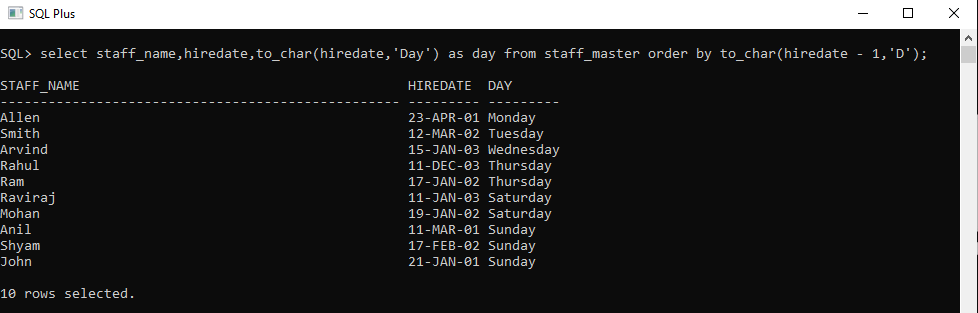
11. Show staff names with the respective numbers of asterisk from Staff table except first and last characters. For example: KING will be replaced with K\*\*G.



12. Show all staffs who were hired in the first half of the month.



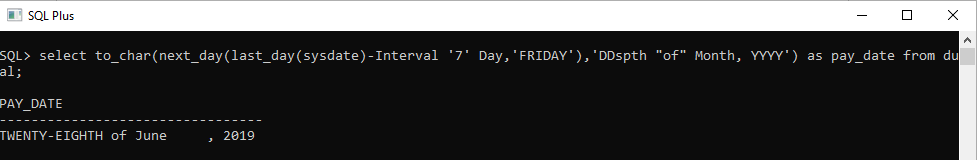
13. Display the staff name, hire date and day of the week on which the staff joined. Order the results by the day of the week starting with Monday.



14. Write a query to find the position of third occurrence of ‘i’ in the given word ‘Mississippi’.

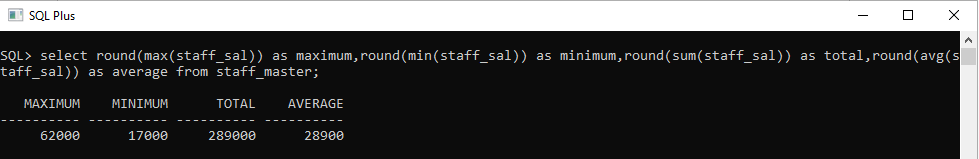


15. Write a query to find the pay date for the month. Pay date is the last Friday of the month. Display the date in the format “Twenty Eighth of January, 2002”. Label the heading as PAY DATE.

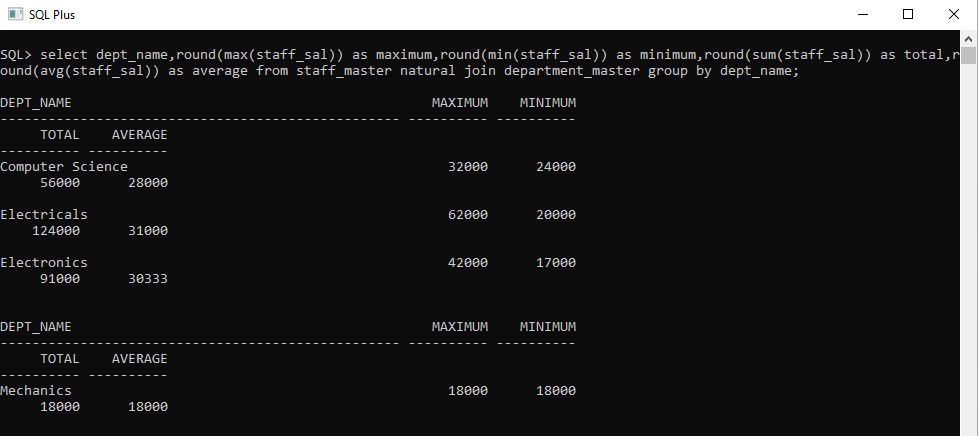


**2.2: Group Functions :**

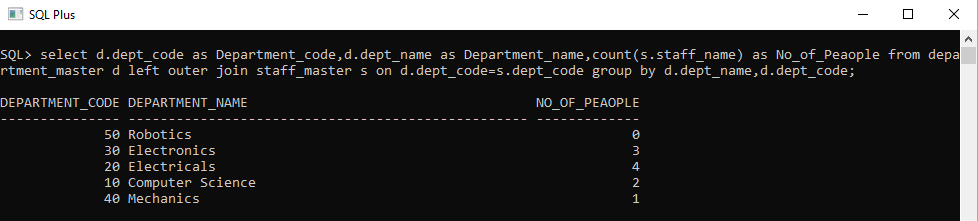
16. Display the Highest, Lowest, Total & Average salary of all staff. Label the columns Maximum, Minimum, Total and Average respectively. Round the result to nearest whole number.



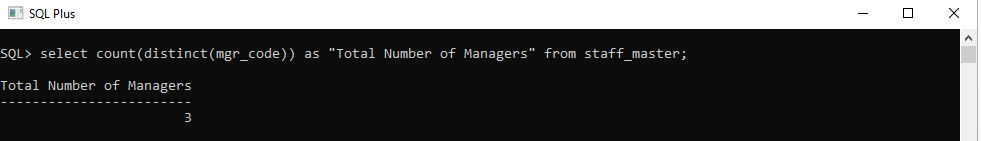
17. Edit the above query and display the same for each Department Name.



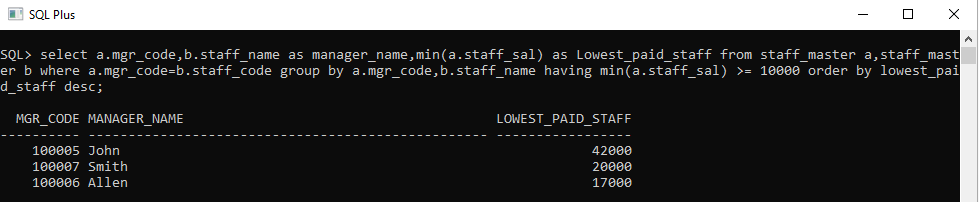
18. Write a query to display number of people in each Department. Output should display Department Code, Department Name and Number of People.



19. Determine the number of managers without listing them. Label the column as ‘Total Number of Managers’.



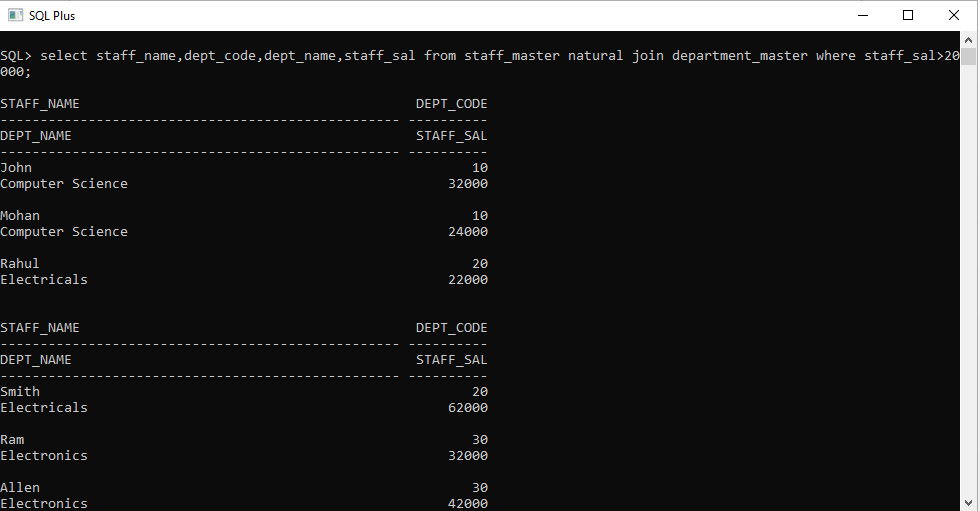
20. Display Manager Code, Manager Name and salary of lowest paid staff in that manager’s team. Exclude any group where minimum salary is less than 10000. Order the result on descending order of salary.



**Lab 3. JOINS AND SUBQUERIES**

**3.1: Joins and Subqueries**

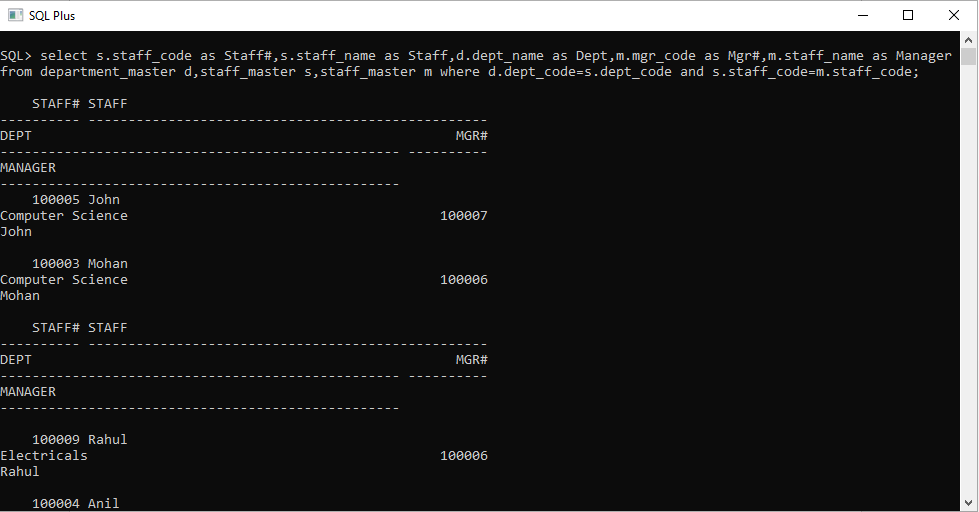
1. Write a query which displays Staff Name, Department Code, Department Name, and Salary for all staff who earns more than 20000.



