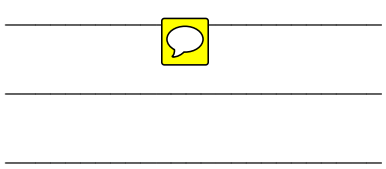
Lab1

Objectives:

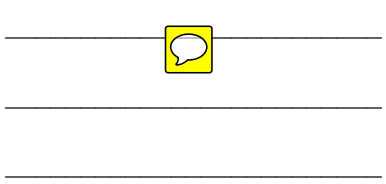
At the end of this lab, you should be able to

- Specify fields and columns of a table.
- To write conditional SQL query statement.

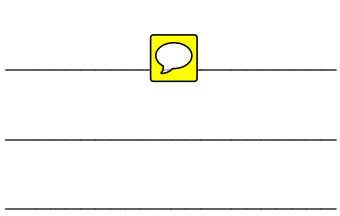
Data represents known facts that can be recorded and that have implicit meaning.
For example:



Database is a collection of related data. Example of database



Database management system (DBMS) is a collection of programs that enables users to create and maintain a databases. Example of DBMS



Manipulating a database includes :
querying the database to retrieve specific data, adding, deleting, or updating.

People who works with database systems:
1-Database administrator (DBA).
2-Database designers
3-Application programmer.
4-End users.

Prepared by : Dr. Saleh Alhazbi

Tables: every database composed of one or more tables, which store the database's data/information. Each table has its own unique name and consists of columns and rows. The columns called fields and the rows called records. A table has a specified number of columns, but can have any number of rows.

Record

A record is the collection of values for all the fields pertaining to one entity: i.e. a person, product, company, transaction, etc.

Field :

A field is an area (within a record) reserved for a specific piece of data. Examples: customer number, customer name, street address,
Fields are defined by:

- Field name
- Data type:Character, Numeric,Date, Logical, ..etc
- Field size

Example: assume you want to save data about students in school, specify some of possible fields that can be in this table, data type, size of each field.
Give two examples of records for that table.

Oracle: is an relational database management system (RDBMS) that provides database tools for storing and managing data. It also provides advanced tools to manage all types of data in web sites.

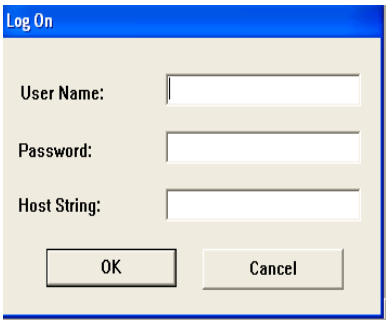
SQL is a command language for communication with the Oracle Server from any tool or application. When you enter a SQL statement, it is stored in a part of memory called the *SQL buffer* and remains there until you enter a new statement.

SQL*Plus is an Oracle tool that recognizes and submits SQL statements to the Oracle Server for execution and contains its own command language. Statements can be executed from the SQL prompt or from a script file.

Logging into SQL*Plus:

To log in through a Windows environment:

1. Click Start—>Programs—>Oracle home92—>Application Development-- > SQL Plus.
2. Fill in username**(scott)**, password(**tiger**) , and database.



Writing SQL Statements

- Within SQL*Plus, a SQL statement is entered at the SQL prompt, and the subsequent lines are numbered. This is called the *SQL buffer*. Only one statement can be current at any time within the buffer.
- SQL statements are not case sensitive, unless indicated.
- SQL statements can be entered on one or many lines.

Tables used in this course:

1- **EMP** table: which gives details of all the employees.
2- **DEPT** table: which gives details of all the departments.
3- **SALGRADE**: which gives details of salaries for various grades.

EMP Table

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	1400		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1800	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1450	500	30
7566	JONES	MANAGER	7839	02-APR-81	3175		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1450	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	3050		30
7782	CLARK	MANAGER	7566	09-JUN-81	2650		10
7788	SCOTT	ANALYST		19-APR-87	3200		20
7839	KING	PRESIDENT	7788	17-NOV-81	5200		10
7876	ADAMS	CLERK	7698	23-MAY-87	1300		20
7900	JAMES	CLERK	7566	03-DEC-81	1150	300	30
7902	FORD	ANALYST	7566	03-DEC-81	3200		20
7934	MILLER	CLERK	7782	23-JAN-82	1500		10

Displaying Table Structure

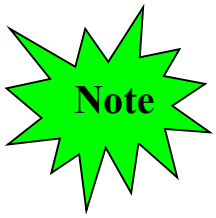
Use the SQL*Plus DESCRIBE command to display the structure of a table.

