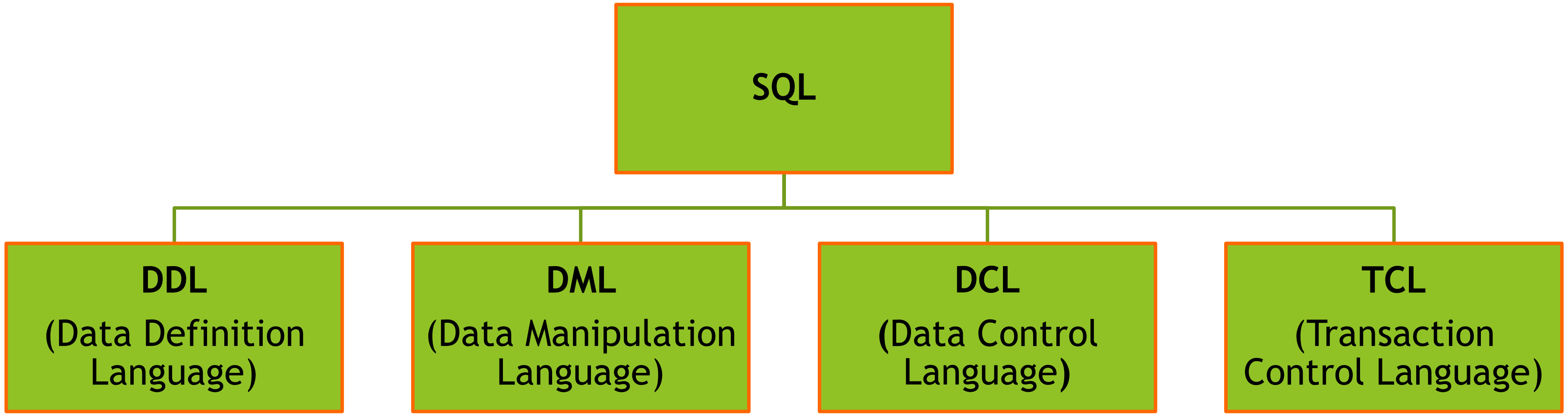
Data Management in RDBMS using SQL

Contents

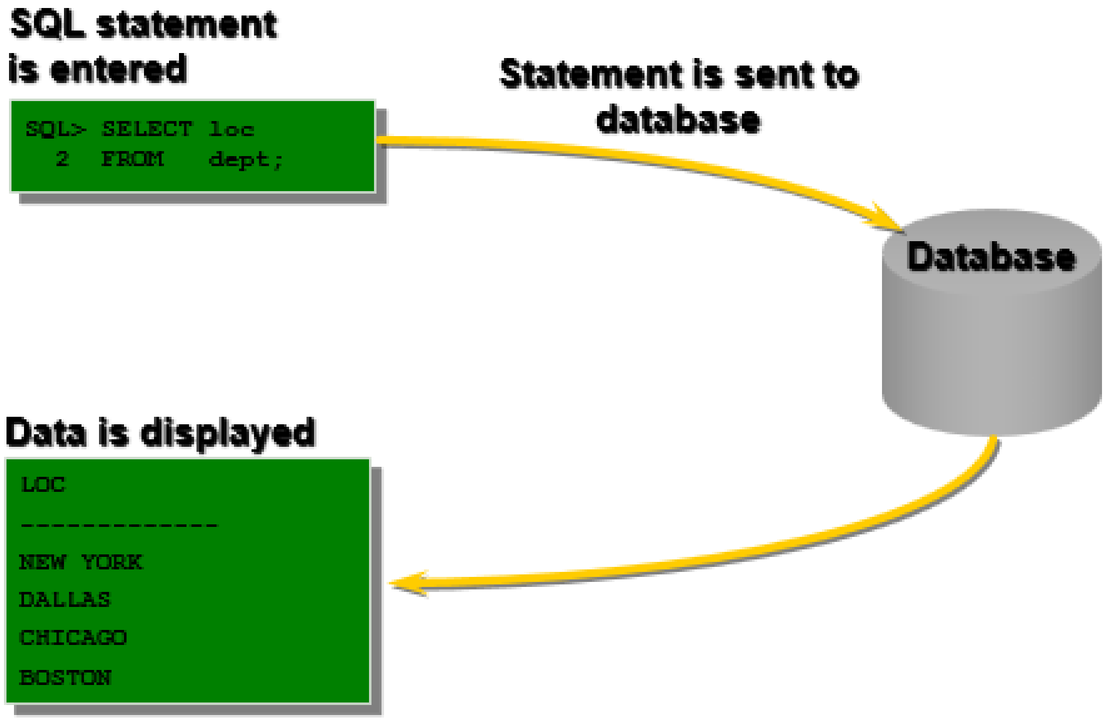
- SQL: Structured Query Language
- Capabilities of SQL
- SQL:DDL, DML,DQL,DCL, TCL
- Data Storing and Retrieving Examples:
    - DDL Operations
    - DML Operations
    - DQL Operattions

