

Introduction to Oracle

Client Server Architecture

Client-server architecture (client/server) is a network architecture in which each computer or process on the network is either client or server. **Servers** are powerful computers or processes dedicated to managing disk drives (file servers), printers (print servers), or network traffic (network servers). **Clients** are PCs or workstations on which users run applications. Clients rely on servers for resources, such as files, devices, and even processing power.

The problems associated with handling large data-centric applications, over the file sharing network led directly to the development of the **Client/Server Architecture** in the early 1980's.

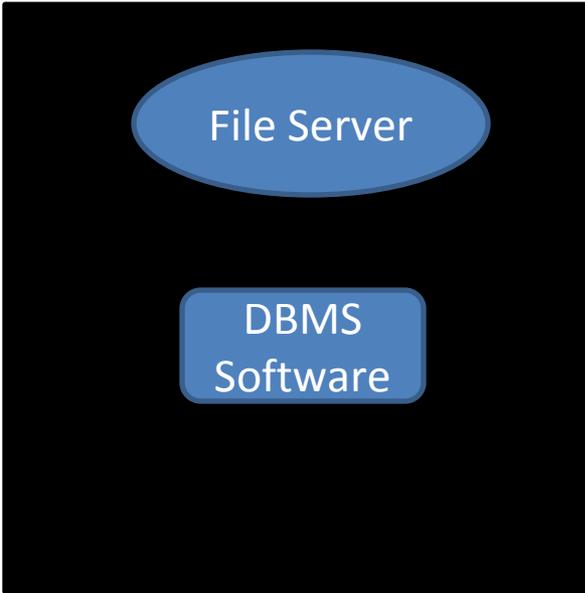
1-Tier Architecture

Single Tier Architecture says that the DBMS and the data reside in one location and the dumb terminals were used to access the DBMS. The theory was that if data are kept in two or more places there is a high probability that two data items which are identical may not be the same. E.g. If a student's record is stored in two files then it is possible that one is to be changed and other to remain the same.

1-Tier Architecture is the simplest, single tier or single user, and is the equivalent of running an application on a personal computer. The entire required component to run the application is located within it. User interface, business logic, and data storage are all located on the same machine. They are the easiest to design.

Personal
Computer

Personal
Computer



2- Tier Architecture

In Two Tier Architecture the workload is divided between the Server(which hosts the database) and the Client (which hosts the User interface) . In reality these are normally located on separate physical machines but there is no absolute requirement for this to be the case. The most basic of client server architecture employs only two types of hosts :