DESC[RIBE] ***tablename***

Example :

```
SQL> DESCRIBE dept;

Name           Null?    Type
-------------- -------- ------------
DEPTNO         NOT NULL NUMBER(2)
DNAME                   VARCHAR2(14)
LOC                     VARCHAR2(13)
```



NULL : indicates whether a column *must* contain data;
NOT NULL: indicates that a column must contain data

The data types are described in the following table:

Datatype	Description
NUMBER(*p*,*s*)	Number value having a maximum number of digits *p*, the number of digits to the right of the decimal point *s*
VARCHAR2(*s*)	Variable-length character value of maximum size *s*
DATE	Date and time value between January 1, 4712 B.C. and December 31, 9999 A.D.
CHAR(*s*)	Fixed-length character value of size *s*

Basic SELECT Statement

```
SELECT [DISTINCT]{*, column] alias[,...}
FROM   table;
```

Examples:

```
SQL> SELECT *
  2  FROM   dept;

SQL> SELECT deptno, loc
  2  FROM   dept;
```

Using Arithmetic Operators

```
SQL> SELECT ename, sal, sal+300
  2  FROM emp;
```

Exercise: Write SQL statement to display employee name and total of his income(salary +commission)

_____

_____

_____

What do you notice ??!!!

_____

Defining a Null Value

- A null is a value that is unavailable, unassigned, unknown, or inapplicable.
- A null is not the same as zero or a blank space.

```
SQL> SELECT ename, job, sal, comm
  2  FROM   emp;
```

Note: *some records do not have values in comm field.*

Duplicate Rows

SQL> SELECT deptno FROM emp;

Eliminating Duplicate Rows

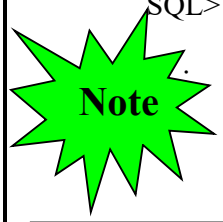
SQL> SELECT **DISTINCT** deptno FROM emp;

Limiting Rows Selected

SELECT	**[DISTINCT] {*| *column* [*alias*], ...}**
FROM	***table***
[WHERE	***condition(s)*]**;

Example:

SQL> SELECT ename, job, deptno FROM emp **WHERE ename ='JAMES'** ;



- Character strings and date values are enclosed in single quotation marks.
- Character values are case sensitive and date values are format sensitive.
- The default date format is DD-MON-YY.

Comparison Operators

Operator	Meaning
=	Equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
<>	Not equal to

Examples:

```
SQL> SELECT ename, sal FROM emp
  WHERE sal BETWEEN 1000 AND 1500;

SQL> SELECT empno, ename, sal, mgr FROM emp
  WHERE mgr IN (7902, 7566, 7788);

SQL> SELECT ename FROM emp
  WHERE ename LIKE 'S%';

SQL> SELECT ename, mgr FROM emp
  WHERE mgr IS NULL;
```

Logical Operators

Operator	Meaning
AND	Returns TRUE if *both* component conditions are TRUE
OR	
NOT	Returns TRUE if *either* component condition is TRUE
Operator	Returns TRUE if the following condition is FALSE

Examples:

```
SQL> SELECT empno, ename, job, sal
  2  FROM   emp
  3  WHERE  sal>=1100
  4  AND    job='CLERK';

SQL> SELECT empno, ename, job, sal
  2  FROM   emp
  3  WHERE  sal>=1100
  4  OR     job='CLERK';

SQL> SELECT ename, job
  2  FROM   emp
  3  WHERE  job NOT IN ('CLERK','MANAGER','ANALYST');
```

ORDER BY Clause

- Sort rows with the ORDER BY clause
- ASC: ascending order, default
- DESC: descending order
- The ORDER BY clause comes last in the SELECT statement.

Examples:

```
SQL> SELECT          ename, job, deptno, hiredate
  2  FROM       emp
  3  ORDER BY hiredate;

SQL> SELECT   ename, job, deptno, hiredate
  2  FROM       emp
  3  ORDER BY hiredate DESC;
```

Sorting by Multiple Columns

```
SQL> SELECT          ename, deptno, sal  FROM     emp
 ORDER BY    deptno, sal DESC;
```

Practice

1. Show the structure of the EMP table. Create a query to display the name, job, hire date, and employee number for each employee.

2. Create a query to display the name and salary of employees earning more than $2850.

3. Create a query to display unique jobs from the Emp table;

4. Create a query to display the name and salary for all employees whose salary is not in the range of $1500 and $2850.

5. Display the name and job title of all employees who do not have a manager.

6. Display the name, salary, and commission for all employees who earn commissions. Sort data in descending order of salary and commissions.

7. Display the names of all employees where the third letter of their name is an *A*.