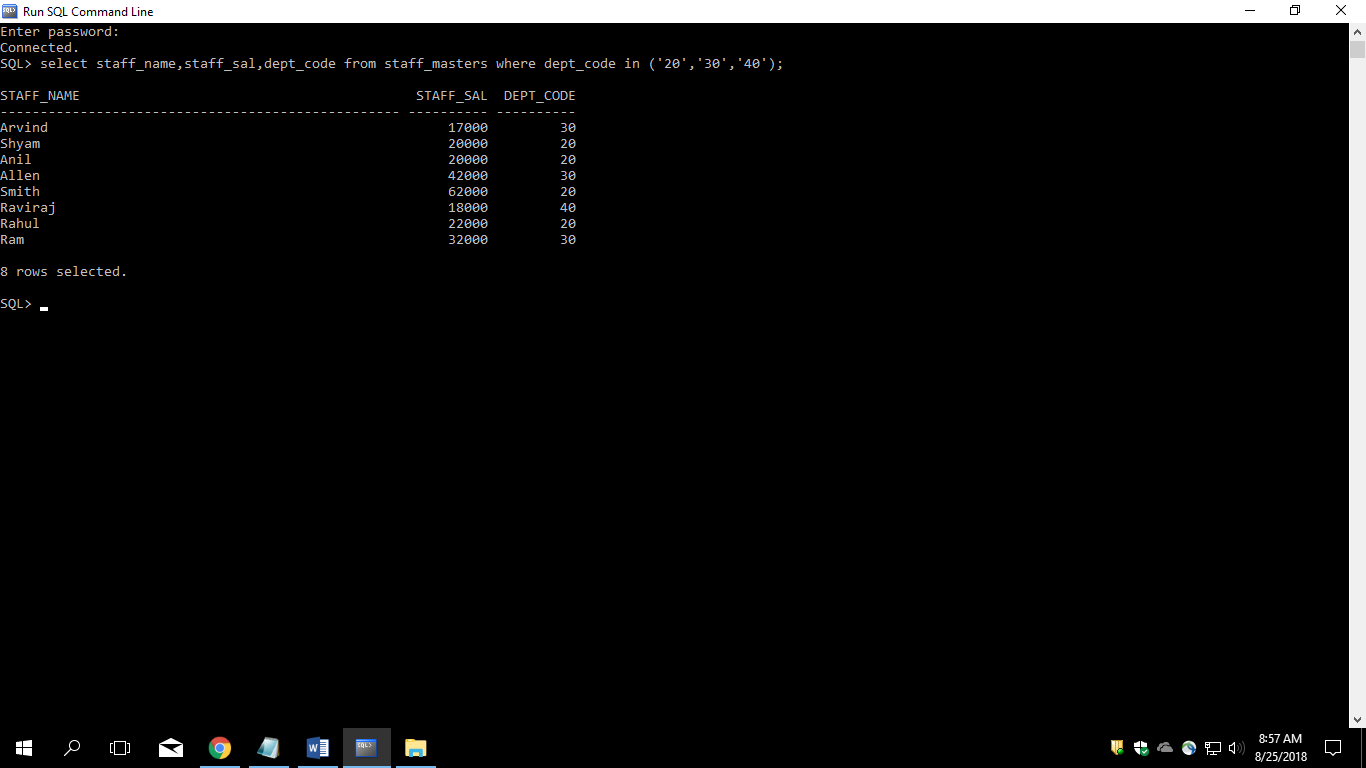
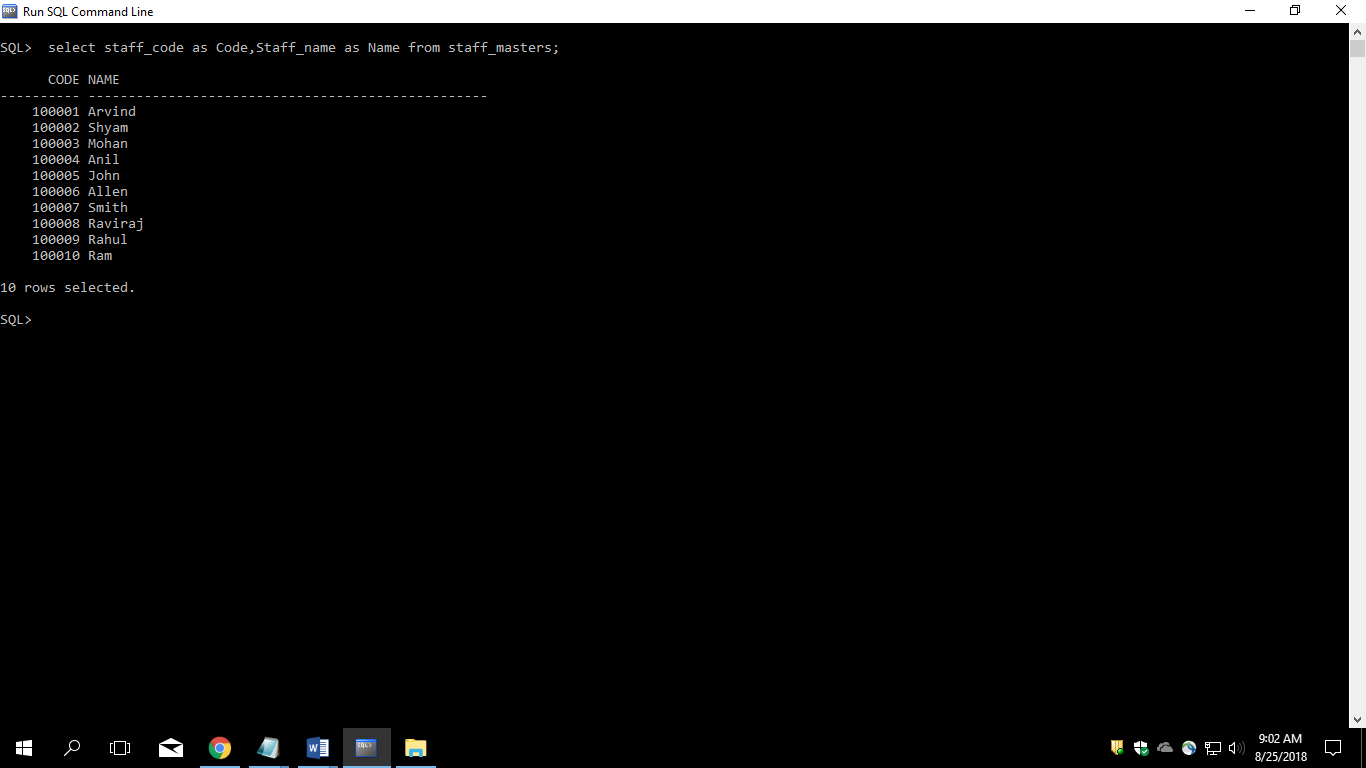
Lab1

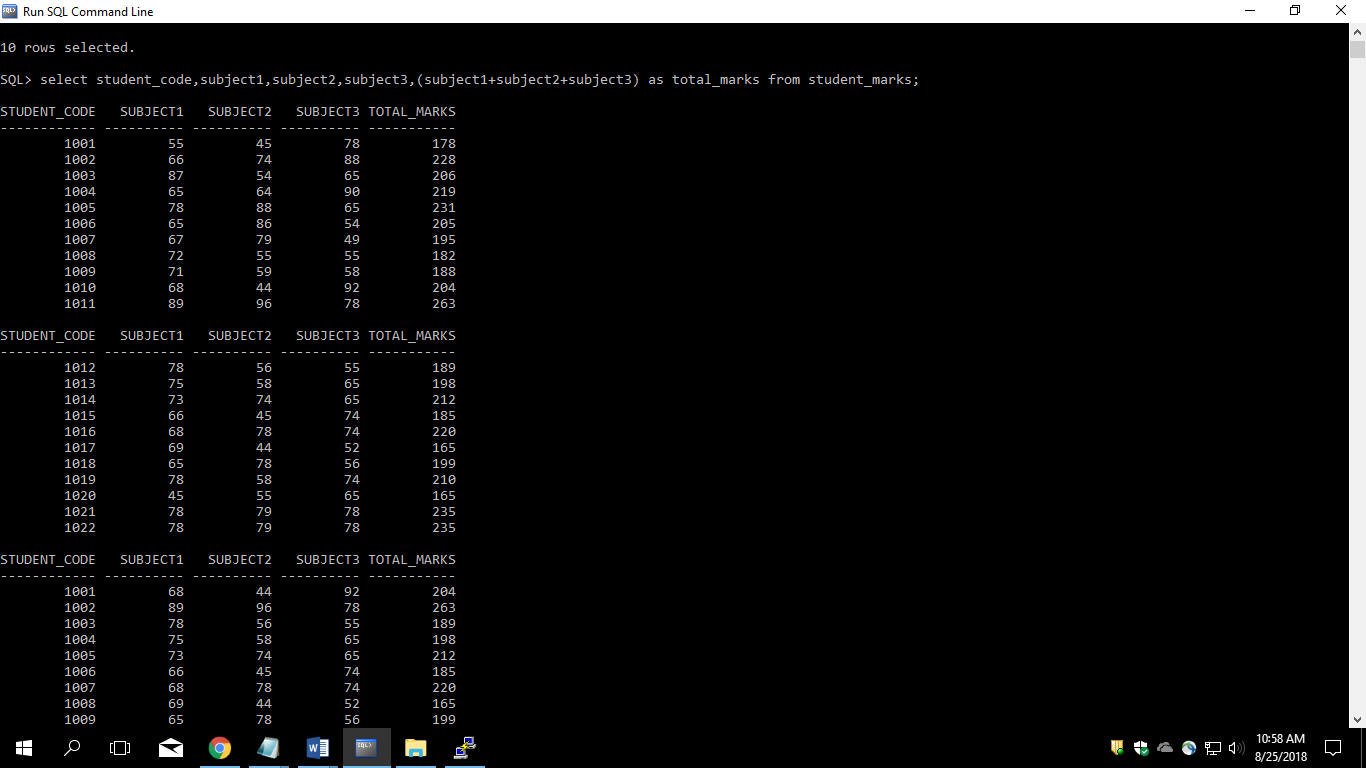
. 1. Retrieve the details (Name, Salary and dept code) of the employees who are working in department 20 , 30 and 40.

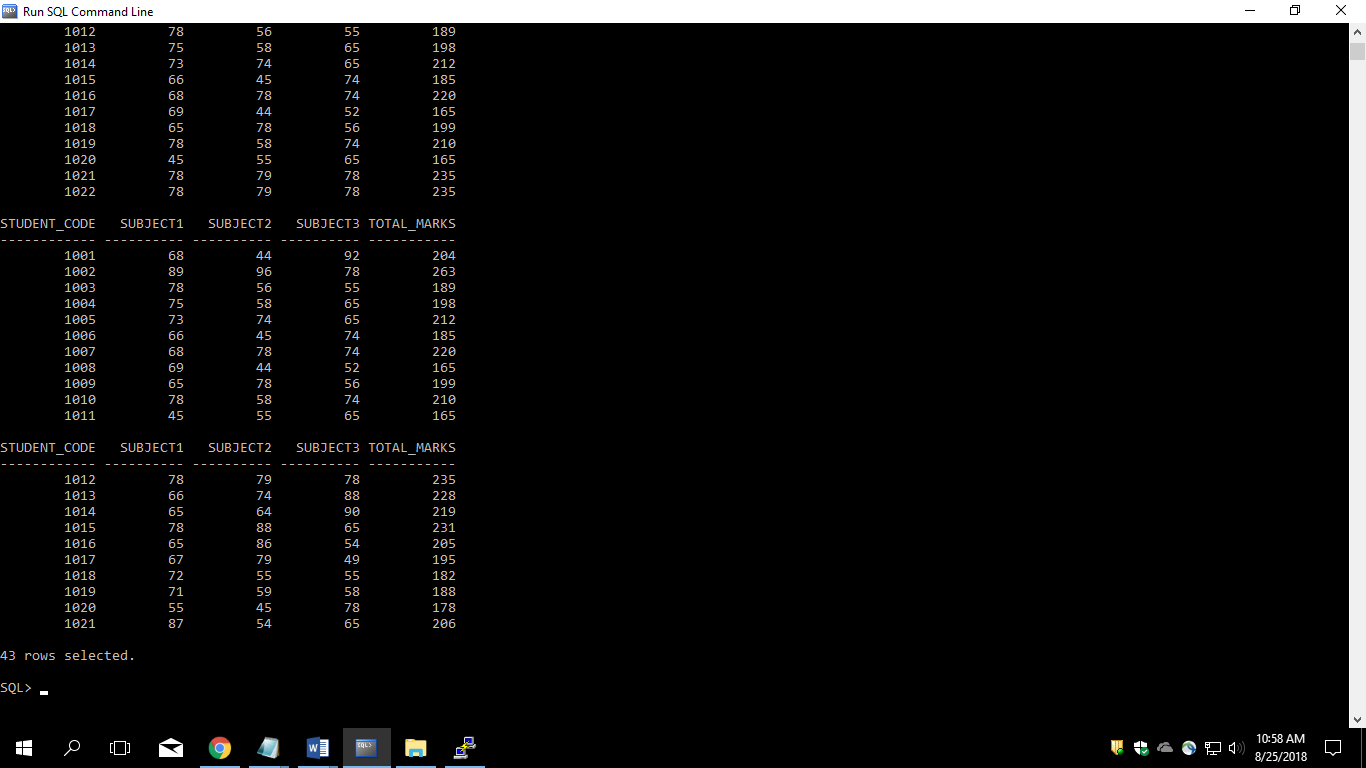


2. List the details of the employees with user defined Column headers.

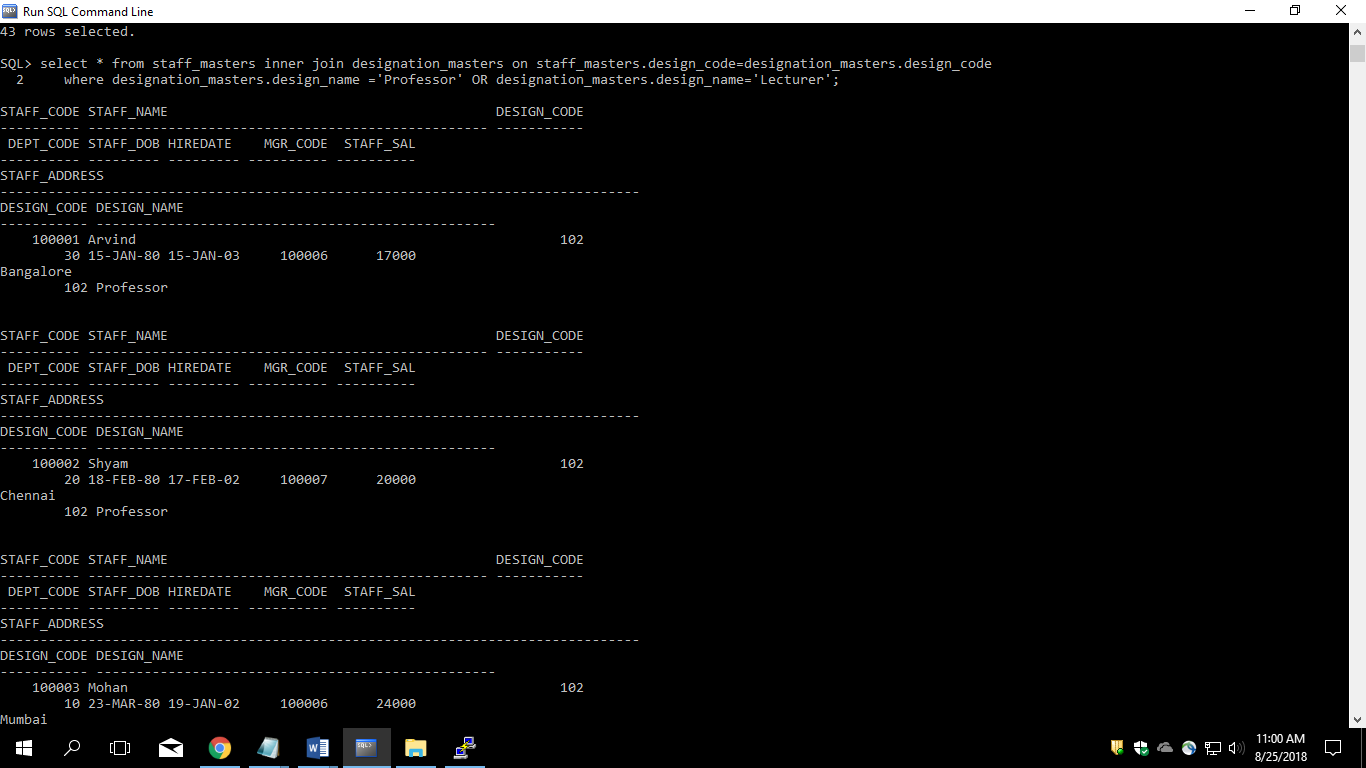


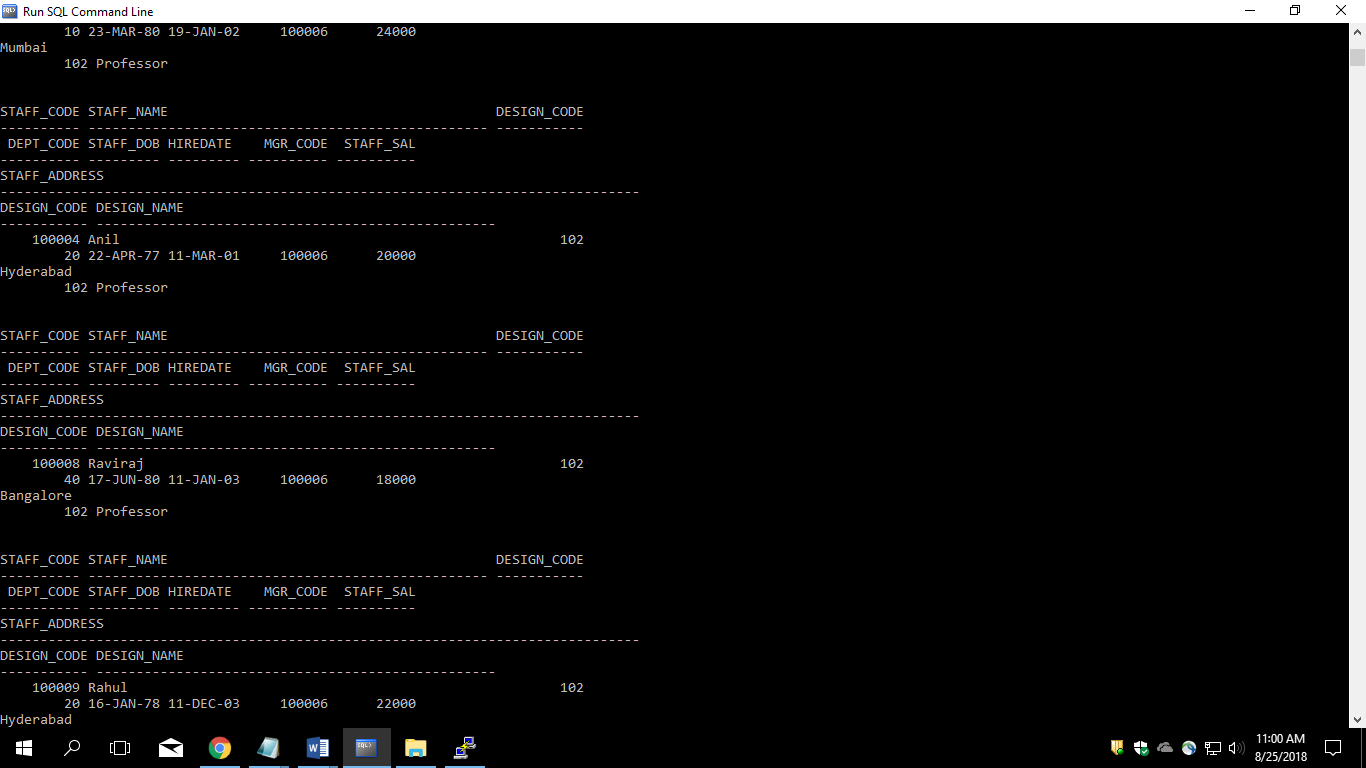
3. Display the code, subjects and total marks for every student. Total Marks will be calculated as Subject1+Subject2+Subject3. (Refer Student\_Marks table)



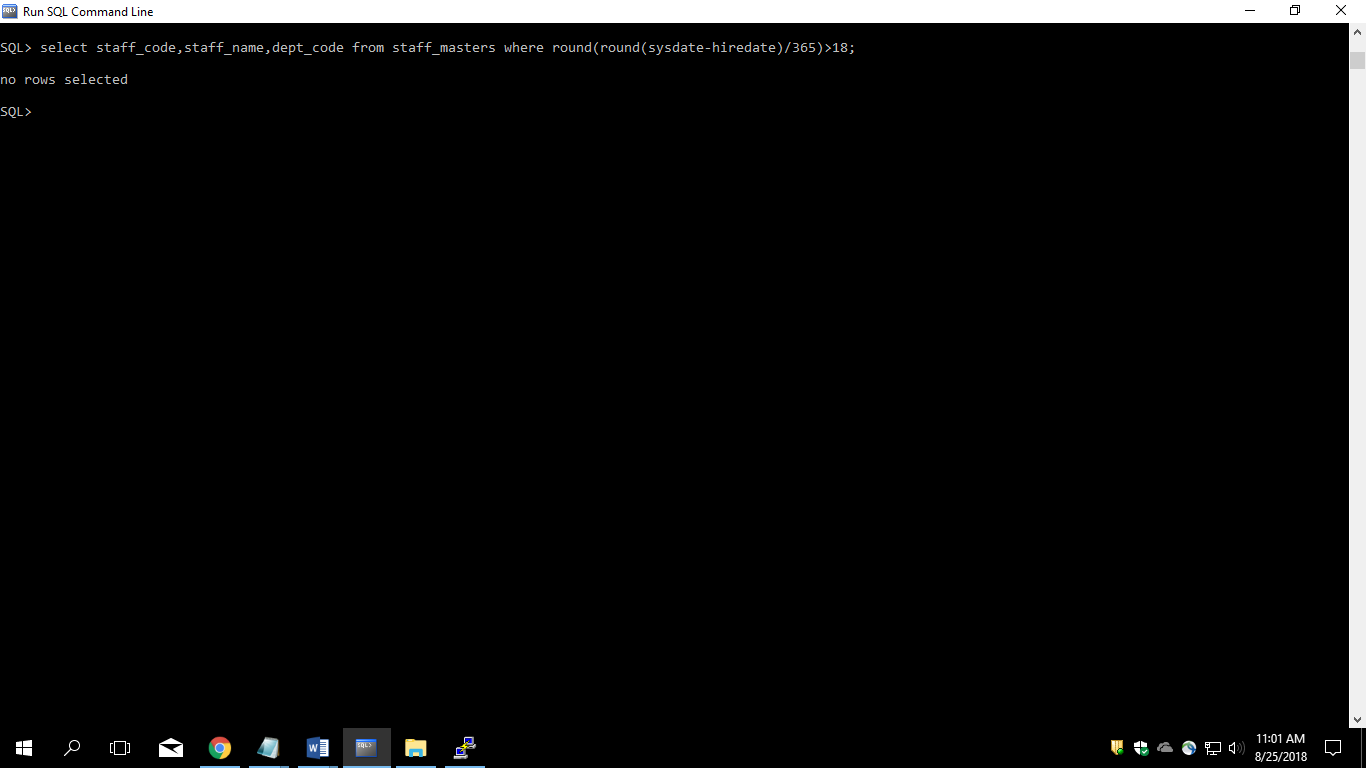


4. List the details of the staff whose designations are either PROFESSOR or LECTURER.

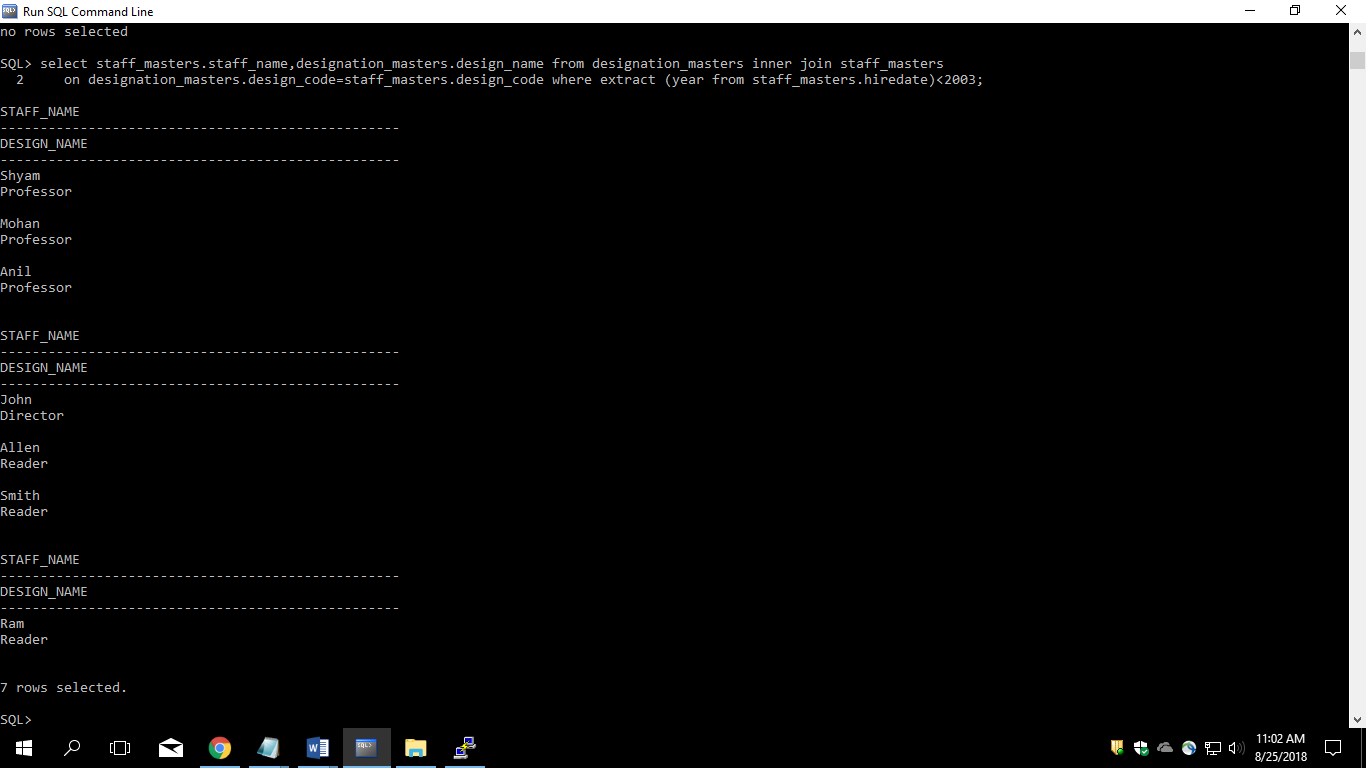




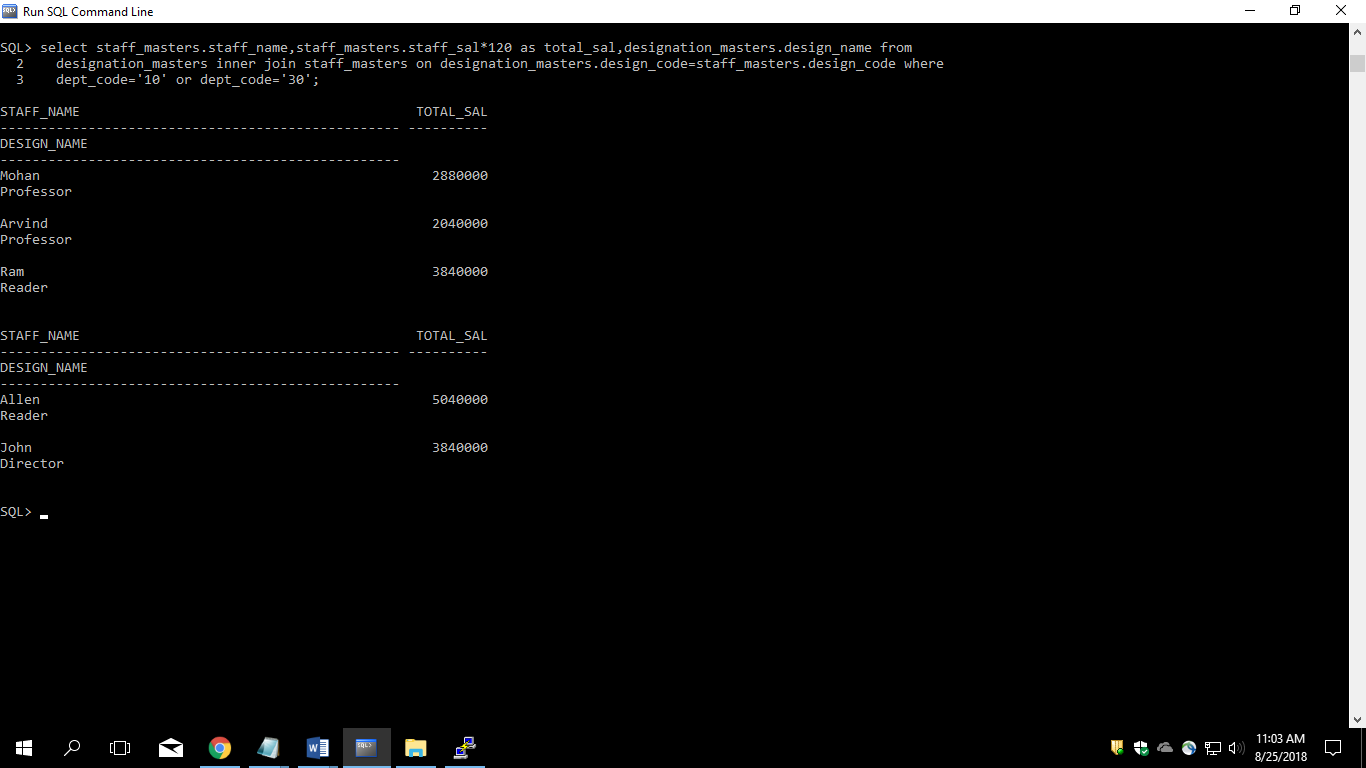
5. List the code, name, and department number of the employees who have experience of more than 18 years.



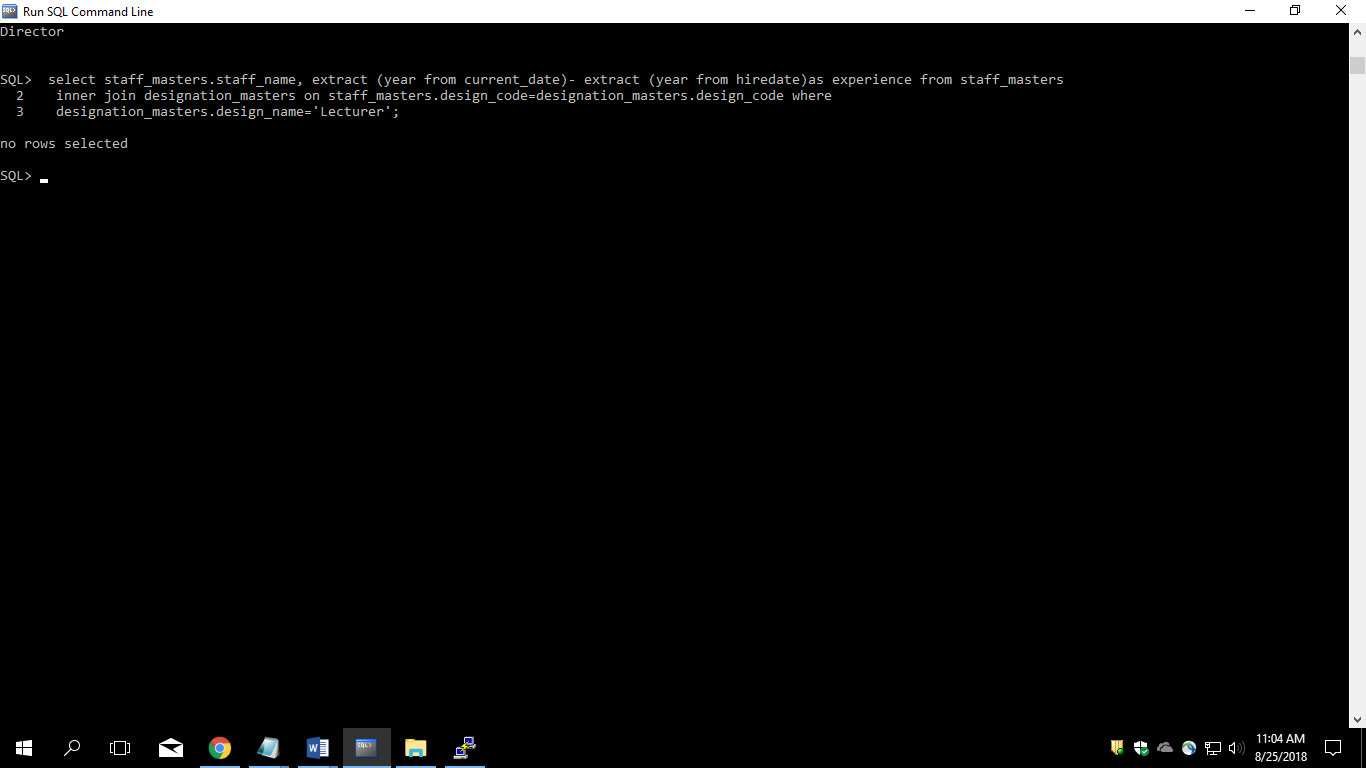
6. List the name and Designations of the staff who have joined before Jan 2003.



