

CHAPTER 4

DFC2083

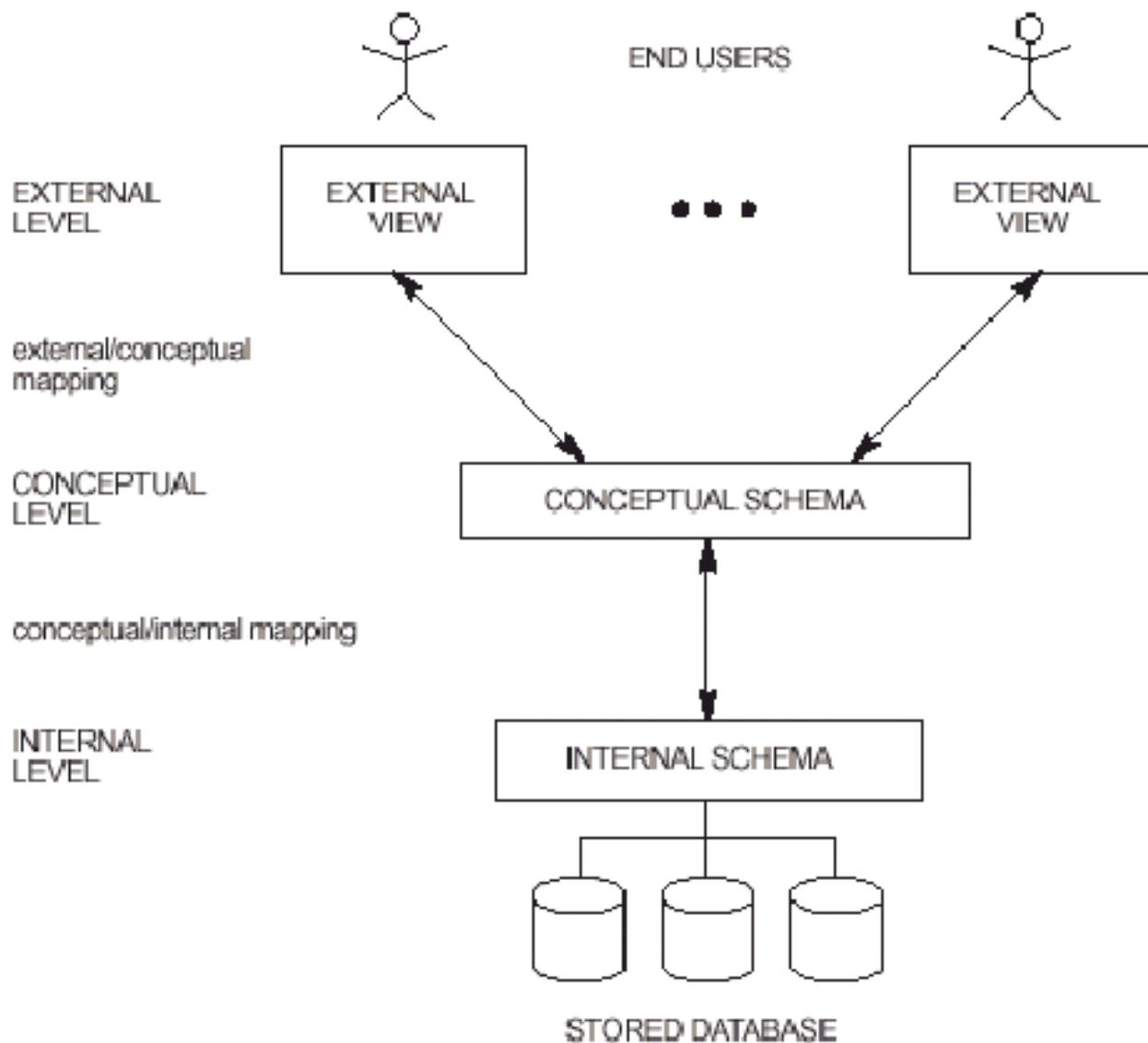
DATABASE DESIGN

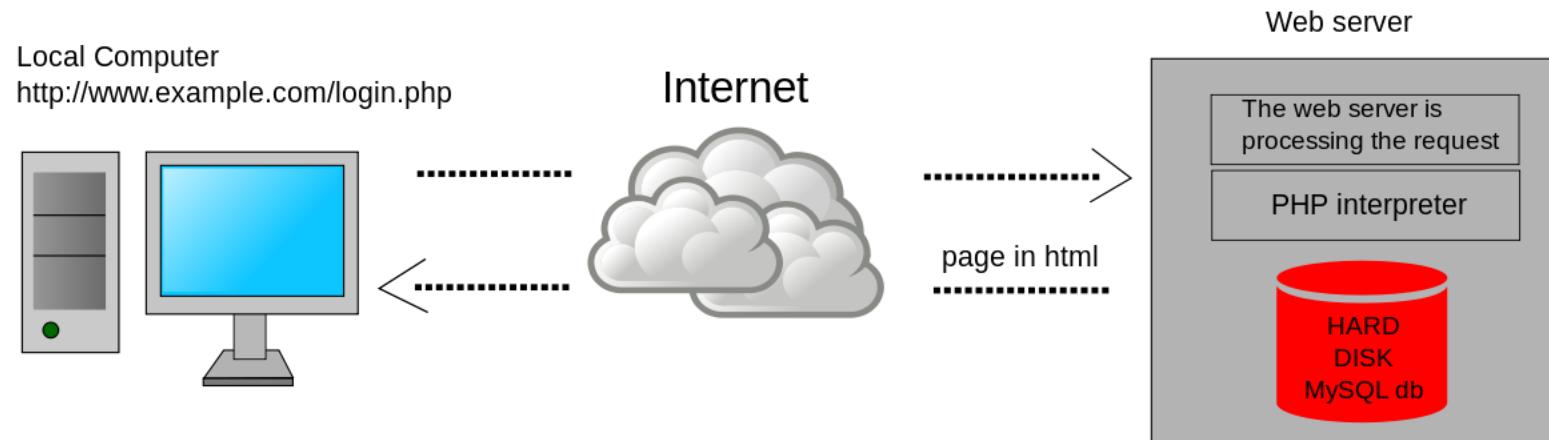
STRUCTURED QUERY LANGUAGE

Madam Zati

RDBMS

- RDBMS stands for **Relational Database Management System.**
- RDBMS is the basis for SQL, and for all modern database systems such as MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.
- The data in RDBMS is stored in **database objects called tables.**
- A table is a collection of related data entries and it consists of columns and rows.





SQL as Query Language :

- Purpose is to query data from relational tables
- **Declarative** : Instead of specifying how to retrieve the data, we declare what data we want
- A combination of two formalisms known as Relational Algebra and Relational Calculus

MYSQL :

- ✓ MySQL is a **database**.
- ✓ The data in MySQL is stored in database **objects** called **tables**.
- ✓ A table is a collections of related data entries and it consists of **columns** and **rows**.
- ✓ Databases are useful when **storing information categorically**.
- ✓ A company may have a database with the following tables: "Employees", "Products", "Customers" and "Orders".

MYSQL

- MySQL is a database system **used on the web**
- MySQL is a database system that **runs on a server**
- MySQL is ideal for **both small and large applications**
- MySQL is very **fast, reliable, and easy to use**
- MySQL uses **standard SQL**
- MySQL **compiles** on a number of platforms
- MySQL is **free to download** and use
- MySQL is developed, distributed, and supported by
Oracle Corporation

WHAT CAN YOU DO WITH SQL?

- ✓ SQL can **execute** queries against a database
- ✓ SQL can **retrieve** data from a database
- ✓ SQL can **insert** records in a database