Database Language: SQL (Structured Query Language)


SQL
DDL
(Data Definition Language)
DML
(Data Manipulation Language)
DCL
(Data Control Language)
TCL
(Transaction Control Language)

DQL is a special type of DML that is used for querying data from the existing Database table(s)

Communicating with RDBMS using SQL



SQL statement is entered

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

Statement is sent to database

Data is displayed

```
LOC
-------------
NEW YORK
DALLAS
CHICAGO
BOSTON
```

SQL Statements

- SQL statements are not case sensitive.
- SQL statements can be on one or more lines.
- Keywords cannot be abbreviated or split across lines.
- Clauses are usually placed on separate lines.
- Tabs and indents are used to enhance readability.

DDL: Data Definition Languages

Followed by Database Schema, Data Definition Language defines a set of database tables for a database and during this, the integrity constraints (both Domain constraints and Referential constraints) are also another major concerns.

For example, defining table department "DEPT" will be,

```
Create table DEPT ( DEPTNO integer(10) PRIMARY KEY,
                    DNAME char(15),
                    Loc char(15));
```

Different DDL operations are:

CREATE, ALTERIM, DROP, RENAME, TRUNCATE

DDL

Statements	Description
Create	Defines the table
Alter	Modifies structure of the already created table
Drop	Removes the entire table
Rename	Changes the name of the already defined table or, even view or sequence
Truncate	Removes all the stored values of the table and eventually releases the storage space

Example: DDL Statements

```
CREATE TABLE [schema.]table
 (column datatype [DEFAULT expr][, ...]);
```

For Example,

```
CREATE TABLE dept (deptno NUMBER (2),
dname VARCHAR (14),
loc VARCHAR (13));
```

Example: DDL Statements

```
ALTER TABLE table
COMMAND   (column datatype [DEFAULT expr]
                [, column datatype] ...);
```

****** By Alteration, columns (attributes) can be added/dropped/modified. Alterations can be done on the database constraints as well.**

For Example, if it’s necessary to add any column to the table then,

```
ALTER TABLE dept
ADD (address VARCHAR (9));
```

Example: DDL Statements

```
DROP TABLE table_name;
```

For example,

```
DROP TABLE Dept;
```

This is how the entire table will be dropped/deleted from the database.

Example: DDL Statements

```
RENAME TABLE table_name1 TO table_name2
```

For Example,

```
RENAME TABLE dept to Department;
```

The existing table name will be changed from the way it was defined to the new name

Example: DDL Statements

```
TRUNCATE TABLE table_name;
```

For Example,

```
TRUNCATE TABLE Department;
```

This is how the values stored in the database table Department will be removed but the table will not be deleted, freeing the storage of the data it used to hold.

Database Constraints

- **Primary Key**
- **Foreign Key**
- **Unique Key**
- **NOT NULL**

Discussion and Examples will be shown during the Lab Today…

Relating Multiple Tables

Each row of data in a table is uniquely identified by a Primary Key (PK).