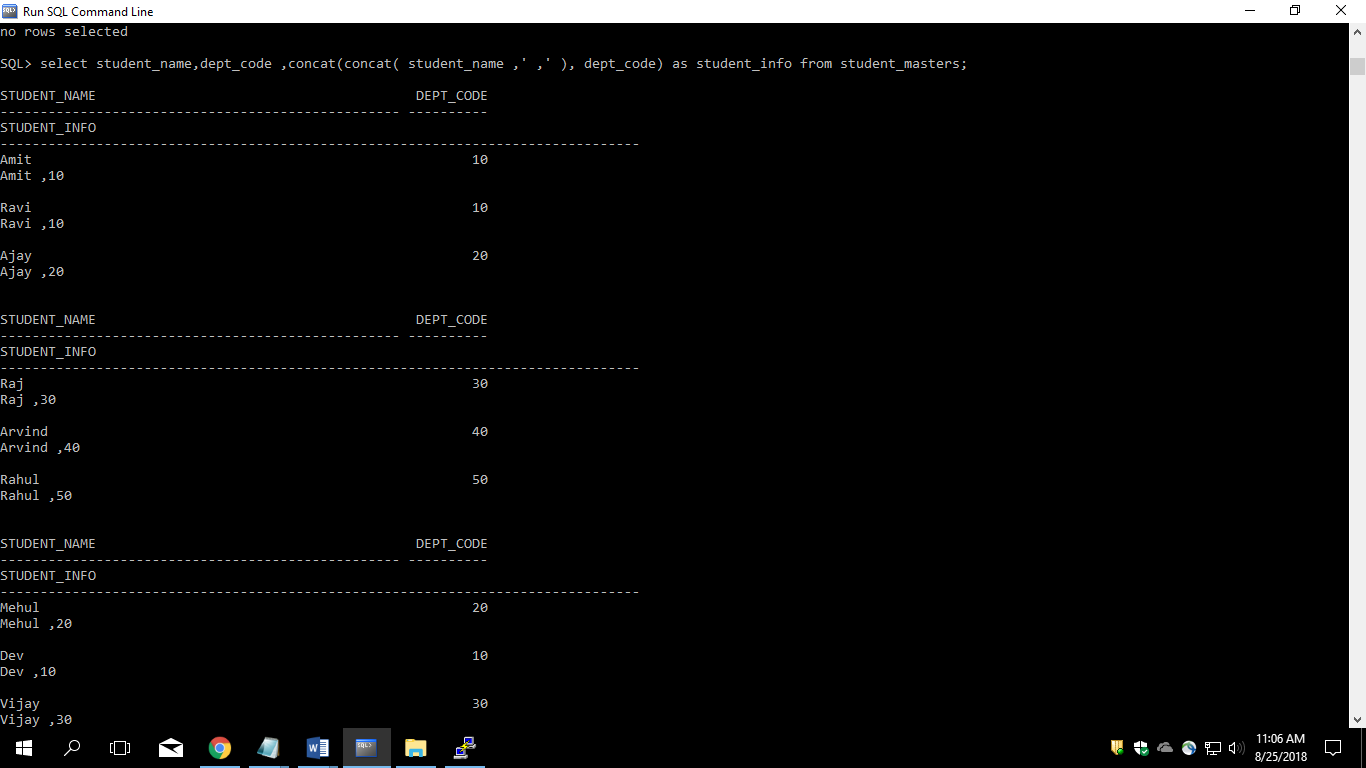
7. List the name, designation, and income for 10 years of the employees who are working in departments 10 and 30.

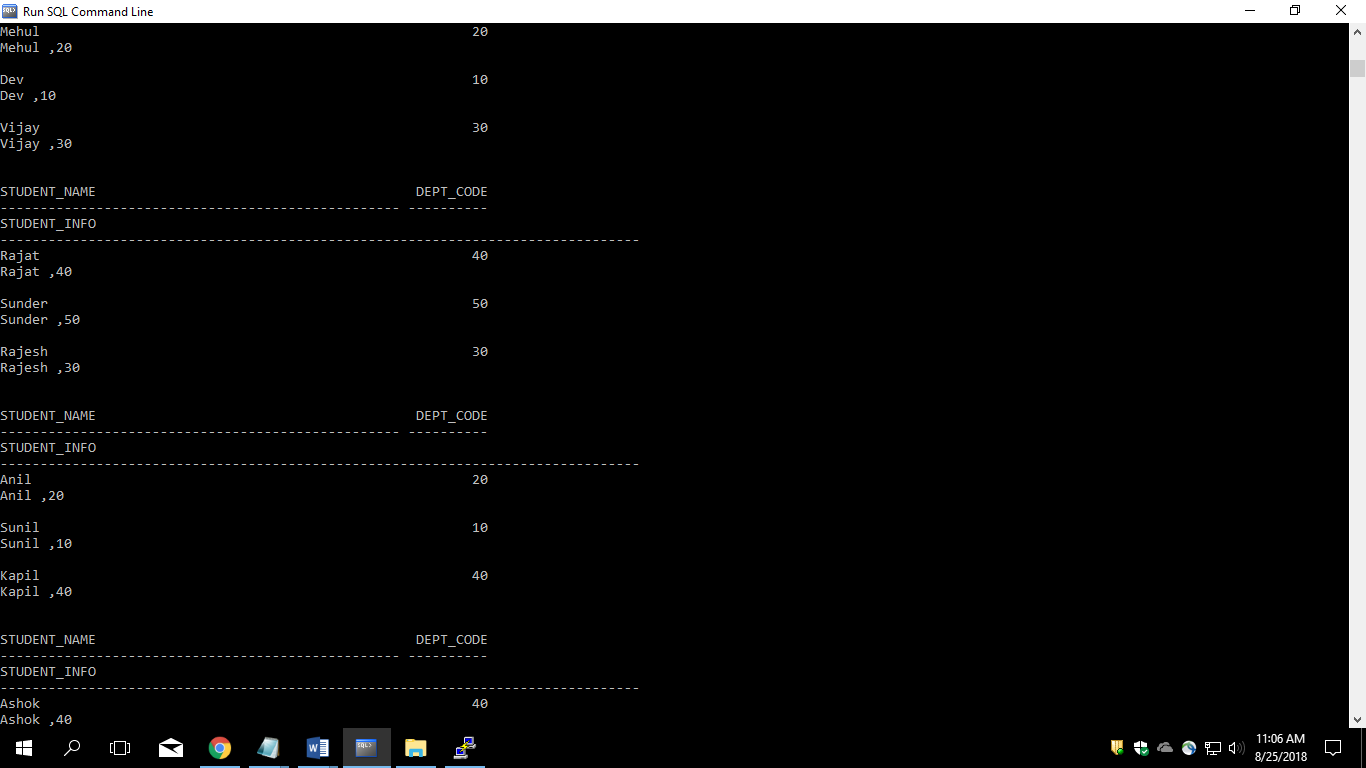


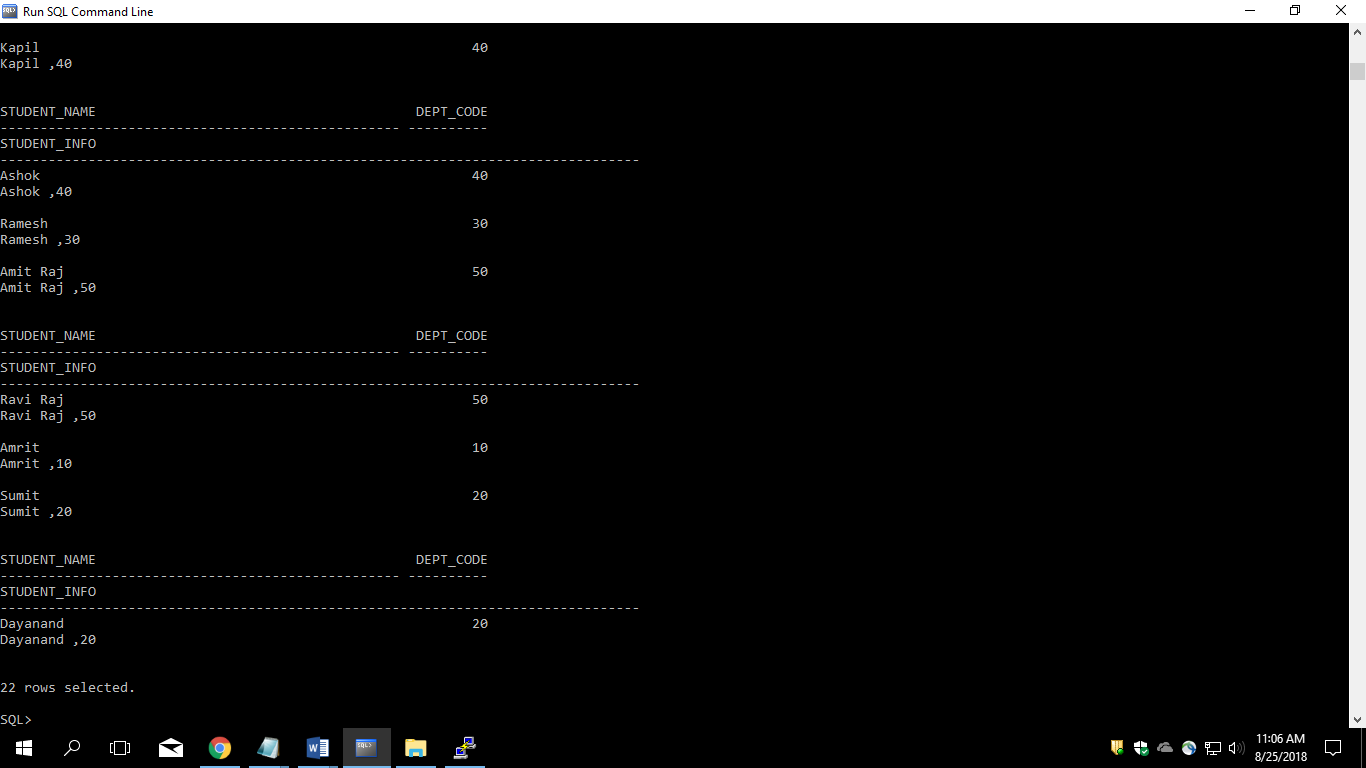
8.List the name and experience (in years) of employees who are working as LECTURER.



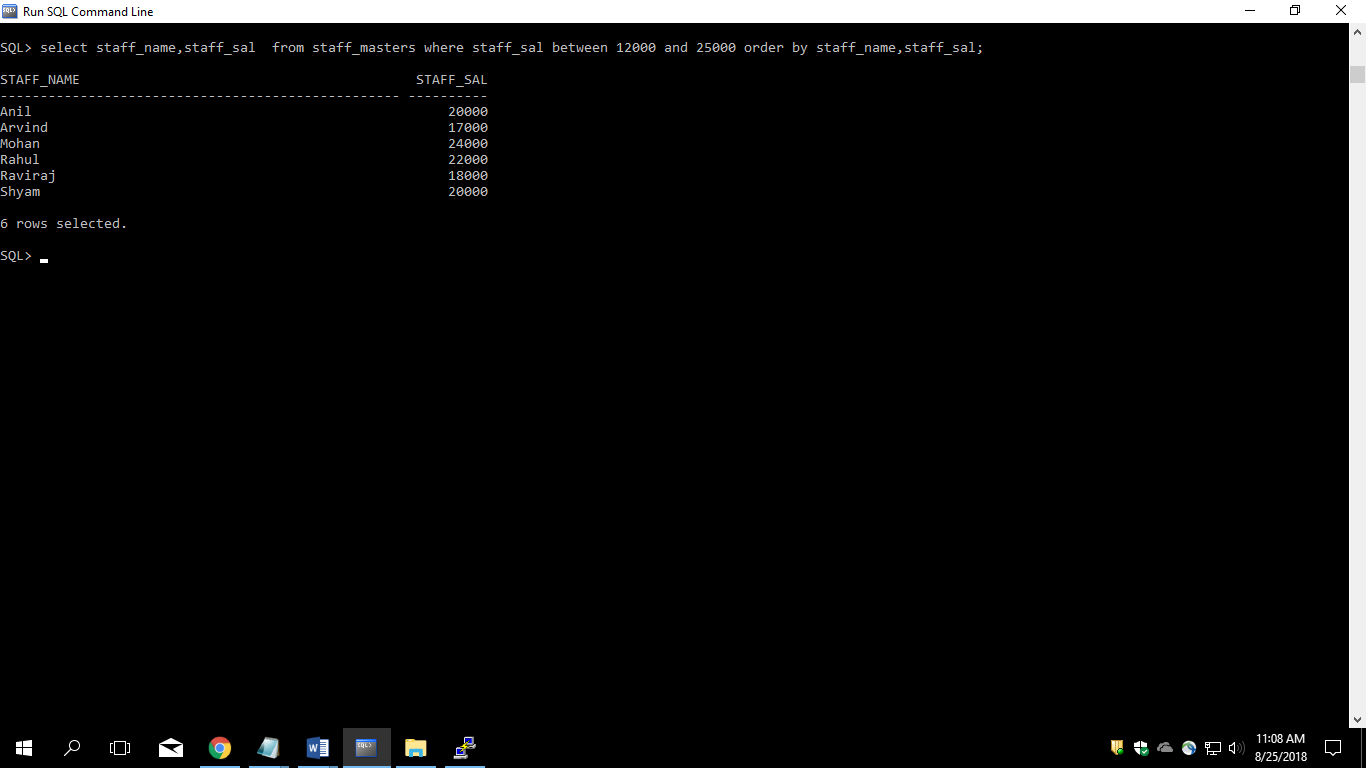
9. Display name concatenated with dept code separated by comma and space. Name the column as ‘Student Info’



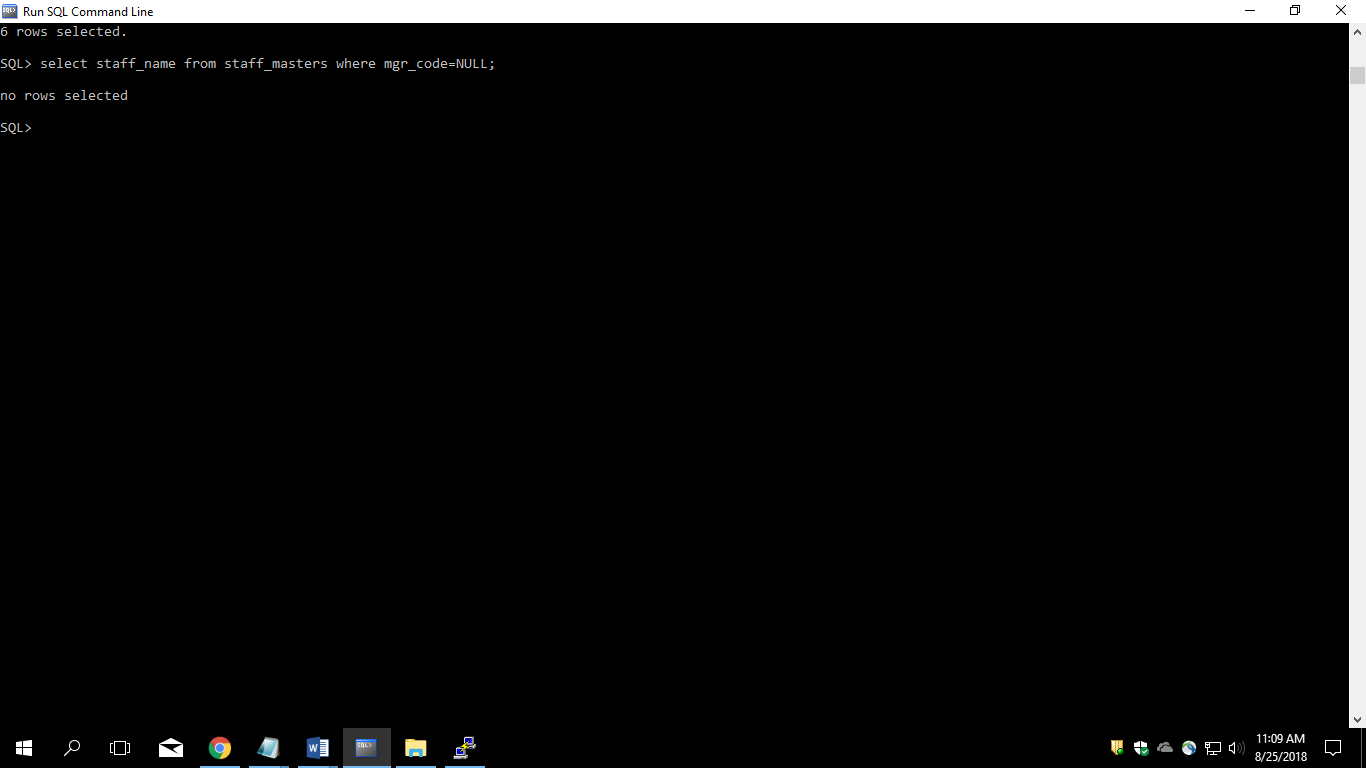
.



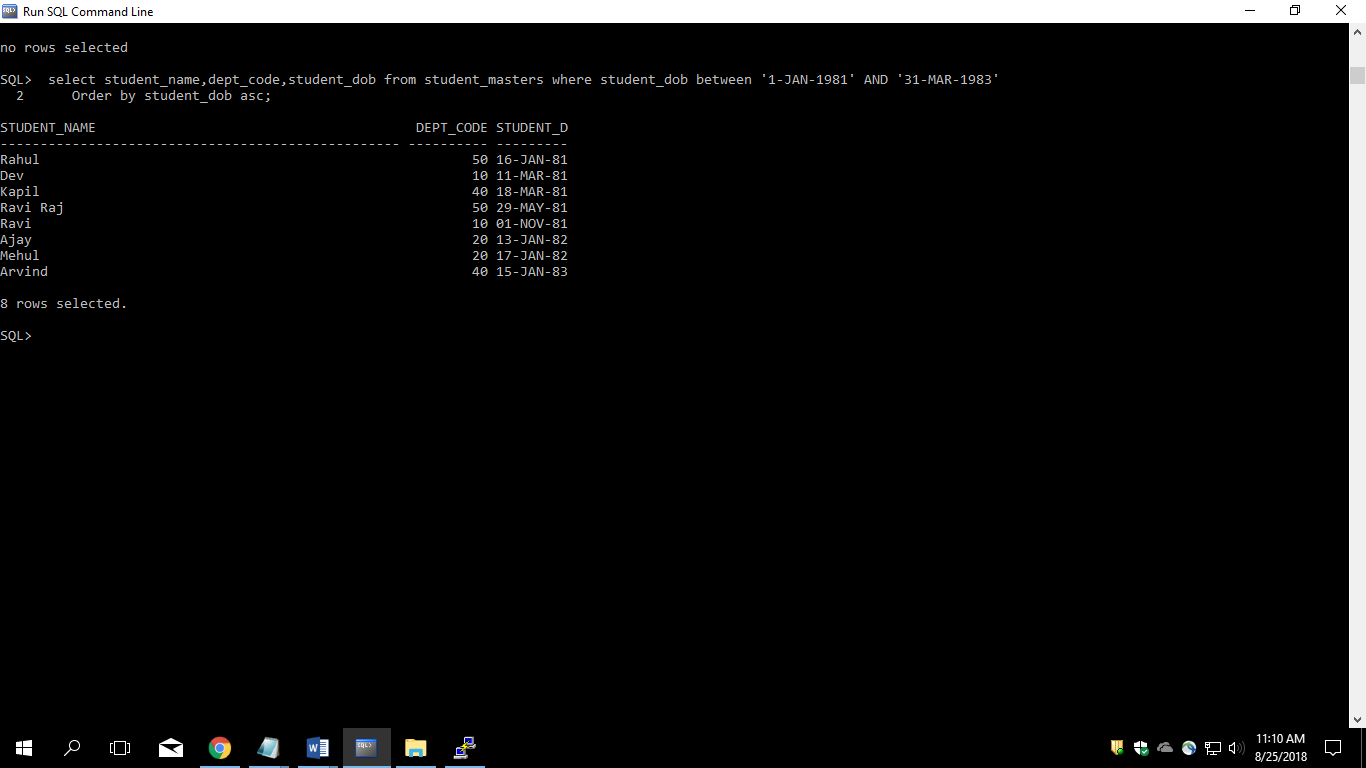
10. List the Name and Salary of the staff who are earning between 12000 and 25000. Sort them based on their salaries and name.



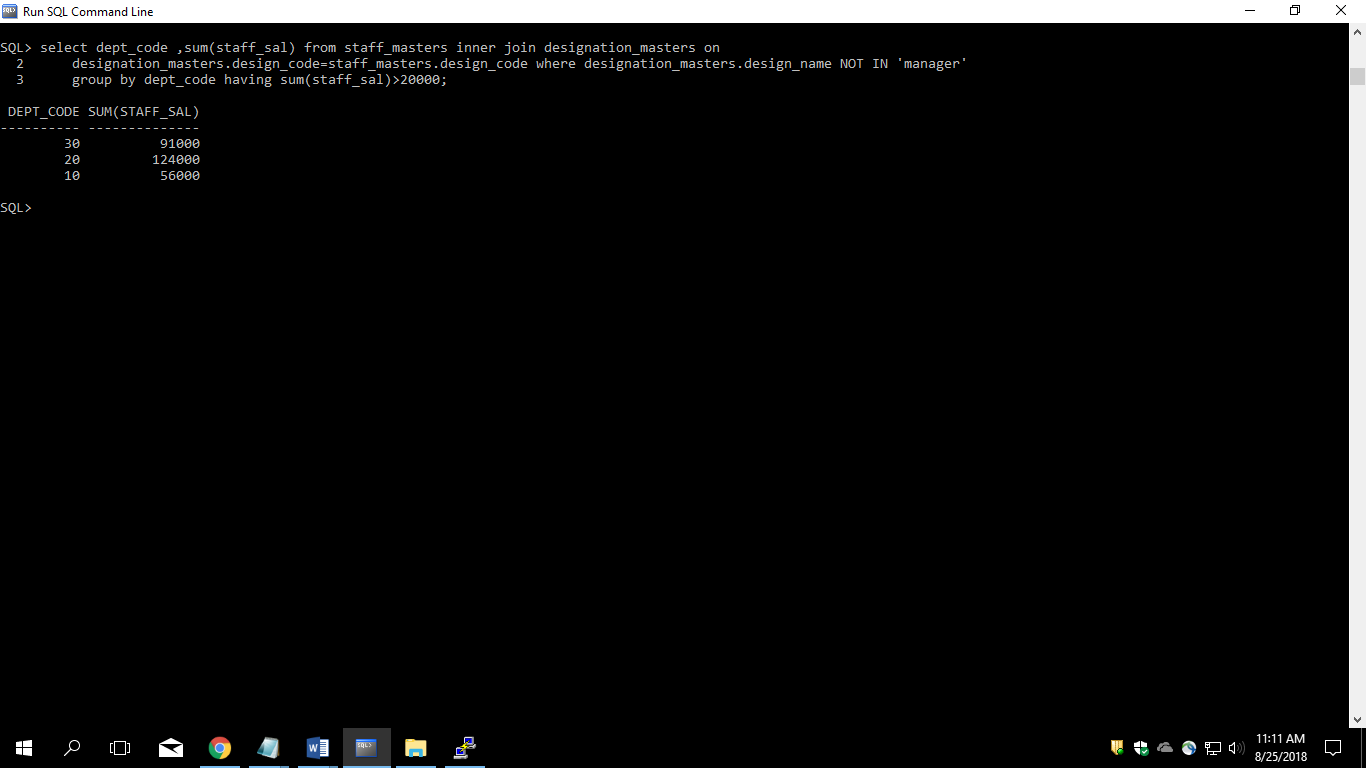
11. Display employees who do not have manager.



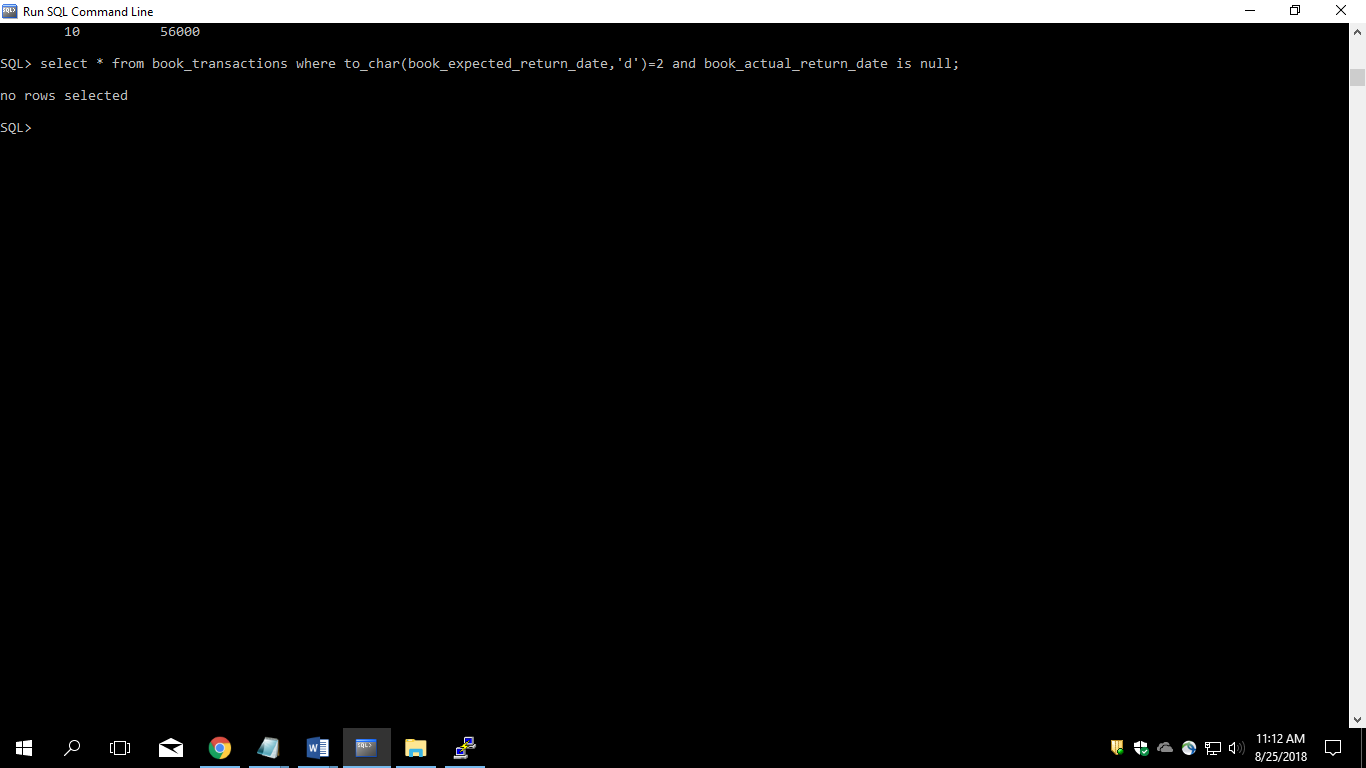
12. Write a query which will display name, department code and date of birth of all students who were born between January 1, 1981 and March 31, 1983. Sort it based on date of birth (ascending).



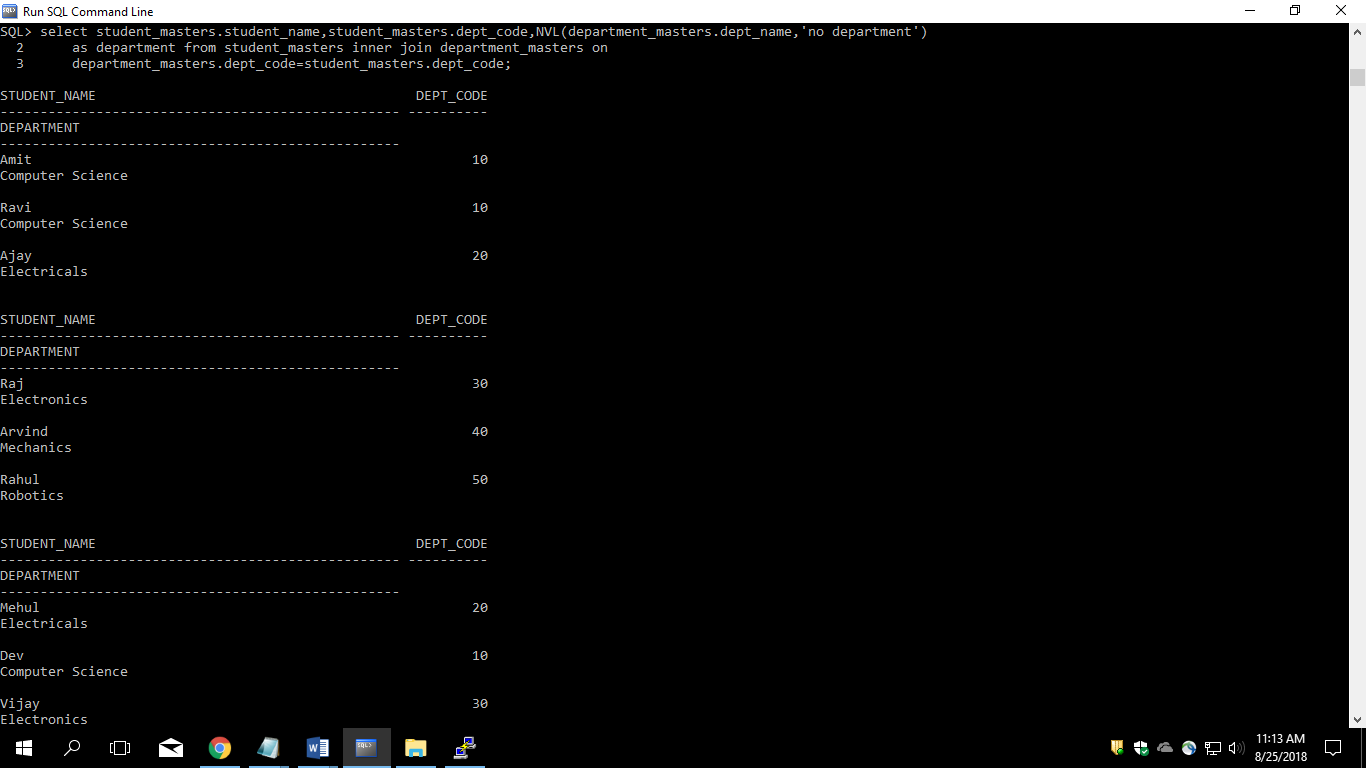
13. Get the Department number, and sum of Salary of all non managers where the sum is greater than 20000.