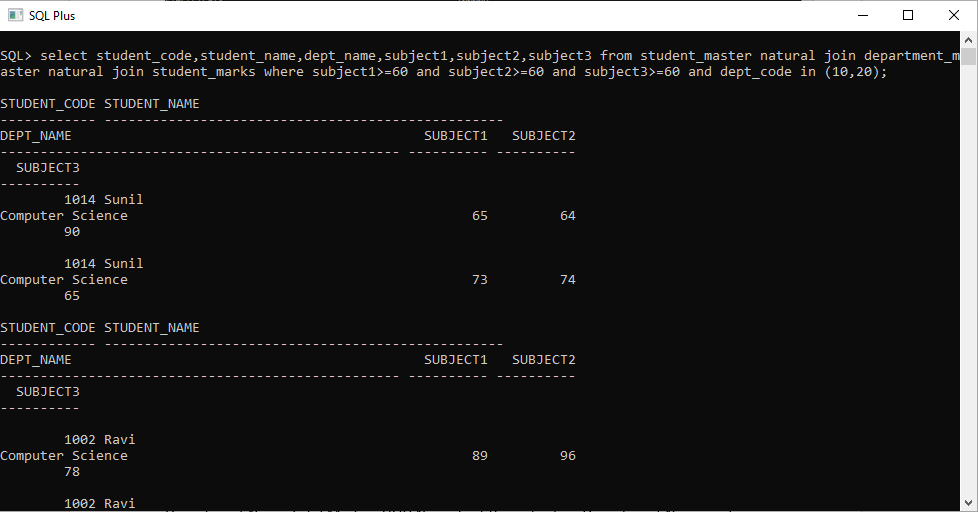
2. Write a query to display Staff Name, Department Code, and Department Name for all staff who do not work in Department code 10 and have ‘A’ in their name.



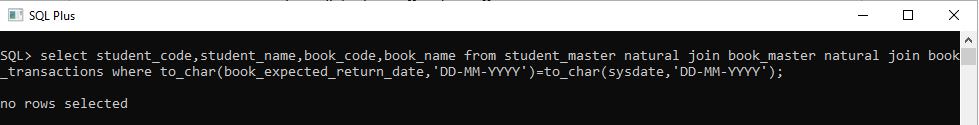
3. Display Staff Code, Staff Name, Department Name, and his manager’s number and name. Label the columns Staff#, Staff, Mgr#, Manager.



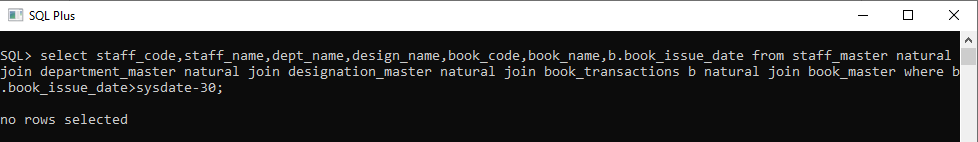
4. Create a query that will display Student Code, Student Name, Department Name, Subjec1, Subject2, and Subject3 for all students who are getting 60 and above in each subject from department 10 and 20.



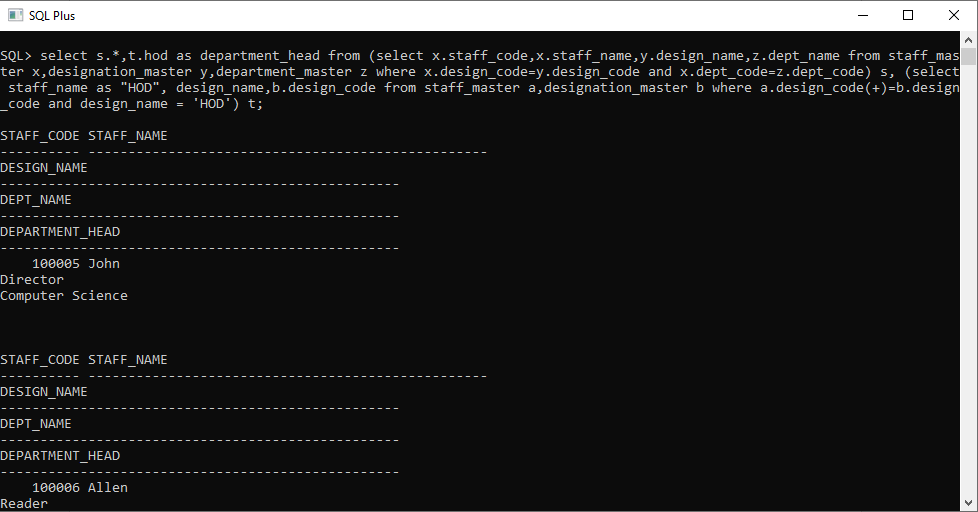
5. Create a query that will display Student Code, Student Name, Book Code, and Book Name for all students whose expected book return date is today.



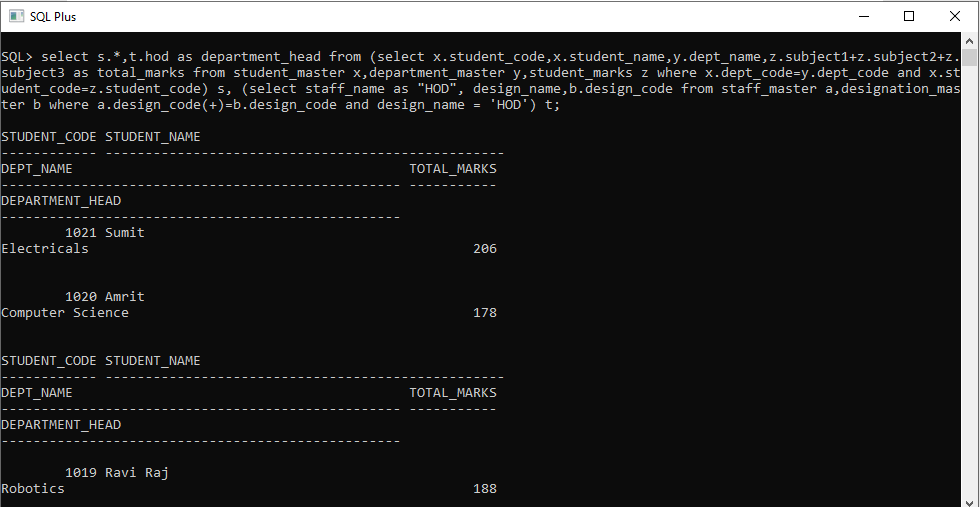
6. Create a query that will display Staff Code, Staff Name, Department Name, Designation, Book Code, Book Name, and Issue Date. For only those staff who have taken any book in last 30 days.



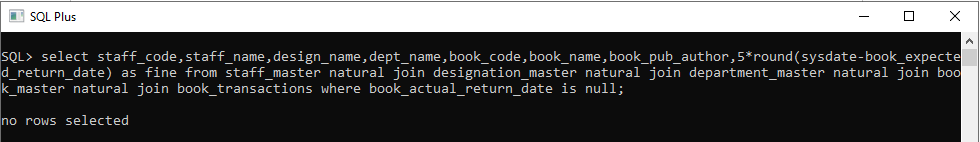
7. Generate a report which contains the following information. Staff Code Staff Name Designation Department Name Department Head For all staff excluding HOD (List should not contain the details of Department head).



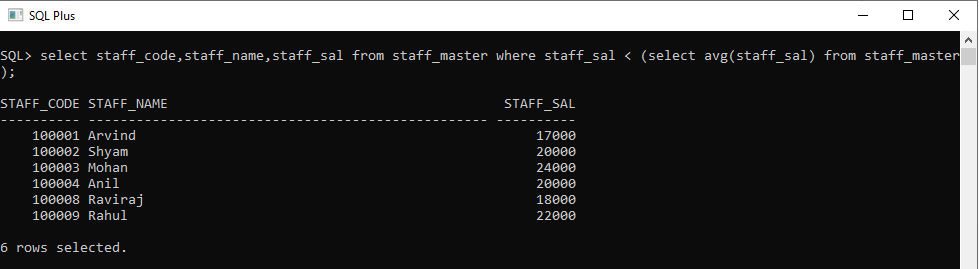
8. Generate a report which contains the following information Student Code Student Name Department Name Total Marks HOD Name Sort the output on Department Name and Total Marks.



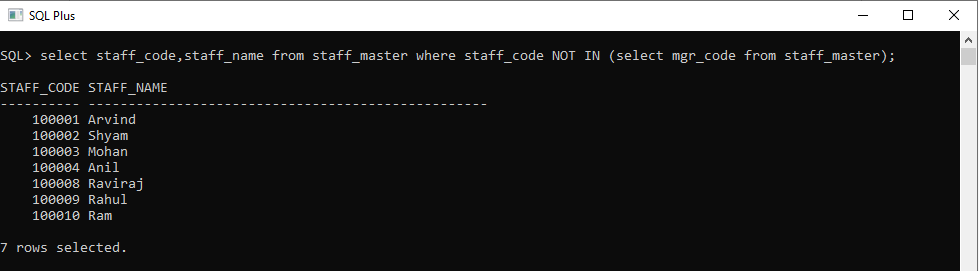
9. Generate a report which contains the following information. Staff Code, Staff Name, Designation, Department, Book Code, Book Name, Author, Fine For the staff who have not return the book. Fine will be calculated as Rs. 5 per day. Fine = 5 \* (No. of days = Current Date – Expected return date).