(a) Client's

(b) Server's

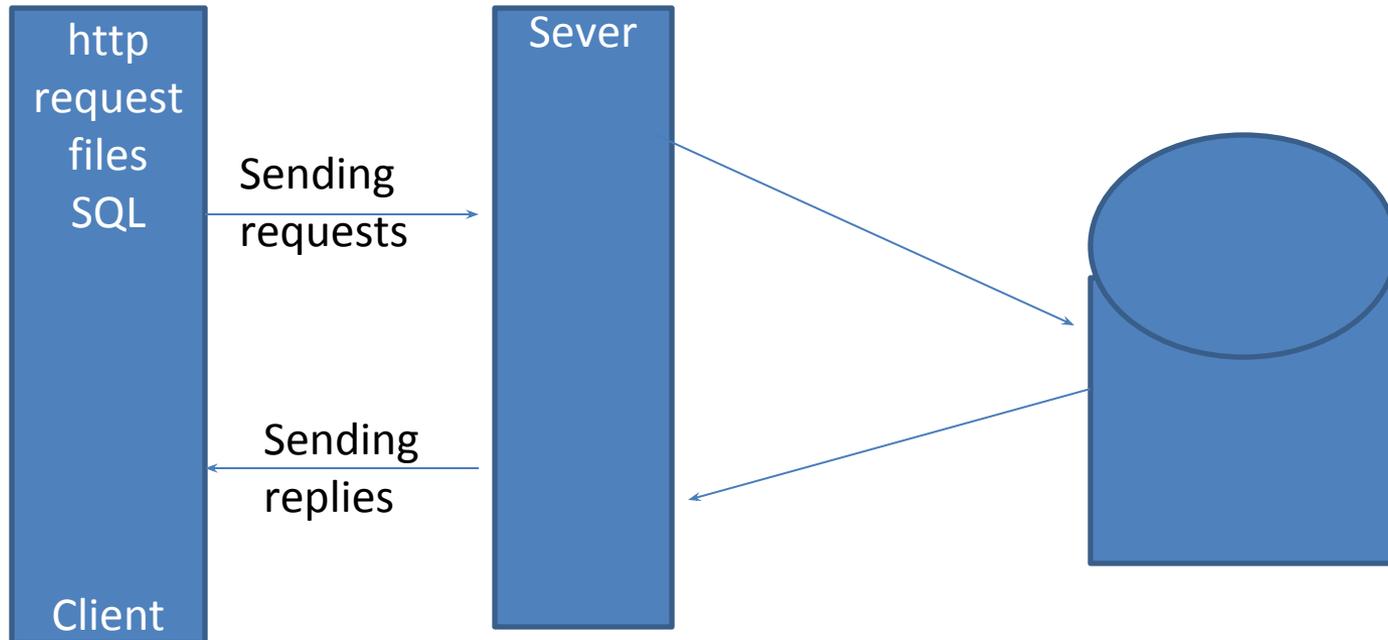
This type of architecture is sometimes referred to as two-tier. It allows devices to share files and resources. The two tier architecture means that the client acts as one tier and application in combination with server acts as another tier.

The 2-tier architecture is used to describe client/server systems where the client requests resources and the server responds directly to the request, using its own resources. The user interface and application programs are placed on the client side and the database system on the server side. The application programs that reside at the client side invoke the DBMS at the server side. The application program interface standards like Open Database

connectivity (ODBC) and Java Database Connectivity (JDBC) are used for interaction between client and server.

Level 1

Level 2



3-Tier Architecture

In a 3-Tier architecture , there is an intermediate level, it adds intermediate layer known as application server between the client and the database server. The architecture is generally split up between:

- (a) A **Client** , i.e. The computer, which requests the resources, equipped with a user interface (usually a web browser) for presentation purposes.
- (b) The **application server** , (also called middleware), whose task it is to provide the requested resources, but by calling on another server.
- (c) The **data server**, which provides the application server with the data it requires.

The client communicates with the application server, which in turn communicates with the database server. The application server stores the business rules(procedures and constraints) used for accessing data from database server. It checks the client's credentials before forwarding a request to database server. When a client requests for information, the application server accepts the request, processes it, and sends corresponding database commands to database server. The database server sends the result back to application server which is converted into GUI format and presented to the client.