- ✓ SQL can **update** records in a database
- ✓ SQL can **delete** records from a database
- ✓ SQL can **create** new databases
- ✓ SQL can **create** new **tables** in a database
- ✓ SQL can **create stored procedures** in a database
- ✓ SQL can **create views** in a database
- ✓ SQL can set permissions on tables, procedures, and views

TYPES OF SQL STATEMENT

1. Data Definition Language (DDL)
2. Data Manipulation Language (DML)
3. Transaction Control Language (TCL)
4. Data Control Language (DCL)

DATA DEFINITION LANGUAGE (DDL)

A language that allows the DBA or user to describe and name the entities, attributes, and relationship required for the application, together with any associated integrity and security constraints.

DDL statement can define data tables, index and database relations.

DDL commands:

- CREATE : to define a new database/table/view
- USE : to select a database
- ALTER : to modify table/column definition
- DROP : to drop database/table.

DATA MANIPULATION LANGUAGE (DML)

- ❖ A language that provides a set of operators to support the basic data manipulation operations on the data held in the database.

- ❖ DML statement used to select, insert, update and delete information stored in data tables

DML commands:

- + INSERT : to insert data into a table
- + SELECT : to query data in the database
- + UPDATE : to update data in a table
- + DELETE : to delete data from a table

Writing SQL Commands :

- SQL commands can be written in one line or many.
- SQL commands are case insensitive.
- SQL commands must end with semicolon (;) .

SQL Identifiers :

- An identifier can be no longer than 128 characters
- An identifier must start with a letter.
- An identifier cannot contain spaces.