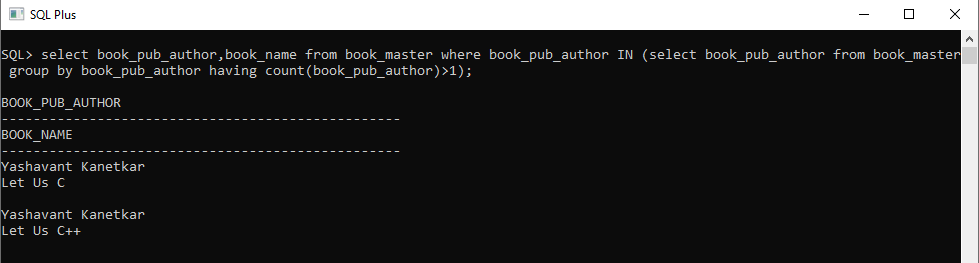
10. List Staff Code, Staff Name, and Salary for those who are getting less than the average salary of organization.



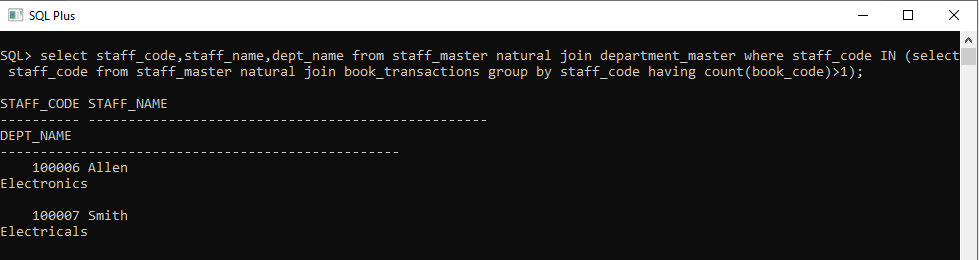
11. List the Staff Code, Staff Name who are not Manager.



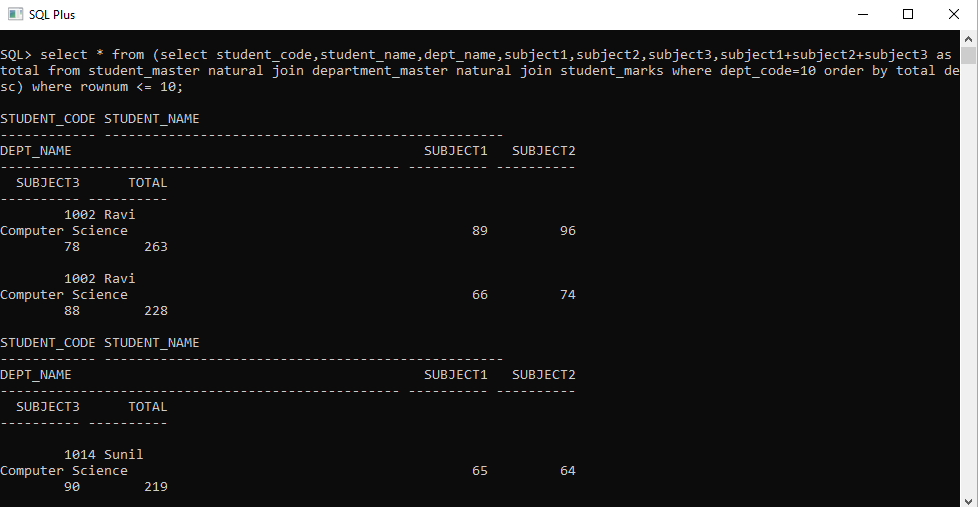
12. Display Author Name, Book Name for those authors who wrote more than one book.



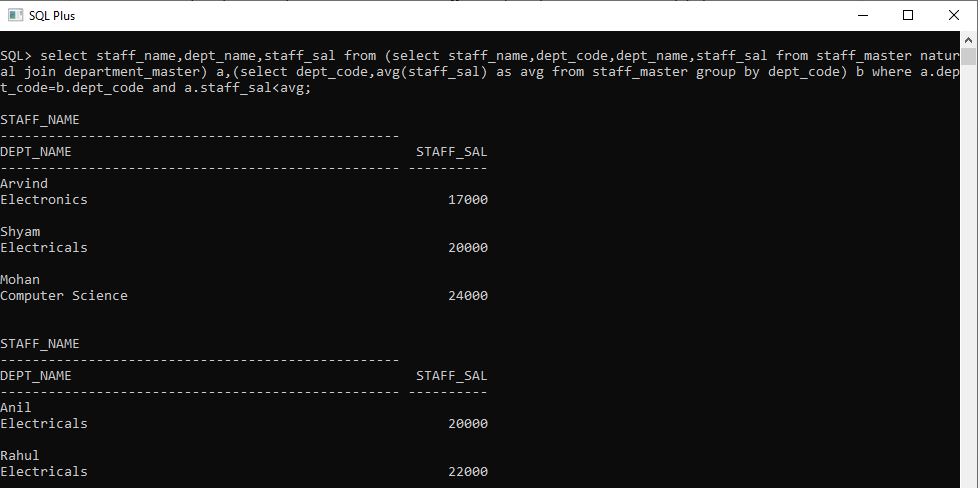
13. Display Staff Code, Staff Name, and Department Name for those who have taken more than one book.



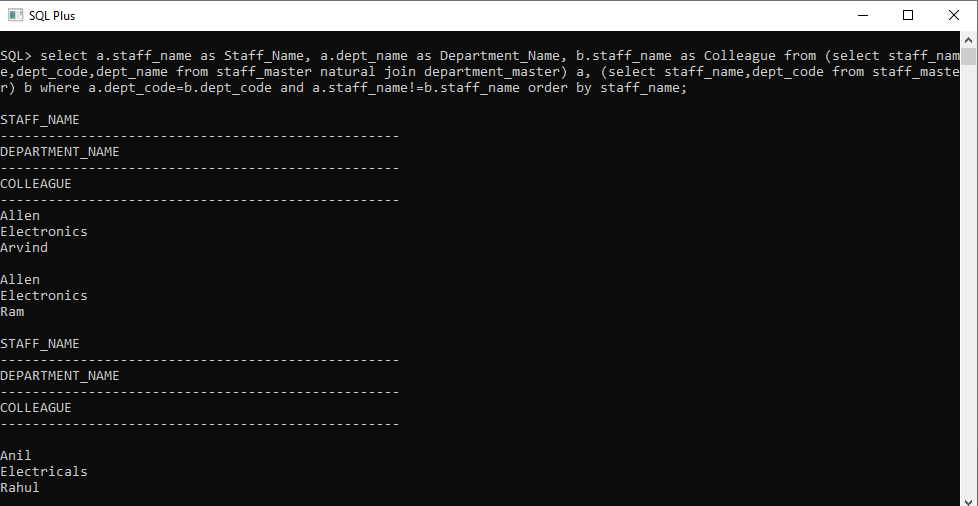
14. Display top ten students for a specified department. Details are: Student Code, Student Name, Department Name, Subject1, Subject2, Subject3, Total.



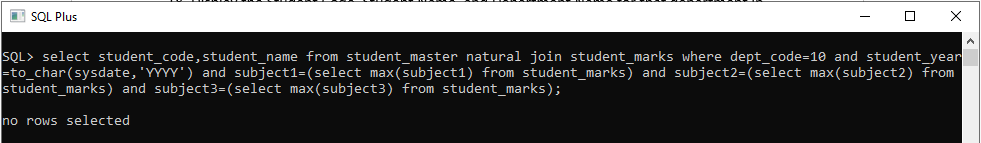
15. Display the Staff Name, Department Name, and Salary for those staff who are getting less than average salary in their own department.



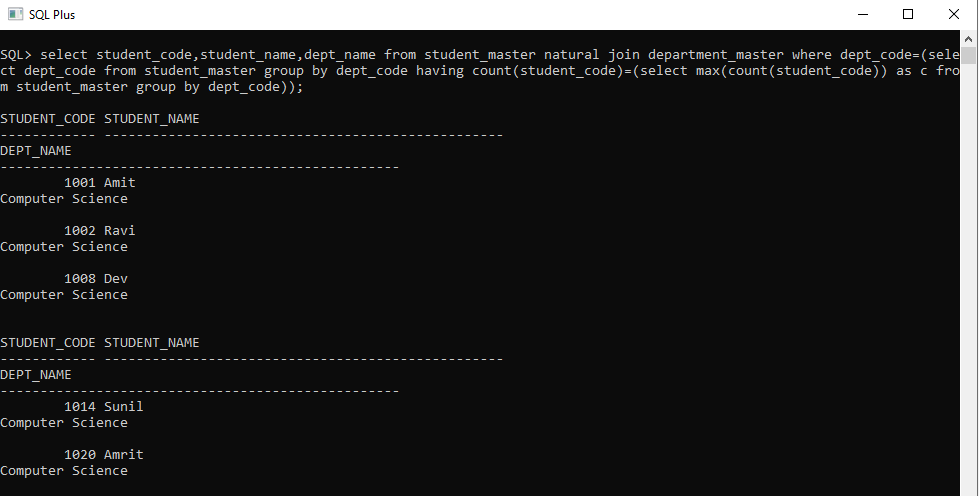
16. Create a query that will display the Staff Name, Department Name, and all the staff that work in the same department as a given staff. Give the column as appropriate label.



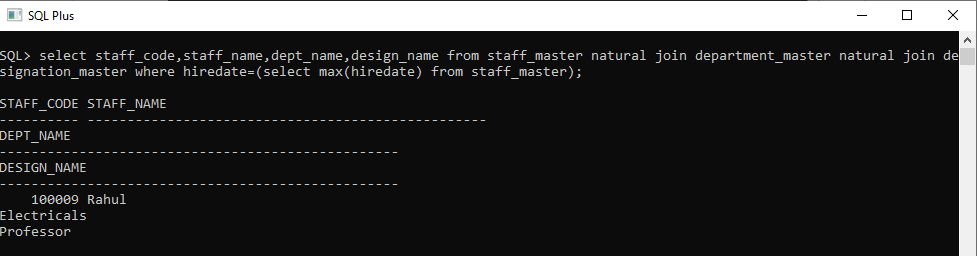
17. List the Student Code, Student Name for that student who got highest marks in all three subjects in Computer Science department for current year.



