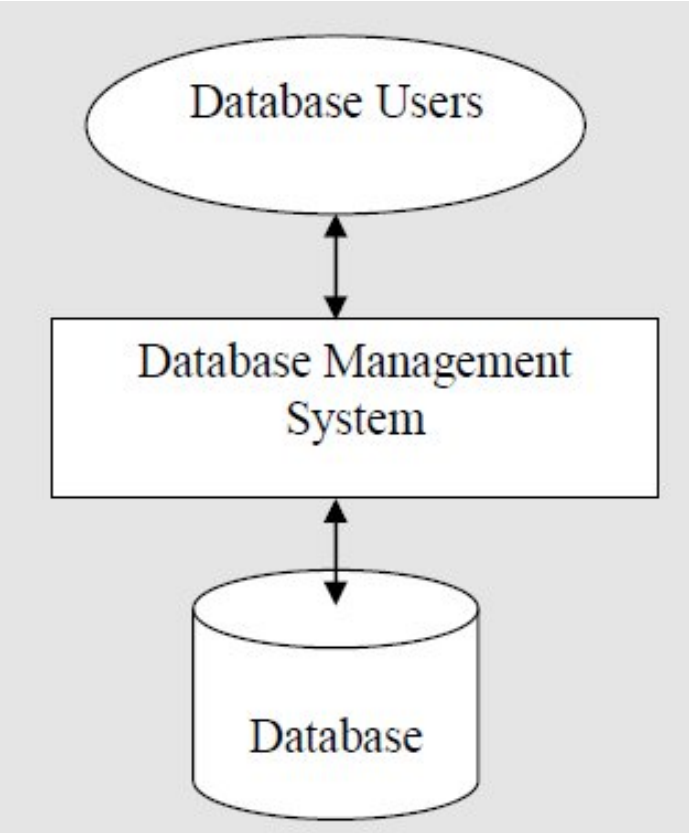
What is a database?

A ***database*** is an organized collection of interrelated data. In order to manage databases, we need Database Management Systems (DBMS). A DBMS is a program that stores, retrieves and modifies data in the database on user's request as illustrated in figure



What is a Relational Database?

A ***Relational database*** uses relations or two dimensional Tables to store Data. proposed by Dr. E .F. Codd

Database can be accessed by executing SQL (structured Query Language) statements.

Basic storage structure of RDBMS is TABLE which holds all the real world data. Example employees, departments and students.

Oracle 7 was just a relation DBMS where as oracle 8 & oracle 9i are Object relational database

Object relational database combine the features of Relational Databases and object oriented programming like inheritance(develop classes of your datatypes), polymorphism (shape table can be operated as Circle, Triangle and rectangle) and Encapsulation. Advantage of object relational database is its advance searching.

What is a Relational Database?

We will use oracle 11g in our labs.

Tables that we will use

EMP

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-NOV-81	5000		10
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20
7876	ADAMS	CLERK	7788	12-JAN-83	1100		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

There are total 8 columns(attributes)

There are total fourteen rows and Each row is identified by a primary key which is ***EMPNO***.

DEPT NO is a foreign key. Foreign key is a column or a set of columns that refers to a primary key or unique key of same or another table.

Tables that we will use

EMP

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-NOV-81	5000		10
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20
7876	ADAMS	CLERK	7788	12-JAN-83	1100		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

Intersection of each row and column is a ***cell*** & There can only be one value or no value(Null value) in a cell.

In employee table only salesman has a value in COMM(commission) field

Tables that we will use

DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

SALGRADE

GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

JOB_HISTORY

EMPNO	JOB	START_DATE	END_DATE
7698	ASSISTANT	04-MAR-80	30-APR-81
7654	RECEPTIONIST	13-JAN-80	09-SEP-80
7654	SALESMAN	10-SEP-80	20-SEP-81
7788	PROGRAMMER	13-FEB-80	03-DEC-82
7876	TYPIST	12-APR-80	13-NOV-81
7876	OPERATOR	15-NOV-81	11-JAN-83
7839	ANALYST	13-JUN-78	10-OCT-81

Guidelines for primary and foreign keys

1. No Duplicate values are allowed in primary key
2. Primary Keys generally cannot be changed
3. Foreign keys are based on data values and are purely logical, not physical pointers.
4. A foreign key value must match an existing primary key value or unique key value, or else be null

SQL(Structured Query Language)

Almost DBMS supports query language and SQL is the most influential commercially marketed product

SQL is a non procedural Language.