As a result, logically related data from multiple tables can be ‘connected’ by using Foreign Keys (FK).

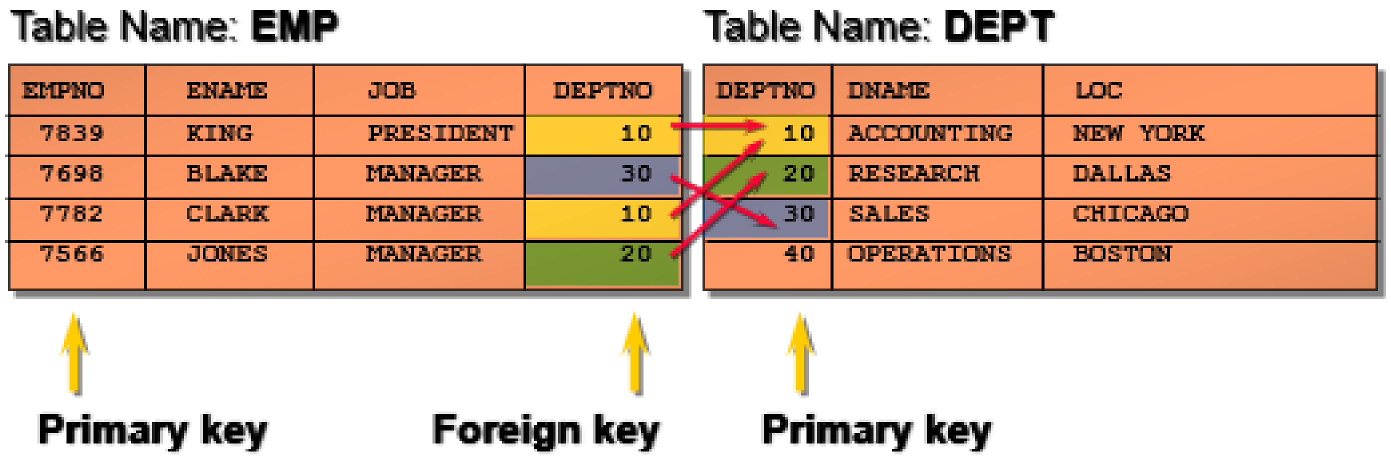


Table Name: **EMP**

EMPNO	ENAME	JOB	DEPTNO
7839	KING	PRESIDENT	10
7698	BLAKE	MANAGER	30
7782	CLARK	MANAGER	10
7566	JONES	MANAGER	20

Table Name: **DEPT**

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

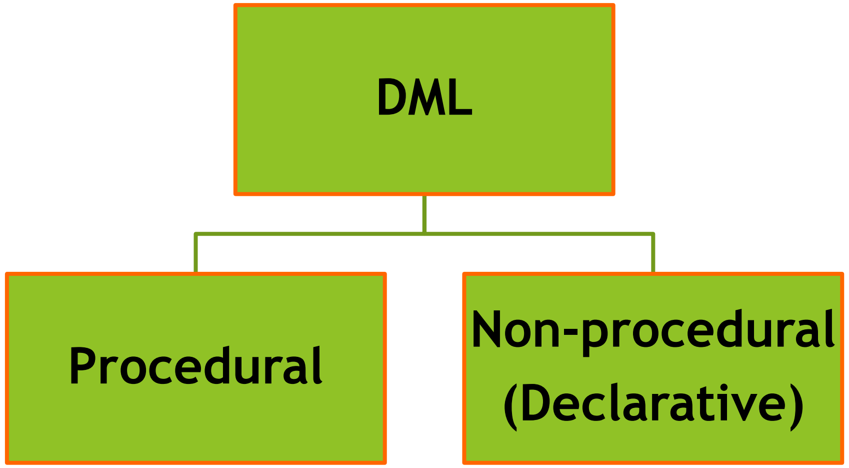
Primary key
Foreign key
Primary key

The actual content of the database at a point in time which are stored as real values (the data).