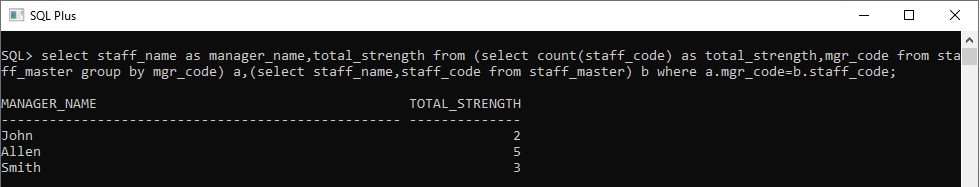
18. Display the Student Code, Student Name, and Department Name for that department in which there are maximum number of student are studying.



19. Display Staff Code, Staff Name, Department Name, and Designation for those who have joined most recently.

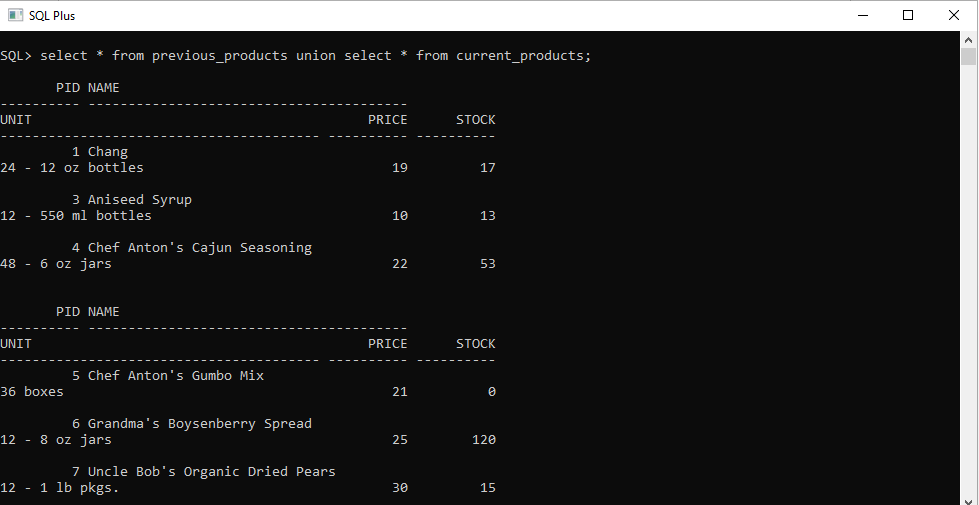


20. Display the Manager Name and the total strength of his/her team.

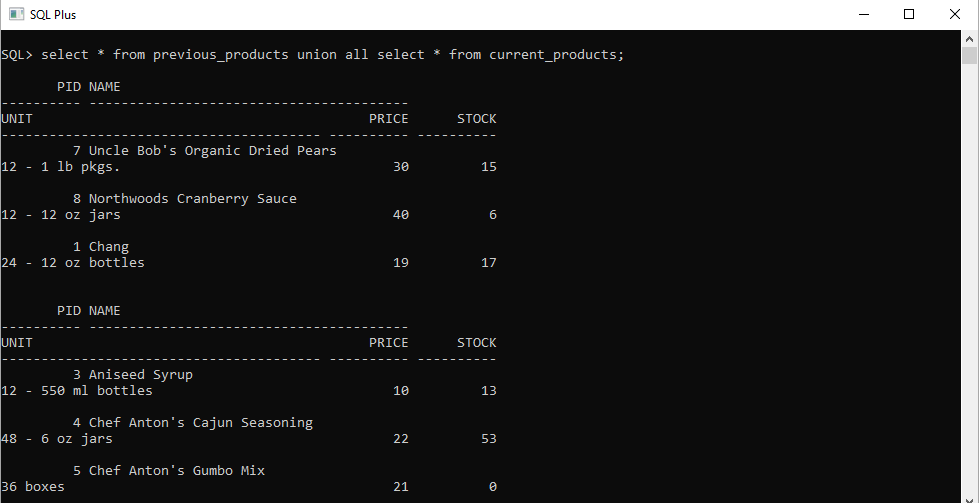


**3.2: Set Operators**

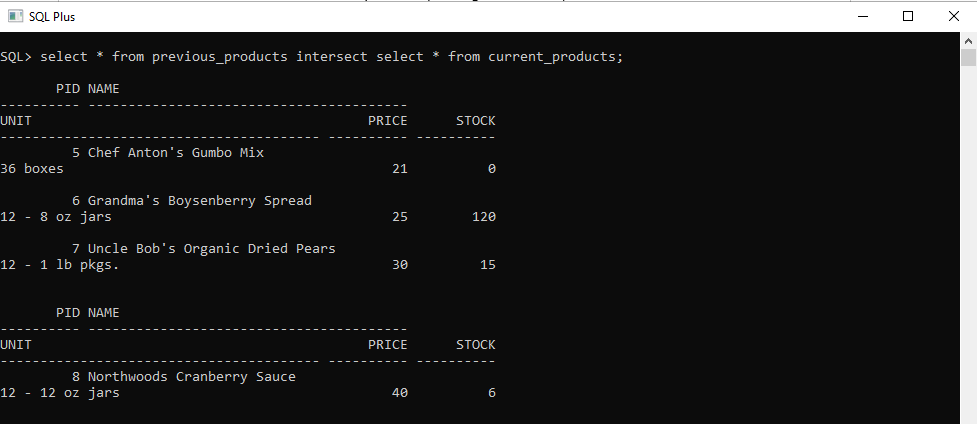
1. Get the details of all products irrespective of the fact whether they are in previous set or current set.



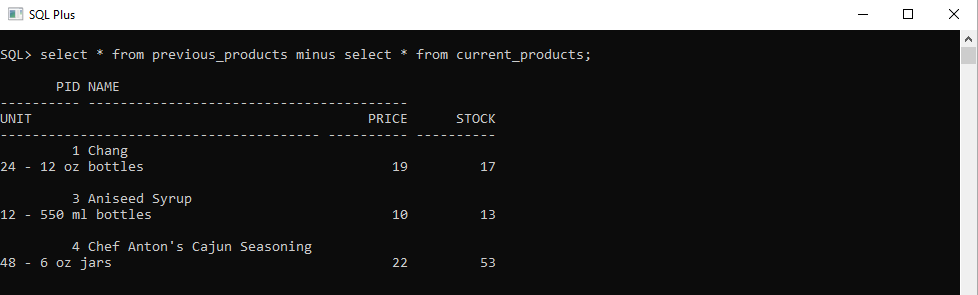
2. Get the details of all products along with the repetition of those that were present both in the previous and current sets.



3. Get the details of only those products which were present in the previous set and are still continuing.



4. Get the details of all obsolete products (no longer continued).



**Lab 4. Database Objects**

**4.1: Database Objects**

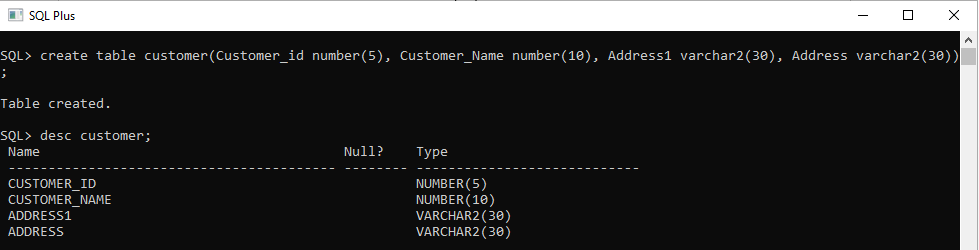
1. Create the Customer table with the following columns.

Customerid Number(5)

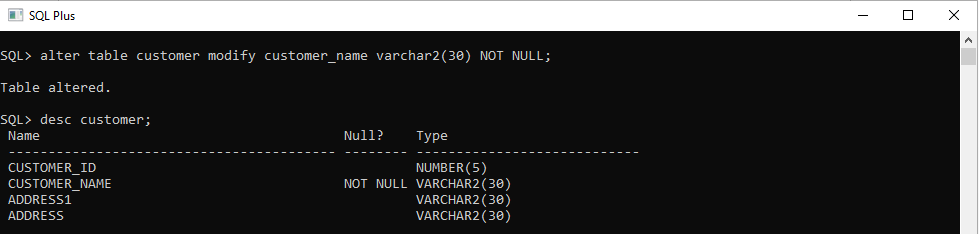
CustomerName Number(10)

Address1 Varchar2(30)

Address2 Varchar2(30)



2. Modify the Customer table CustomerName column of datatype with Varchar2(30). CustomerName should not accept Nulls.

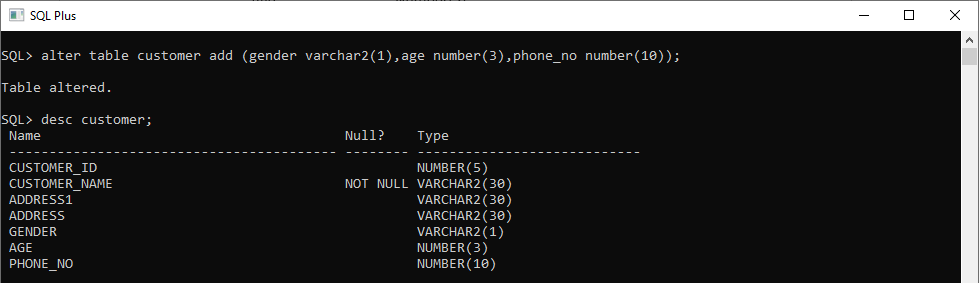


3. Add the following Columns to the Customer table.

Gender Varchar2(1)

Age Number(3)

PhoneNo Number(10)