Non procedural Language is where you just specify what information you require instead of how to get information. Means in SQL we write a query to just get the information but as far as front end languages are concern like c#, C++ & visual basic we write code for how to retrieve the data.
Non procedural Language does not focus as much on individual processes needed to get to the conclusion but rather on ways to get to the conclusion itself

Scope

All type of Database activities by all types of users:

- System administrator
- Database administrator
- Security administrator
- Application Programmer
- Decision support system user
- End users

Language Components

Data Retrieval: Query language based on relation algebra & tuple relational calculus

Relation Algebra Example:
all loans of over $1200
σ amount > 1200 (loan)
loan number for each loan of an amount greater than $1200
π loan_number (σamount > 1200 (loan))
σ= Selection & π = projection

Tuple relational calculus Example:
{t/student(t) And t.percentage > 60}
Student(t) is the range of relation of t

U will cover above topics in detail in theory classes

Language Components

Data Definition Language(DDL):

DDL statements create, modify, and remove database objects such as tables, indexes, and users. Common **DDL** statements are CREATE, ALTER, and DROP.

Data Manipulation Language(DML):

DML statements used for inserting, deleting and updating data in a database

Embedded DML:

Designed for only general purpose programming languages such as cobol pascal & C language.

View definition:

The SQL DDL includes commands for Defining views for example:

```
CREATE VIEW view_name AS
SELECT column_name(s)
FROM table_name
WHERE condition
```

Language Components

Authorization:

SQL DDL includes commands for specifying access rights for views & relations

Integrity:

SQL DDL includes commands for specifying integrity constraints. Updates(DML) that violates integrity are not allowed

Transaction Control:

SQL includes commands for specifying the beginning and ending of transaction:

```
IF EXISTS (SELECT * FROM table_name WHERE id = ?)
  BEGIN
  --do what needs to be done if exists
  END
ELSE
  BEGIN
  --do what needs to be done if not
  END
END
```

It also includes explicit locking of data for concurrency like commit & rollback

Basic Data Retrieval

The basic structure of an SQL query consists of three clauses: SELECT, FROM and WHERE

Query Examples:

1. Select All columns from a table(DEPT)

   SELECT * FROM DEPT;

2. Select names of all jobs in a department

   SELECT DISTINCT JOB FROM EMP;

   The DISTINCT clause before a column name suppresses duplicate values

3. Select all employees whose salary is greater than 2200.

   *SELECT **
   FROM EMP
   WHERE SAL > 2200;

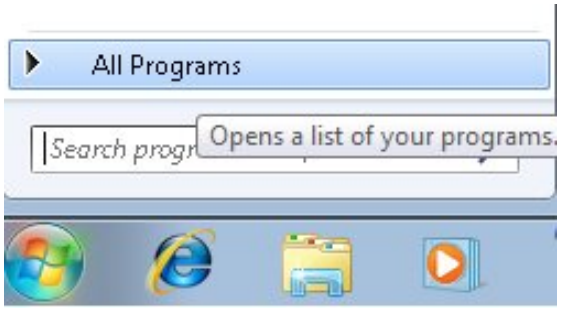
SQL * Plus

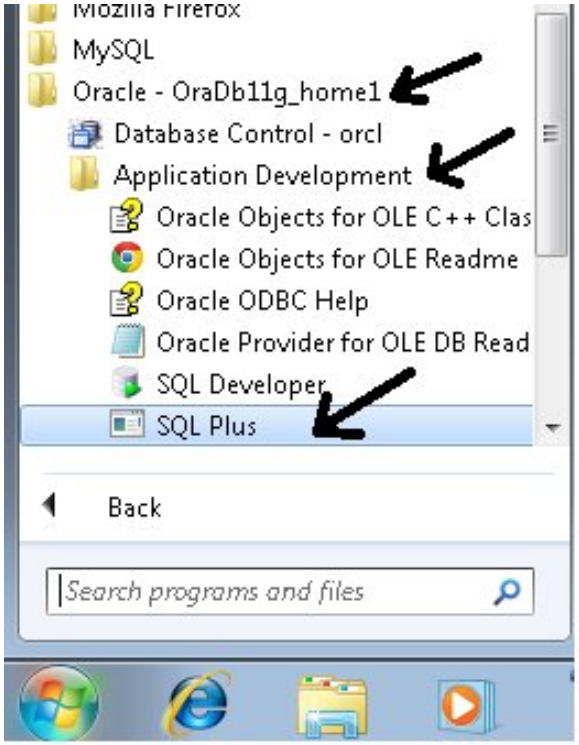
SQL * plus is oracle tool that identifies and submits SQL & PL/SQL statements to the server for execution.