DML: Data Manipulation Languages

Data Manipulation Language or, Query Language are used for accessing and manipulating the data organized by the appropriate data model in a database system.

Two classes of Data Manipulation Languages are used :



DML

Procedural

Non-procedural (Declarative)

Different DML operations are: INSERT, UPDATE, DELETE, MERGE

DML: Data Manipulation Languages

A DML statement is executed when you:

- Add new rows to a table
- Modify existing rows in a table
- Remove existing rows from a table

A *transaction* consists of a collection of DML statements that form a logical unit of work.

DML: “Insertion” Operation

- Insert a new row containing values for each column.
- List values in the default order of the columns in the table.
- Optionally list the columns in the INSERT clause.

***Enclose character and date values within single quotation marks.

Example DML: “Insertion” Operation

INSERT INTO *table* VALUES (*value* [, *value*...]);

Only one row is inserted at a time with this syntax.

For Example,

INSERT INTO dept VALUES (50, 'DEVELOPMENT', 'DETROIT');

Example DML: “Insertion” Operation

Implicit method: Omit the value from the column list by keeping the quotation empty , e.g, ‘ ‘

```
INSERT INTO dept VALUES (60, ‘MIS’, ‘ ‘);
```

Explicit method: Specify the NULL keyword as below

```
INSERT INTO dept VALUES(70, ‘FINANCE’, NULL);
```

Example DML: “Insertion” Operation- Alternative Syntax

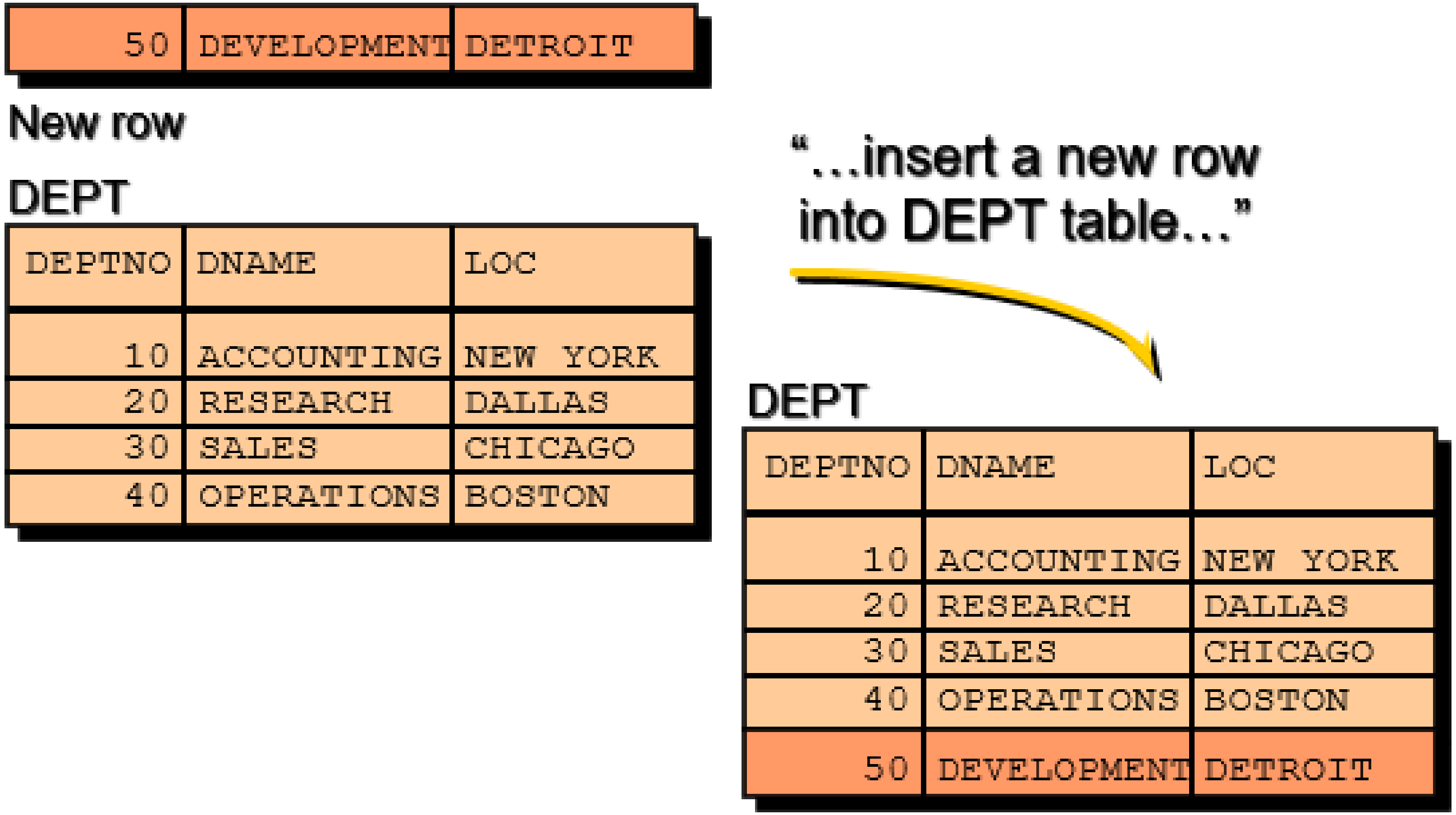
```
INSERT INTO dept (deptno, dname, loc ) VALUES (60,'MIS’, ‘Brentford’);
```