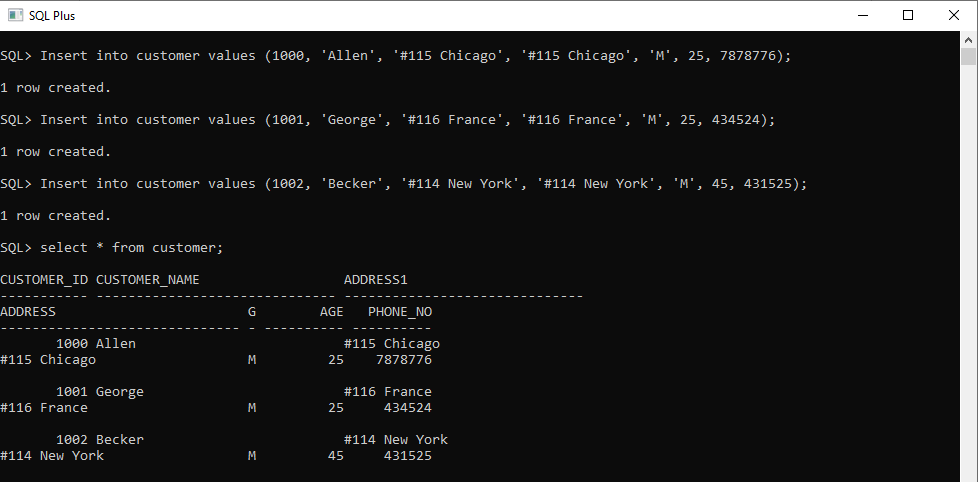


4. Insert rows with the following data in to the Customer table.

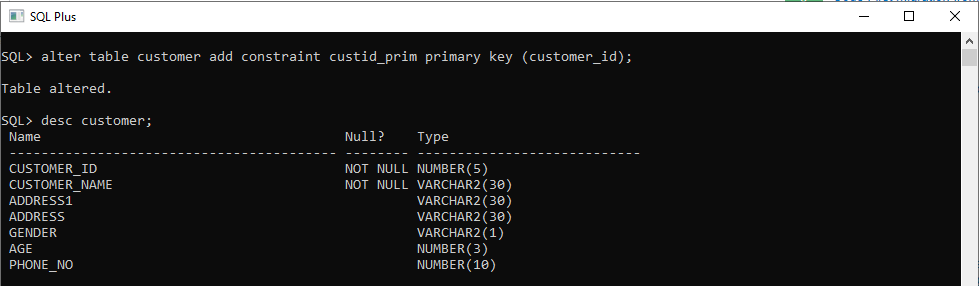
Insert into customer values: (1000, ‘Allen’, ‘#115 Chicago’, ‘#115 Chicago’, ‘M’, ‘25, 7878776’)

In similar manner, add the below records to 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

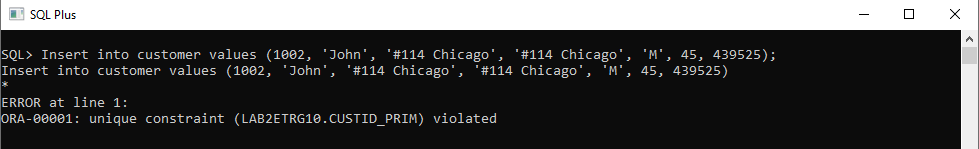


5. Add the Primary key constraint for Customerld with the name Custld\_Prim.



6. Insert the row given below in the Customer table and see the message generated by the Oracle server.

1002, John, #114 Chicago, #114 Chicago, M, 45, 439525

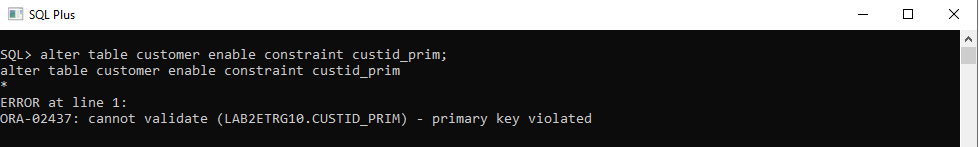


7. Disable the constraint on CustomerId, and insert the following data:

* 1002, Becker, #114 New York, #114 New york , M, 45, 431525
* 1003, Nanapatekar, #115 India, #115 India , M, 45, 431525

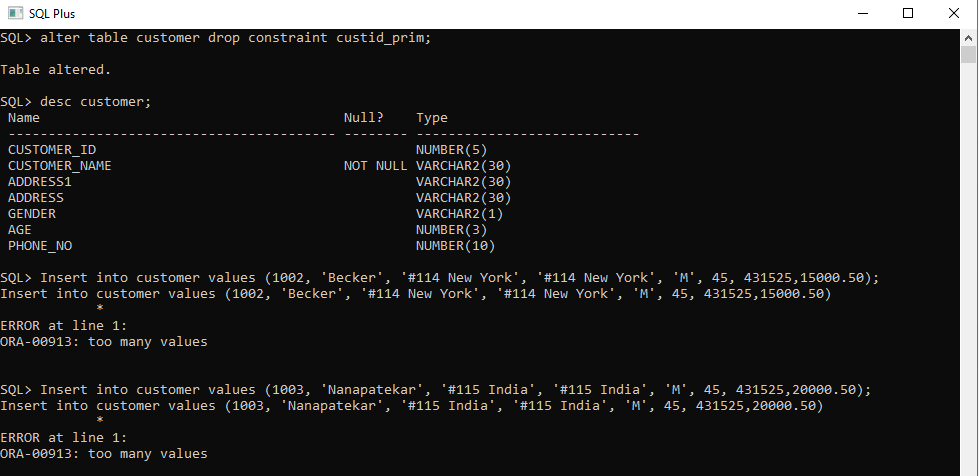


8. Enable the constraint on CustomerId of the Customer table, and see the message generated by the Oracle server.



9. Drop the constraint Custld\_Prim on CustomerId and insert the following Data. Alter Customer table, drop constraint Custid\_Prim.

* 1002, Becker, #114 New York, #114 New york , M, 45, 431525, 15000.50
* 1003, Nanapatekar, #115 India, #115 India , M, 45, 431525, 20000.50



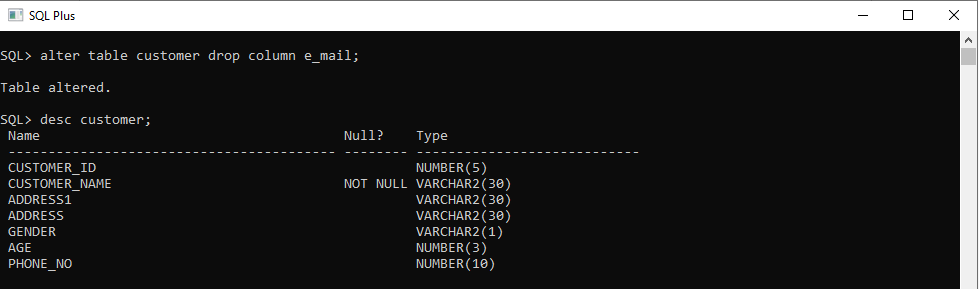
10. Delete all the existing rows from Customer table, and let the structure remain itself using TRUNCATE statement.



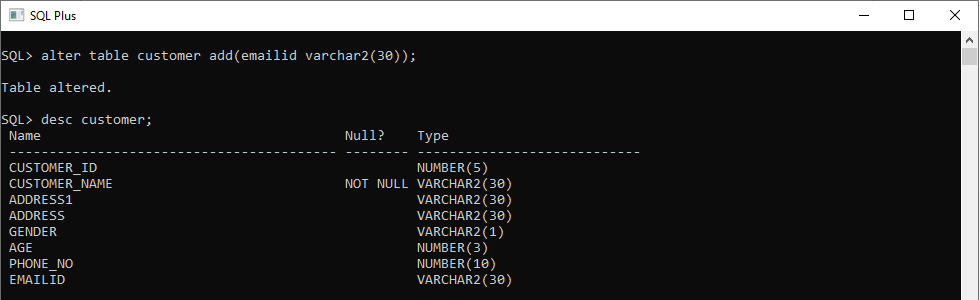
11. In the Customer table, add a column E\_mail.



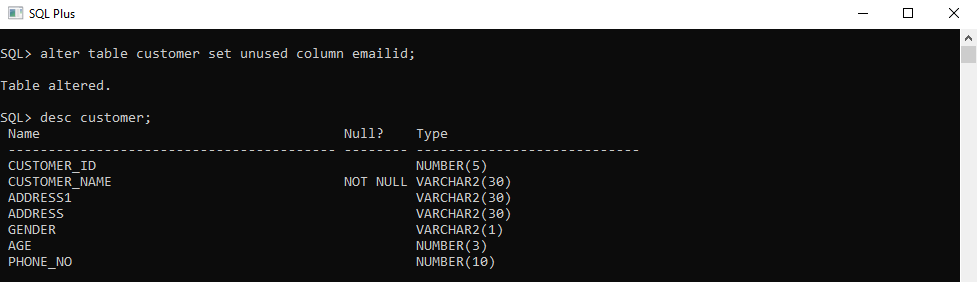
12. Drop the E\_mail column from Customer table.



13. Add a new column EmailId to Customer table.



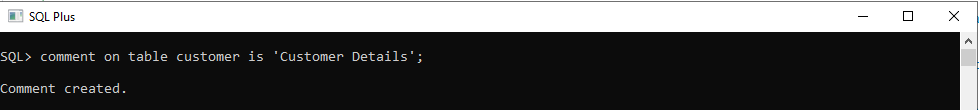
14. Mark EmailId column as unused before dropping it.