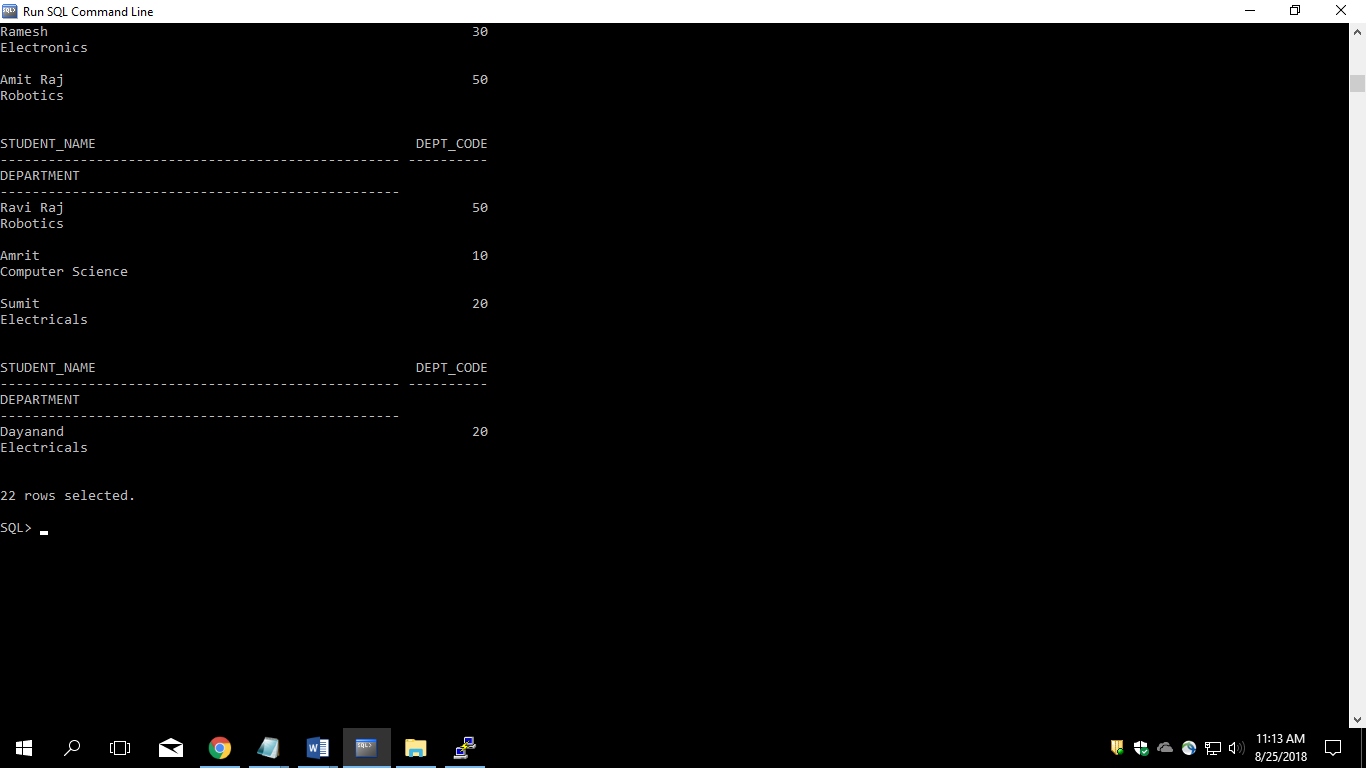
14. Display the details of books that have not been returned and expected return date was last Monday.



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)





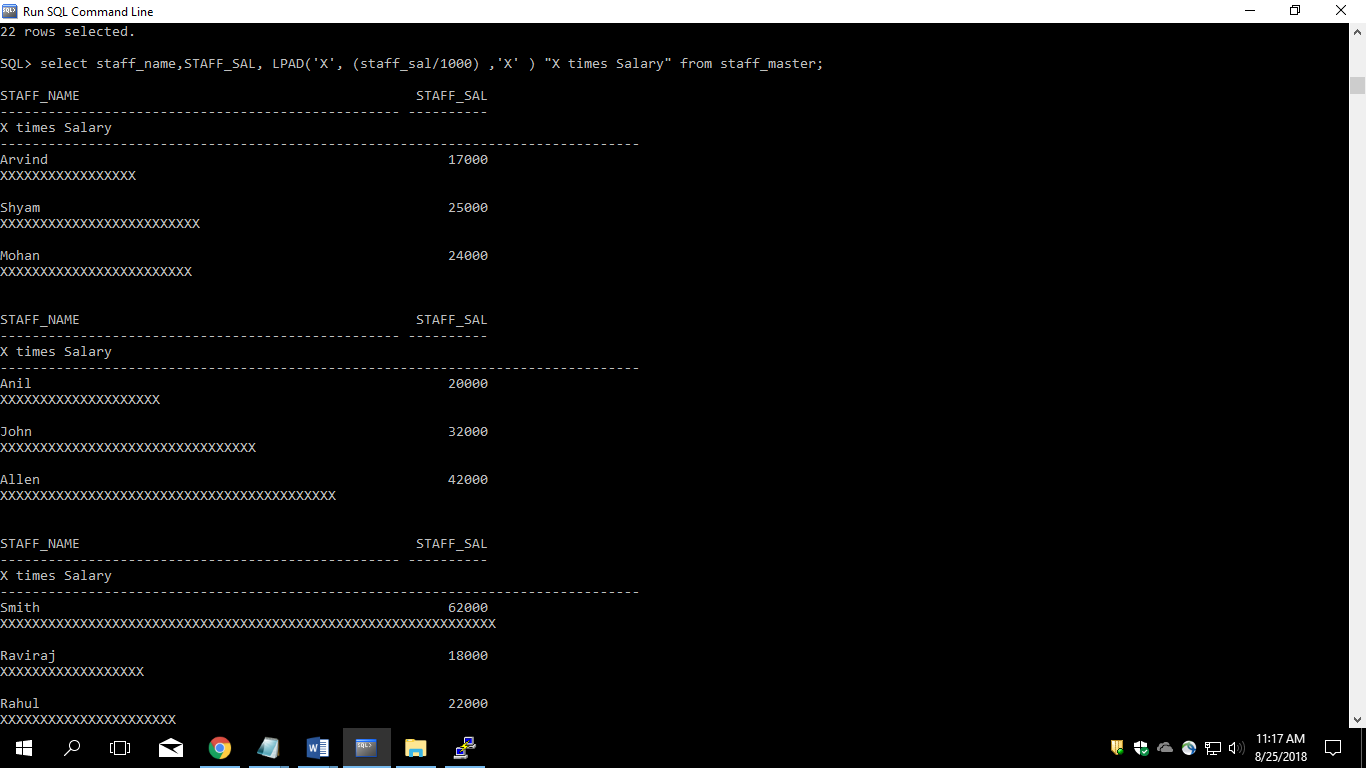


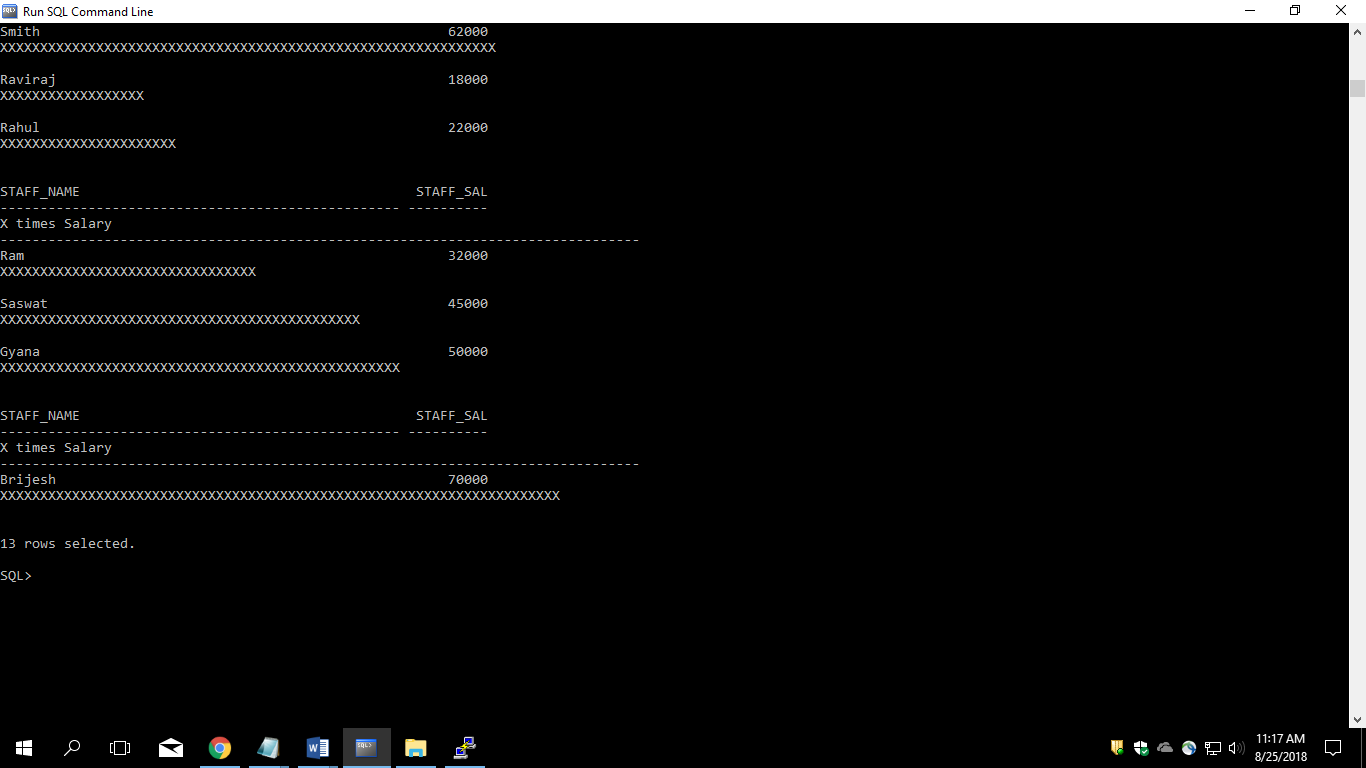
16. Display the name and salary of the staff. Salary should be represented as X. Each X represents a 1000 in salary.

Sample Output

JOHN 10000 XXXXXXXXXX

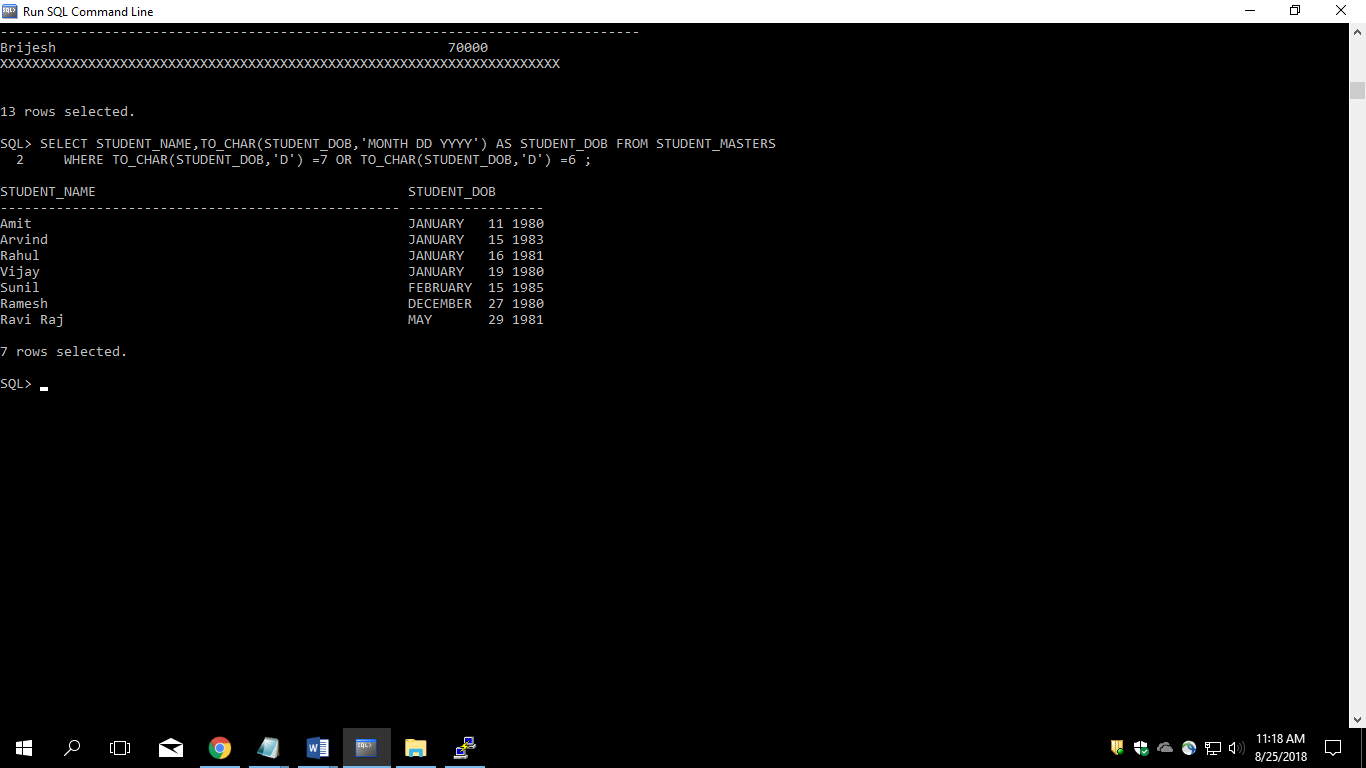
ALLEN 12000 XXXXXXXXXXXX



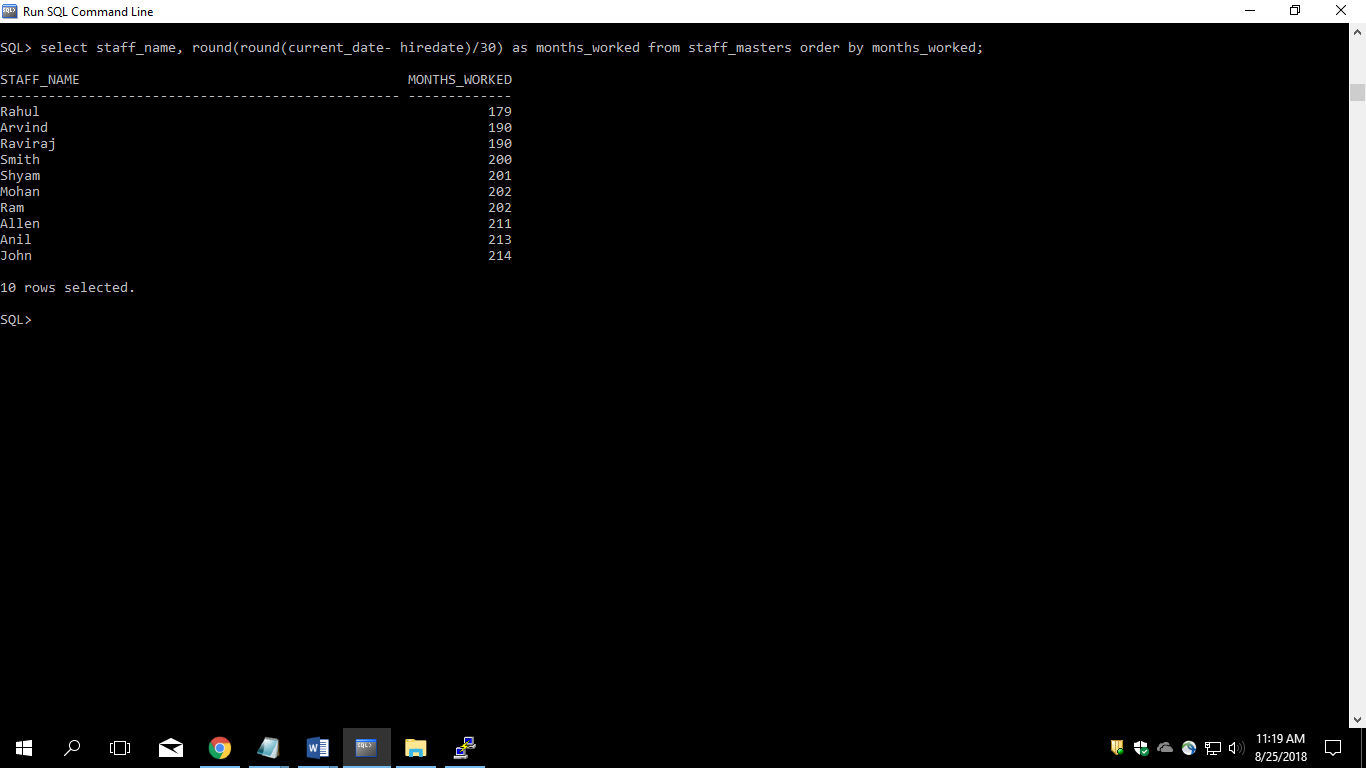


**Lab2**

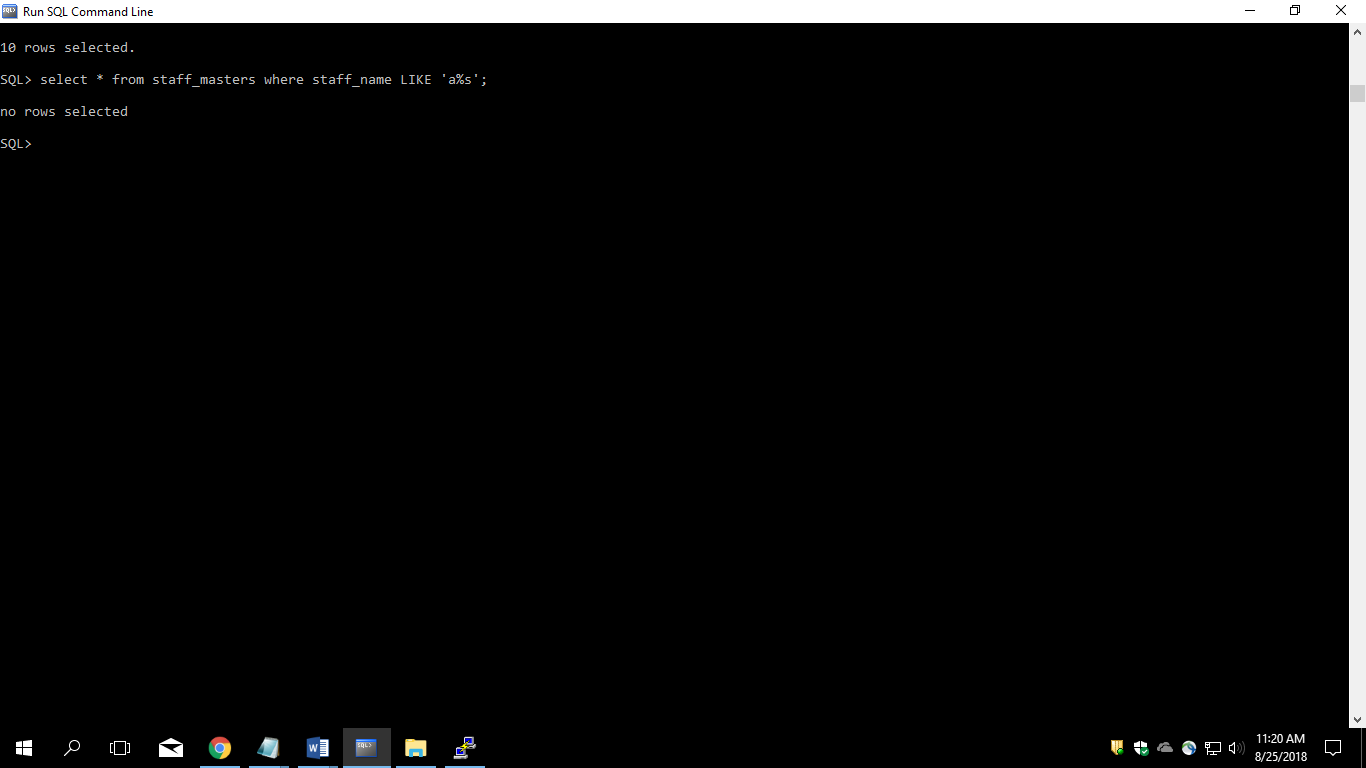
1. Display name and date of birth of students where date of birth must be displayed in the format similar to “January, 12 1981” for those who were born on Saturday or Sunday.



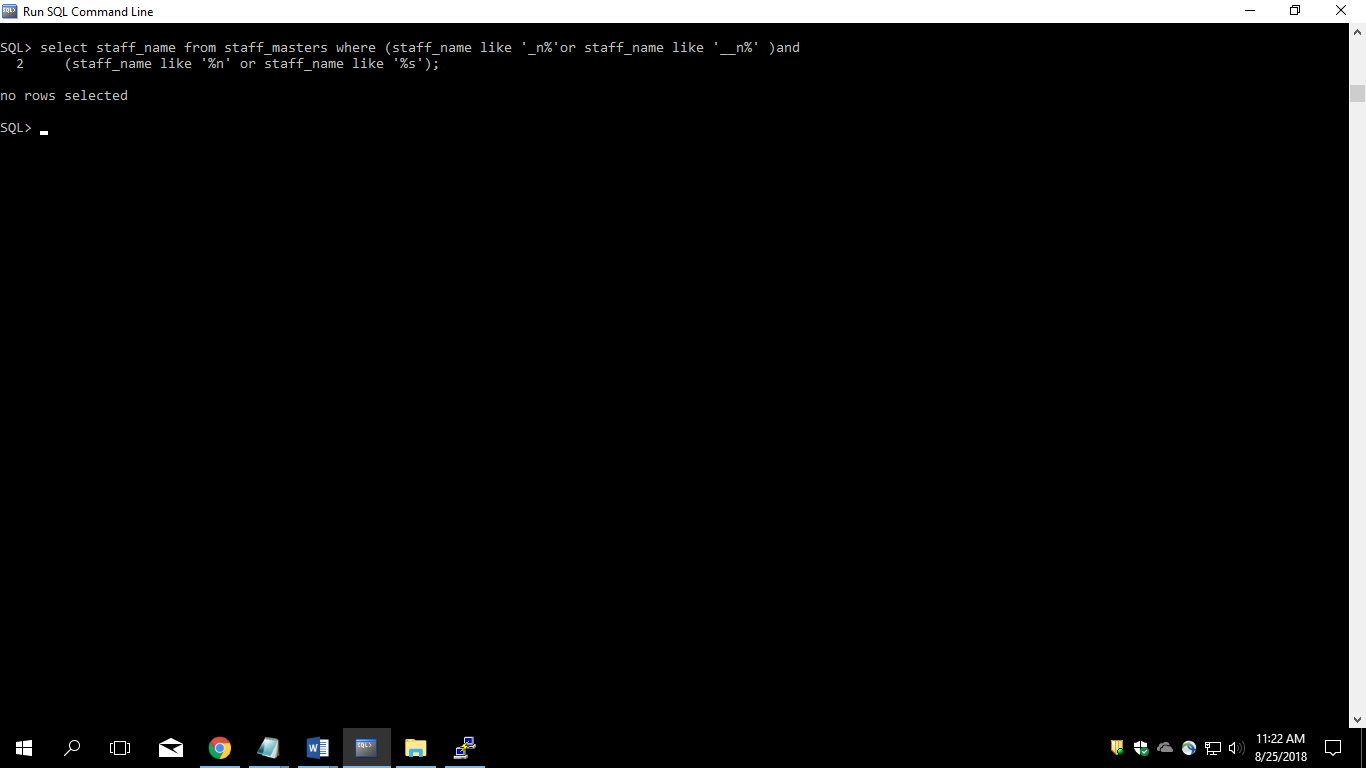
2. Display each staff name and number of months they worked for the organization. Label the column as ‘Months Worked’. Order your result by number of months employed. Round the number of months to closest whole number.



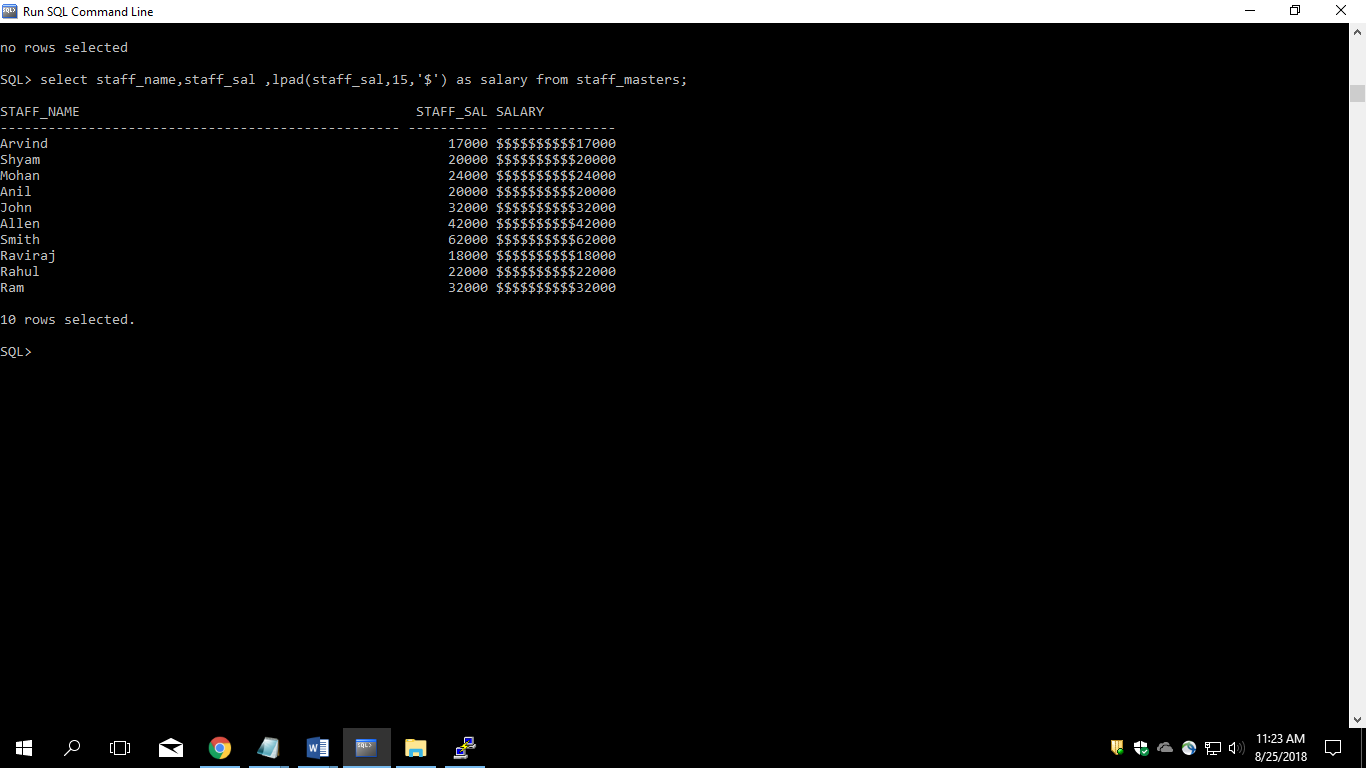
3. List the details of the employees, whose names start with ‘A’ and end with ‘S’.



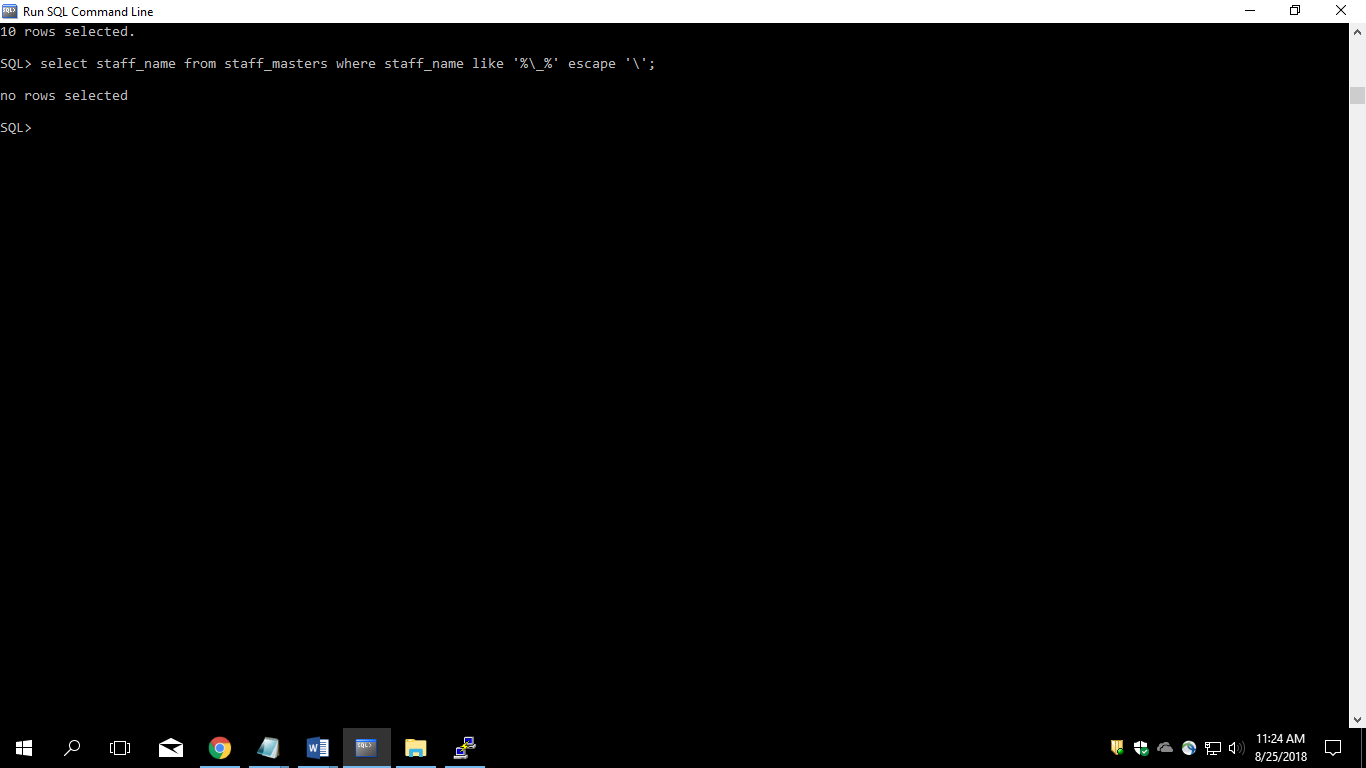
4. List the name and job of the employees whose names should contain N as the second or third character, and ending with either ‘N’ or ‘S’.



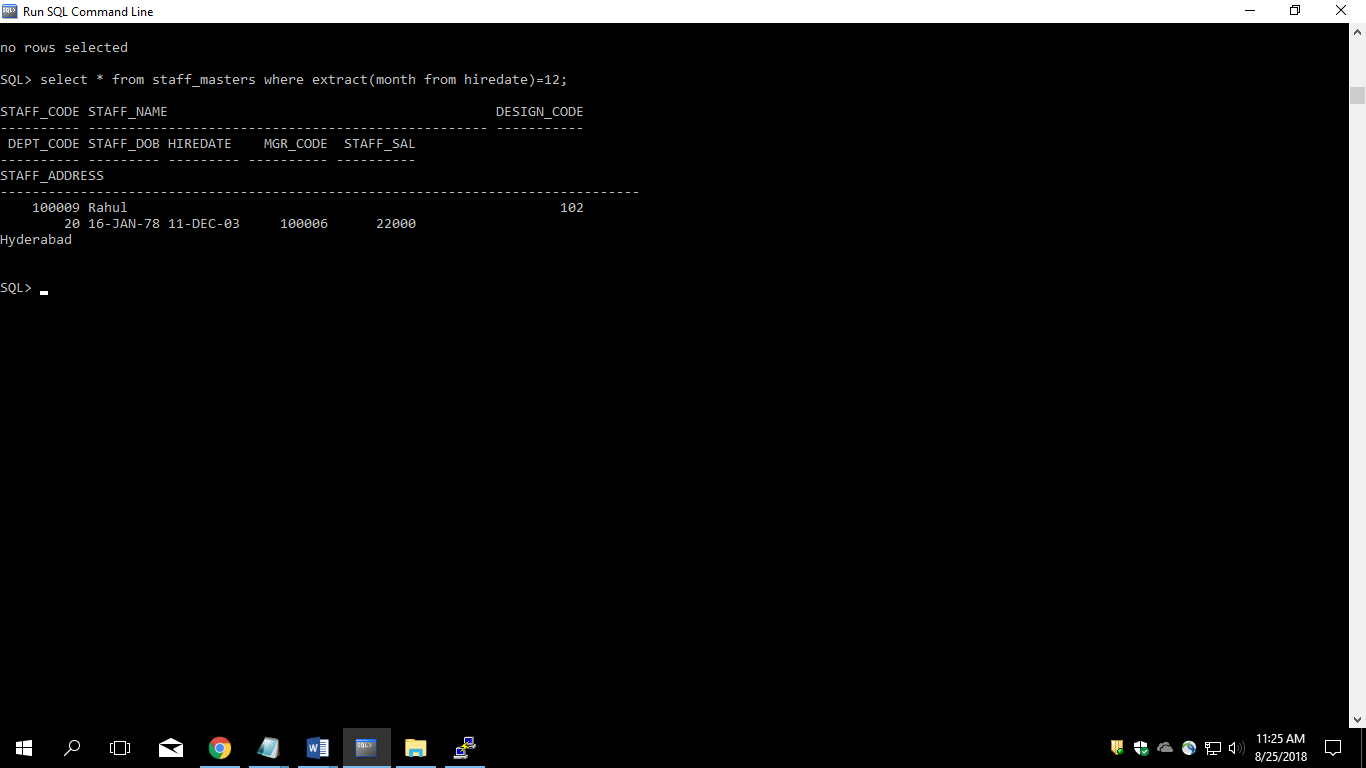
5. Create a query which will display Staff Name, Salary of each staff. Format the salary to be 15 character long and left padded with ‘$’.