As the column names are mentioned the values will be inserted accordingly with the specified order.

However, we can OMIT the column from the column list as,

```
INSERT INTO     dept  VALUES (60,'MIS’, ‘Brentford’);
```

Insertion operation on a table



50	DEVELOPMENT	DETROIT

New row

DEPT

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

"…insert a new row into DEPT table…"

DEPT

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

DML: “Update” Operation

Update operation modifies existing rows with the UPDATE statement.

```
UPDATE     table_name
SET column_name = value [ column = value, ...]
[WHERE       condition];
```

Example DML: “Update” Operation

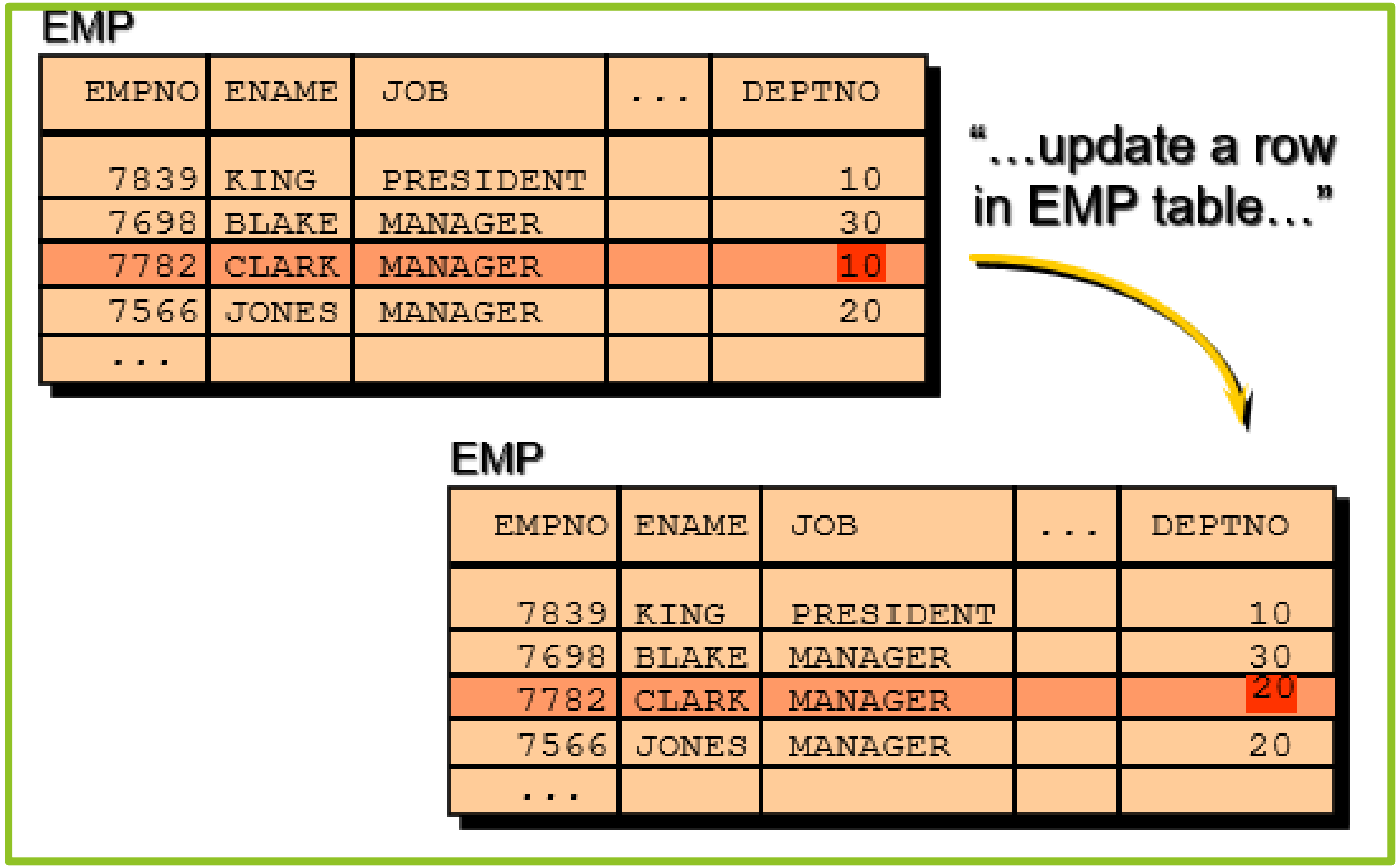
Specific row or rows are modified when you specify with the WHERE clause.

```
UPDATE     dept
SET        loc = ‘Dallas’
WHERE      deptno = 20;
```

All rows in the table are modified if you omit the WHERE clause.

```
UPDATE     dept
SET        loc = Dallas;
```

Update Operation on a table



EMP

EMPNO	ENAME	JOB	...	DEPTNO
7839	KING	PRESIDENT		10
7698	BLAKE	MANAGER		30
7782	CLARK	MANAGER		10
7566	JONES	MANAGER		20
...				

"...update a row in EMP table..."

EMP

EMPNO	ENAME	JOB	...	DEPTNO
7839	KING	PRESIDENT		10
7698	BLAKE	MANAGER		30
7782	CLARK	MANAGER		20
7566	JONES	MANAGER		20
...				

Here, on a table called "Emp" the deptno for a specific employee has been updated.

DML: “Delete” Operation

Existing Row or, set of rows of the table can be removed using DELETE operation

```
DELETE [FROM]   table_name
[WHERE    condition];
```