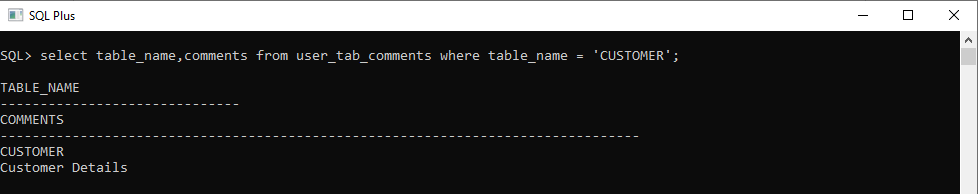
15. Drop the unused EmailId column from the Customer table.



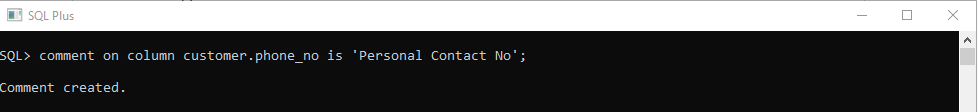
16. Define the COMMENT ‘Customers Details’ for Customer table.



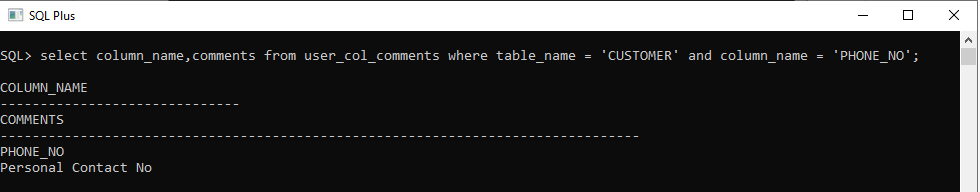
17. Use Data Dictionary USER\_TAB\_COMMENTS to view the created comment.



18. Define the COMMENT ‘Personal Contact no’ for the phoneno column of the Customer table.

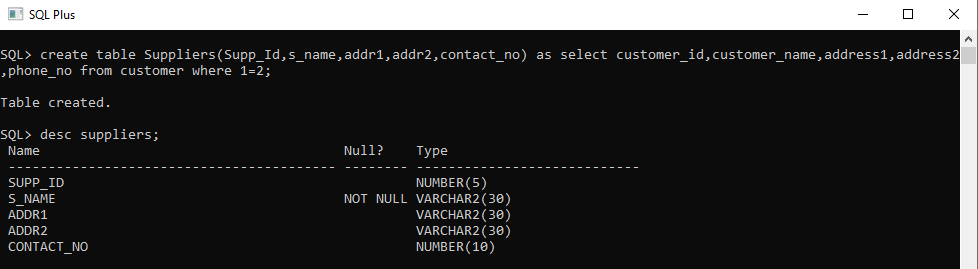


19. Use Data Dictionary USER\_COL\_COMMENTS to view the created comment.



20. Create the Suppliers table based on the structure of the Customer table. Include only the CustomerId, CustomerName, Address1, Address2, and phoneno columns.

Name the columns in the new table as SuppID, SName, Addr1, Addr2, and Contactno respectively.



21. Drop the above table and recreate the following table with the name CustomerMaster.

CustomerId Number(5) Primary key(Name of constraint is CustId\_PK)

CustomerName Varchar2(30) Not Null

Addressl Varchar2(30) Not Null

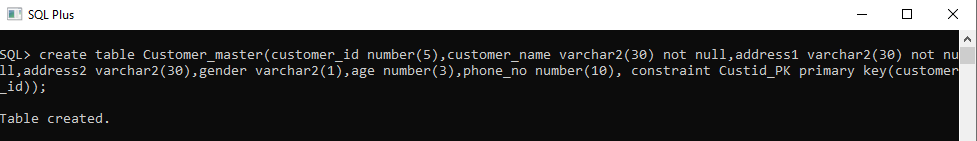
Address2 Varchar2(30)

Gender Varchar2(l)

Age Number(3)

PhoneNo Number(10)





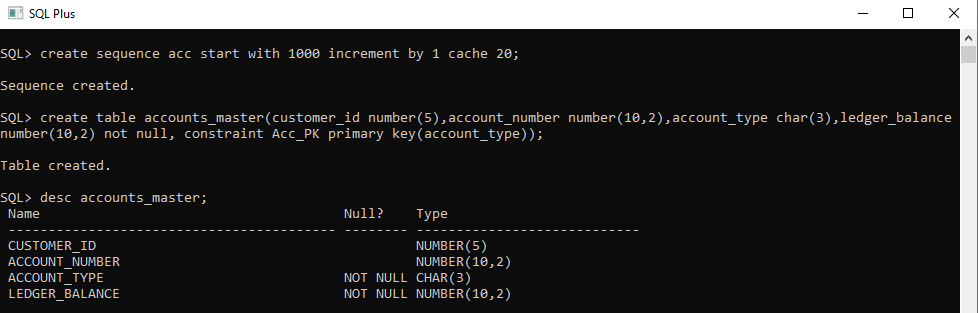
22. Create the AccountsMaster table with the following Columns. Use sequence to generate Account number

Customerld Number(5)

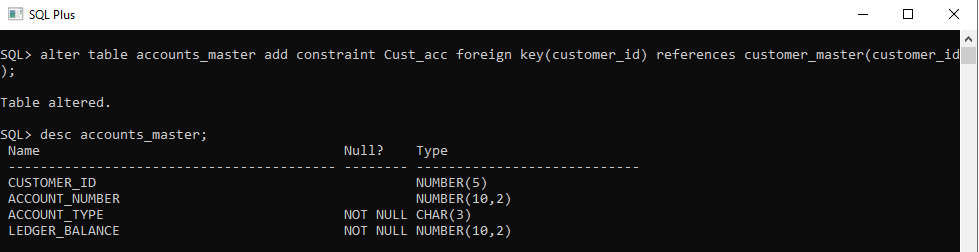
AccountNumber Number(10,2) Primary key(Name of constraint is Acc\_PK)

AccountType Char(3)

LedgerBalance Number(10,2) Not Null

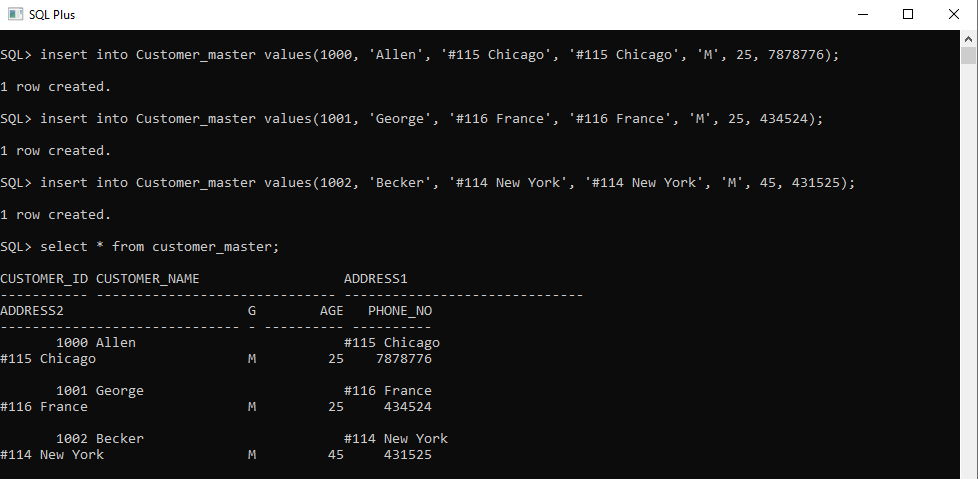


23. Relate AccountsMaster table and CustomerMaster table through Customerld column with the constraint name Cust\_acc.

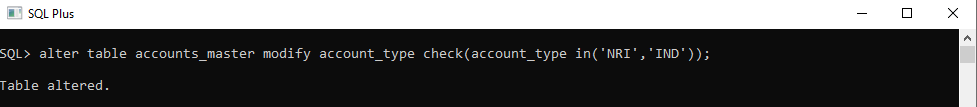


24. Insert the following rows to the CustomerMaster table:

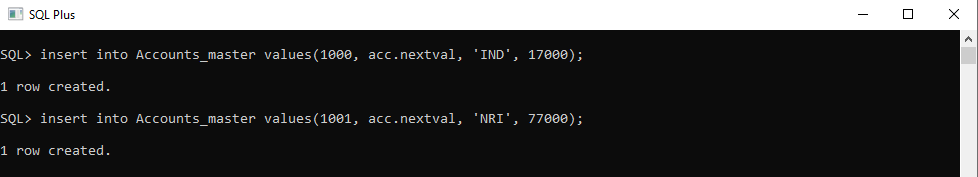
* 1000, Allen, #115 Chicago, #115 Chicago, M, 25, 7878776
* 1001, George, #116 France, #116 France, M, 25, 434524
* 1002, Becker, #114 New York, #114 New York, M, 45, 431525



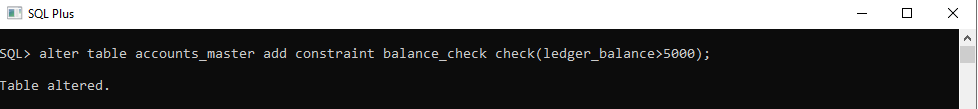
25. Modify the AccountMaster table with the Check constraint to ensure AccountType should be either NRI or IND.



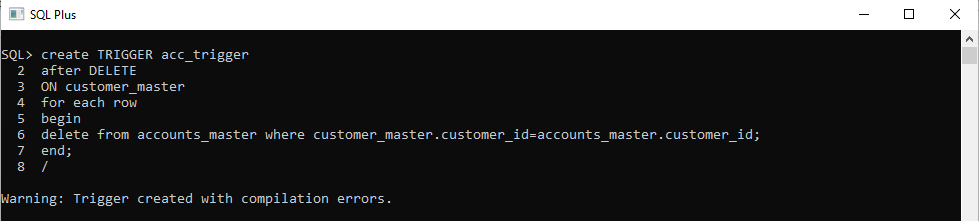
26. Insert 5 rows into the AccountsMaster table.