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

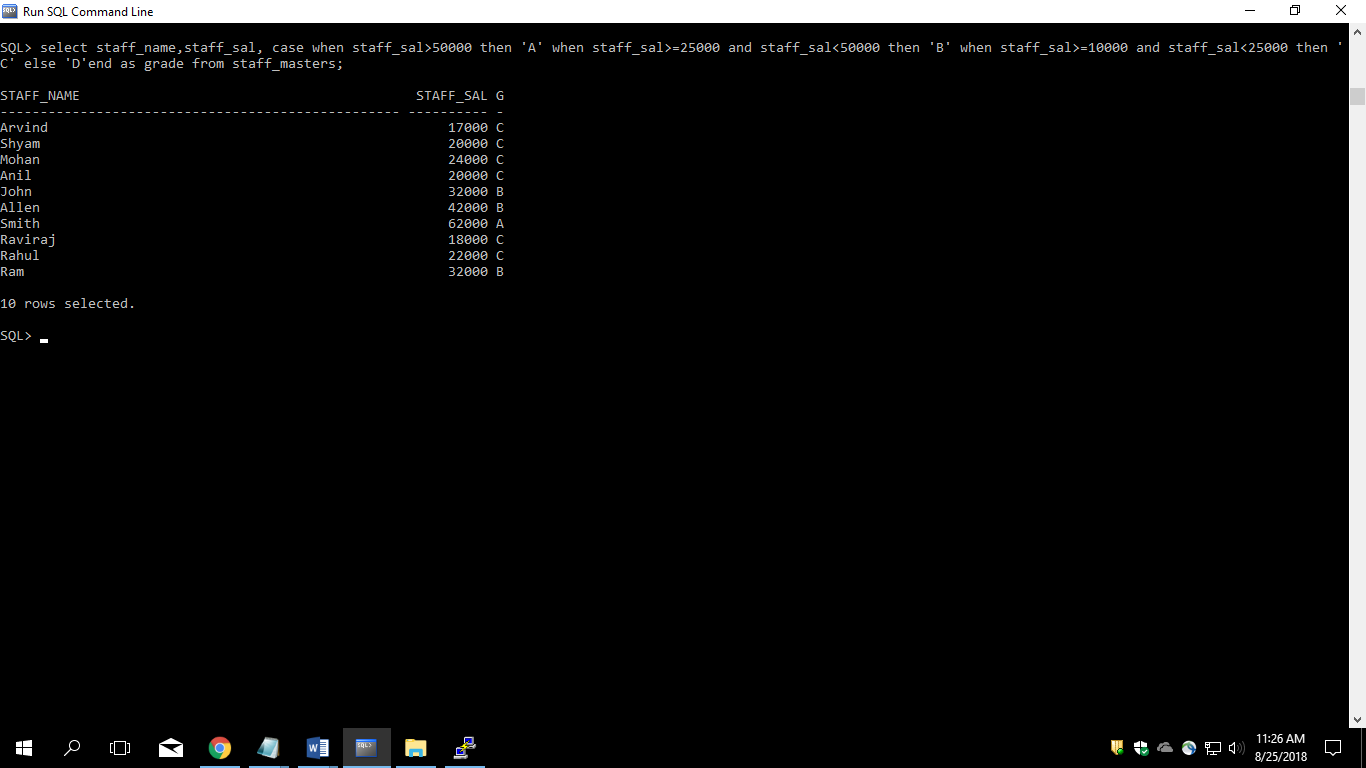


7. List the details of the employees who have joined in December (irrespective of the year).

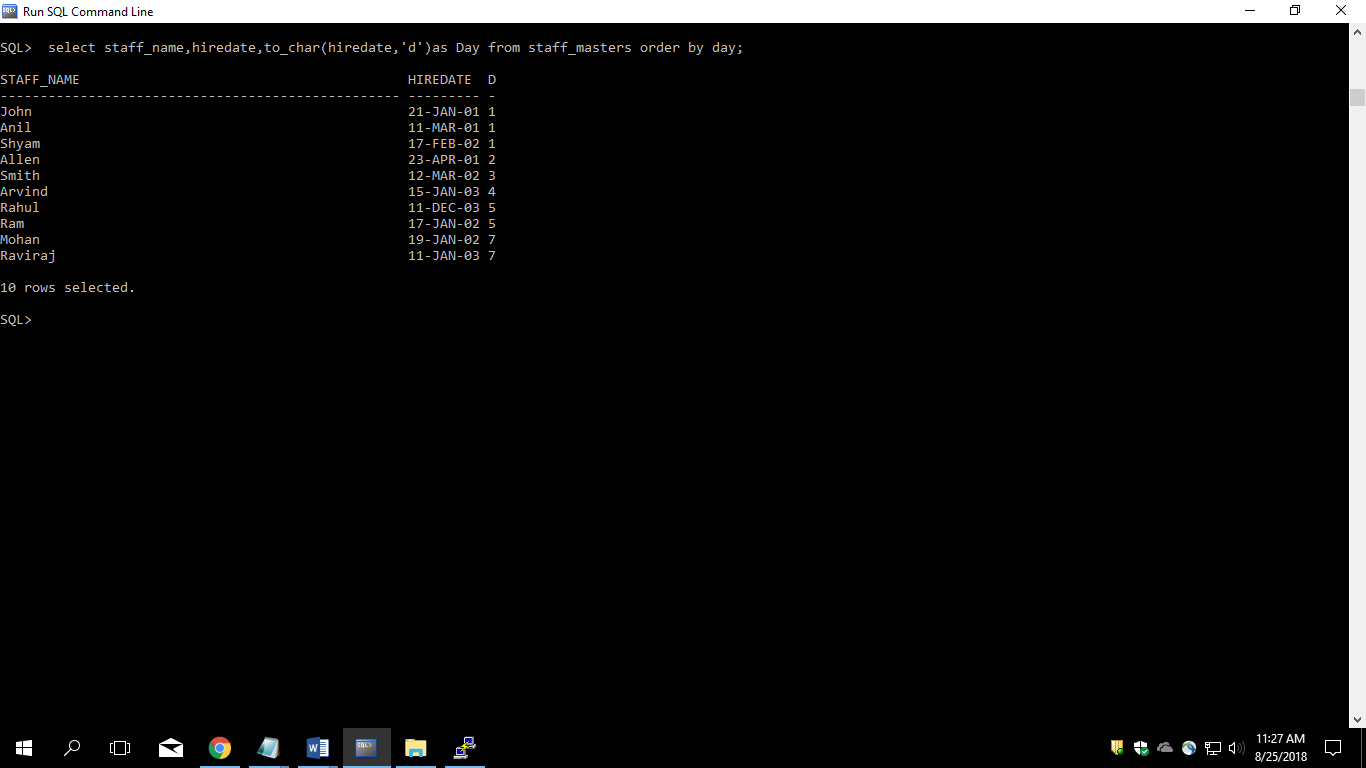


8. Write a query that displays Staff Name, Salary, and Grade of all staff. Grade depends on the following table.

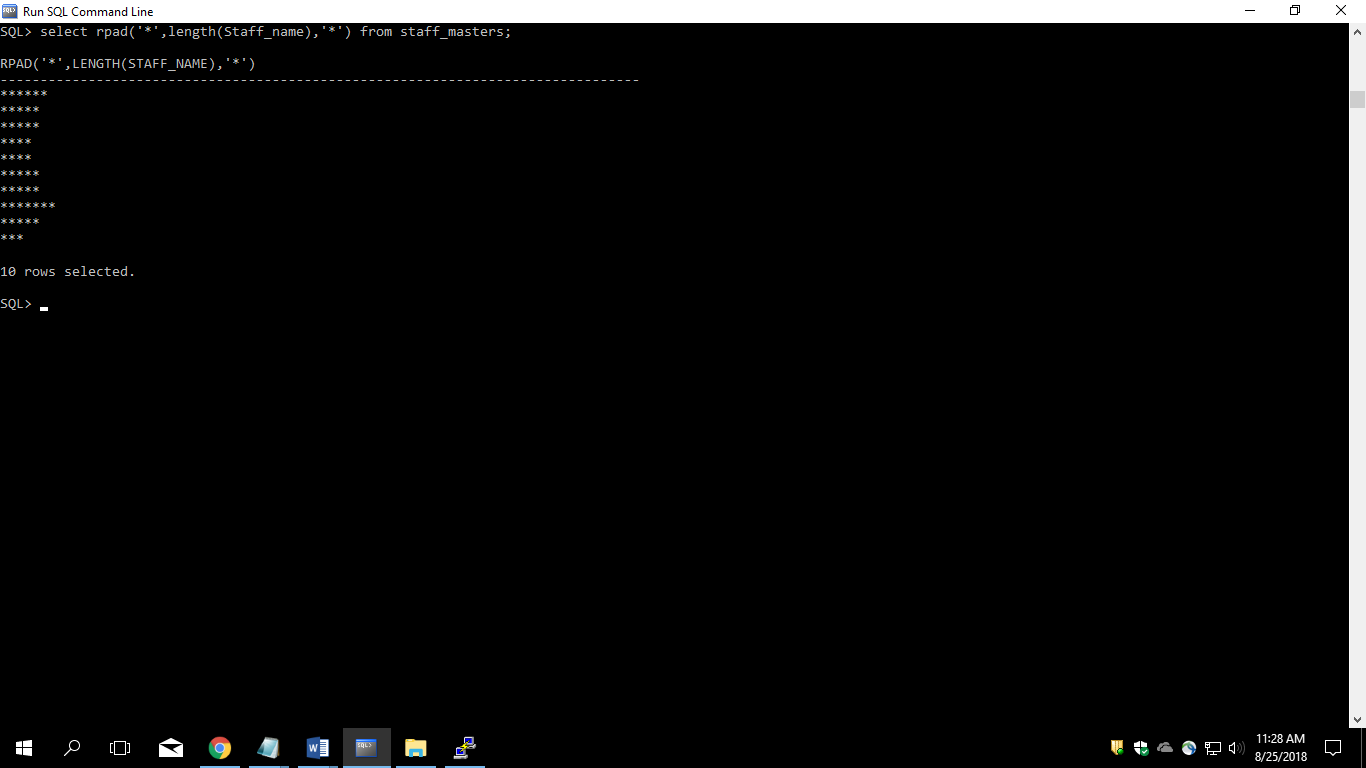
|  |  |
| --- | --- |
| Salary | Grade |
| Salary >=50000 | A |
| Salary >= 25000 < 50000 | B |
| Salary>=10000 < 25000 | C |
| OTHERS | D |

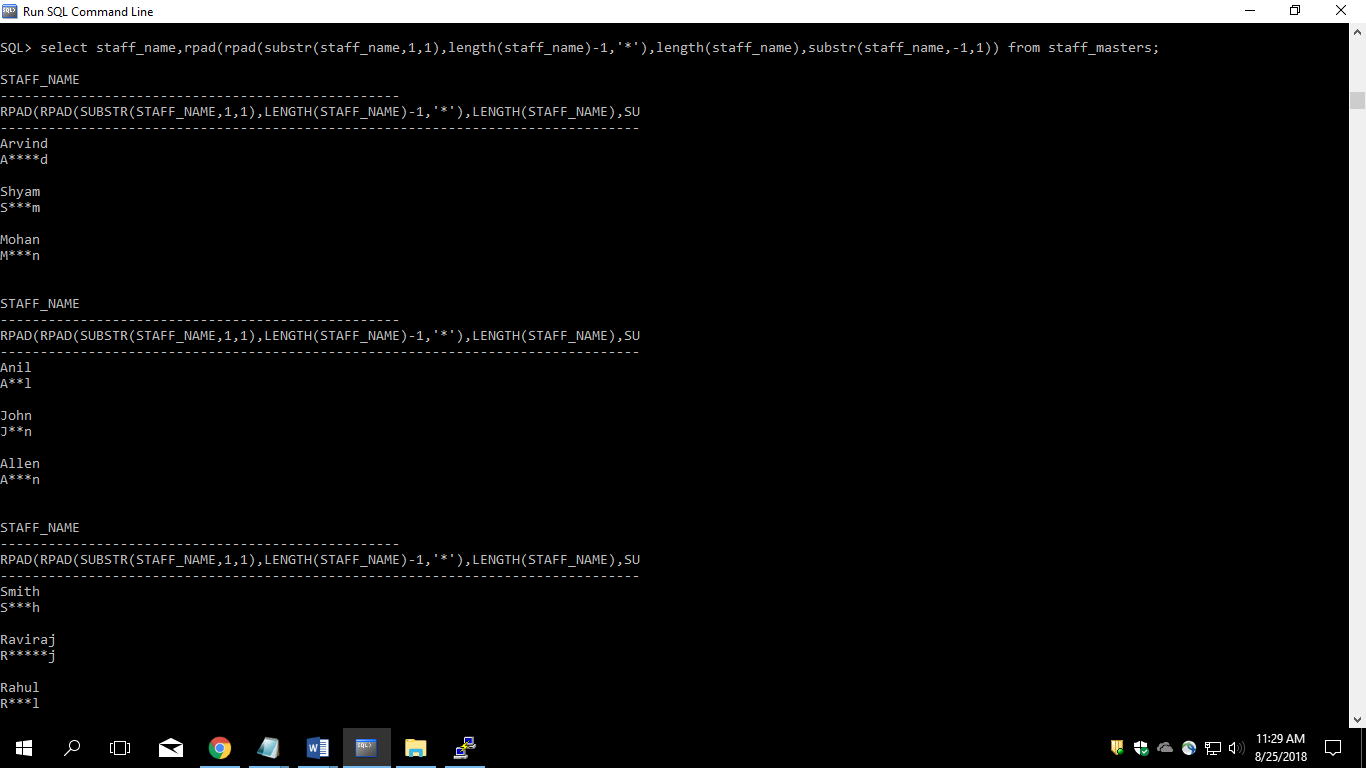


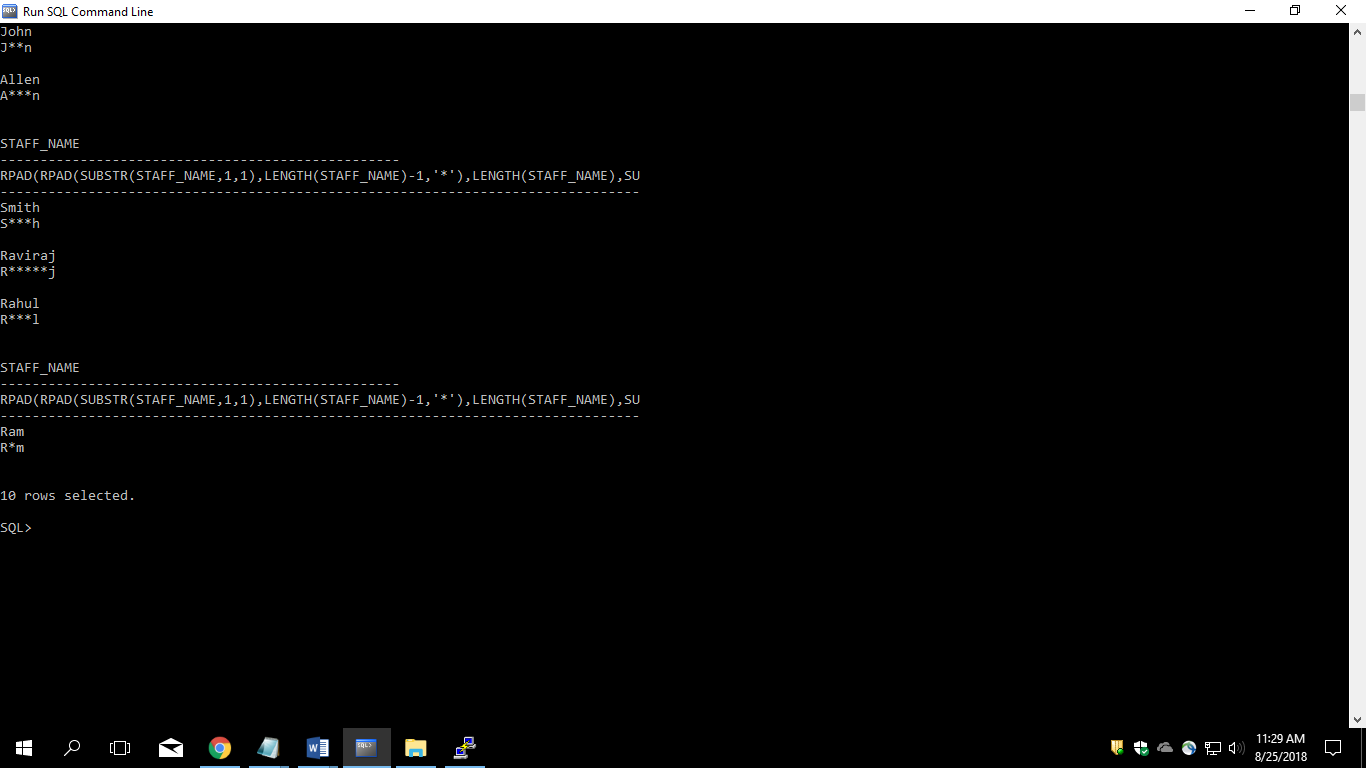
9. Display the Staff Name, Hire date and day of the week on which staff was hired. Label the column as DAY. Order the result by the day of the week starting with Monday.



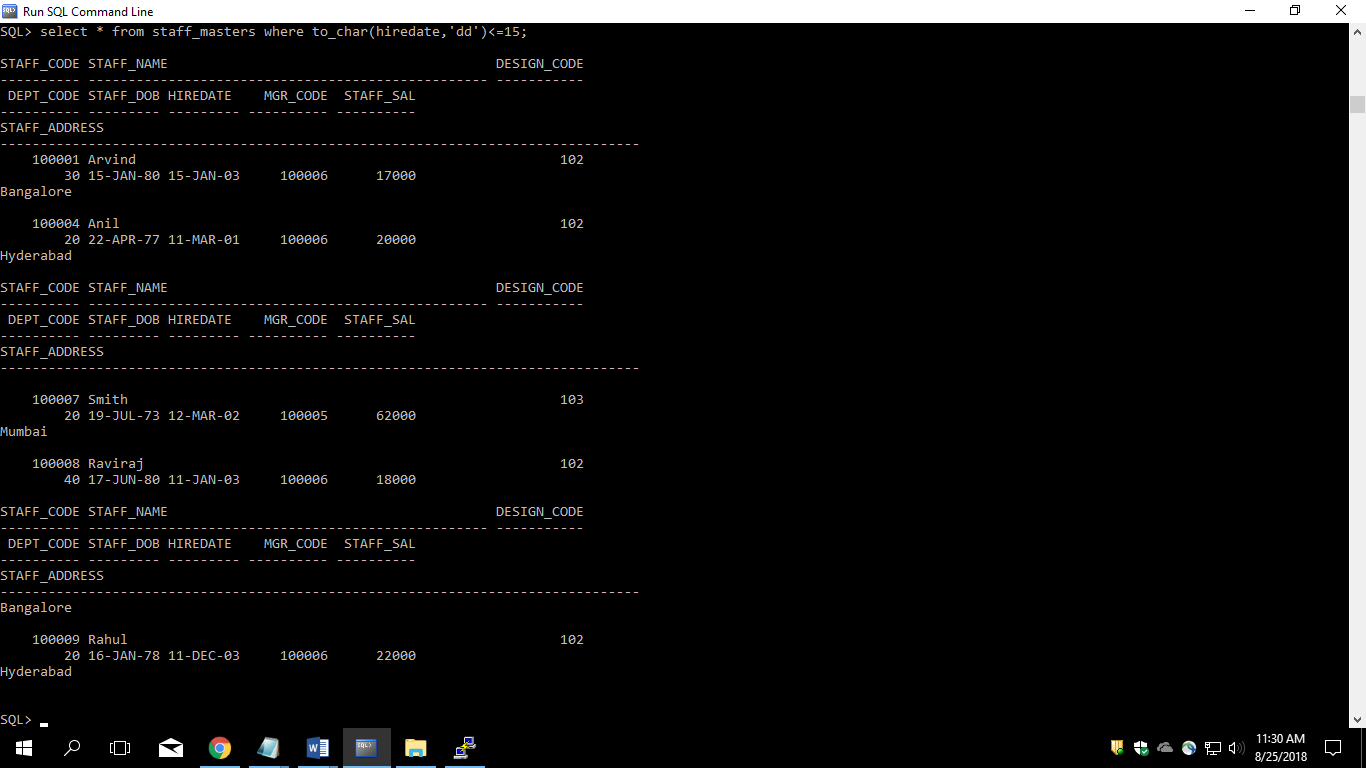
10. Show staff names with the respective numbers of asterisk from Staff table.



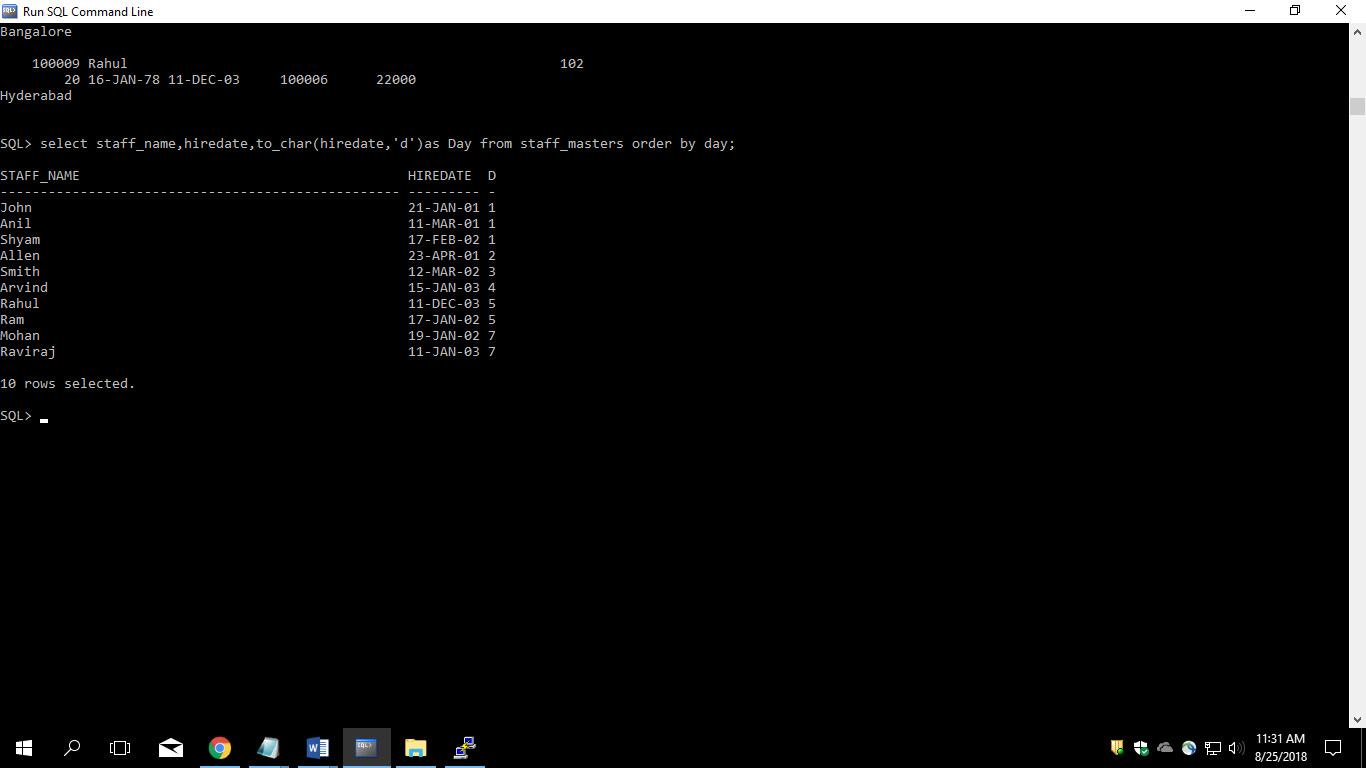
11. Show staff names with the respective numbers of asterisk from Staff table except first and last characters. For example: KING will be replaced with K\*\*G.   




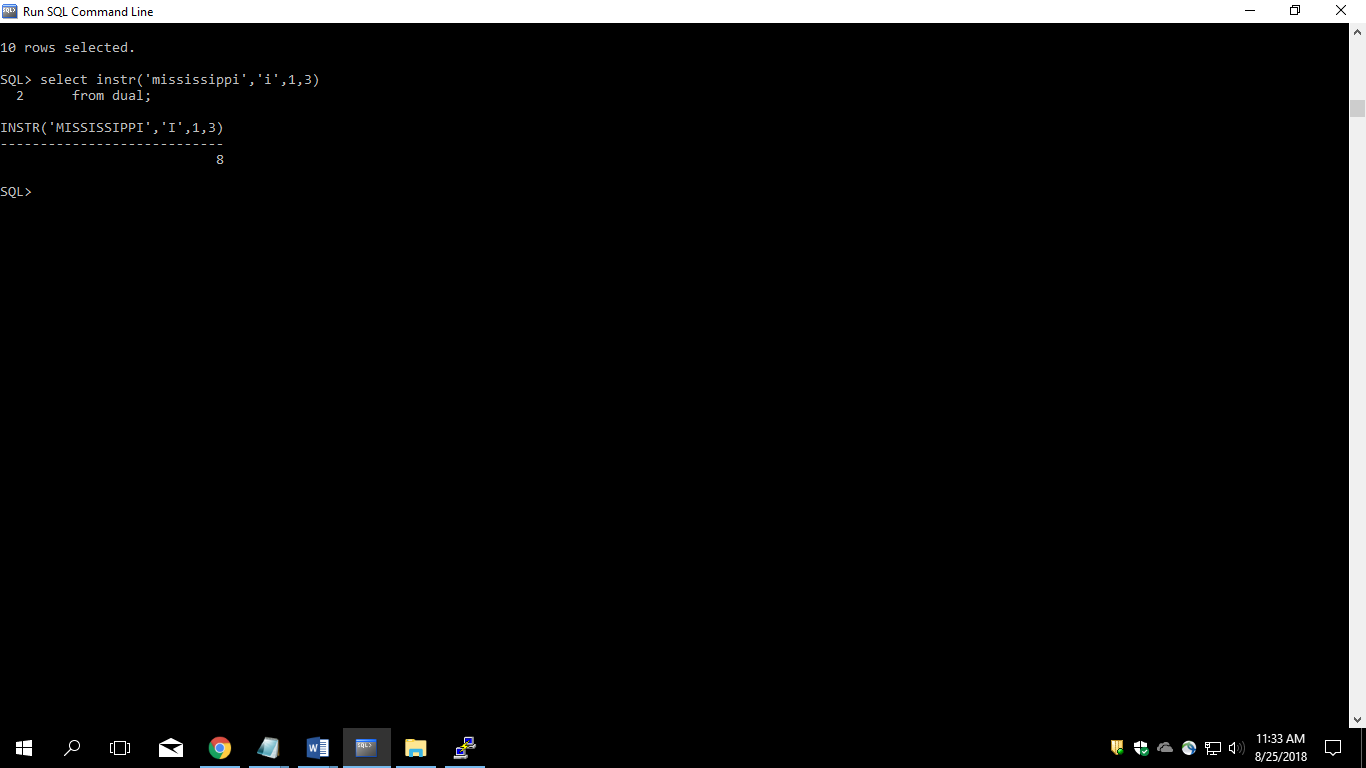
12. Show all staffs who were hired in the first half of the month.



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.