Example DML: “Delete” Operation

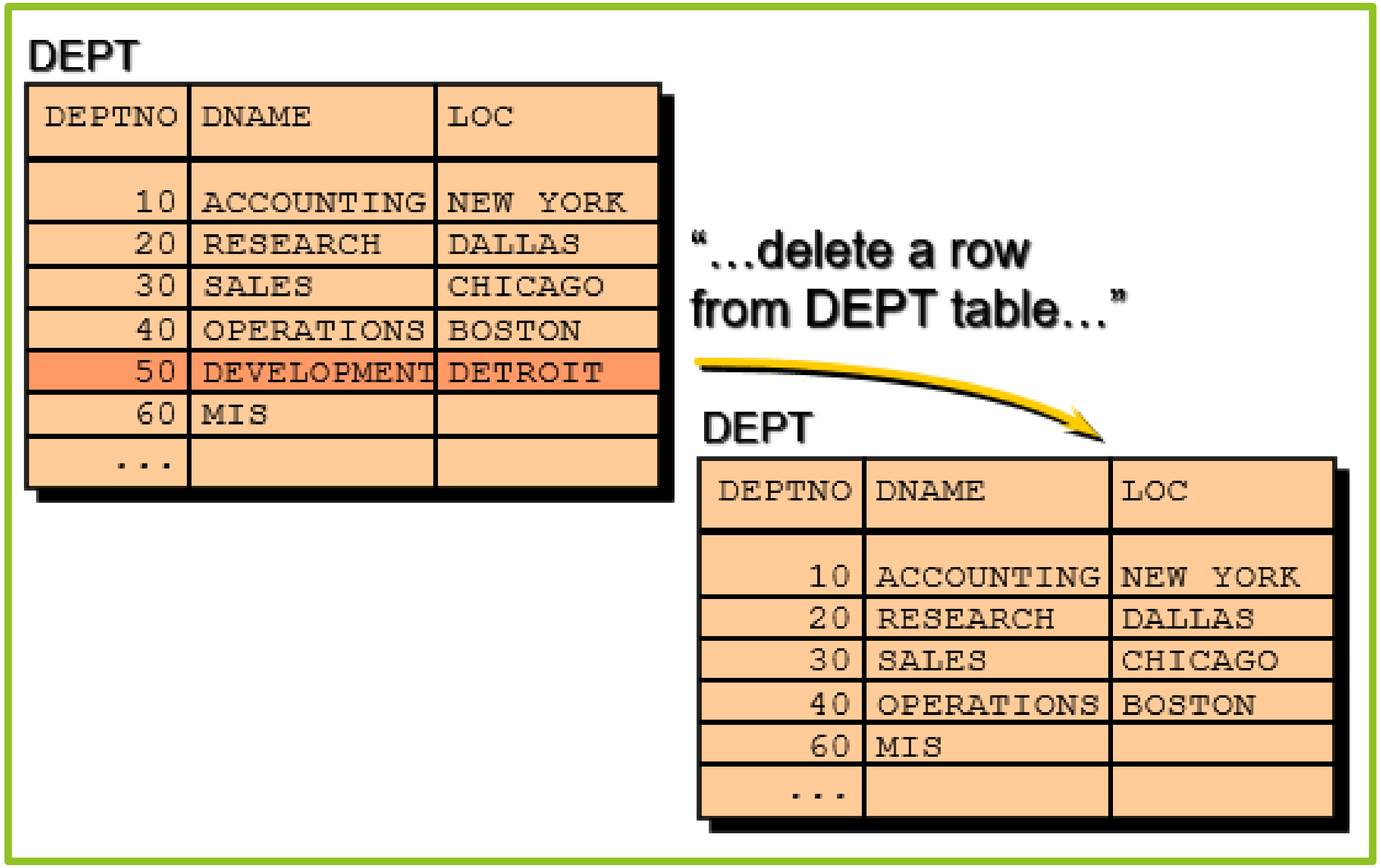
Specific rows are deleted when you specify with the WHERE clause.

```
DELETE FROM department
WHERE  dname = 'DEVELOPMENT’;
```

All rows in the table are deleted if you omit the WHERE clause.

```
DELETE FROM department;
```

Example DML: “Delete” Operation



DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON
50	DEVELOPMENT	DETROIT
60	MIS	
...		

