**Objectives:**

* **Explain Client Server architecture**
* **Describe Types of database systems**
* **Explain Database system concepts**

**Two-tier architecture**

* Consists of two software components the client and the server.
* The application on the client’s machine has the ability to invoke database system functionality from the server through SQL statements.
* This architecture has the advantage of being simple and having compatibility with existing systems.

**Three-tier architecture (client server architecture)**

* Adds a mediating layer between the client and database server.
* The client does not have direct access to database calls, instead they communicate with an application server.
* The application server then communicates with the database to retrieve data requested by the client.
* This architecture is more appropriate for the large applications that run on the internet.

**Advantages:**

**-** Added security.

**-** More flexibility.

**Disadvantages:**

**-**Unable to respond to multiple request, efficiently.

**-**Slower than two-tier architecture.

**ANSI-SPARC Model**

This model provides a conceptual framework for database management systems.

It defines **3** levels of **abstraction**:

* ***Internal level –*** this is the ***lowest*** level. It details how data is ***physically*** ***stored*** in the database, as well as ***optimizing*** ***performance*** and ***storage*** efficiency.
* ***Conceptual level –*** its ***higher*** than the internal level. Represents the overall ***logical structure*** of the database. It ***describes*** entities, relationships and constraints.
* ***External level –*** the ***highest*** level. Provides a ***view*** of the entire ***database*** to end-users. Defining ***interfaces***, ***views*** and ***access rights*** ***specific*** to the different user groups.

**Types of database systems**

**Relation model**

* Consists of tables representing both the data and the relationship amongst the data.
* Each table has multiple columns, and each column has a unique attribute name.
* An attribute value is the value contained in the specific column.

**One-to-One (1:1) Entity Relationship**

One entity is related to only one other entity

**One-to-Many (1:M) Entity Relationship**

A single entity in one table can have a relation with more than one other entity in another table.

**Many-to-Many (M:M) Entity Relationship**

A relationship between two entities, where table A can have a relation with one or more items in table B, and vice versa.

**Many-to-One (M:1) Entity Relationship**

When table A has multiple entries that are linked to one entry in table B, however table B only makes a relation with a single item in table A.

**Keys**

A key is a data item which uniquely identifies a record/row.

**Primary key** is a predetermined attribute that uniquely identifies an entity/row.

**Candidate key** is an attribute which has the characteristics to uniquely identify a row or entity in a database, and has the ability to become a primary key.

**Foreign key** its a column/attribute in table A which can uniquely identify a record/row in table B, often times this attribute has the ability to act as the primary key of table B.

**Network model**

* This model has a structure often associated with the many-to-many relationship, allowing a record to have more than one parent segment.

**Database system concepts**

**Personal databases**

A database designed for a single-user where only one computer is involved and supports usage of only one user at a time.

e.g. Microsoft Access.

**Organizational databases**

These databases have multiple computers and applications, allowing multiple users to interact with the database simultaneously. These databases can have several databases and hundreds of table and column entries.

**Centralized databases**

When several systems, computers or users connects back to a single centralized computer which contains the database server and its management system.

**Distributed databases**

Involves a network of database servers and management systems which has been spread out and isolated into different sites. Each site has a database server and management system which can access database functions independent of the other, however changes made to one site will still apply to all other sites.

If all sites use the same DBMS software then the distribution is **homogenous**. However if the DBMS software differs between sites its known as a **heterogeneous** distribution.