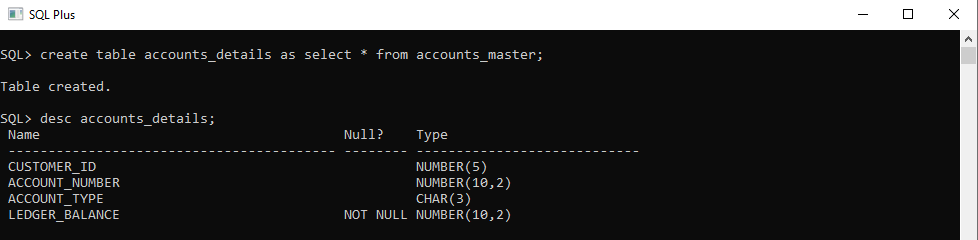
27. Modify the AccountsMaster table keeping a Check constraint with the name Balance\_Check for the Minimum Balance which should be greater than 5000.



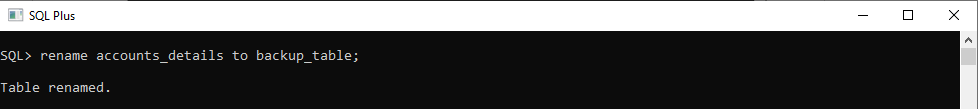
28. Modify the AccountsMaster table such that if Customer is deleted from Customer table then all his details should be deleted from AccountsMaster table.



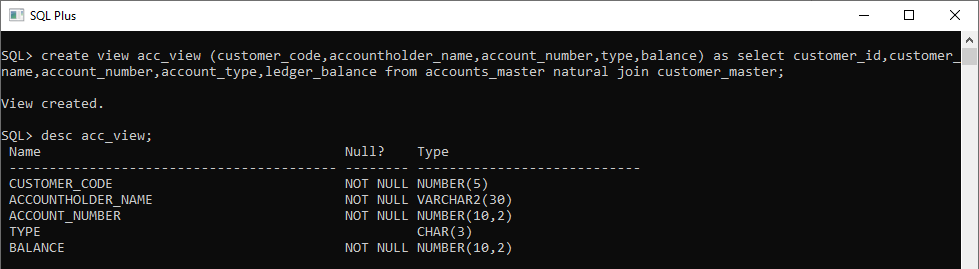
29. Create Backup copy for the AccountsMaster table with the name ‘AccountDetails’.



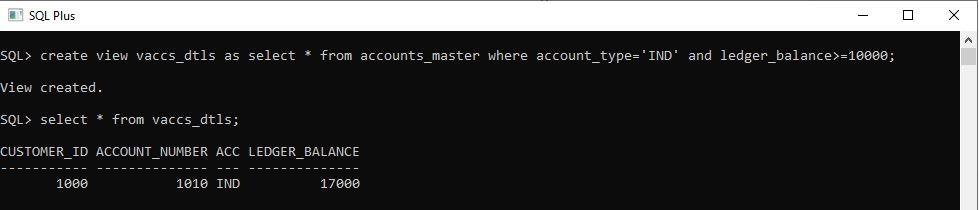
30. Change the name of the AccountDetails table to ‘BackUpTable’ table.



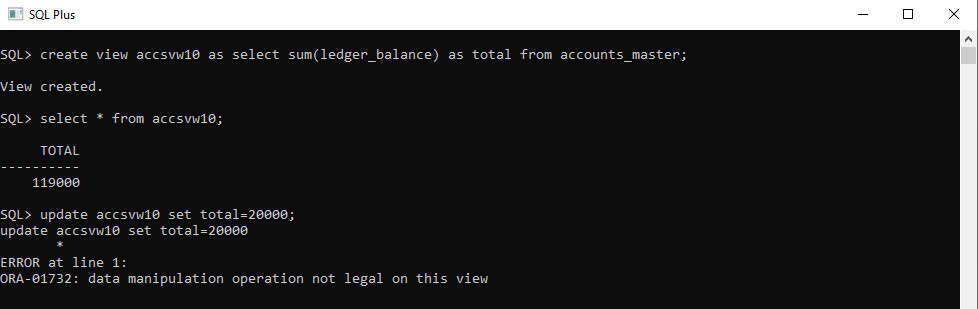
31. Create a view ‘Acc\_view’ with columns Customerld, CustomerName, AccountNumber, AccountType, and LedgerBalance from AccountsMaster. In the view Acc\_view, the column names should be CustomerCode, AccountHolderName, AccountNumber, Type, and Balance for the respective columns from AccountsMaster table.



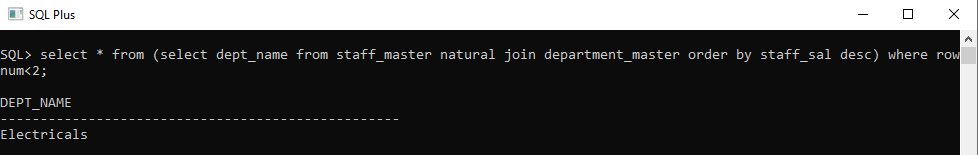
32. Create a view on AccountsMaster table with name vAccs\_Dtls. This view should list all customers whose AccountType is ‘IND’ and their balance amount should not be less than 10000. Using this view any DML operation should not violate the view conditions.



33. Create a view accsvw10 which will not allow DML statement against it.

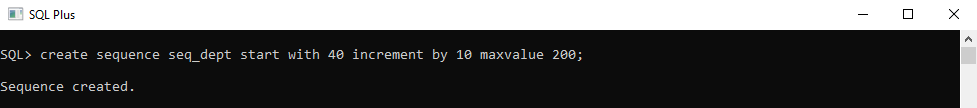


34. Display the department from Staff table which has the highest salary by using Inline View.

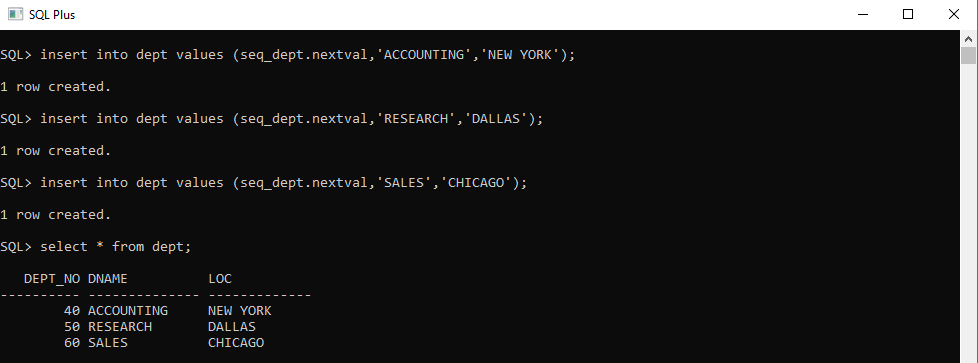


35. List the top two highest earning employees in each department.

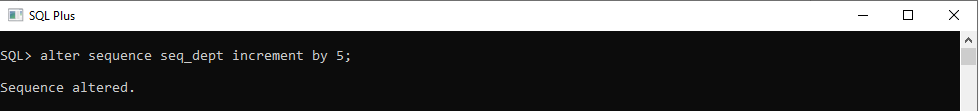
36. Create a Sequence with the name Seq\_Dept on Deptno column of Dept table. It should start from 40 and stop at 200. Increment parameter for the sequence Seq\_Dept should be in step of 10.



37. Insert three sample rows by using the above sequence in Dept table.



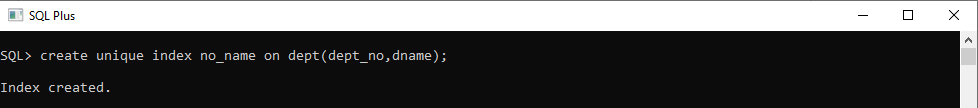
38. Alter the above specified sequence with an increment by 5.



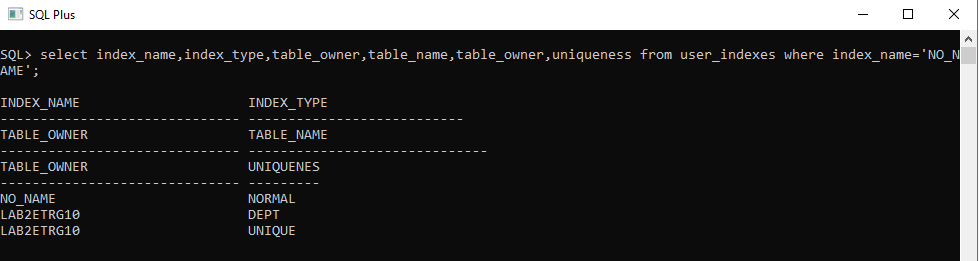
39. Drop the Seq\_Dept sequence.



40. Create a Unique index with the name No\_Name on DeptNo and Dname of Dept table.



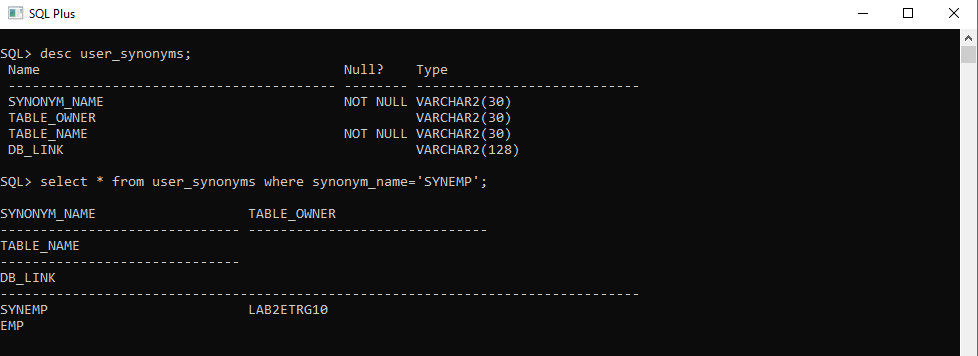
41. Get information on the index No\_Name from the Data Dictionary.