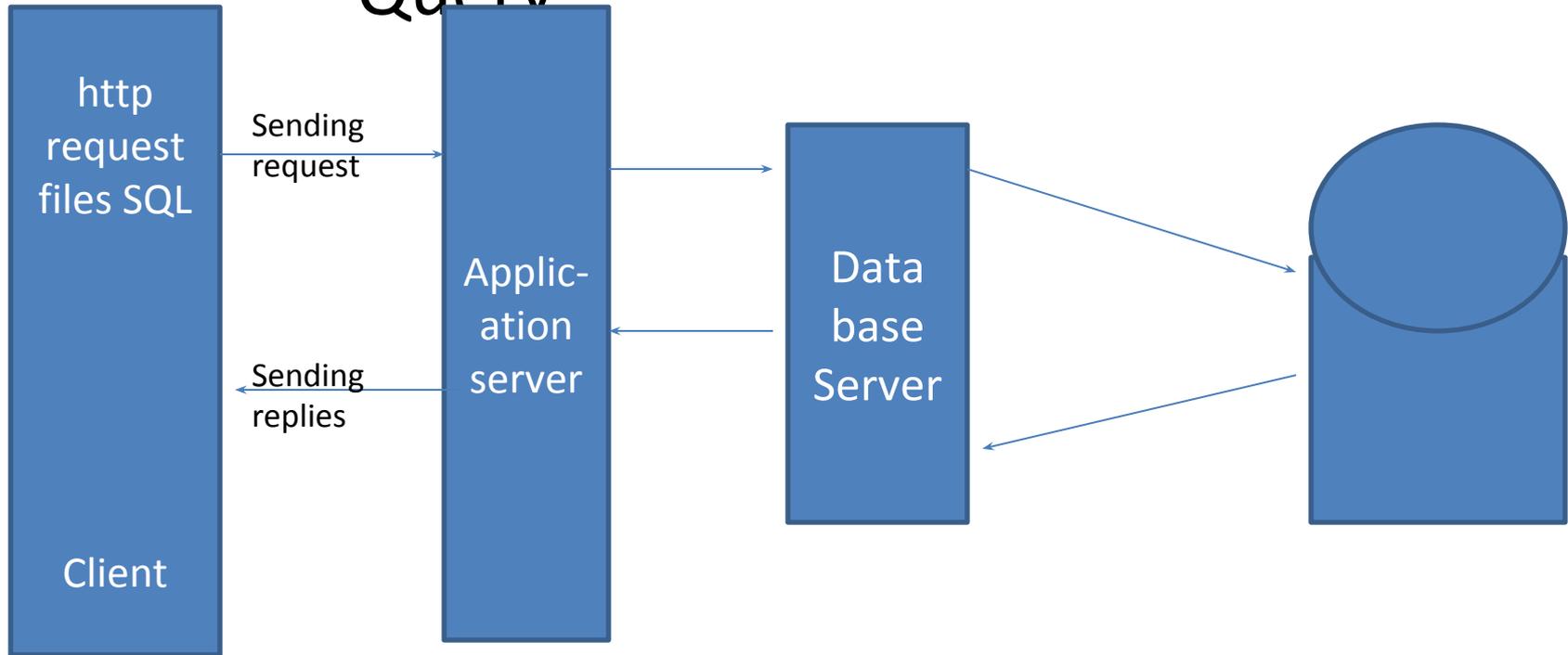
Level 1

Level 2

Level 3

SQL

Query



SQL Fundamentals

SQL is a short for Structured Query Language and is a widely used database language, providing means of data manipulation (store, retrieve, update, delete) and database creation. It was developed by IBM Research in the mid 70's and standardized by ANSI in 1986. Donald D. Chamberlin and Raymond F. Boyce developed the first version of SQL, also known as Sequel as part of the System R project in the early 1970s at IBM.

SQL Language Statements :- SQL is a simple and powerful language used to perform the various operations like create, retrieve and manipulate the data in the database. SQL is an easy understandable language. The various categories of SQL statements are as under :-

- (1) Data Definition Language
- (2) Data Manipulation Language
- (3) Data Control Language
- (4) Transaction Control Language

Data Definition Language (DDL)

A Data Definition Language (DDL) is a computer language for defining data structures. The term was first introduced in relation to the Codasyl database model, where the schema of the database was written in a Data Definition Language describing the records, fields, and “sets” making up the user Data Model. DDL statements create, modify, and remove database objects such as tables, indexes, and users. Common DDL statements are Create, Alter and Drop.

A Data Definition Language has a pre-defined syntax for describing data. The following are the various DDL commands :-

1. **CREATE** – to create objects in the database
2. **ALTER** – alters the structure of the database
3. **DROP** – delete objects from the database
4. **RENAME** – rename an object

Data Manipulation Language

It **access** and **manipulate** data in existing schema objects. These statements do not implicitly commit the current transaction. Data manipulation language is a family of computer languages used by computer programs and/or database users to retrieve data. The following operations are performed by DML :-

1. **SELECT** – retrieve data from a database
2. **INSERT** – insert data into a database
3. **UPDATE** – updates existing data within a database
4. **DELETE** – deletes all records from a table, the space for the records remain

Data Control Language

It is a computer Language and a subset of SQL, used to control access to data in a database. DCL statements are used to provide a kind of security to the database. Users can be granted privileges or roles to allow restricted access to the database. Grant command is used to give the permission to the user.