PL/SQL is advance form of SQL that supports basic programming language features like Loops ,IF & Else statements.

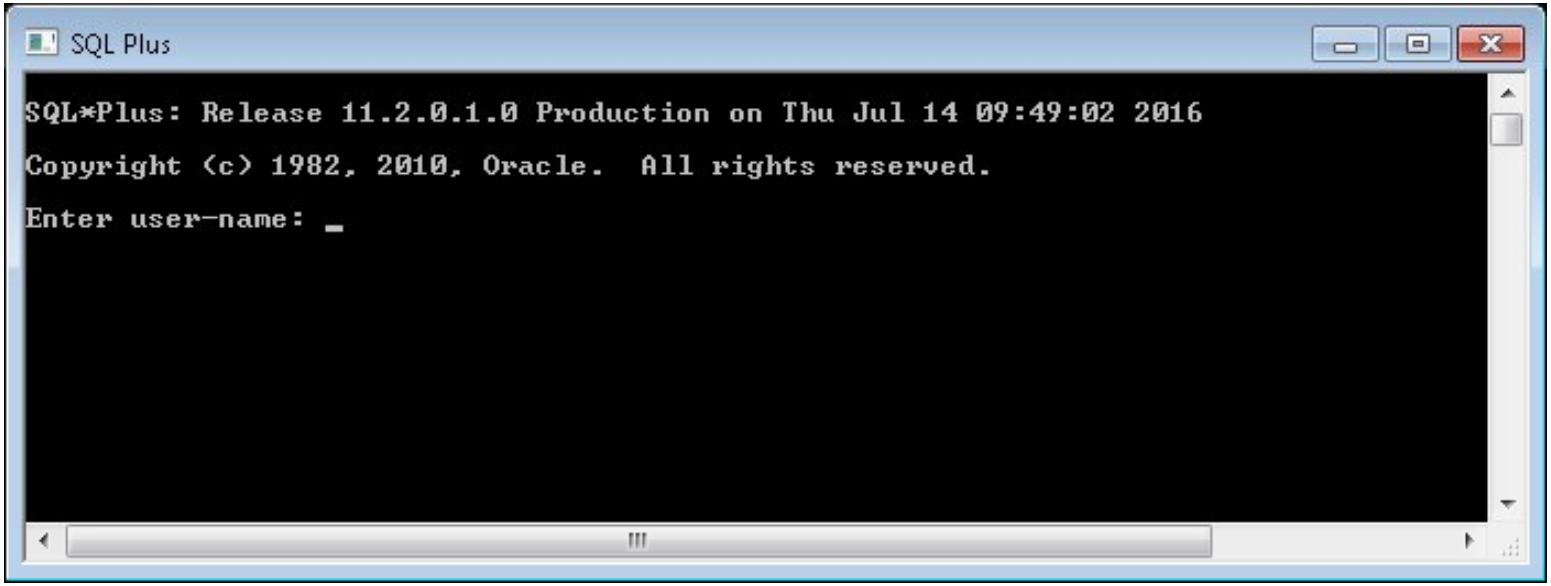
SQL * plus accepts query from editor as well as from SQL input from files

Lets start SQL plus and see its environment





SQL * Plus



We will enter from Scott(User ID)/Tiger(PWD)

Lets take a review of few basic SQL*Plus commands

Note *commands have few conventions like DESC[RIBE]=DESC means in bracket element shows you either syntax of same command*

SQL * Plus

Let take a review of few basic SQL*Plus commands

DESC[RIBE]: To display the structure of a table e.g. SQL> DESC EMP

```
SQL> describe emp
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPNO                                     NOT NULL NUMBER(4)
 ENAME                                              VARCHAR2(10)
 JOB                                                VARCHAR2(9)
 MGR                                                NUMBER(4)
 HIREDATE                                           DATE
 SAL                                                NUMBER(7,2)
 COMM                                               NUMBER(7,2)
 DEPTNO                                             NUMBER(2)

SQL> _
```

