

UIT University

**Department of Engineering and Technology
CET 224 Database Application**

Lab#1

Objective:

1. Overview of the features of SQL and SQL*Plus.

Name of Student: _____

Roll No: _____ **Sec.** _____

Date of Experiment: _____

.....

Marks Obtained/Remarks: _____

Signature: _____

THEORY

What is a database?

Every organization needs information for making effective decisions. A library keeps a list of books, members, due dates and fines. A company needs to save information about employees, departments and salaries. The collection of facts about different entities in an organization is called *data*.

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 1.1 below.

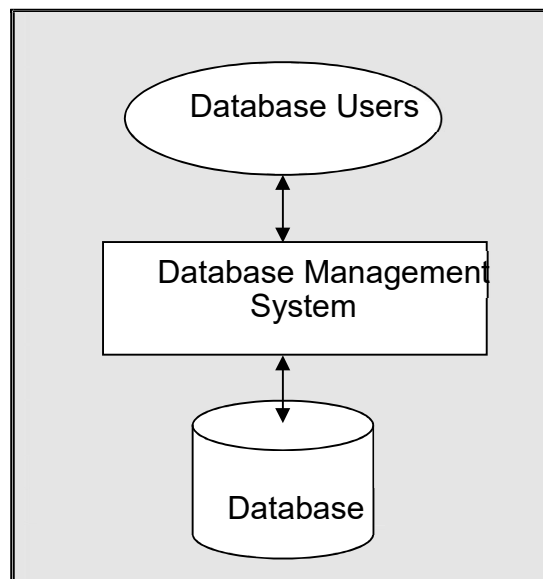


Figure 1.1: Users access data in a database through DBMS

Relational Database Concept

A *relational database* uses *relations* or *two-dimensional tables* to store data. Such a database can be accessed and modified by executing *Structured Query Language (SQL)* statements. **Dr. E. F. Codd** proposed the relational model for database systems in 1970. Relational Database Management Systems (RDBMS) soon became very popular, especially for their *ease of use* and *flexibility in structure*. In addition, a number of innovative vendors such as Oracle, supplemented the RDBMS with a suite of powerful application development and user products, providing a total solution.

A table is the basic storage structure of an RDBMS. A table holds all the data necessary about something in the real world – for example, employees, invoices, or customers.

Oracle 7 is a relational database management system whereas *Oracle 8* and *Oracle 9i* are object relational database management systems. *Oracle 9i* includes everything needed to develop, deploy and manage internet applications. There are two products *Oracle 9i Application Server* and *Oracle*

Lab No 1

9i Database that provide a complete and simple infrastructure for internet applications. The Oracle 9i application server (Oracle *9iAS*) runs all your applications. The Oracle 9i database stores all your data.

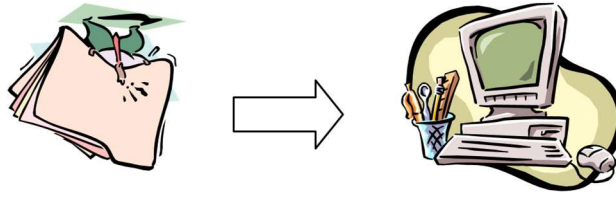


Figure 1.2: Databases allow compact and efficient storage of data

The table 1.1 below shows the contents of the EMP table or relation that stores data about employees presently working in an organization.

- The table has eight columns namely EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO storing the different attributes of an employee.
- The table has fourteen rows each representing all data that is required for a particular employee. Each row in a table should be identified by a **primary key**, which allows no duplicate rows. In table 1.1, EMPNO is the primary key where as in table 1.2, DEPTNO is the primary key.
- A **foreign key** is a column or a set of columns that refers to a primary key or a unique key in the same table or another table. In EMP table, DEPTNO is the foreign key.
- A field can be found at the intersection of a row and a column. There can be only one value in it.
- A field may have no value in it. This is called *null value*. In the EMP table, only employees who have a role of *salesman* have a value in the COMM (Commission) field.

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

Table 1.1

The table DEPT below contains data about different departments in an organization and SALGRADE gives details of salaries for various grades. The table JOB_HISTORY stores data about the job history of employees.