“…delete a row from DEPT table…”

DEPT

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

Comparison Between DDL and DML

DDL	DML
Data Definition Language	Data Manipulation Language
Defines database structure	Manages data within the schema objects
Defines the data structure	Manipulates existing database
Changes the whole table	Changes rows (records) or set of rows(records)
DDL cannot be classified	DML can be classified as “Procedural” and “Non-Procedural” Query Language
Examples: CREATE, DROP, ALTER, RENAME, TRUNCATE	Examples: INSERT, UPDATE, DELETE

DQL: “Select” Statement

SELECT [DISTINCT] {*, *column* [*alias*],...}

FROM *table*;

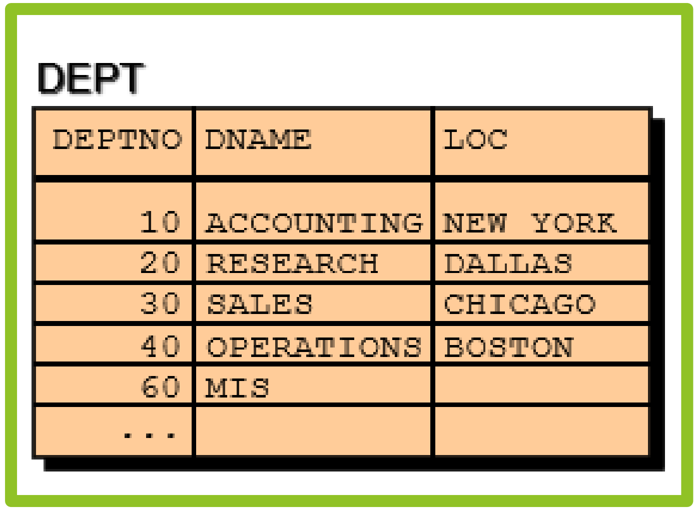
- SELECT identifies ***what*** columns
- FROM identifies ***which*** table

Example DQL operation: “Select” Statement

Here, for example we want to retrieve all the data stored in the “DEPT” table

```
SELECT *
FROM Dept;
```

The return of the above execution be like,



DEPT

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

Data Retrieval using “Select” Statement

By Using Select Statements we can,

- Limit the rows retrieved by a query
- Sort the rows retrieved by a query

Limiting Rows in “Select”

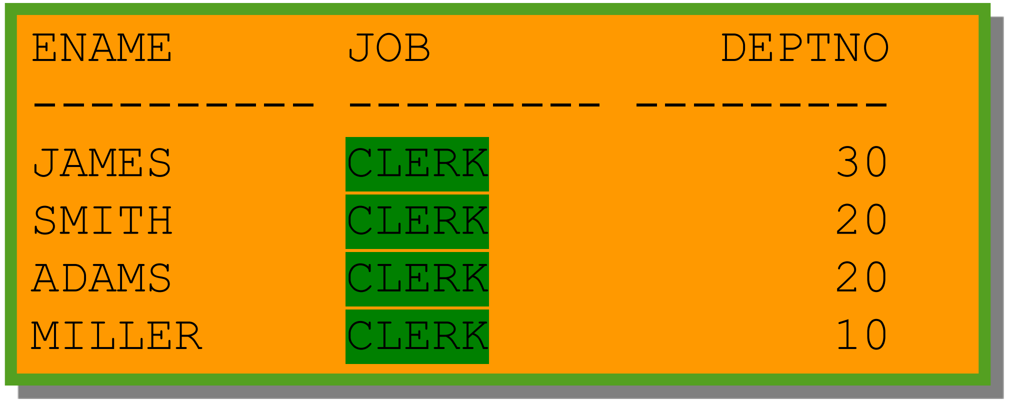
Restrict the rows returned by using the WHERE clause.

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

*****The WHERE clause follows the FROM clause.**

Example DQL operation: “Select” Statement

```
SELECT ename, job, deptno
FROM   emp
WHERE  job='CLERK’;
```



```
ENAME      JOB          DEPTNO
---------- --------- ---------
JAMES      CLERK            30
SMITH      CLERK            20
ADAMS      CLERK            20
MILLER     CLERK            10
```

DCL & TCL

Data Control Language deals with providing privileges to users in Relational Database Management System objects.

To perform the above-mentioned operation both “GRANT” and “REVOKE” statements are used.

Transaction Control Language includes COMMIT, ROLLBACK and SAVEPOINT to ensure the successful transaction any database user intends to make while querying.

Do you have any
QUESTIONS?