NUMBER(9)	7456124
NUMBER(9,2)	7456123.89

SQL * Plus

SAV[E] *filename*[.ext]: Saves current contents of SQL buffer to a file e.g.
SQL>SAVE D:\DATA\FINDSAL

```
SQL> select * from emp;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7924 MICHEAL    COOK            7811 26-JUL-81       9000                    30
      7345 SANIA      ENGINEER        7904 18-DEC-90       2340        600         30
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

16 rows selected.

SQL> save D:\DATA\FINDSAL
Created file D:\DATA\FINDSAL.sql
SQL> _
```

Note *if extension will not be mention then by default extention will be .sql*

SQL * Plus

GET *filename*[.ext]: Writes the contents of a previously saved file to the SQL buffer.
SQL> GET D:\DATA\FINDSAL

```
SQL> get D:\DATA\FINDSAL
  1* select * from emp
```

"@" Runs a previously saved command file e.g. SQL>@ *filename*

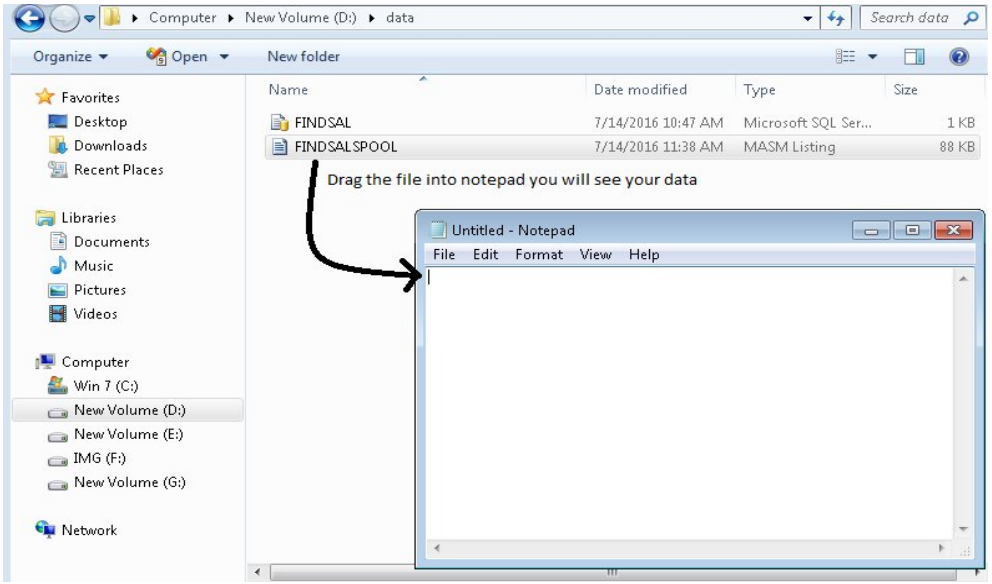
```
SQL> @D:\DATA\FINDSAL.sql

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7924 MICHEAL    COOK            7811 26-JUL-81       9000                    30
      7345 SANIA      ENGINEER        7904 18-DEC-90       2340        600         30
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

16 rows selected.
```

SQL * Plus

SPO[OL]: Stores query results in a file e.g. SQL>SPOOL *filename.ext*

```
SQL> spool D:\DATA\FINDSALSPOOL
SQL> select * from dept;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        50 MARKETING1     SAN DIEGO1
        60 MIS
        70 FINANCE
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON

7 rows selected.

SQL> spool off
```

To read file open the notepad file and place the respective created file into it



SQL * Plus

SPOOL OFF: Closes the spool file as we have seen above

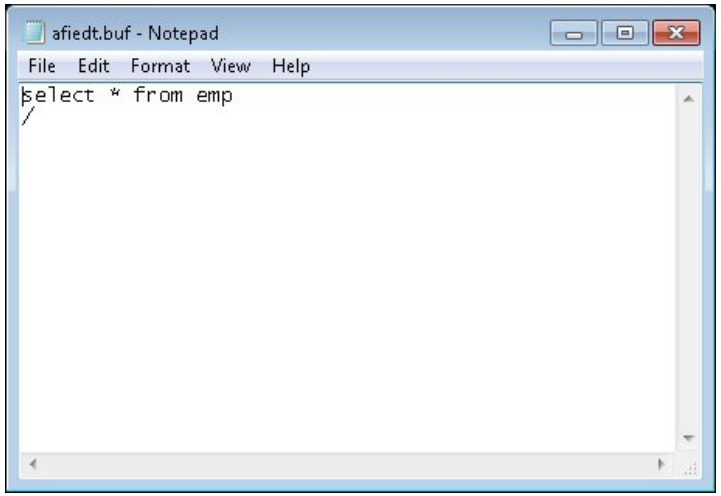
SPOOL OUT: Closes the spool file and sends the file results to the system printer(Default printer)