DEPT

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

Table 1.2

SALGRADE

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

Table 1.3

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

Table 1.4

Relating Multiple Tables

Each table contains data that describes exactly one entity. For example, the EMP table contains information about employees. Since data about different entities is stored in different tables, it may be needed to combine two or more tables to answer a particular question. For example, someone may want to know the location of the department where an employee works. In this scenario, information is needed from both the EMP and DEPT table. An RDBMS enables to relate the data in one table to the data in another table by using the foreign keys.

Guidelines for Primary and Foreign Keys

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

SQL (Structured Query Language)

Most commercial database management systems support a query language, SQL, which is the most influential commercially marketed product. SQL is a nonprocedural language: you specify *what* information you require, rather than *how* to get it. In other words, SQL does not require you to specify the access methods to the data. As a result, it doesn't provide traditional programming structures. Instead, it is a language for specifying the operations at an unusually high level. The details of the implementation are left to the DBMS.

In Oracle, SQL is a language for communication with the Oracle Server from any tool or application. Oracle SQL has many extensions. Although we refer to the SQL language as a *query language*, it contains many other capabilities besides querying a database. It includes features for defining the structure of the data, for inserting and modifying data in the database, and for specifying security constraints.

It has following advantages:-

- Efficient
- Easy to learn and use
- Functionally complete (SQL allows to define, retrieve, and manipulate data in the tables.)

Background of SQL

The SQL language, originally called SEQUEL, was developed by IBM as part of the extensive work on the relational model in 1970s at their San Jose Research Laboratory, California. In order to test the viability of implementing the relational model in a DBMS, workers at the San Jose Research Laboratory undertook the project of the development of a prototype RDBMS named *System R*. This project took place from 1974 to 1979. The System R project led to the development of SEQUEL (Structured English Query Language) which was later renamed as SQL. Because System R was well received at the user sites where it was installed, other vendors began developing relational products that used SQL. In the late 1970s, the database management system *Oracle* was produced by what is now called the Oracle Corporation and was probably the first commercial implementation of a relational DBMS based on SQL. ORACLE is now available in mainframe, client-server and PC-based platforms for many operating systems including DOS, OS/2, various UNIX operating systems, Windows, VAX/VMS and MVS. Numerous RDBMS products now support the SQL language.

In 1986, the American National Standards Institute (ANSI) and the International Standards Organization (ISO) published an SQL standard, called SQL-86. IBM published its own corporate SQL standard, the *Systems Application Architecture Database Interface* (SAASQL) in 1987. An extended standard for SQL, SQL-89, was published in 1989. The ISO and ANSI committees created SQL-92 which was a more extensive expansion of SQL-86. SQL92 which was in fact a first major revision of the SQL standard is sometimes referred to as SQL2. It was not until 1999 that the next release of the standard was formalized, commonly referred to as SQL3. This release contains additional features that support object-oriented data management.

Scope

SQL is used for all types of database activities by all types of users including:

- System administrators
- Database administrators
- Security administrators
- Application programmers
- Decision support system personnel
- Many other types of end users

Language Components

The SQL language has several parts: -

Data Retrieval: The SQL includes a query language based on both the relational algebra and the tuple relational calculus.

Data-definition language (DDL): The SQL DDL provides commands for defining relation schemas, deleting relations, creating indices, and modifying relation schemas.

Interactive data-manipulation language (DML): It includes commands to insert tuples into, delete tuples from, and to modify tuples in the database.

Embedded DML: The embedded form of SQL is designed for use within general purpose programming languages, such as Cobol, Pascal and C.

View definition: The SQL DDL includes commands for defining views.

Authorization: The SQL DDL includes commands for specifying access rights for relations and views.

Integrity: The SQL DDL includes commands for specifying integrity constraints that the data stored in the database must satisfy. Updates that violate integrity constraints are disallowed.

Transaction control: SQL includes commands for specifying the beginning and ending of transaction. Several implementations also allow explicit locking of data for concurrency control.

Basic Data Retrieval

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

```
SELECT * | {[DISTINCT] column | expression [alias], ...}  
FROM table  
[WHERE condition];
```

Note: The keywords and clauses enclosed in square brackets are *optional*.