Examples of DCL commands include :-

1. **GRANT** :- to allow specified users to perform specific tasks.
2. **REVOKE** :- to cancel previously granted or denied permissions.

Transaction Control Language Statements

TCL statements are used to manage the changes made by DML statements. It allows statements to be grouped together into logical transactions.

1. **COMMIT** :- save work done
2. **ROLLBACK** :- restore database to original since the last COMMIT (like undo)
3. **SET TRANSACTION** :- change transaction options like isolation level and what rollback segment to use.

The Syntax for the CREATE TABLE Statement is:

```
CREATE TABLE table_name  
  
(column_name1 datatype,  
  
column_name2 datatype,  
  
... column_nameN datatype  
  
);
```

For Example: If you want to create the employee table, the statement would be like,

```
CREATE TABLE employee  
  
( id number(5), name char(20), dept char(10), age number(2),  
  
salary number(10), location char(10)  
  
);
```

The sql insert query will be as follows

```
INSERT INTO TABLE_NAME (Column Name(s))  
VALUES (value1, value2, value3,...valueN);
```

For Example: If you want to insert a row to the employee table, the query would be like,

```
INSERT INTO employee (id, name, dept, age, salary  
location) VALUES (105, 'Srinath', 'Aeronautics', 27,  
33000,'CHandigarh');
```

The Syntax for SQL UPDATE Command is:

```
UPDATE table_name  
  
SET column_name1 = value1,  
column_name2 = value2, ...  
  
[WHERE condition]
```

For Example: To update the location of an employee, the sql update query would be like,

```
UPDATE employee  
  
SET location = 'Mysore'  
  
WHERE id = 101;
```

Syntax of a SQL DELETE Statement

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

SQL DELETE Example

To delete an employee with id 100 from the employee table, the sql delete query would be like,

```
DELETE FROM employee WHERE id = 100;
```

To delete all the rows from the employee table, the query would be like,

```
DELETE FROM employee;
```

The DROP TABLE Statement

The DROP TABLE statement is used to delete a table.

```
DROP TABLE table_name
```

The DROP DATABASE Statement

The DROP DATABASE statement is used to delete a database.

```
DROP DATABASE database_name
```

Syntax for using SQL ORDER BY clause to sort data is:

```
SELECT column-list
```

```
FROM table_name [WHERE condition]
```

```
[ORDER BY column1 [, column2, .. columnN] [DESC]];
```

For Example: If you want to sort the employee table by salary of the employee, the sql query would be.

```
SELECT name, salary FROM employee ORDER BY salary;
```

SQL datatypes

Data Type	Description
CHARACTER(n)	Character string. Fixed-length n
VARCHAR(n)	Character string. Variable length. Maximum length n
BOOLEAN	Stores TRUE or FALSE values
INTEGER	Integer numerical (no decimal)
DATE	Stores year, month, and day values
TIME	Stores hour, minute, and second values

Arithmetic Functions

1. The AVG() Function

The AVG() function returns the average value of a numeric column.

SQL AVG() Syntax

```
SELECT AVG(column_name) FROM table_name
```

2. SQL COUNT(column_name) Syntax

The COUNT(column_name) function returns the number of values (NULL values will not be counted) of the specified column:

```
SELECT COUNT(column_name) FROM table_name;
```

SQL COUNT(*) Syntax

The COUNT(*) function returns the number of records in a table:

```
SELECT COUNT(*) FROM table_name;
```

```
SELECT COUNT(DISTINCT column_name) FROM table_name;
```