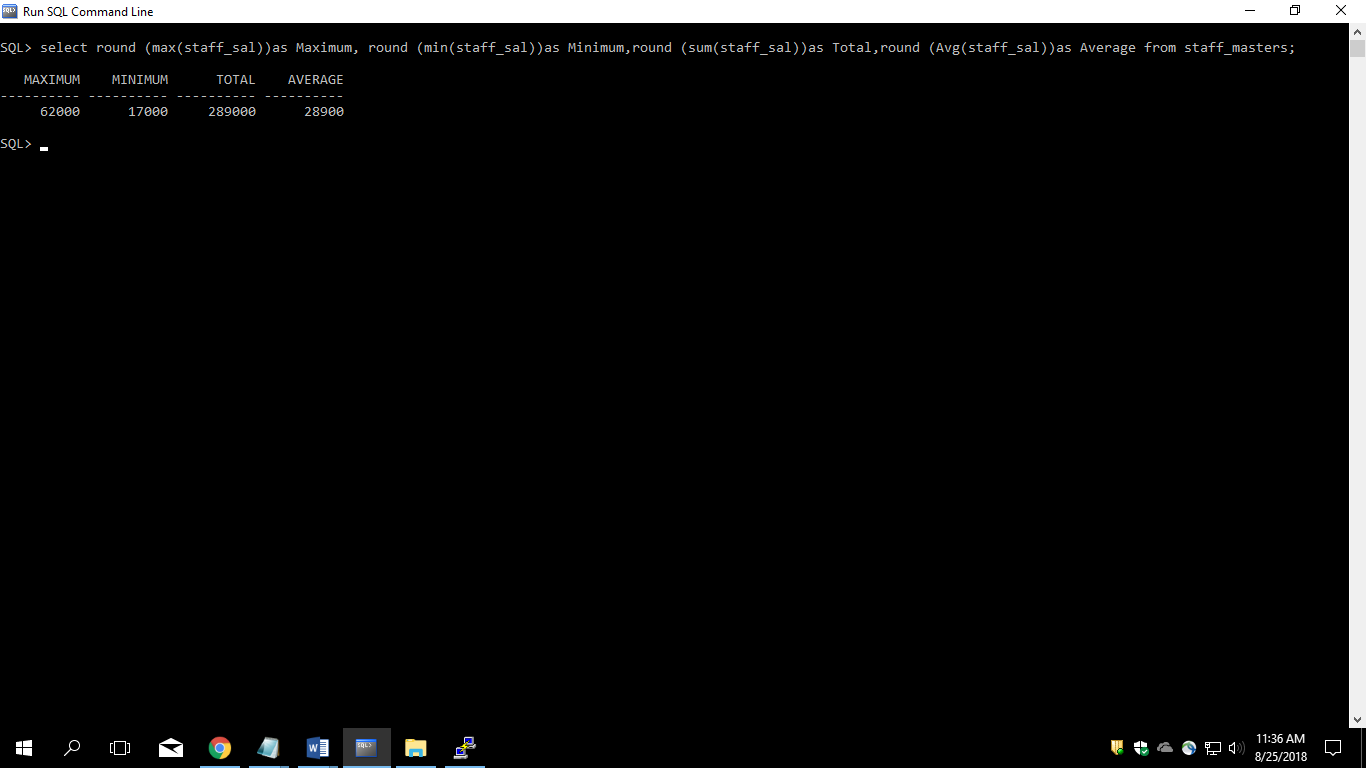


14. Write a query to find the position of third occurrence of ‘i’ in the given word ‘Mississippi’.

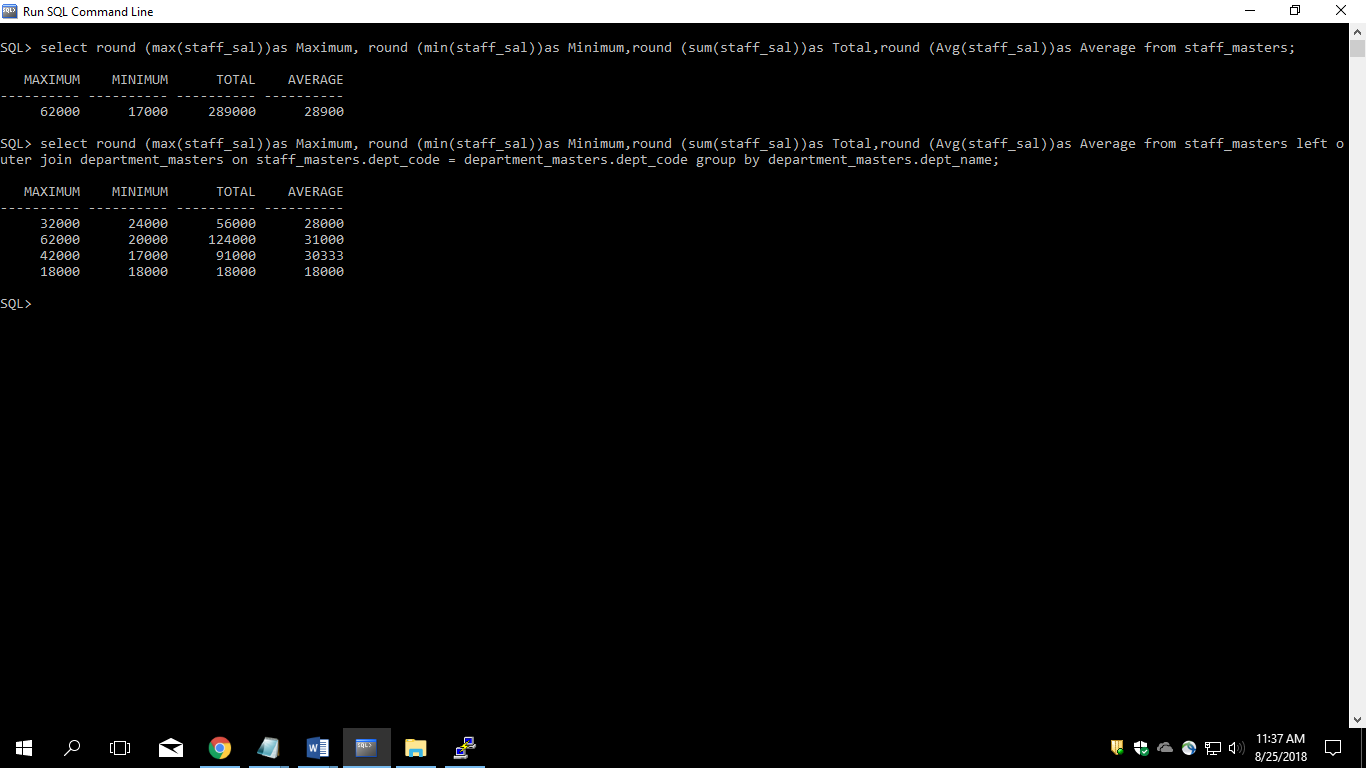


15. Write a query to find the pay date for the month. Pay date is the last Friday of the month. Display the date in the format “Twenty Eighth of January, 2002”. Label the heading as PAY DATE.

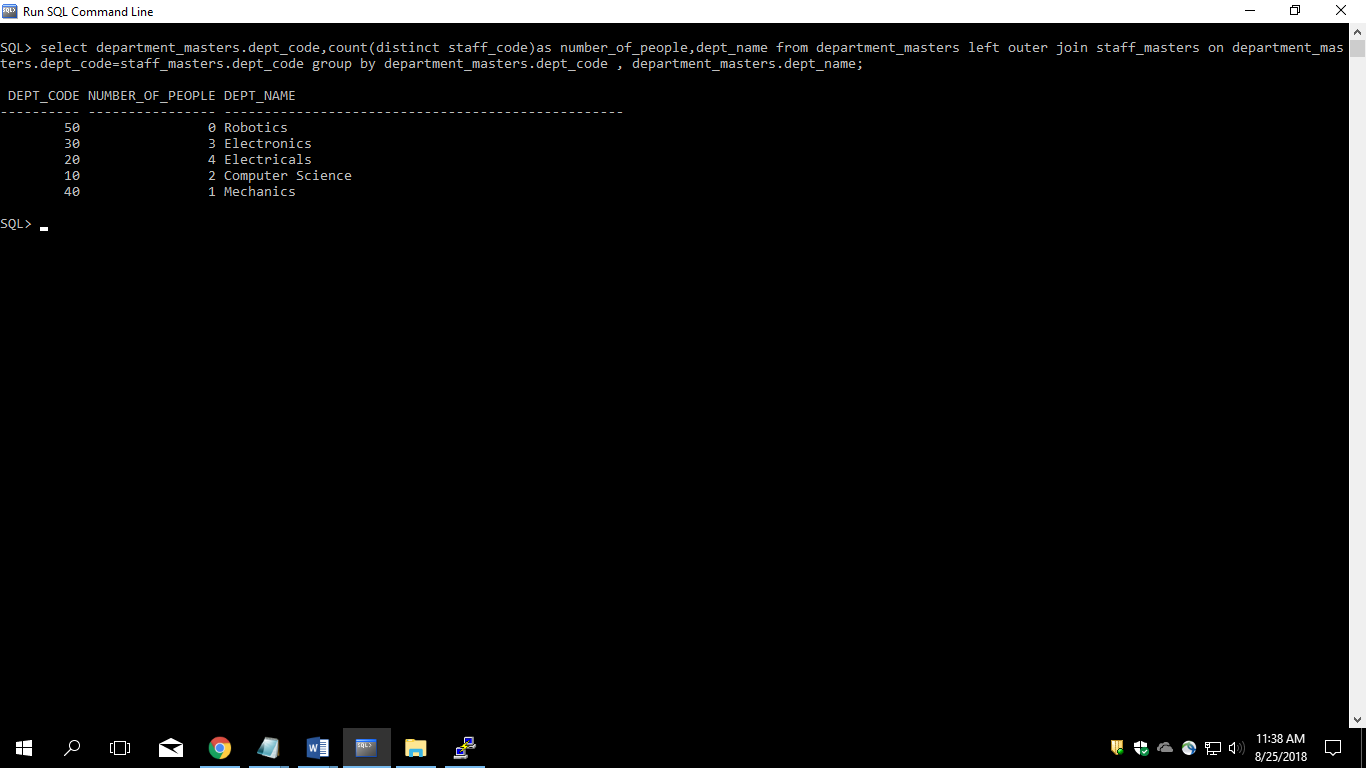
16. Display the Highest, Lowest, Total & Average salary of all staff. Label the columns Maximum, Minimum, Total and Average respectively. Round the result to nearest whole number.



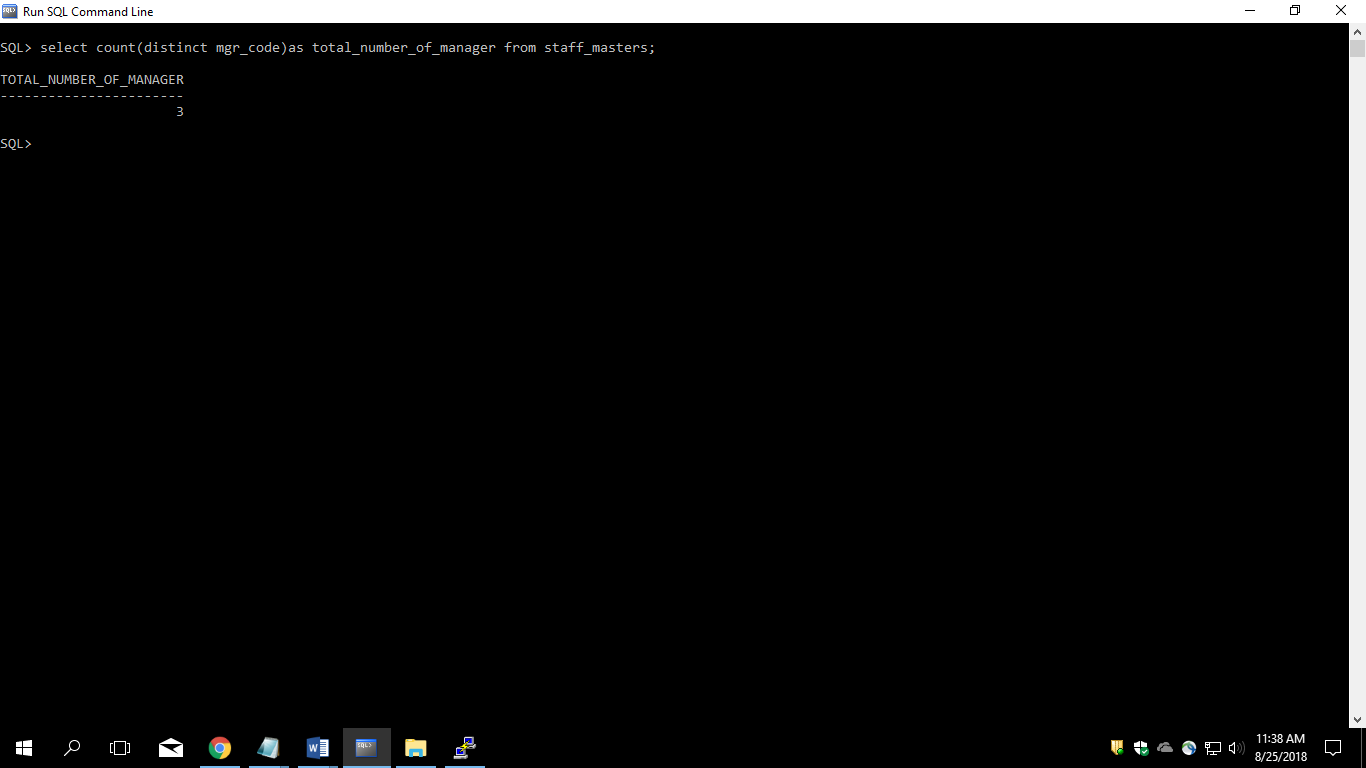
17. Edit the above query and display the same for each Department Name.



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’.

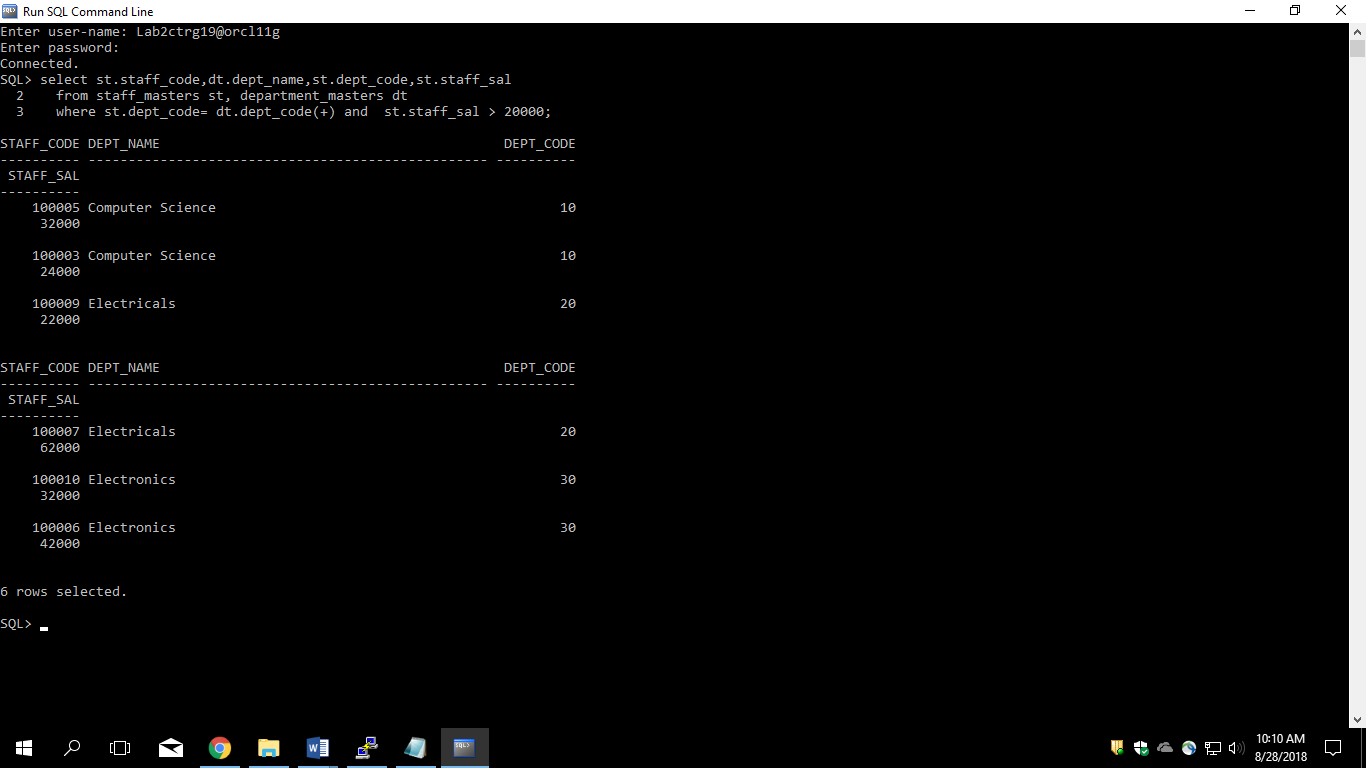


20. Display Manager Code, Manager Name and salary of lowest paid staff in that manager’s team. Exclude any group where minimum salary is less than 10000. Order the result on descending order of salary.

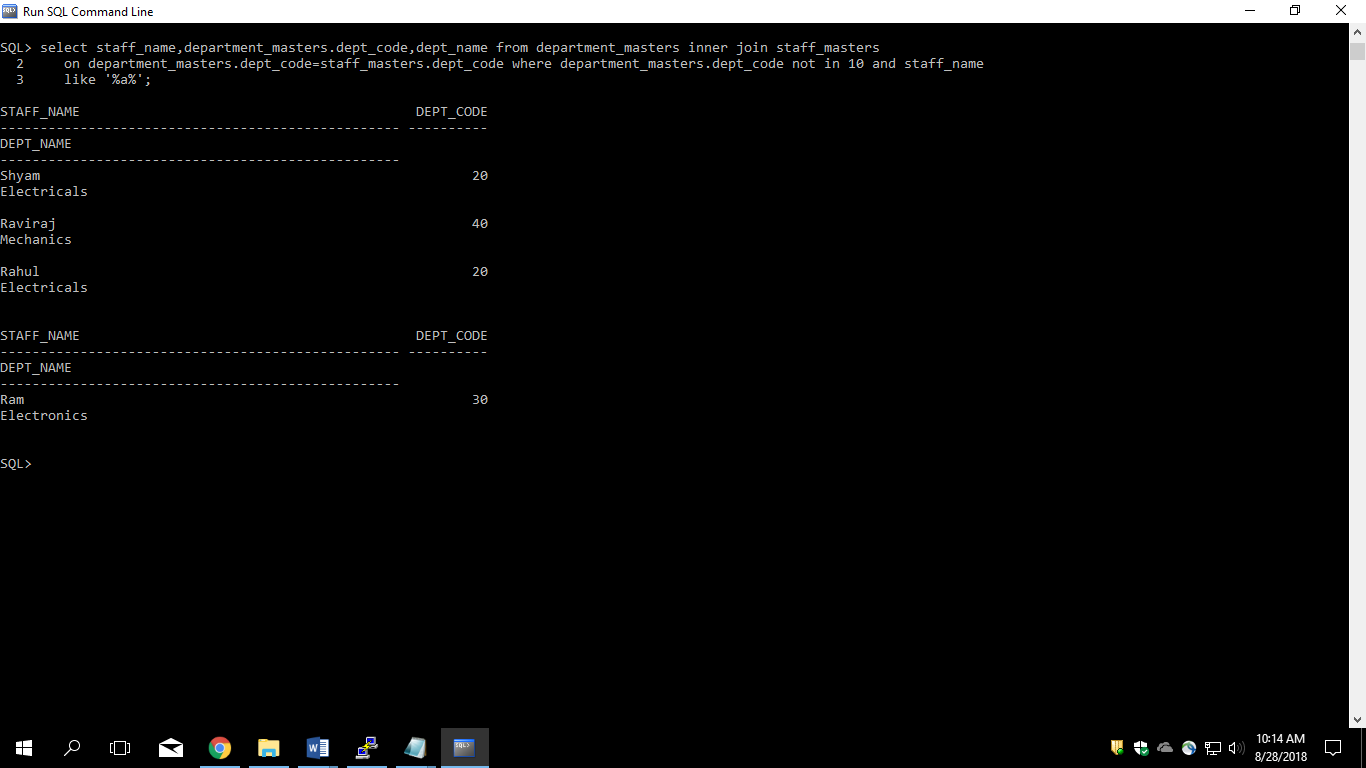
Select mgr\_code,min(staff\_sal) from staff\_masters group by mgr\_code having min(staff\_sal)>10000 order by min(staff\_sal) desc;

Lab3

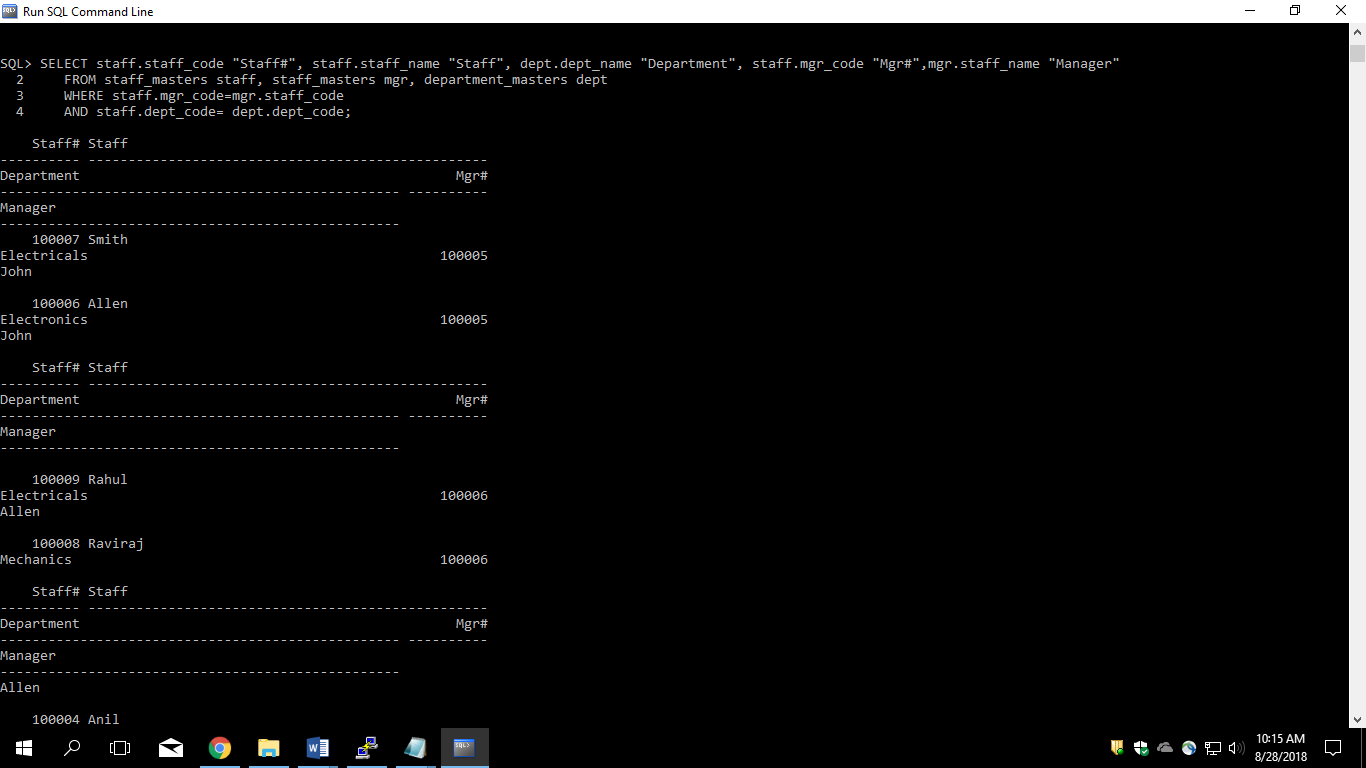
1. Write a query which displays Staff Name, Department Code, Department Name, and Salary for all staff who earns more than 20000.



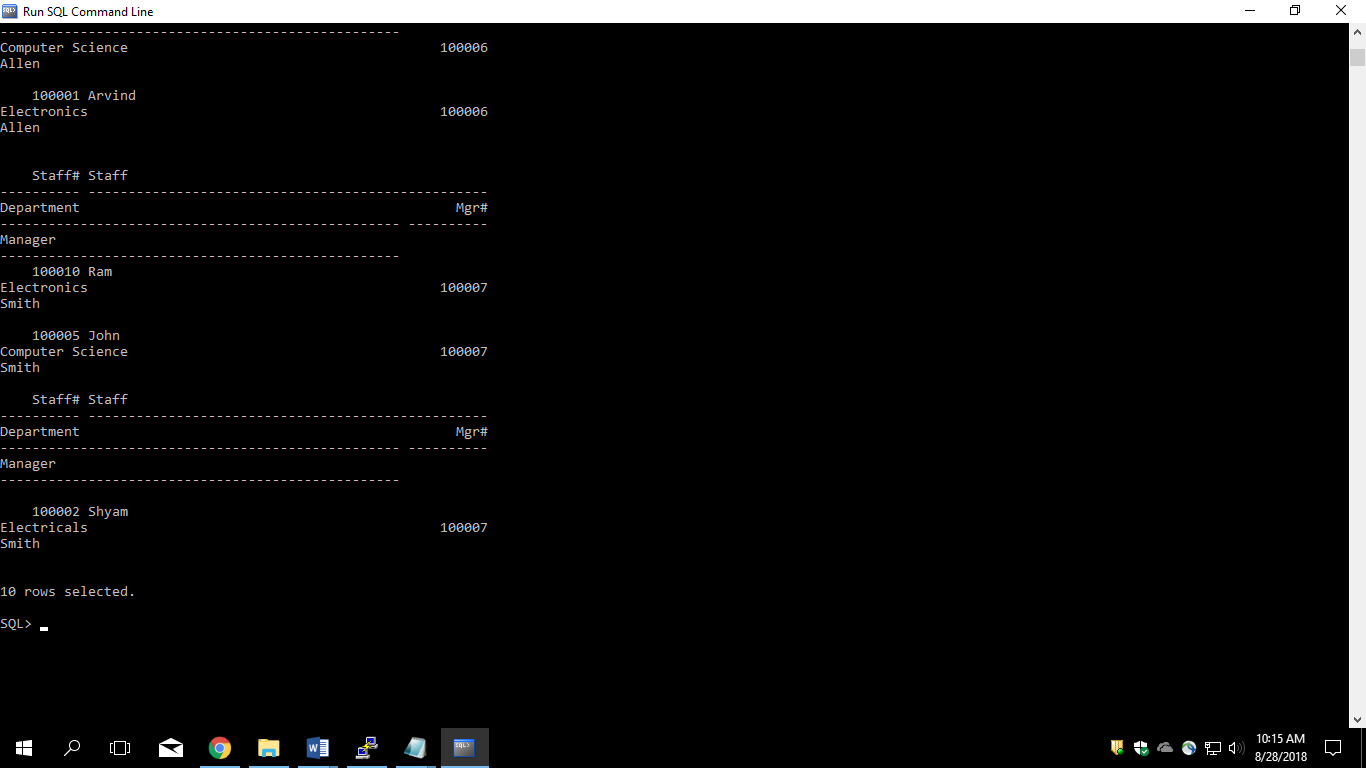
2. Write a query to display Staff Name, Department Code, and Department Name for all staff who do not work in Department code 10 and have ‘A’ in their name.



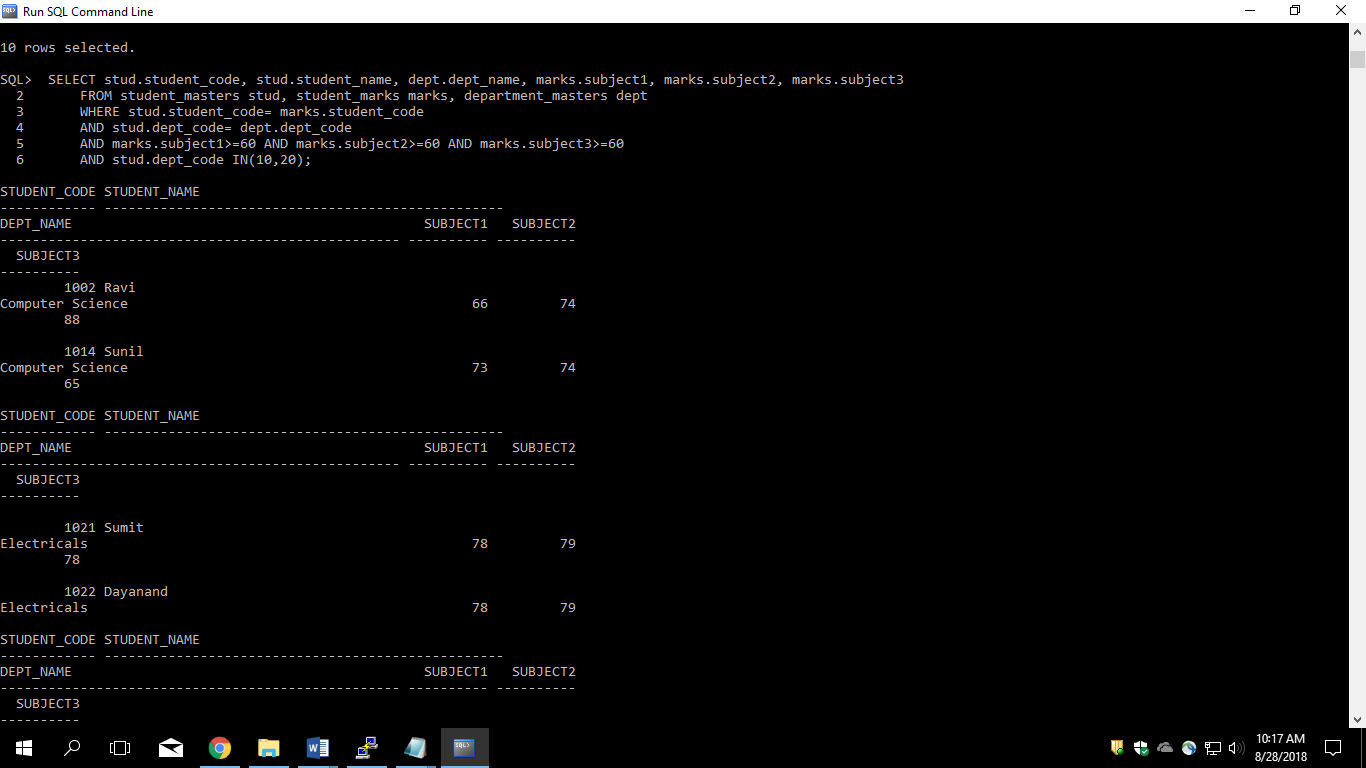
3. Display Staff Code, Staff Name, Department Name, and his manager’s number and name. Label the columns Staff#, Staff, Mgr#, Manager.

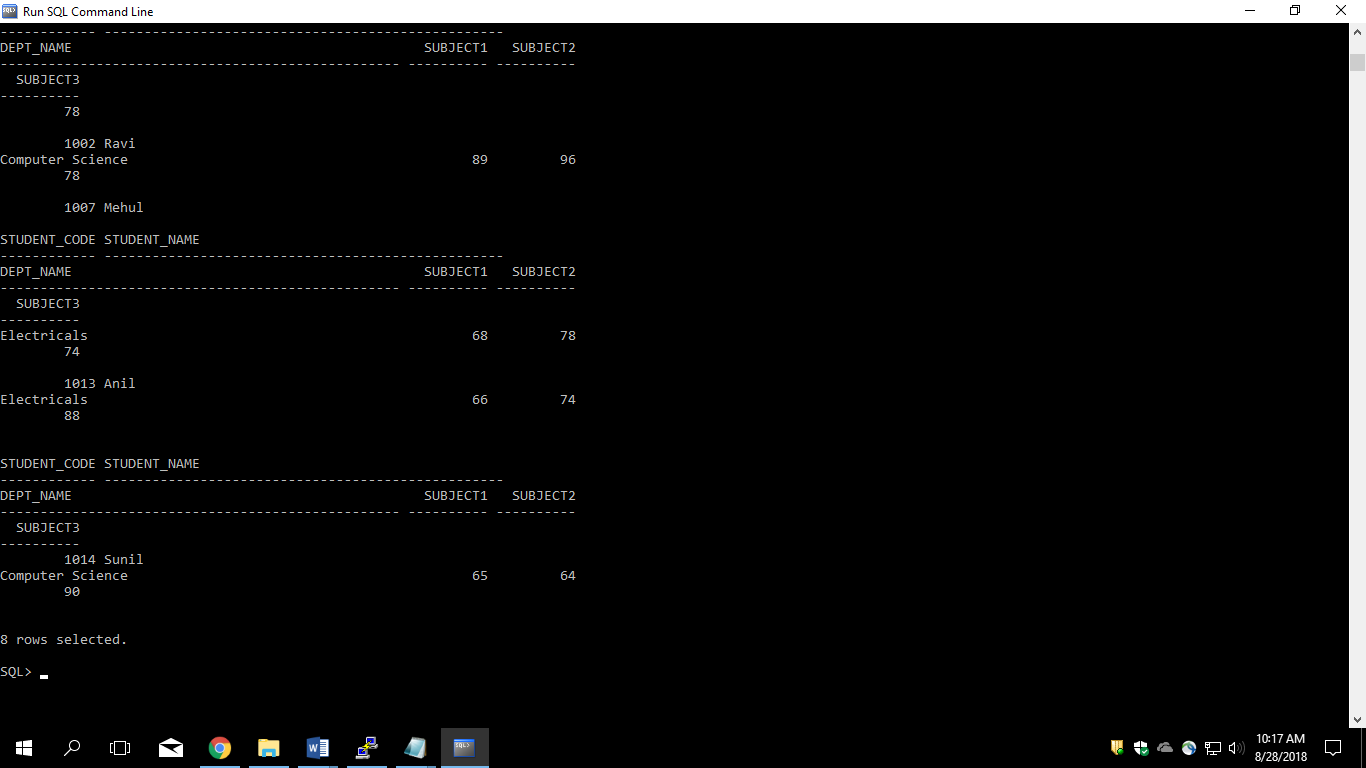




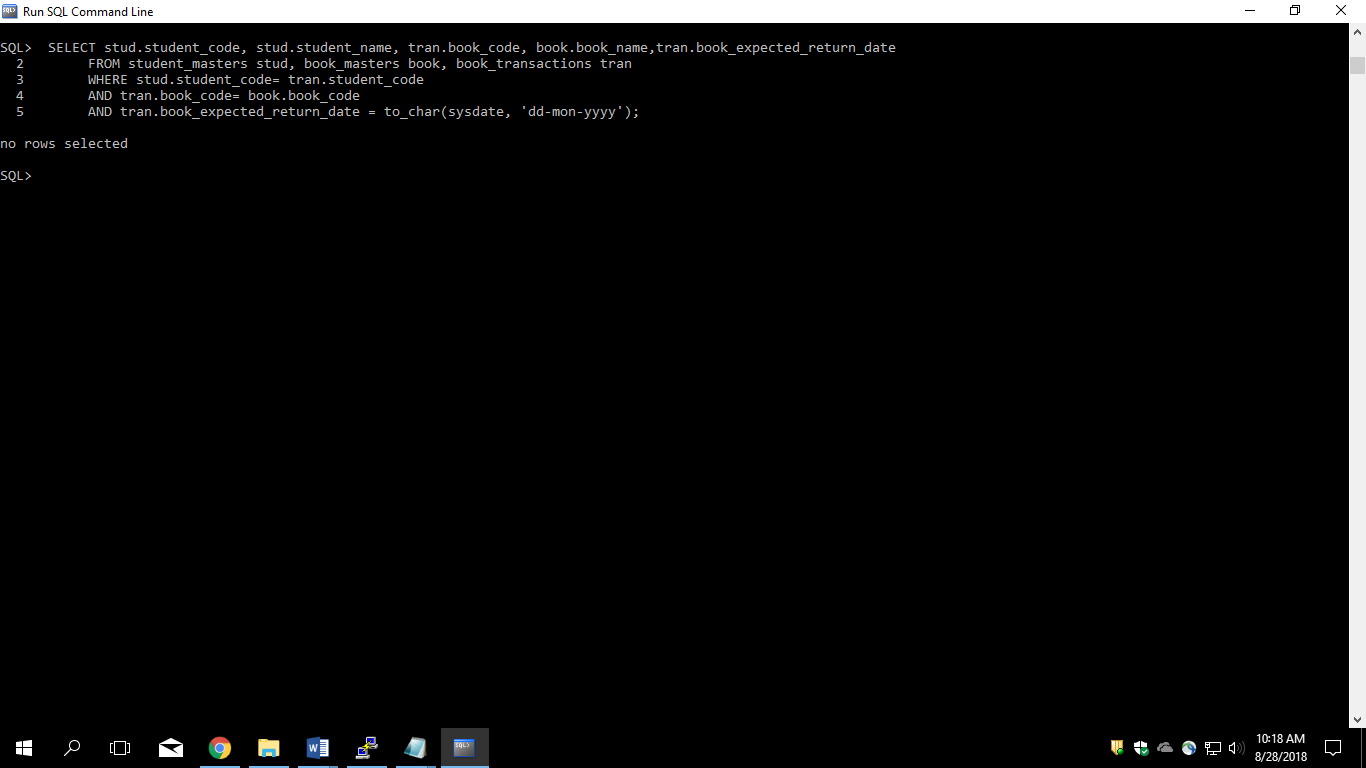


4. Create a query that will display Student Code, Student Name, Department Name, Subjec1, Subject2, and Subject3 for all students who are getting 60 and above in each subject from department 10 and 20.