Examples

- i. Selecting all columns from a table

```
SELECT *  
FROM DEPT;
```

Lab No 1

ii. To select names of all jobs in a department, use

```
SELECT DISTINCT JOB  
FROM EMP;
```

Note: The DISTINCT clause before a column name suppresses duplicate values

iii. To select all employees whose salary is greater than 2200.

```
SELECT *  
FROM EMP  
WHERE SAL > 2200;
```

iv. To display the name and department number of employees who were hired before 12th May, 1981.

```
SELECT ENAME, DEPTNO  
FROM EMP  
WHERE HIREDATE < '12-MAY-1981';
```

v. To display the name and job of employees using literal character strings and concatenation operators.

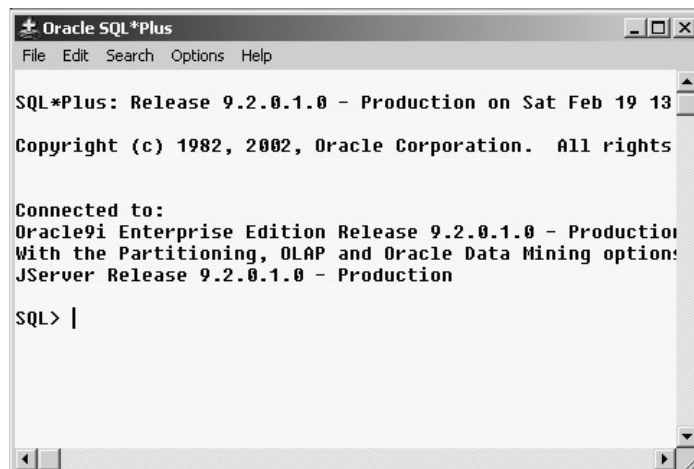
```
SELECT ENAME || ' is a ' || JOB  
AS "Employee Details"  
FROM EMP;
```

vi. Retrieving data from multiple tables: To select employee name, job and department name,

```
SELECT E.ENAME, E.JOB, D.DNAME  
FROM EMP E, DEPT D  
WHERE E.DEPTNO = D.DEPTNO;
```

SQL*Plus

SQL*Plus is an Oracle tool that recognizes and submits SQL and PL/SQL statements to the server for execution, and contains its own command language. It accepts *ad hoc* query statements as well as SQL input from files. It provides a line editor for modifying SQL statements and formats query results into a basic report. The SQL*Plus environment is shown below in figure 1.3.



In contrast to SQL statements, SQL*Plus commands do not allow the manipulation of values in the database and are not stored in the SQL buffer. Commands are entered one line at a time and have a dash (-) as a continuation character if the command is longer than one line. It uses commands to format data retrieved through SQL statements. SQL*Plus commands can be abbreviated, whereas SQL statements cannot. A few commands of SQL*Plus are as follows: -

DESC[RIBE]: To display the structure of a table

e.g. SQL> DESC EMP

SAV[E] *filename* [.ext]: Saves current contents of SQL buffer to a file

e.g. SQL>SAVE

D:\DATA\FINDSAL

GET *filename* [.ext]: Writes the contents of a previously saved file to the SQL buffer.

The default extension for the file is .sql. e.g. SQL> GET D:\DATA\FINDSAL @: Runs a previously saved command file e.g. SQL>@ *filename*

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

SPOOL OFF: Closes the spool file

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

ED[IT]: Invokes the editor and saves the buffer contents to a file named afiedt.buf **ED[IT]** *filename* [.ext]: Invokes editor to edit contents of a saved file

EXIT: Leaves SQL*Plus

Logging in to SQL*Plus

To log in through a windows environment:

- Click Start/Programs/Oracle for Windows/SQL*Plus
- Fill in username, password, and database in the window shown in figure 1.4.

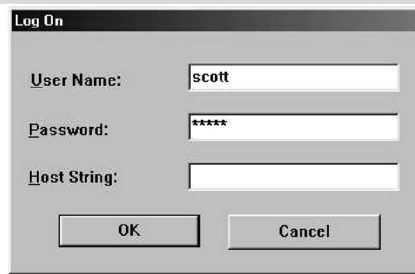


Figure 1.4

EXERCISES

1. What are relational databases? Describe, with the help of an example, how an RDBMS enables to relate data in one table to the data in another table.

2. Why SQL is called a nonprocedural language?

3. Write down a brief history of SQL and explain why its standardization was necessary.

[illegible]

4. Write a query to display the employee number, name and salary of all managers.
5. Write a query to display the name and department number of all employees with salary incremented by 300 also apply aliasing.
6. Write a query to display the name and job title of employee using concatenation.

* * * * *