Difference between spool and save:

Save only saves last entered SQL command or the command that is in SQL buffer where as spool saves all commands and their outputs after writing **spool** command, unless **spool off** has been written

ED[IT]: Invokes the editor and saves the buffer contents to a file named afiedt.buf

```
SQL> ed
```



SQL * Plus

As soon as you will close and save the file you will find queries as follows

```
SQL> ed
Wrote file afiedt.buf

  1* select * from emp
SQL>
```

Now to execute the buffer query we simply need to type "/"

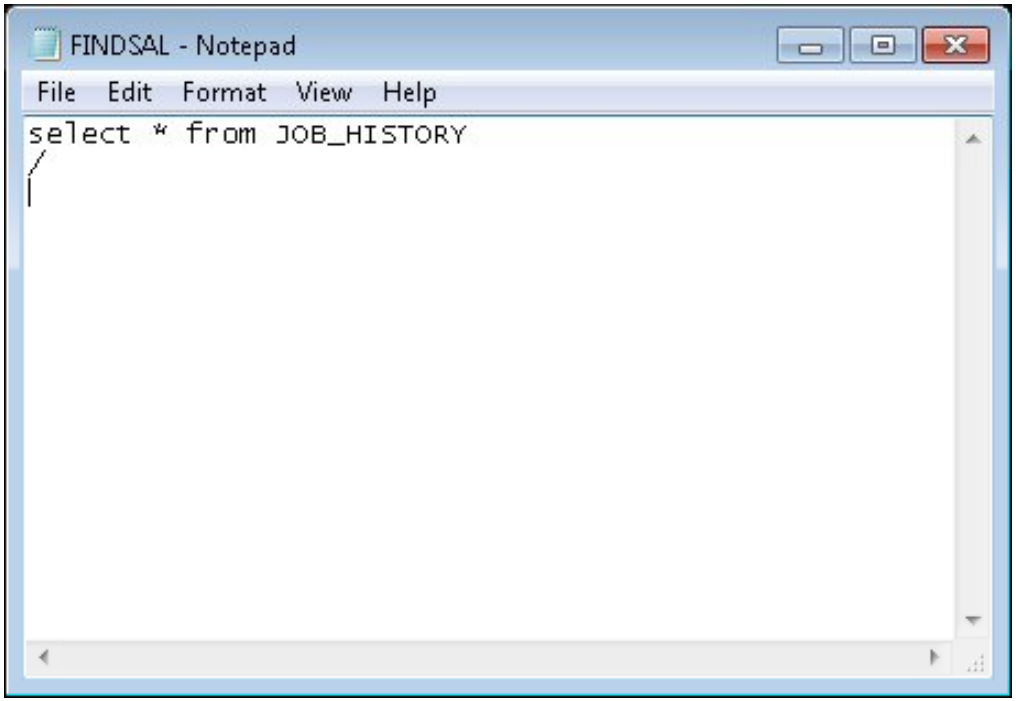
```
SQL> /

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7924 MICHEAL    COOK            7811 26-JUL-81       9000                    30
      7345 SANIA      ENGINEER        7904 18-DEC-90       2340        600         30
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

16 rows selected.
```

SQL * Plus

ED[IT] **[***filename*[.ext]**]:** Invokes editor to edit contents of a saved file

```
SQL> ed D:\data\FINDSAL
```

We changed EMP table to JOB_HISTORY



SQL * Plus

```
SQL> /

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7924 MICHEAL    COOK            7811 26-JUL-81       9000                    30
      7345 SANIA      ENGINEER        7904 18-DEC-90       2340        600         30
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK           7702 23-JAN-02       1300                    10

16 rows selected.
```

EXIT: leaves SQL * Plus

SQL * Plus

1. Write a query to display the employee number, name and salary of all managers

select empno, ename, sal from emp where job='MANAGER'

2. Write a query to display the name of all managers in department 20.

select ename,deptno from emp where hiredate>='1-JAN-83'

3. Write a query to display the name and department number of all employees who were

hired after 1982.

select ename from emp where job = 'MANAGER' and deptno=20

4. Display the one month salary of employees written as:

KING: 1 Month salary = 5000

Hint: use literal character strings

select ename ||': 1 Month Salary ='|| sal from emp