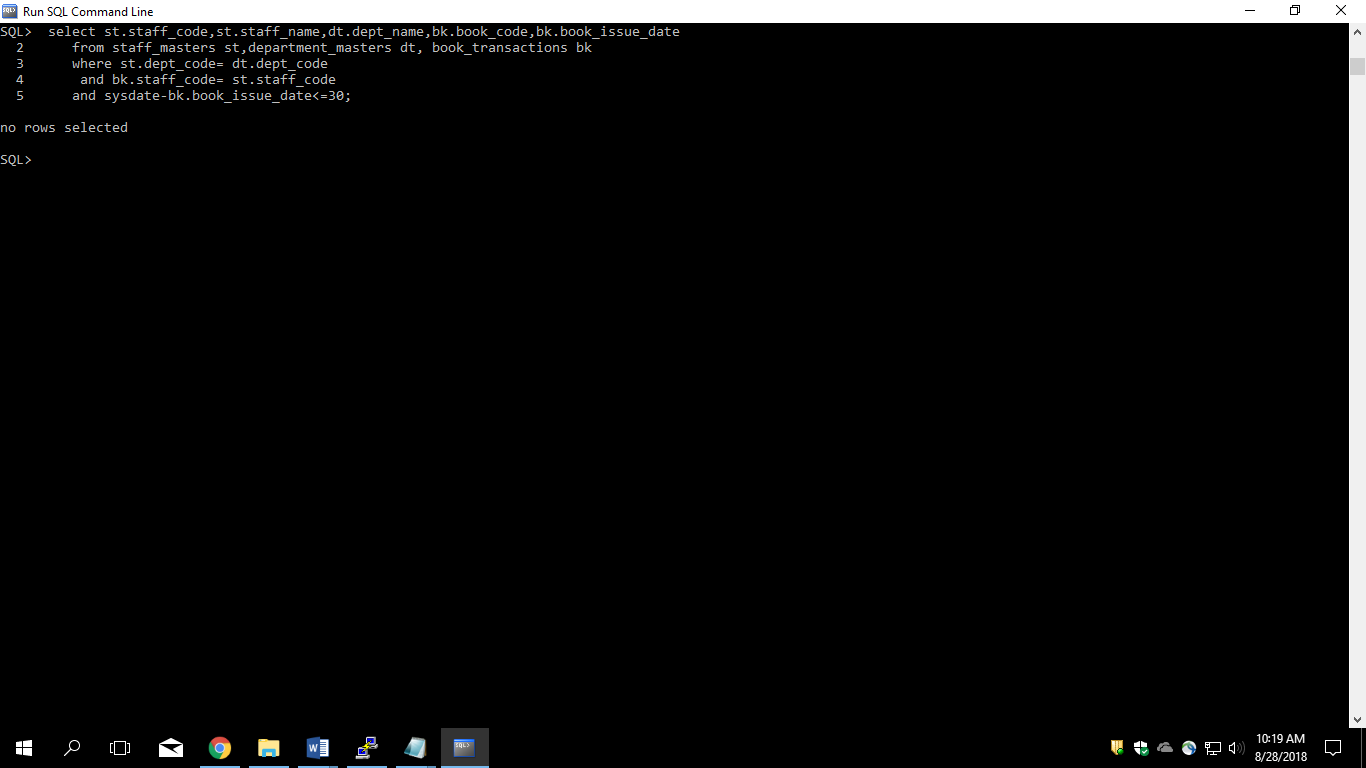




5. Create a query that will display Student Code, Student Name, Book Code, and Book Name for all students whose expected book return date is today.



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.

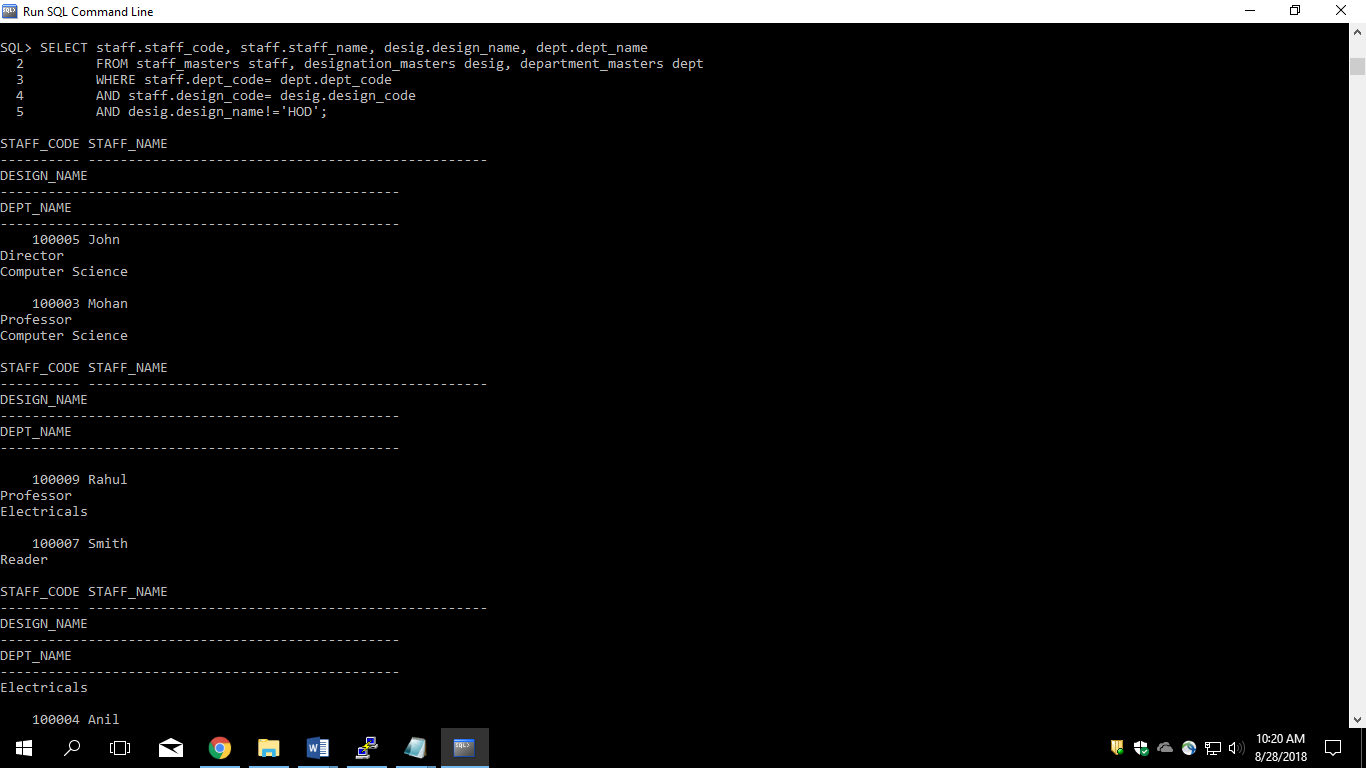


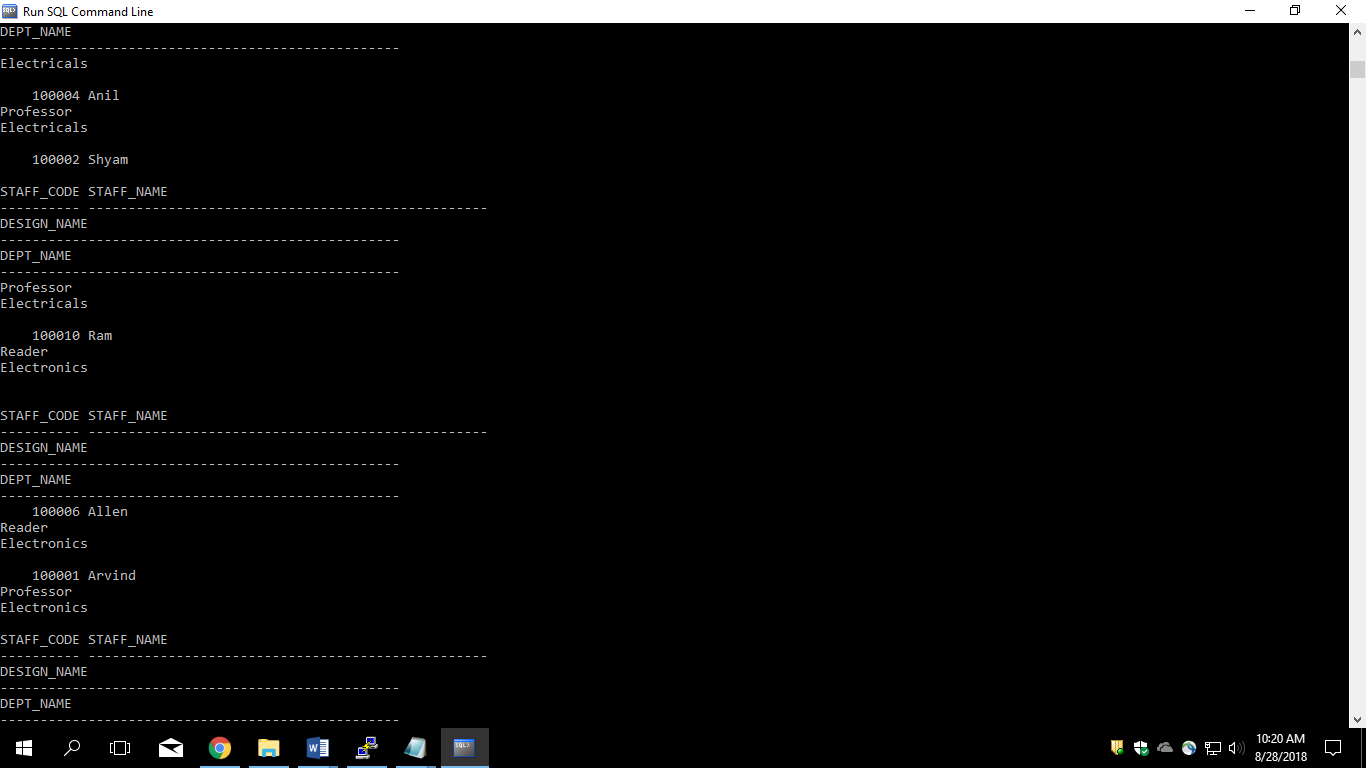
7. Generate a report which contains the following information.

Staff Code Staff Name Designation Department Name

Department Head

For all staff excluding HOD (List should not contain the details of Department head).



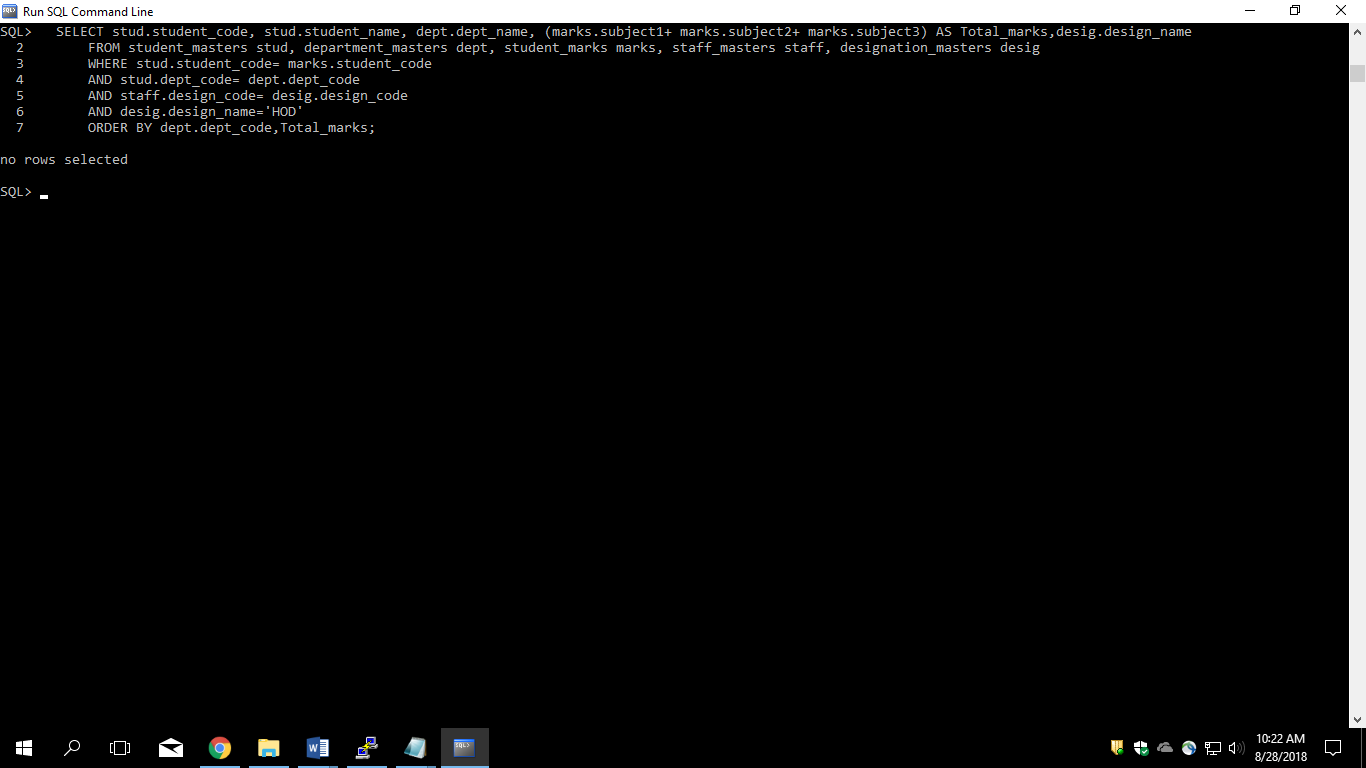


8. Generate a report which contains the following information

Student Code Student Name Department Name Total Marks

HOD Name

Sort the output on Department Name and Total Marks.



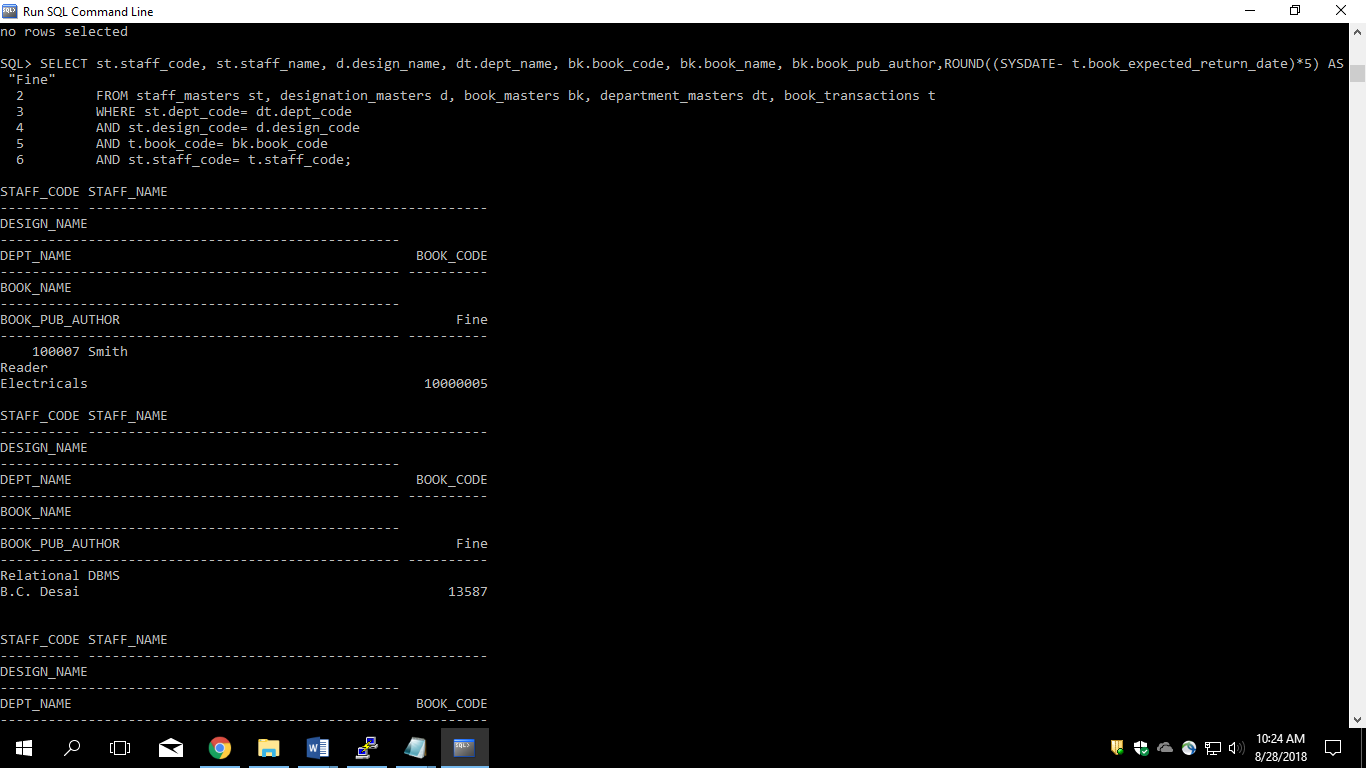
9. Generate a report which contains the following information.

Staff Code, Staff Name, Designation, Department, Book Code, Book Name,

Author, Fine

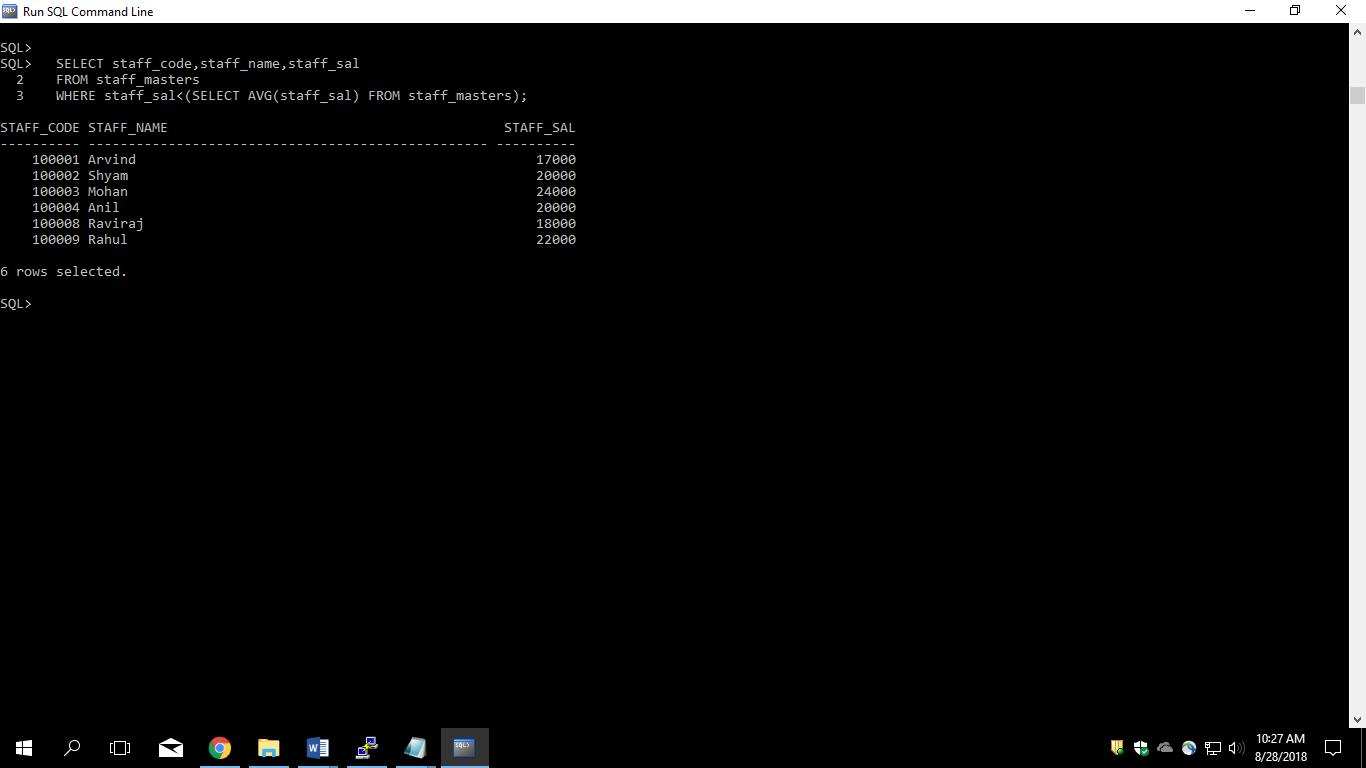
For the staff who have not return the book. Fine will be calculated as Rs. 5 per day.

Fine = 5 \* (No. of days = Current Date – Expected return date).

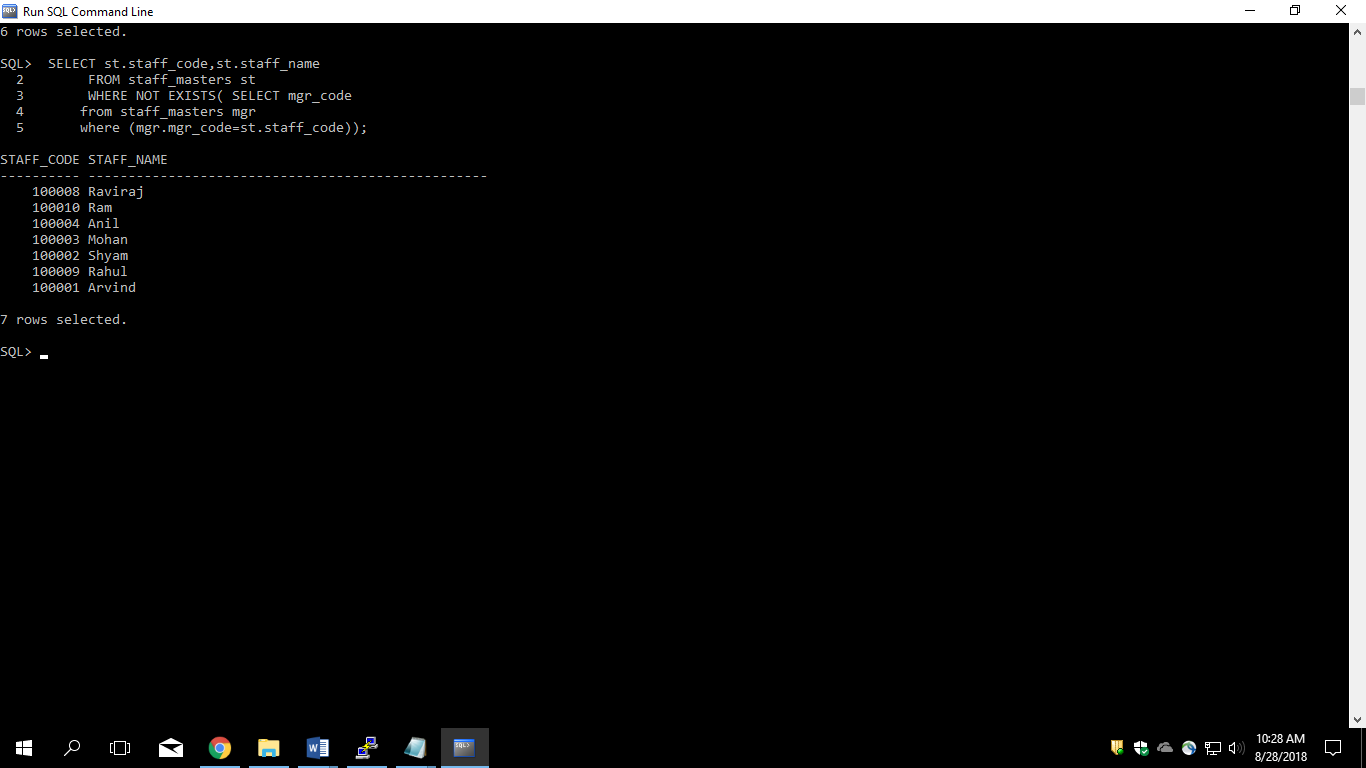


Continue…

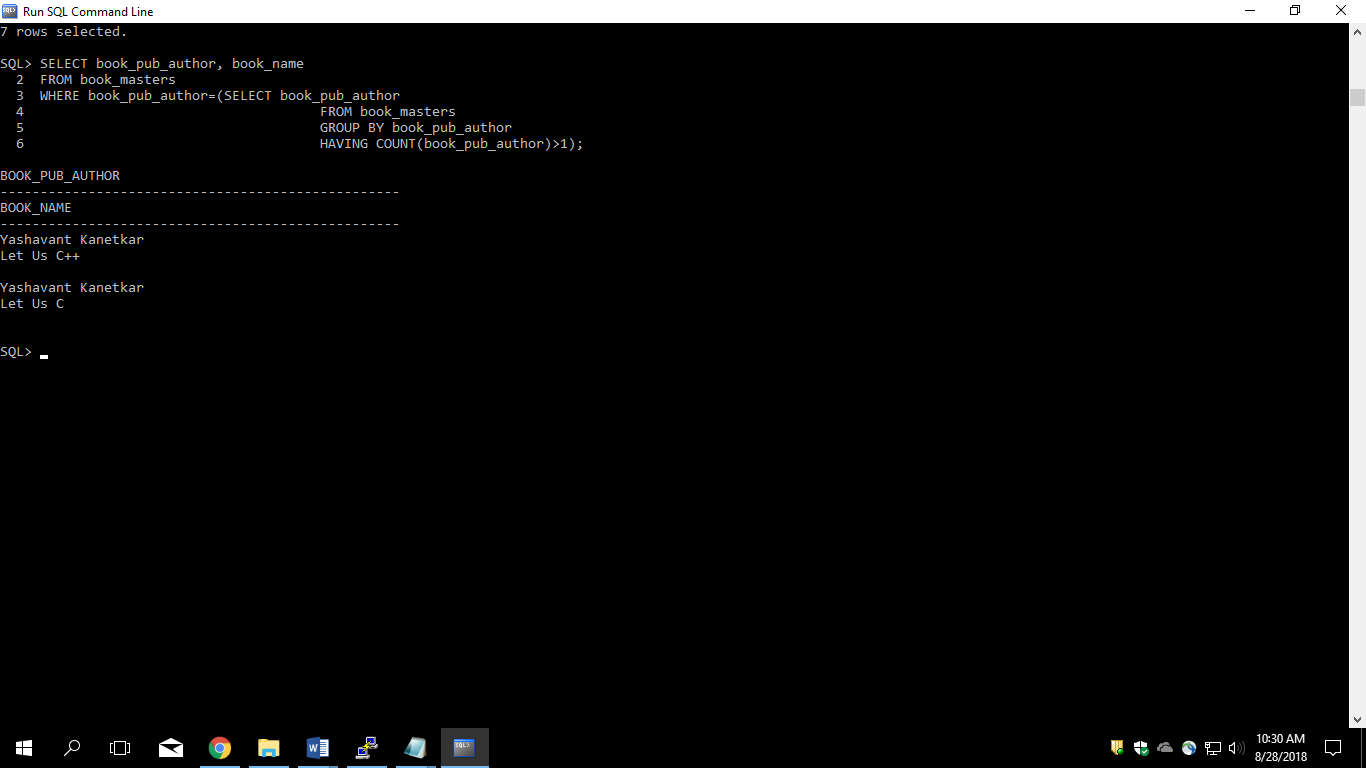
10. List Staff Code, Staff Name, and Salary for those who are getting less than the average salary of organization.



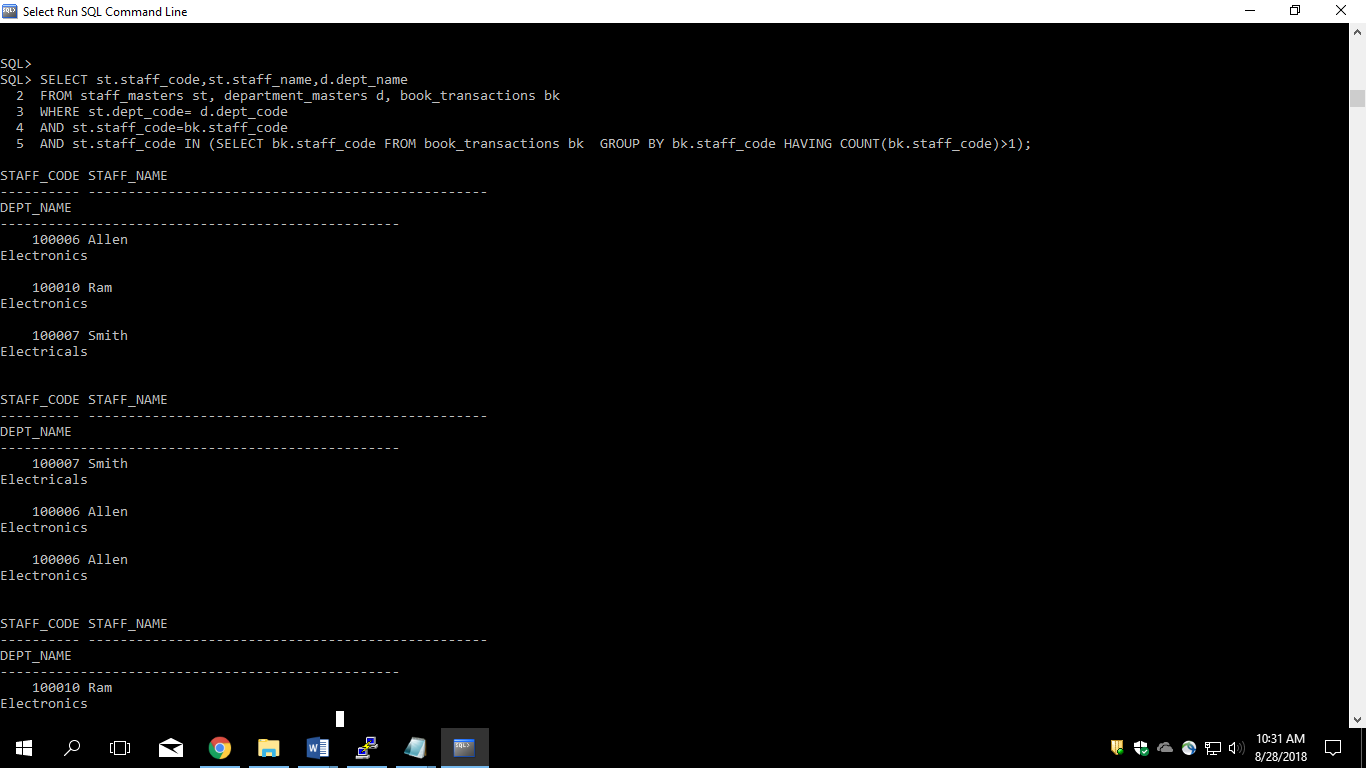
11. List the Staff Code, Staff Name who are not Manager.