Create Database

- To see how many databases you have on your system:

show databases;

- To create database:

create database <database name>;

- To specify database:

use <database>;

Create Table

- Database store all data in tables or relations
- Table consist of column and row
- Each column define data according to data type
- Each row contains individual records.
- Database can have many inter related tables which contains the real data.

Create Table

Name	Age	Country	Email
Manish Sharma	28	India	<u>manish@simplygraphix.com</u>
John Doe	32	Australia	<u>j.dow@nowhere.com</u>
John Wayne	48	U.S.A.	<u>jw@oldwesterns.com</u>
Alexander	19	Greece	<u>alex@conqueror.com</u>

Create Table (1st method)

```
CREATE TABLE employee_data(  
    emp_id int unsigned not null  
    auto_increment primary key,  
    f_name varchar(20),  
    l_name varchar(20),  
    title varchar(30),  
    age int,  
    yos int,  
    salary int,  
    perks int,  
    email varchar(60));
```

Create Table (2nd method)

```
CREATE TABLE employee_data(  
emp_id int not null unique,  
f_name varchar(20),  
l_name varchar(20),  
title varchar(30),  
age int,  
yos int,  
salary int,  
perks int,  
email varchar(60),  
PRIMARY KEY (emp_id));
```

Create Table (3rd method)

```
CREATE TABLE employee_per(
e_id int unsigned not null primary key,
address varchar(60),
phone int,
p_email varchar(60),
birth_date DATE,
sex ENUM('M', 'F'),
m_status ENUM('Y', 'N'),
s_name varchar(40),
children int);
```

Create Table (4th method)

```
CREATE TABLE employee_per (
e_id int not null unique,
address varchar(60),
phone int,
p_email varchar(60),
birth_date DATE,
sex ENUM('M', 'F'),
m_status ENUM('Y', 'N'),
s_name varchar(40),
children int,
FOREIGN KEY (e_id) references employee_data
ON DELETE RESTRICT
ON UPDATE CASCADE);
```

Create Table

Command and column name in MySQL is case insensitive

- Every column name is followed by column type
- Column type define data type for each column:

<column name> <column type>(<maximum number>);

Create Table

- **Varchar(variable characters)** : column type consist of character/small text type
- **int**: column type consist of integer (number)
- **unsigned**: determine no sign to number (positive)
- **not null**: value cannot be null (empty); each rows in column must have a value

Show table

- To show tables, to check whether is in the list:
`show tables;`
- To display the column details of the tables:
`describe <table name>;`

Alter Command : Rename Table

- To rename a table:

```
alter table <table name>
```

```
rename to <newtable name>;
```

Edit Column in a Table

- To add column:

```
alter table <table name>
```

```
add column(<column name> <column  
type>(m));
```

Edit Column in a Table

- To drop a column:

```
alter table <table name>  
drop column <column name>;
```

Edit Column in a Table

- To alter column name

```
alter table <table name>
```

```
change <column name> <new column  
name><new column type>(m);
```

Edit Column in a Table

- To alter new column type

alter table < table name >

modify <column name><new_column_type>(m);

Drop table

- To drop a table, DROP table command is used:

DROP TABLE employee_data;

DML commands : **INSERT**

- SQL statement to insert data into tables is as below:

```
INSERT into table_name(col1,col2,...colN)  
values (val1,val2,...valN);
```

- Table_name ⑦ the table name used to **insert** data
- column1, column2 etc. ⑦ are column names
- value1, value2 etc. ⑦ is the value for each column according to the right sequence

INSERT DATA

- Example of inserting data into table:

```
mysql> INSERT INTO employee_data  
-> (f_name, l_name, title, age, yrs, salary,  
-> perks, email) values  
-> ("Manish", "Sharma", "CEO", 28, 4,  
200000, 50000, "manish@bignet.com");
```

SELECT STATEMENTS

- Data is retrieved from table using SQL **SELECT** statement as below:

```
SELECT column_name  
FROM table_name  
[WHERE ...condition];
```

- **Condition** is NOT compulsory, while Column depends on users.

```
SELECT f_name, l_name  
FROM employee_data;
```

SELECT STATEMENTS

- To list all columns in a table:

```
SELECT emp_id, f_name, l_name, title, age,  
      yrs, salary, perks, email  
FROM employee_data;
```

OR

```
SELECT * from employee_data;
```

- * means **ALL COLUMNS**.
- The statement above will list **all records in a table**.

SELECT.. CONDITION

- **WHERE** clause

```
SELECT column_name  
FROM table_name  
[WHERE ...condition];
```

- The select statements with condition will display records according to certain conditions.