42. Create public synonym synEmp for the EMP table.



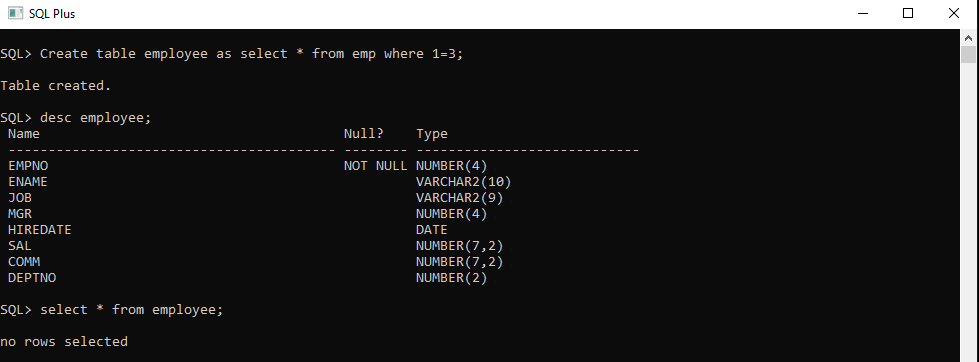
43. Get Information on synonym synEmp from the Data Dictionary.



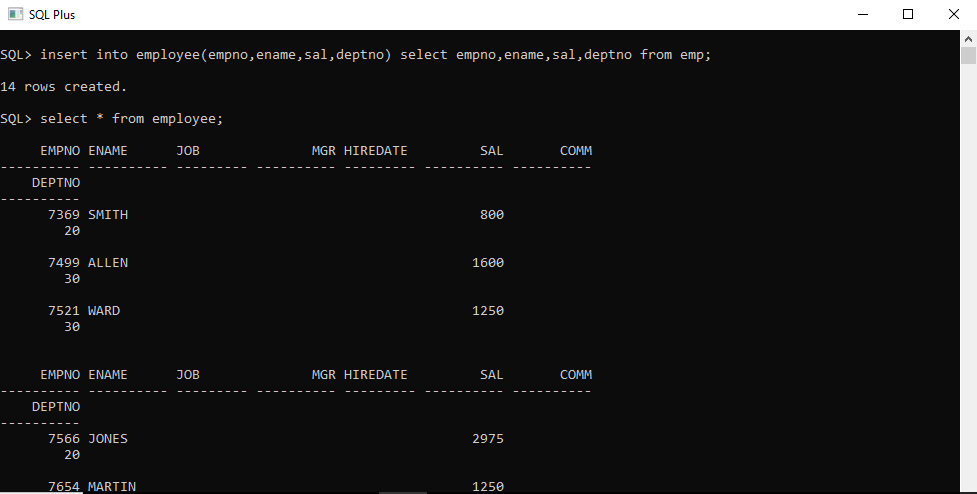
**Lab 5. Data Manipulation Language**

**5.1: Data Manipulation Language**

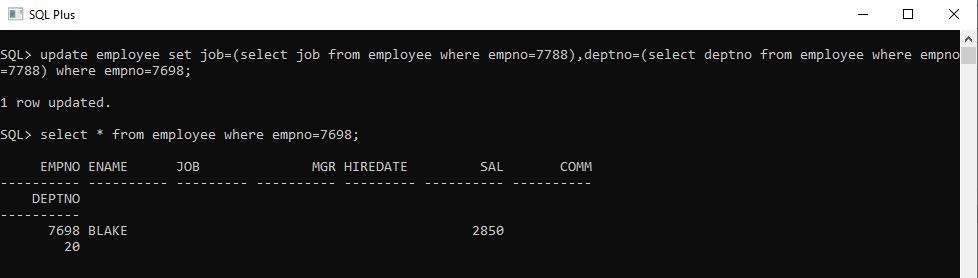
1. Create Employee table with same structure as EMP table.



2. Write a query to populate Employee table using EMP table’s empno, ename, sal, deptno columns.



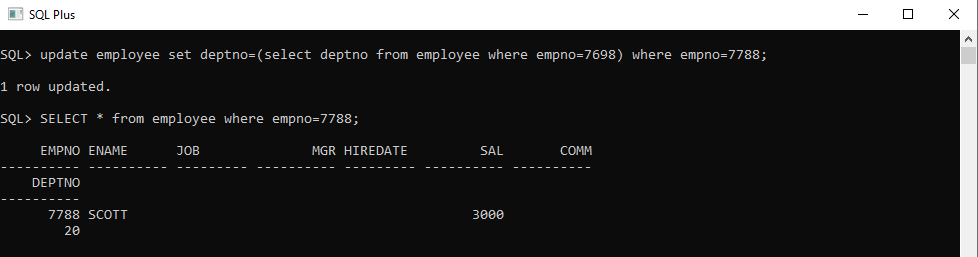
3. Write a query to change the job and deptno of employee whose empno is 7698 to the job and deptno of employee having empno 7788.



4. Delete the details of department whose department name is ‘SALES’.

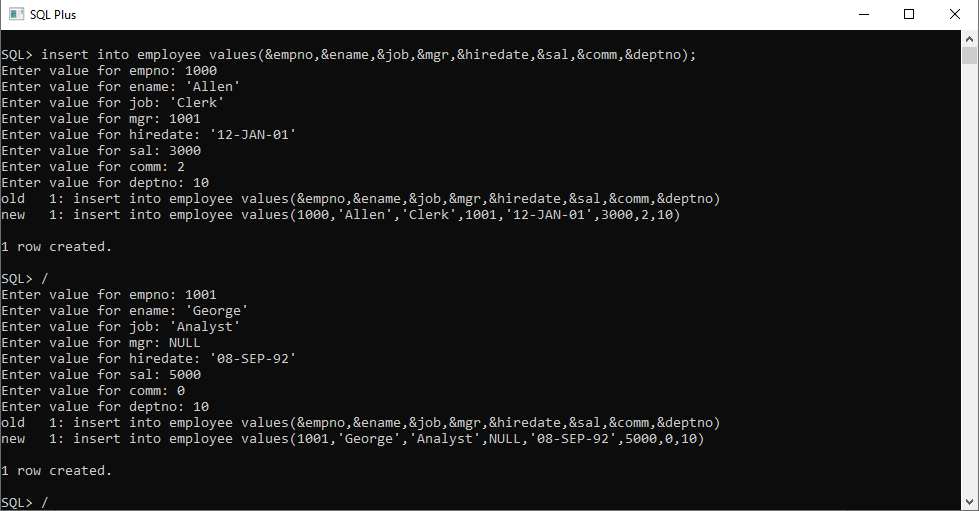


5. Write a query to change the deptno of employee with empno 7788 to that of employee having empno 7698.



6. Insert the following rows to the Employee table through parameter substitution.

* 1000,Allen, Clerk,1001,12-jan-01, 3000, 2,10
* 1001,George, analyst, null, 08 Sep 92, 5000,0, 10
* 1002, Becker, Manager, 1000, 4 Nov 92, 2800,4, 20
* 1003, 'Bill', Clerk, 1002, 4 Nov 92,3000, 0, 20



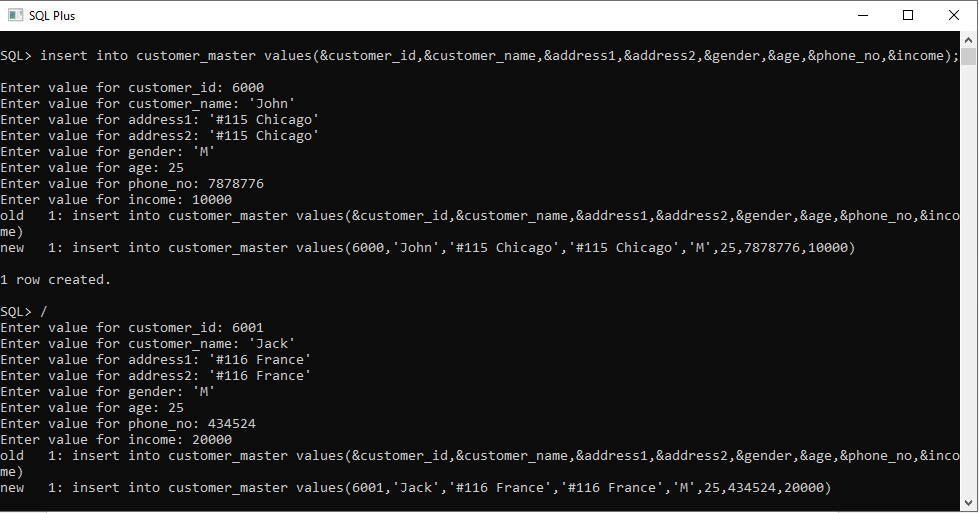
**Lab 6. Transaction Control Language Statements**

**6.1: Transaction Control Language Statements**

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

* 6000, John, #115 Chicago, #115 Chicago, M, 25, 7878776, 10000
* 6001, Jack, #116 France, #116 France, M, 25, 434524, 20000
* 6002, James, #114 New York, #114 New York, M, 45, 431525, 15000.50

Use parameter substitution.

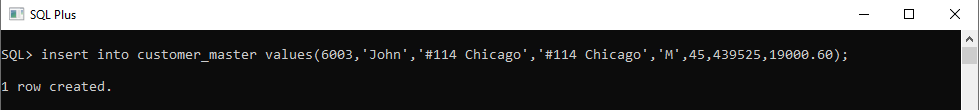


2. Create a Savepoint named ‘SP1’ after third record in the Customer table .



3. Insert the below row in the Customer table.

* 6003, John, #114 Chicago, #114 Chicago, M, 45, 439525, 19000.60



4. Execute rollback statement in such a way that whatever manipulations done before Savepoint sp1 are permanently implemented, and the ones after Savepoint SP1 are not stored as a part of the Customer table.