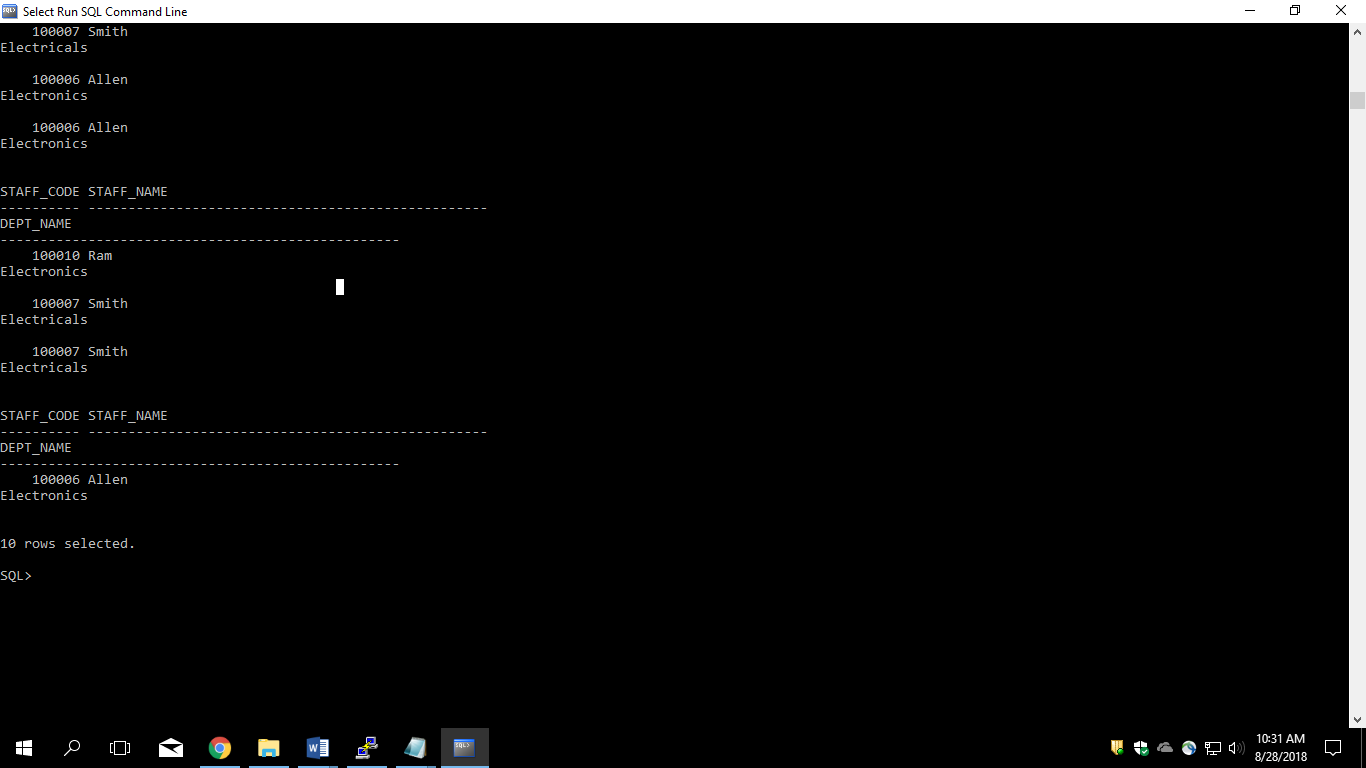


12. Display Author Name, Book Name for those authors who wrote more than one book.



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

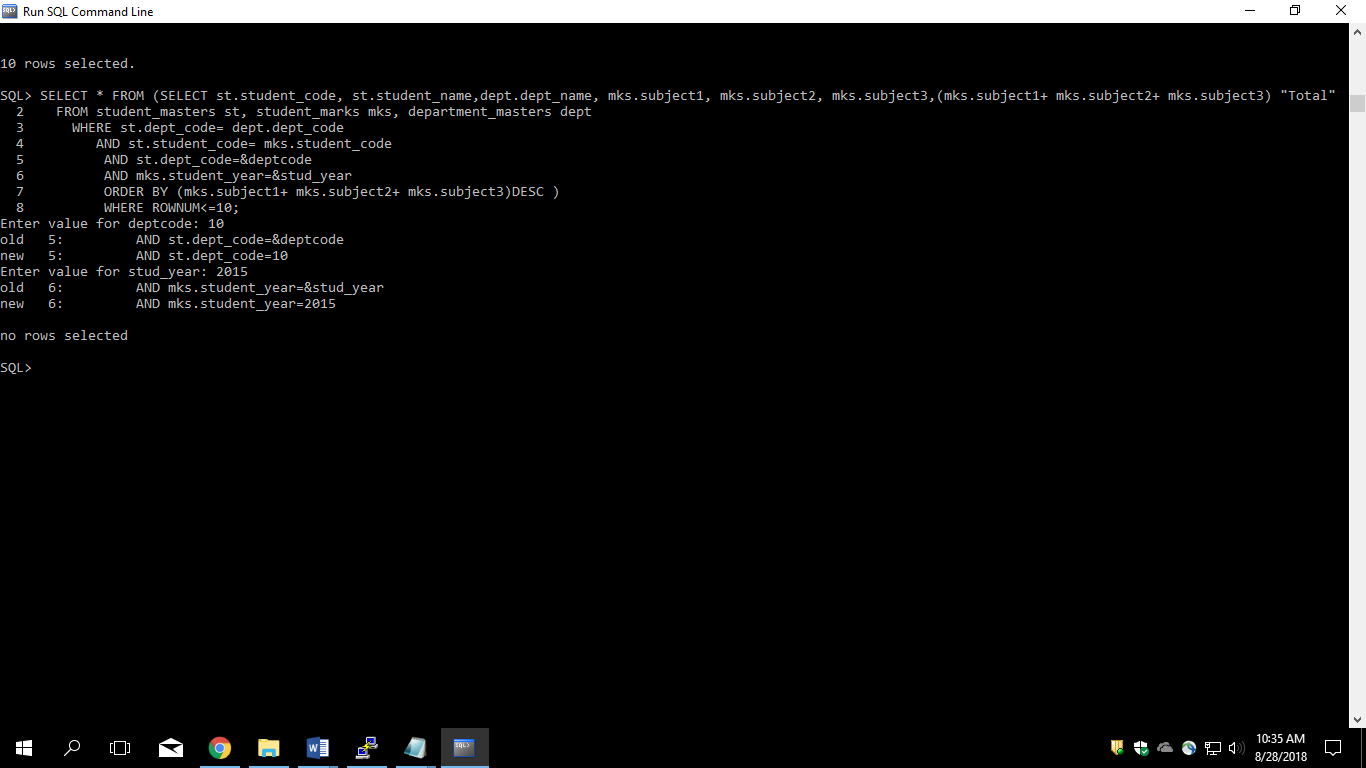




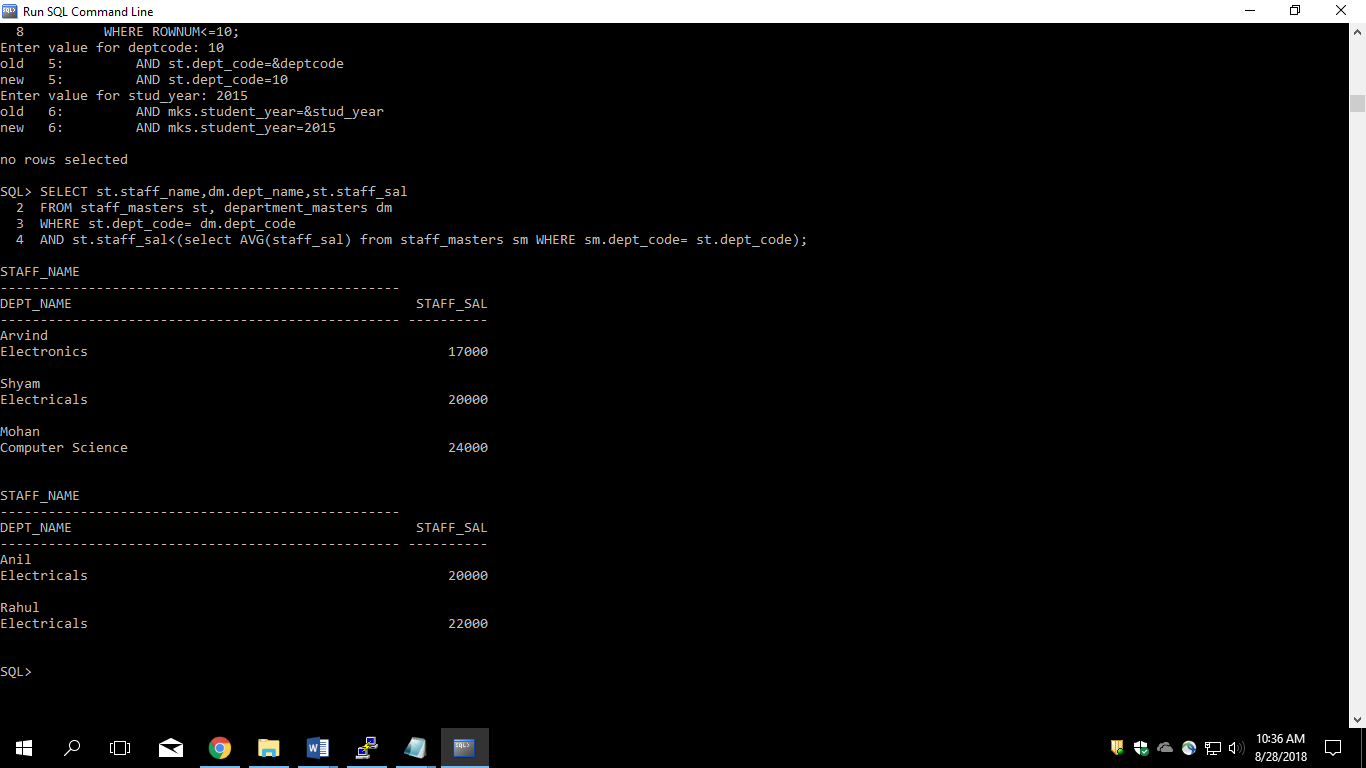
14. Display top ten students for a specified department. Details are:

Student Code, Student Name, Department Name, Subject1, Subject2,

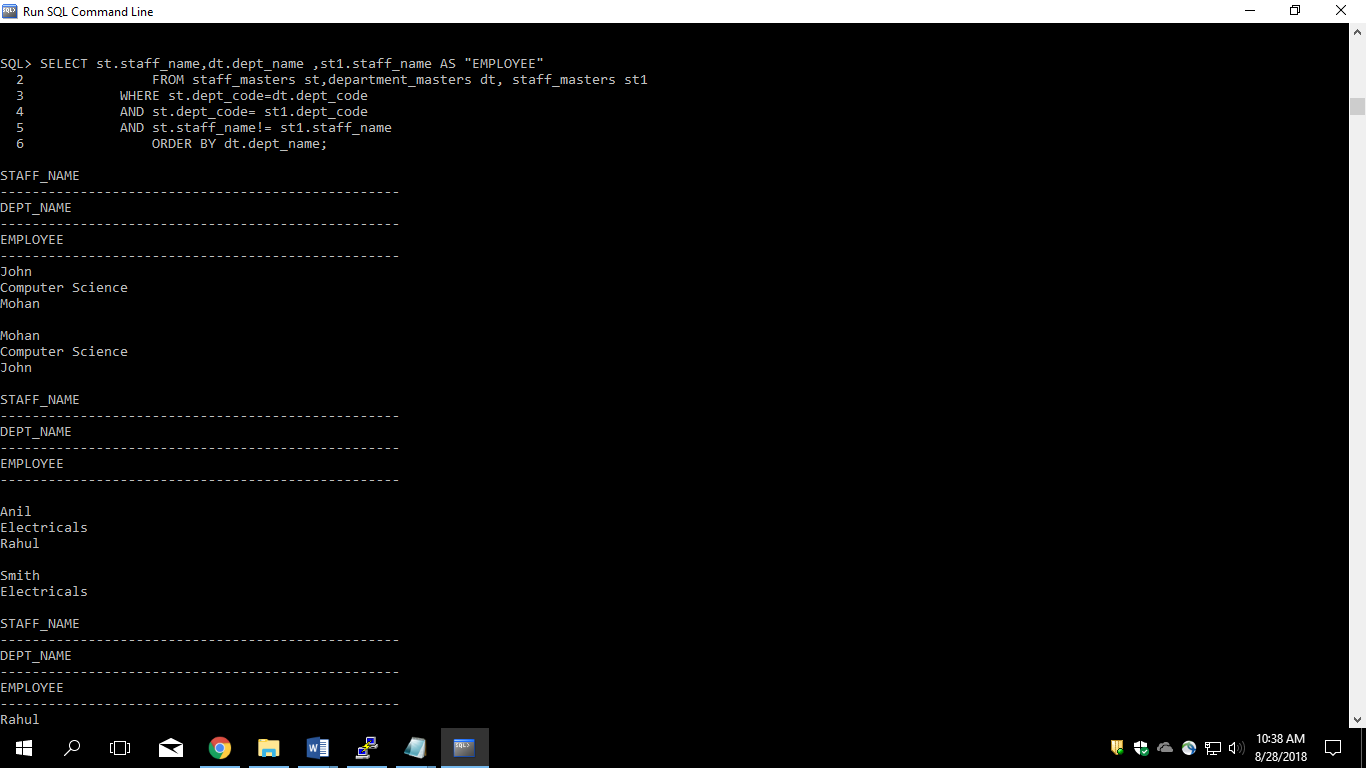
Subject3, Total.

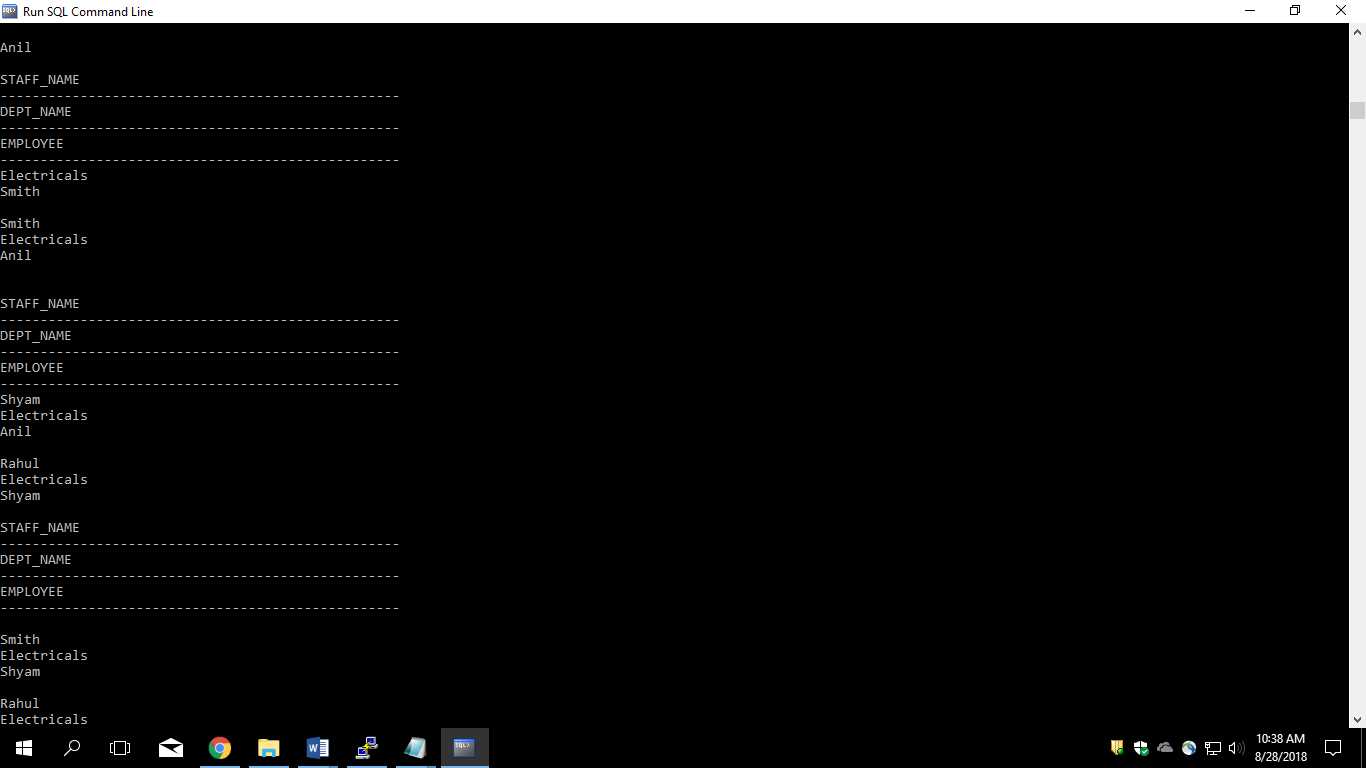


15. Display the Staff Name, Department Name, and Salary for those staff who are getting less than average salary in their own department

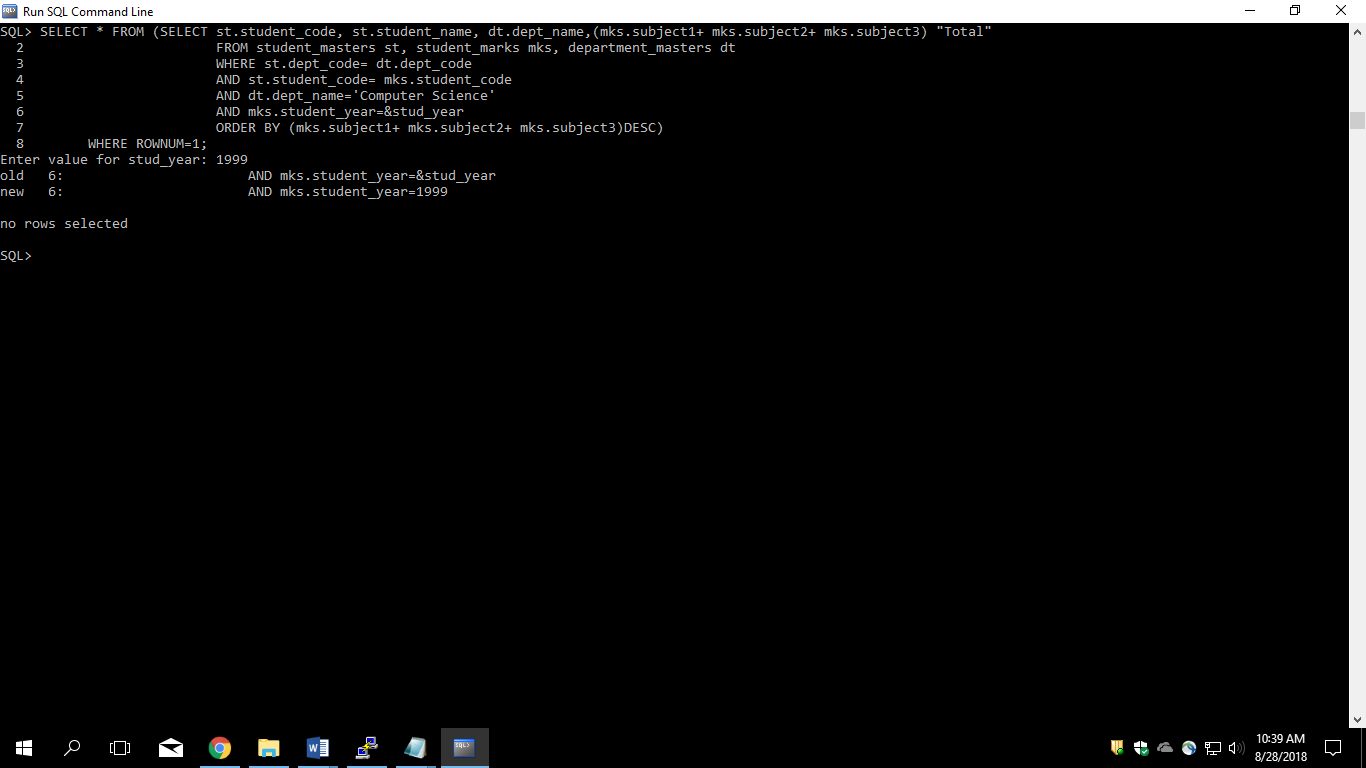


16. Create a query that will display the Staff Name, Department Name, and all the staff that work in the same department as a given staff. Give the column as appropriate label.

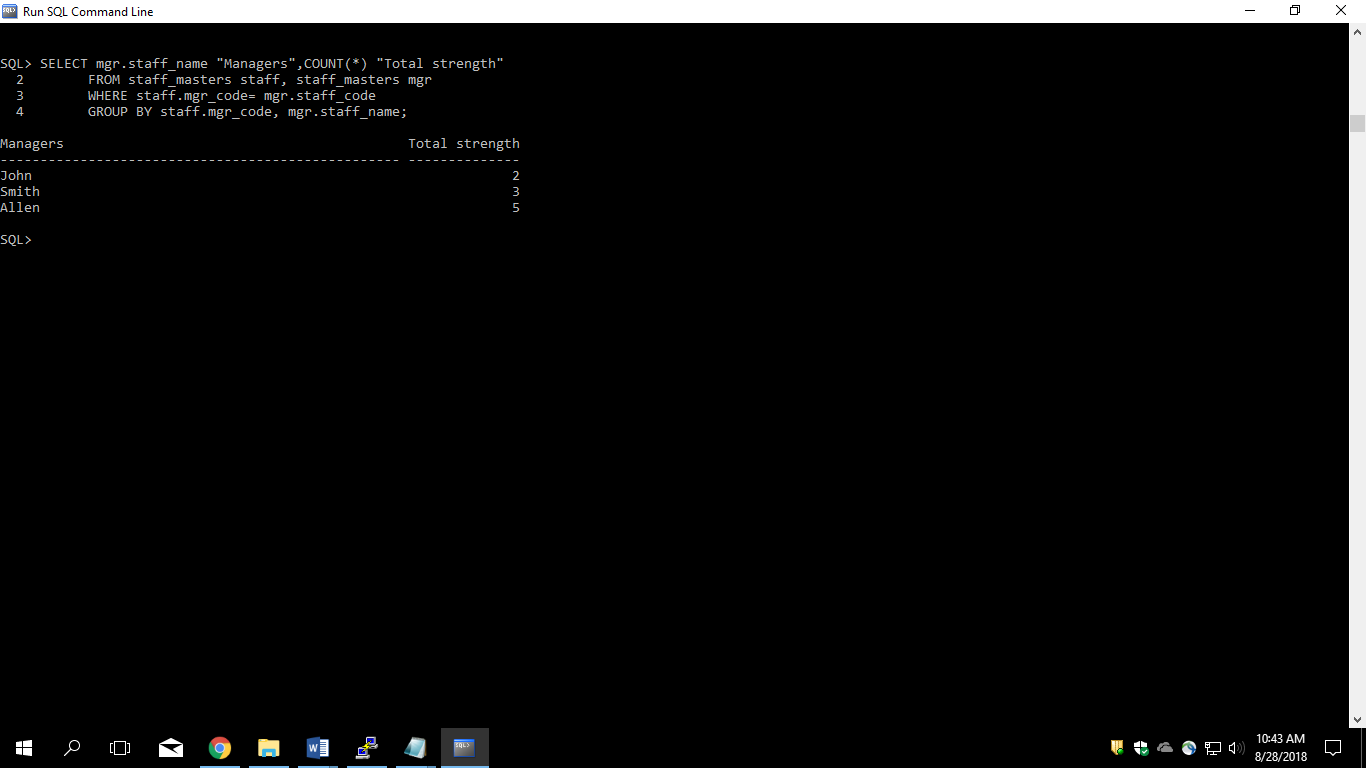




17. List the Student Code, Student Name for that student who got highest marks in all three subjects in Computer Science department for current year.

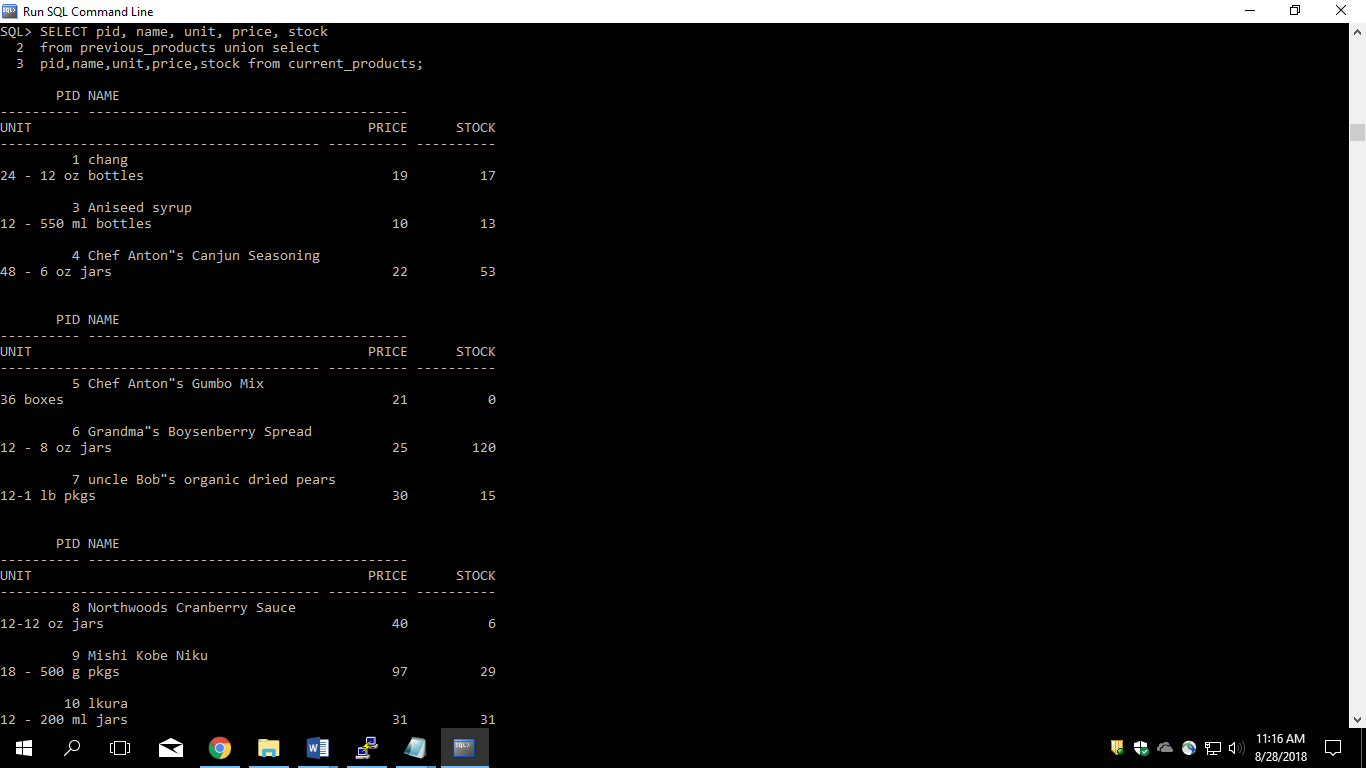


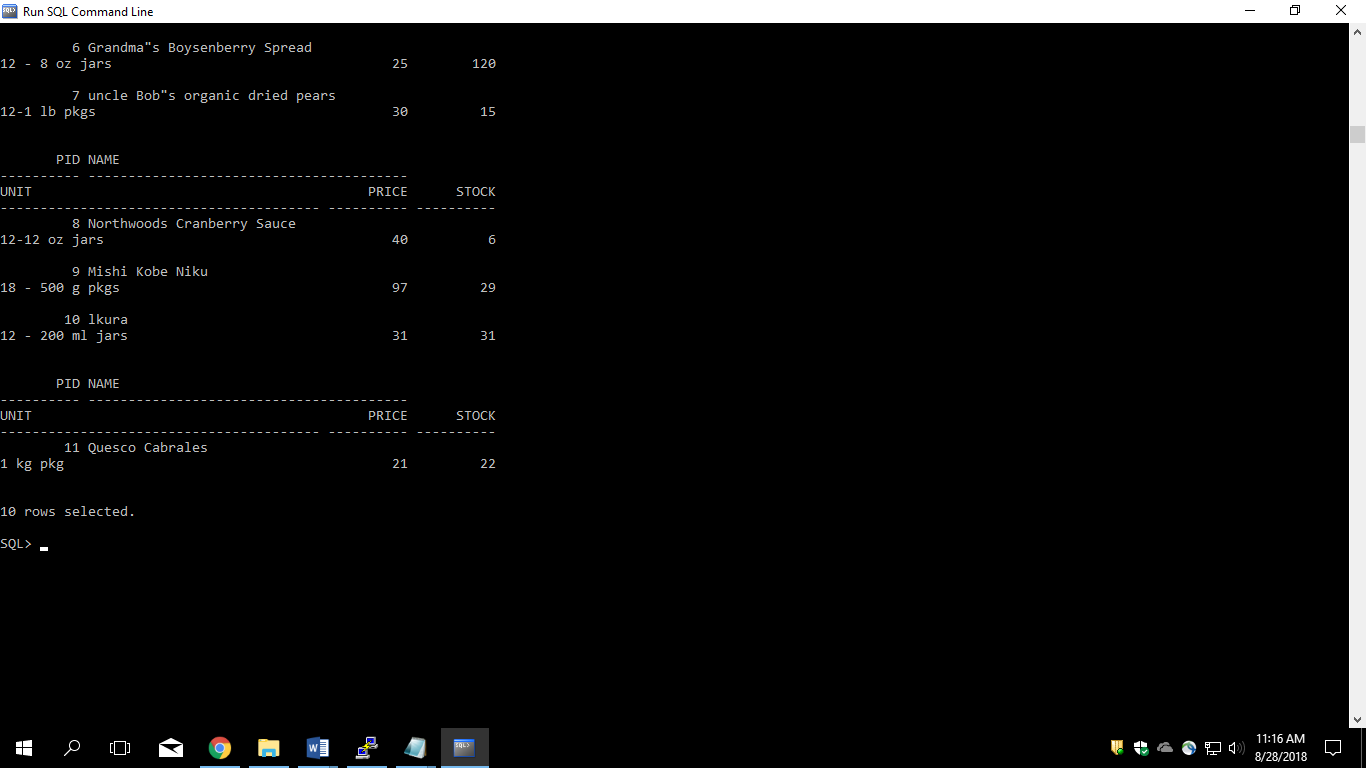
20.//Display the Manager Name and the total strength of his/her team.



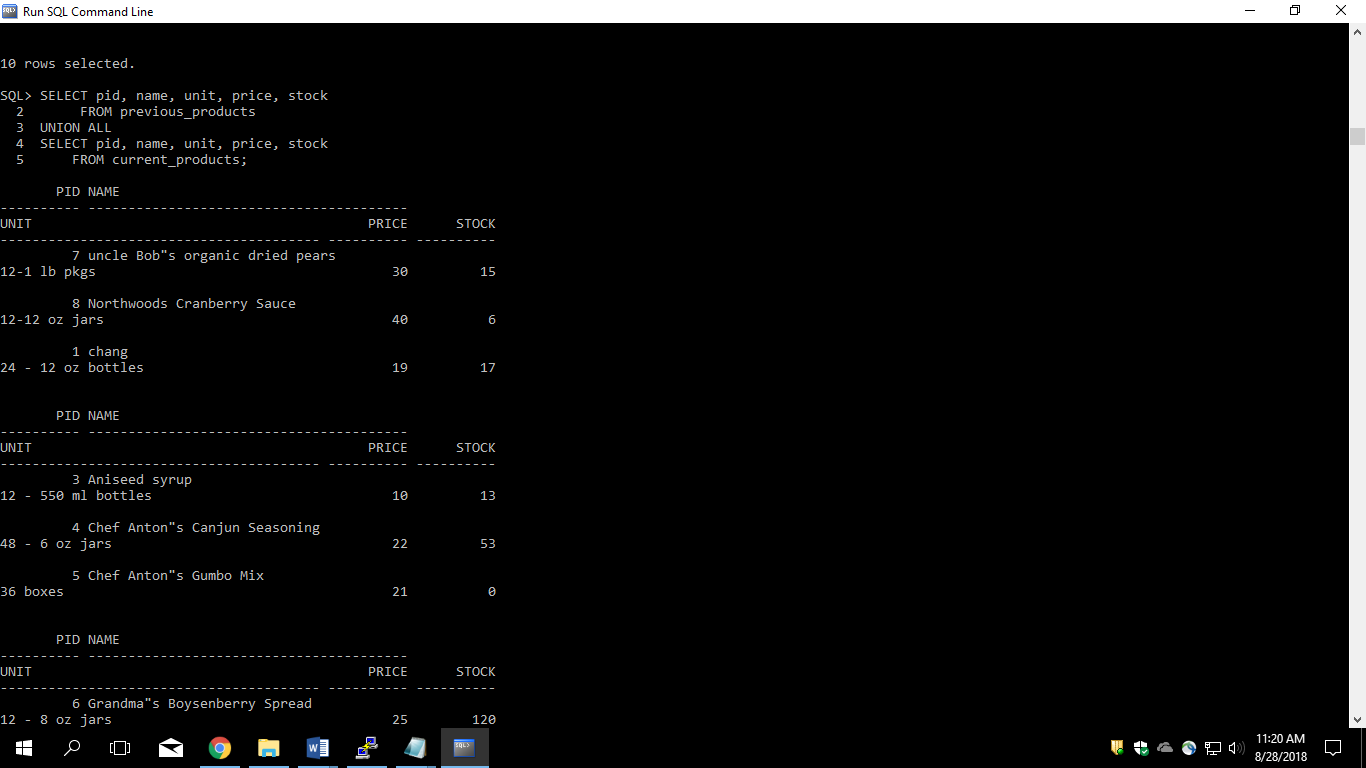
## . 3.2: Set Operators

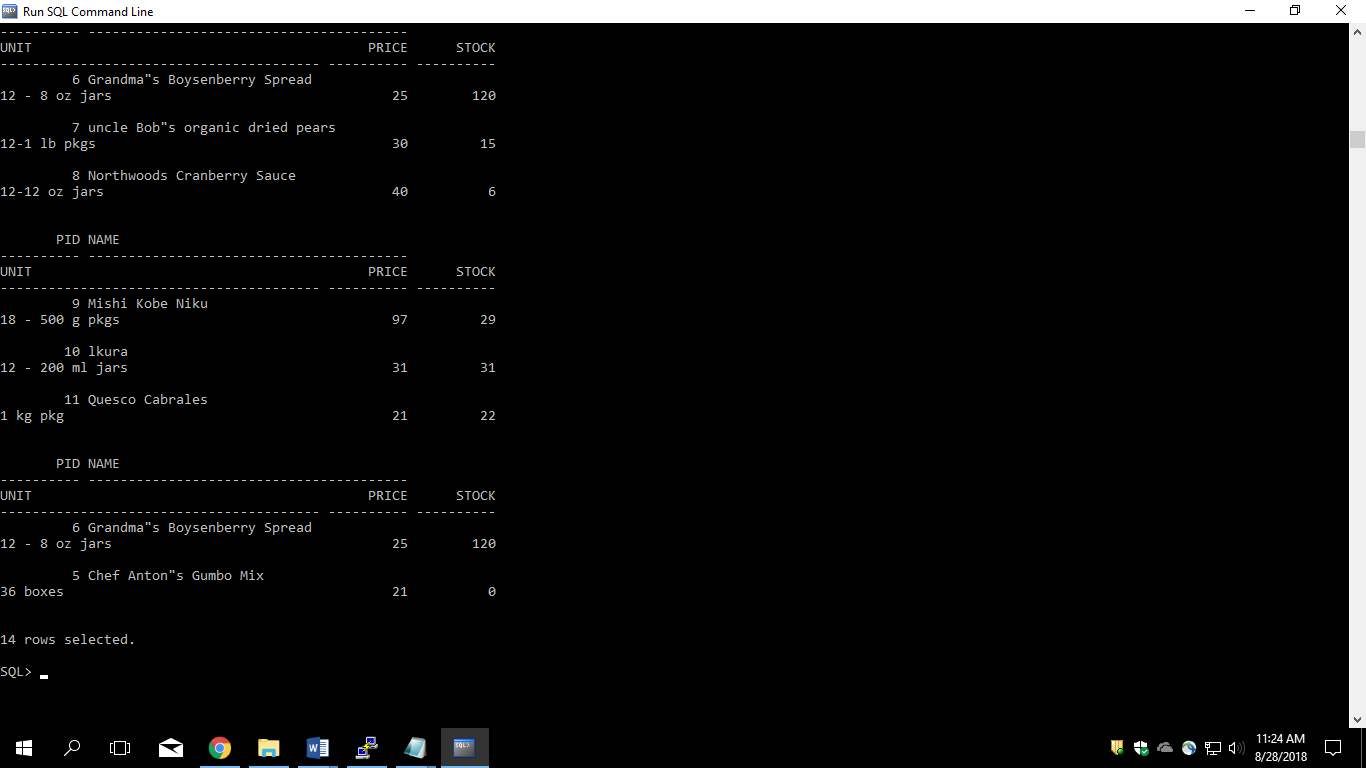
1. Get the details of all products irrespective of the fact whether they are in previous set or current set.



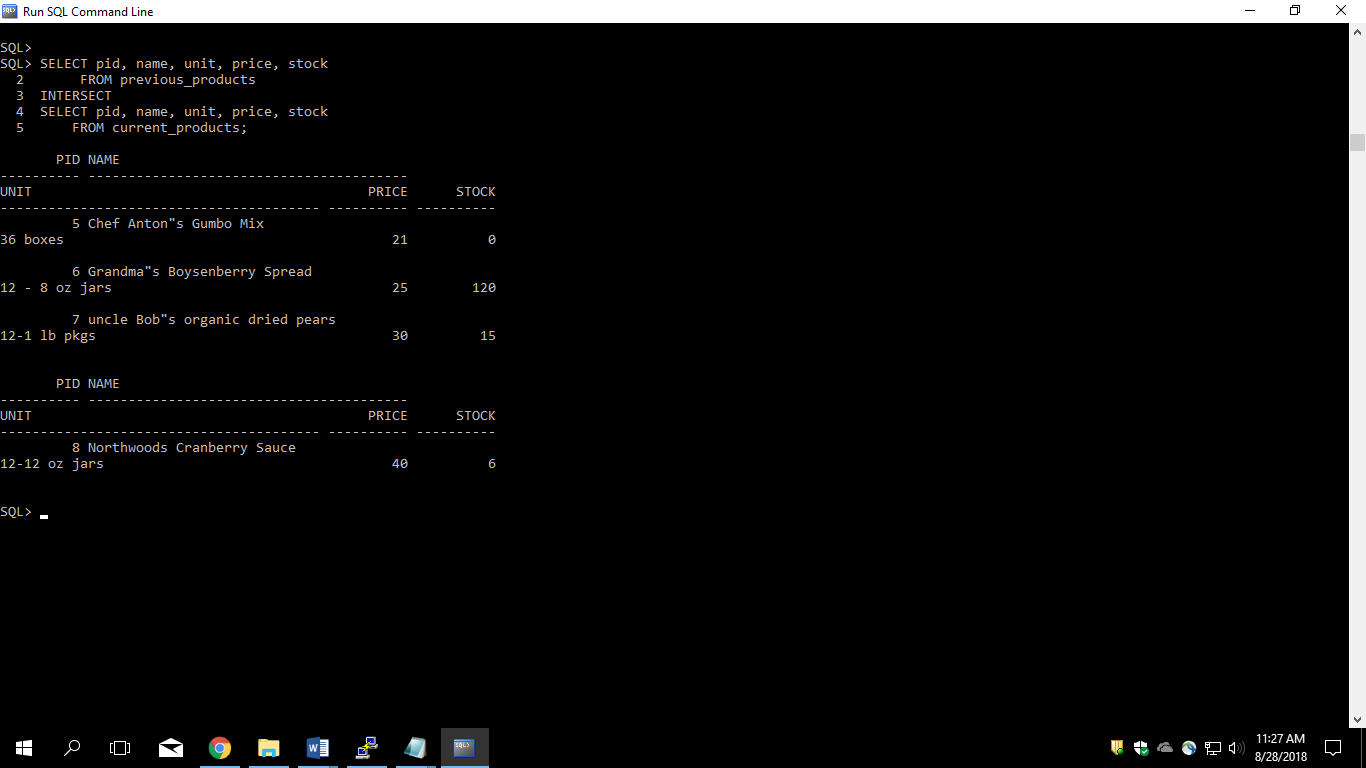


2. Get the details of all products along with the repetition of those that were present both in the previous and current sets.

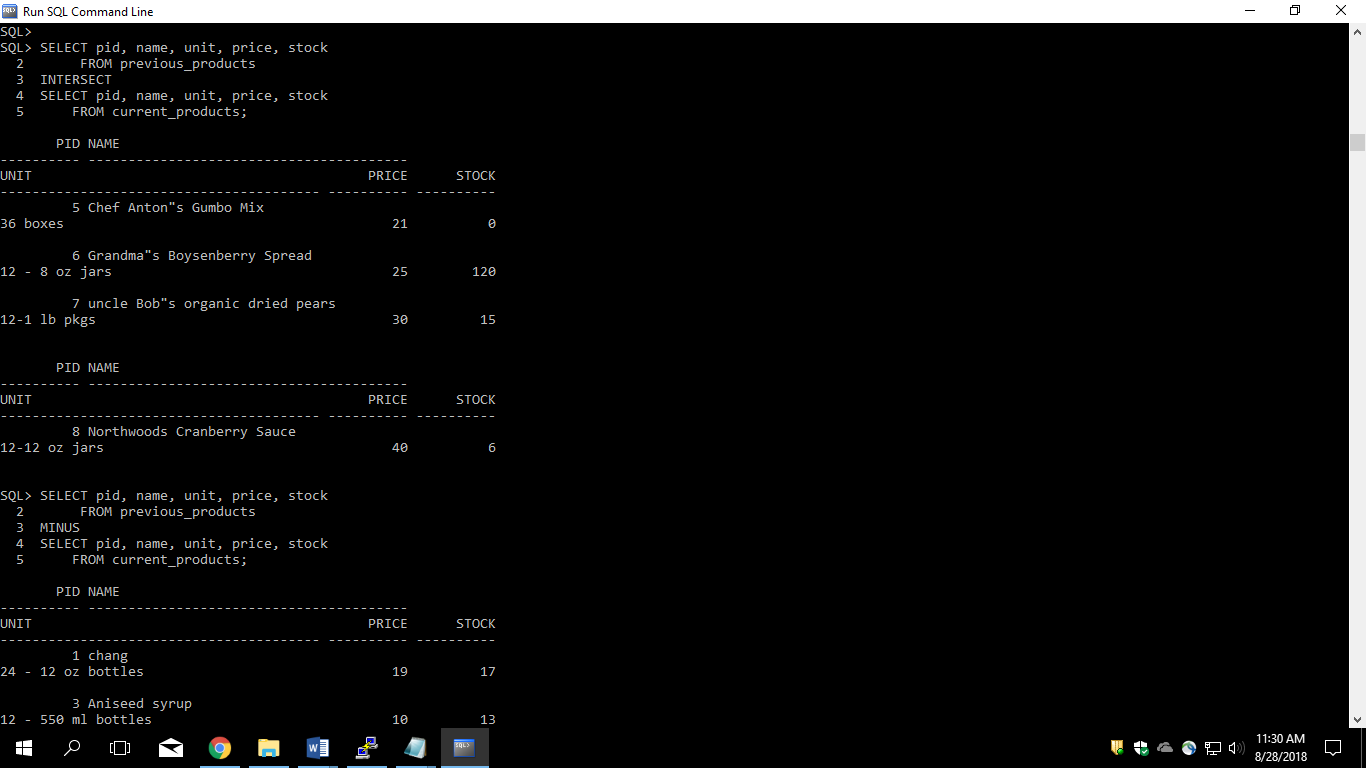


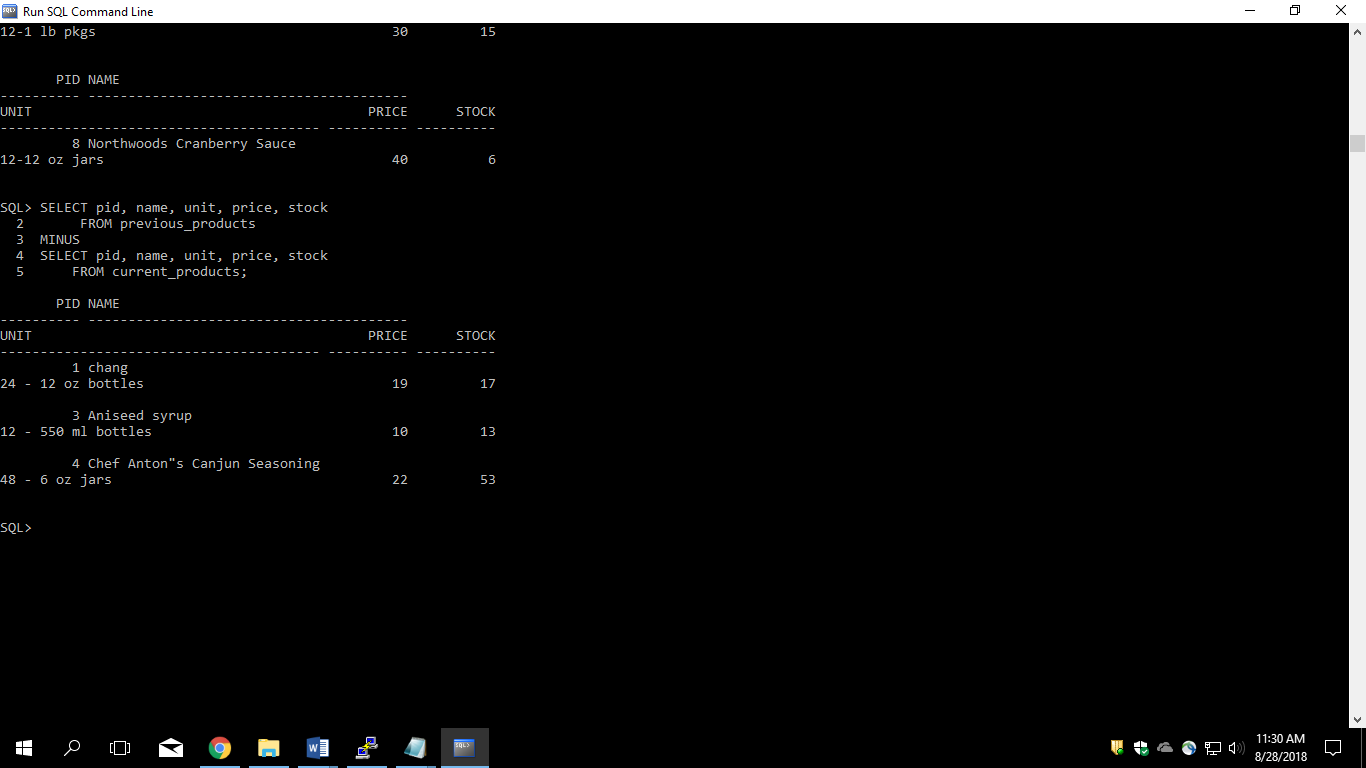


3. Get the details of only those products which were present in the previous set and are still continuing.



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





Lab4

1. Create the Customer table with the following columns.

Customerid Number(5)

CustomerName Number(10)

Address1 Varchar2(30)

Address2 Varchar2(30)

.CREATE TABLE

CUSTOMER

(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.

ALTER TABLE CUSTOMER

MODIFY

CUS\_NAME VARCHAR2(30);

ALTER TABLE CUSTOMER

RENAME

COLUMN CUS\_NAME TO CUSTOMER\_NAME;

ALTER TABLE CUSTOMER

MODIFY

CUSTOMER\_NAME VARCHAR2(30) NOT NULL;

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

Gender Varchar2(1)

Age Number(3)

PhoneNo Number(10)

Query:- ALTER TABLE CUSTOMER

ADD(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

Query:- INSERT INTO CUST\_TABLE VALUES (1000, 'ALLEN', '#115 CHICAGO', '#115 CHICAGO', 'M', 25, 7878776);

INSERT INTO CUST\_TABLE VALUES (1000, 'GEORGE', '#116 FRANCE', '#116 FRANCE', 'M', 25, 434524);

INSERT INTO CUST\_TABLE VALUES (1002, 'BECKER', '#114 NEW YORK', '#114 NEW YORK', 'M', 45, 431525);

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

Query:- ALTER TABLE CUST\_TABLE

ADD( CONSTRAINT CUSTID\_PRIM PRIMARY KEY(CUSTOMERID));

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

Query:- INSERT INTO CUST\_TABLE

VALUES(1002, 'JOHN', '#114 CHICAGO','#114 CHICAGO', 'M',45, 439525);

Oracle Srever gives the message "unique constraint violeted".

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

1:- 1002, Becker, #114 New York, #114 New york , M, 45, 431525

2:- 1003, Nanapatekar, #115 India, #115 India , M, 45, 431525

Query:- //Query for disabling constraint

ALTER TABLE CUST\_TABLE DISABLE CONSTRAINT CUSTID\_PRIM;

//Inserting record

INSERT INTO CUST\_TABLE VALUES (1002, 'BECKER', '#114 NEW YORK', '#114 NEW YORK', 'M', 45, 431525);

INSERT INTO CUST\_TABLE VALUES (1003, 'NANAPATEKAR', '#115 INDIA', '#115 INDIA', 'M', 45, 431525);

Query:- //Query for disabling constraint

ALTER TABLE CUST\_TABLE DISABLE CONSTRAINT CUSTID\_PRIM;

//Inserting record

INSERT INTO CUST\_TABLE VALUES (1002, 'BECKER', '#114 NEW YORK', '#114 NEW YORK', 'M', 45, 431525);

INSERT INTO CUST\_TABLE VALUES (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.

Query:- ALTER TABLE CUST\_TABLE ENABLE CONSTRAINT CUSTID\_PRIM;

Oracle srever gives the message "Can not validate primary key violeted".

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

1:-1002, Becker, #114 New York, #114 New york , M, 45, 431525, 15000.50

2:- 1003, Nanapatekar, #115 India, #115 India , M, 45, 431525, 20000.50

Query:- ALTER TABLE CUST\_TABLE DROP CONSTRAINT CUSTID\_PRIM;

//Inserting record

INSERT INTO CUST\_TABLE VALUES (1002, 'BECKER', '#114 NEW YORK', '#114 NEW YORK', 'M', 45, 431525);

INSERT INTO CUST\_TABLE VALUES (1003, 'NANAPATEKAR', '#115 INDIA', '#115 INDIA', 'M', 45, 431525);

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

Query:- TRUNCATE TABLE CUST\_TABLE;

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

Query:- ALTER TABLE CUST\_TABLE

ADD(EMAIL VARCHAR2(40));

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

Query:- ALTER TABLE CUST\_TABLE

DROP COLUMN EMAIL ;

13. Add a new column EmailId to Customer table.

Query:- ALTER TABLE CUST\_TABLE

ADD(EMAILID VARCHAR2(40));

14. Mark EmailId column as unused before dropping it.

Query:- ALTER TABLE CUST\_TABLE

SET UNUSED COLUMN EMAILID;

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

Query:- ALTER TABLE CUST\_TABLE

DROP UNUSED COLUMNS;

16. 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.

Query:- CREATE TABLE SUPPLIERS

(SUPPID NUMBER(5), SNAME VARCHAR2(40), ADDR1 VARCHAR2(40), ADDR2 VARCHAR2(40), CONTACTNO NUMBER(10));

17. 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)

Query:- //Drop table supplier

DROP TABLE SUPPLIERS;

//Create table customer master

CREATE TABLE CUSTOMER\_MASTER (CUSTOMERID NUMBER(5) CONSTRAINT CustId\_PK PRIMARY KEY,

CUSTOMERNAME VARCHAR2(30) NOT NULL, ADDRESS1 VARCHAR2(30) NOT NULL,

ADDRESS2 VARCHAR2(30),

GENDER VARCHAR2(1), AGE NUMBER(1), PHONENO NUMBER(10));

18. 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

Query:- CREATE TABLE ACCOUNTSMASTER(CUSTOMERID NUMBER(5),

ACCOUNTNUMBER NUMBER(10,2) CONSTRAINT ACC\_PK PRIMARY KEY, ACCOUNTTYPE CHAR(3),

LEDGERBALANCE NUMBER(10,2) NOT NULL);

//Create sequence

CREATE SEQUENCE ACNUM

INCREMENT BY 1

START WITH 1000;

INSERT INTO ACCOUNTSMASTER VALUES(101, ACNUM.NEXTVAL, 'S', 123);

INSERT INTO ACCOUNTSMASTER VALUES(102, ACNUM.NEXTVAL, 'S', 103);

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

Query:- ALTER TABLE ACCOUNTS\_MASTER

MODIFY(CUSTOMERID NUMBER(5) CONSTRAINT CUST\_ACC REFERENCES CUSTOMER\_MASTER(CUSTOMERID));

20. Insert the following rows to the CustomerMaster table:

1- 1000, Allen, #115 Chicago, #115 Chicago, M, 25, 7878776

2- 1001, George, #116 France, #116 France, M, 25, 434524

3- 1002, Becker, #114 New York, #114 New York, M, 45, 431525

Query:- INSERT INTO CUSTOMER\_MASTER VALUES(1000, 'ALLEN', '#115 CHICAGO', '#115 CHICAGO', 'M', 25, 7878776);

INSERT INTO CUSTOMER\_MASTER VALUES (1001, 'GEORGE', '#116 FRANCE', '#116 FRANCE', 'M', 25, 434524);

INSERT INTO CUSTOMER\_MASTER VALUES (1002, 'BECKER', '#114 NEW YORK', '#114 NEW YORK', 'M', 45, 431525);

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

Query:- ALTER TABLE ACCOUNTSMASTER

ADD( CONSTRAINT ACC\_TY CHECK(ACCOUNTTYPE ='NRI' OR ACCOUNTTYPE ='IND'));

22. Insert 5 rows into the AccountsMaster table:

Query:- INSERT INTO ACCOUNTSMASTER VALUES( 10101 ,1001, 'NRI', 45871);

INSERT INTO ACCOUNTSMASTER VALUES(1000,'8877','IND','50000.00');

INSERT INTO ACCOUNTSMASTER VALUES(1001,'8879','IND','25000.00');

INSERT INTO ACCOUNTSMASTER VALUES(1002,'8888','NRI','7000');

INSERT INTO ACCOUNTSMASTER VALUES(1003,'8899','NRI','30000');

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

Query:- ALTER TABLE ACCOUNTSMASTER

ADD CONSTRAINT BALANCE\_CHECK CHECK(LEDGERBALANCE>5000);

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

Query:- ALTER TABLE ACCOUNTSMASTER DROP CONSTRAINT CUST\_ACC;

ALTER TABLE ACCOUNTSMASTER ADD CONSTRAINT CUST\_ACC FOREIGN KEY (ACCOUNTNUMBER) REFERENCES CustomerMaster(CUSTOMERID) ON DELETE CASCADE;

25. Create Backup copy for the AccountsMaster table with the name âAccountDetailsâ.

Query:- CREATE TABLE ACCOUNTDETAILS

AS

SELECT \*

FROM ACCOUNTSMASTER;

26. Change the name of the AccountDetails table to âBackUpTableâ table.

Query:- RENAME ACCOUNTDETAILS TO BACKUPTABLE;

27. Create a view âAcc\_viewâ 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.

Query:- CREATE OR REPLACE VIEW ACC\_VIEW(CUSTOMERCODE,ACCOUNTHOLDERNAME,

ACCOUNTNUMBER,ACCOUNTTYPE,BALANCE)

AS

SELECT CUST.CUSTOMERID, CUST.CUSTOMERNAME,ACC.ACCOUNTNUMBER,

ACC.ACCOUNTTYPE, ACC.LEDGERBALANCE

FROM CUSTOMER\_MASTER CUST,ACCOUNTSMASTER ACC

WHERE ACC.CUSTOMERID=CUST.CUSTOMERID;

28. Create a view on AccountsMaster table with name vAccs\_Dtls. This view should list all customers whose AccountType is âINDâ and their balance

amount should not be less than 10000. Using this view any DML operation should not violate the view conditions.

Query:- CREATE OR REPLACE VIEW VACCS\_DTLS

AS

SELECT AC.CUSTOMERID,AC.ACCOUNTNUMBER, AC.ACCOUNTTYPE, AC.LEDGERBALANCE

FROM ACCOUNTSMASTER AC, CUSTOMER\_MASTER CU

WHERE AC.CUSTOMERID=CU.CUSTOMERID

AND AC.ACCOUNTTYPE='IND'

AND AC.LEDGERBALANCE>10000;

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

Query:-

CREATE OR REPLACE VIEW accsvw10 AS

SELECT \*

FROM ACCOUNTSMASTER WITH READ ONLY CONSTRAINT viewAccountMasters;

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

Query:-

SELECT DEPT\_CODE

FROM (SELECT DEPT\_CODE, MAX(STAFF\_SAL)

FROM STAFF\_MASTERS

GROUP BY DEPT\_CODE

ORDER BY MAX(STAFF\_SAL) DESC)

WHERE ROWNUM <= 1;

31. List the top two highest earning staff in each department.

Query:- SELECT STAFF\_NAME, MAX(STAFF\_SAL)

FROM STAFF\_MASTERS

WHERE ROWNUM <=2

GROUP BY DEPT\_CODE, STAFF\_NAME;

32. Create a Sequence with the name Seq\_Dept on Deptno column of Department\_Masters table. It should start from 40 and stop at 200.

Increment parameter for the sequence Seq\_Dept should be in step of 10.

Query:-

CREATE SEQUENCE SEQ\_DEPT

INCREMENT BY 10

START WITH 40

MAXVALUE 200;

INSERT INTO DEPARTMENT\_MASTERS VALUES(SEQ\_DEPT.NEXTVAL, 'MEDICAL SCIENCE');

33. Insert three sample rows by using the above sequence in Department\_Masters table.

Query:-

INSERT INTO DEPARTMENT\_MASTERS VALUES(SEQ\_DEPT.NEXTVAL,'CIVIL ENGINEERING');

INSERT INTO DEPARTMENT\_MASTERS VALUES(SEQ\_DEPT.NEXTVAL,'CHEMICAL ENGINEERING');

INSERT INTO DEPARTMENT\_MASTERS VALUES(SEQ\_DEPT.NEXTVAL,'NASA TECHNOLOGY');

34. Alter the above specified sequence with an increment by 5. Insert additional records and check what happens after the max value is reached.

Query:-

ALTER SEQUENCE SEQ\_DEPT

INCREMENT BY 5

MAXVALUE 200;

35. Drop the Seq\_Dept sequence.

Query:-

DROP SEQUENCE SEQ\_DEPT;

36. Create a Unique index with the name No\_Name on DeptNo and Dname of Department\_Masters table.

Query:-

CREATE INDEX No\_Name

ON department\_masters(dept\_code,dept\_name);

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

Query:-

SELECT INDEX\_NAME

FROM USER\_INDEXES

WHERE TABLE\_NAME = 'DEPARTMENT\_MASTERS';

38. Create synonym synEmp for the EMP table.

Query:-

`CREATE SYNONYM synEmp

FOR EMP;

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

Query:-

SELECT \* FROM synEmp;

40. Create a View that has fields for Item Master table (ItemNo, ItemDesc, Rate) using FORCE option.

Query:-

CREATE FORCE VIEW ITEN\_DETAILS(ItemNo, ItemDesc, Rate) AS

SELECT ITEMNO, ITENDESC, RATE

FROM ITEM\_MASTERS;

41. Note: Perform this after creating the Employee Table mentioned in the next Lab assignment.

Create Index on HireDate column and give the name as idx\_emp\_hiredate for this object.

Query:-

CREATE INDEX idx\_emp\_hiredate on EMP(HIREDATE);

42. Create a Sequence with the name Seq\_Emp on Empno column of Employee table. It should start from 1001.

Try to set Minimum value for this sequence which is less than / greater than 1001,

use the sequence to generate Empno while inserting records in Employee table and check the values generated.

Query:-

CREATE SEQUENCE Seq\_Emp

INCREMENT BY 1START WITH 1001;

Lab5

5.1 Create Employee table with same structure as EMP table.

CREATE TABLE employee

(empno NUMBER(4) NOT NULL,ename VARCHAR2(10),job VARCHAR2(50),

mgr NUMBER(4),hiredate DATE,sal NUMBER(7,2),comm NUMBER(7,2),deptno NUMBER(2));

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

INSERT INTO employee(empno,ename,sal,deptno)

VALUES('7369','SMITH',800,20);

INSERT INTO employee(empno,ename,sal,deptno)

VALUES('7499','ALLEN',1600,30);

INSERT INTO employee(empno,ename,sal,deptno)

VALUES('7521','WARD',1250,30);

INSERT INTO employee(empno,ename,sal,deptno)

VALUES('7566','JONES',2975,20);

INSERT INTO employee(empno,ename,sal,deptno)

VALUES('7654','MARTIN',1250,30);

INSERT INTO employee(empno,ename,sal,deptno)

VALUES('7698','BALKE',2850,30);

INSERT INTO employee(empno,ename,sal,deptno)

VALUES('7782','CLARK',2450,10);

INSERT INTO employee(empno,ename,sal,deptno)

VALUES('7788','SCOTT',3000,20);

INSERT INTO employee(empno,ename,sal,deptno)

VALUES('7839','KING',5000,10);

INSERT INTO employee(empno,ename,sal,deptno)

VALUES('7844','TURNER',1500,30);

INSERT INTO employee(empno,ename,sal,deptno)

VALUES('7876','ADMAS',1100,20);

INSERT INTO employee(empno,ename,sal,deptno)

VALUES('7900','JAMES',950,30);

INSERT INTO employee(empno,ename,sal,deptno)

VALUES('7902','FORD',3000,20);

INSERT INTO employee(empno,ename,sal,deptno)

VALUES('7934','MILLER',1300,10);

5.3 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.

UPDATE employee

SET deptno=(SELECT deptno FROM employee WHERE empno=7698)

WHERE empno=7788;

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

DELETE FROM employee where job='SALES';

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

UPDATE employee

SET deptno=(SELECT deptno FROM employee WHERE empno=7788)

WHERE empno=7698

5.6 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

INSERT INTO EMPLOYEE VALUES(&empno,'&ename','&job',&mgr,'&hiredate',&sal,&comm,&deptno);

or

INSERT INTO employee

VALUES('1000','ALLEN','clerk',1001,'12-jan-01',3000,2,10)

INSERT INTO employee

VALUES('1001','GEORGE','analyst',NULL,'08 sep 92',5000,0,10)

INSERT INTO employee

VALUES('1002','DBECKER','manager',1000,'4 NOV 92',2800,4,20)

INSERT INTO employee

VALUES('1003','BILL','clerk',1002,'4 NOV 92',3000,0,20)

Lab6

6.1 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.

INSERT INTO CUSTOMER\_TABLE VALUES(&CUSTOMERID,'&CUSTOMERNAME','&ADDRESS1','&ADDRESS2','&GENDER',&AGE,&PHONENO,&AMOUNT);

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

SAVEPOINT SP1;

6.3 Insert the below row in the Customer table.

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

INSERT INTO CUSTOMER\_TABLE VALUES(6003,'JOHN','#114 CHICAGO','#114 CHICAGO','M',45,439525,19000.60);

6.4 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.

ROLLBACK TO SP1;