Chapter 2

Database Environment
Objectives of Three-Level Architecture

- All users should be able to access same data.  
  (provided users are authorized to do so)

- A user’s view is immune to changes made in other views.

- Users should not need to know physical database storage details.
Objectives of Three-Level Architecture

- DBA should be able to change database storage structures without affecting the users’ views.

- Internal structure of database should be unaffected by changes to physical aspects of storage.

- DBA should be able to change conceptual structure of database without affecting all users.
This is a simplified view. Even though there appear to be four levels, there are really only three levels of schema.

**ANSI-SPARC Three-Level Architecture**

![Diagram of the ANSI-SPARC Three-Level Architecture](image)
ANSI-SPARC Three-Level Architecture

- **External Level** *(some call this the view level)*
  - Users’ view of the database.
  - Describes that part of database that is relevant to a particular user.

- **Conceptual Level** *(some call this the logical level)*
  - Community view of the database.
  - Describes what data is stored in database and relationships among the data.
ANSI-SPARC Three-Level Architecture

**Internal Level** (some call this the physical level)

- Physical representation of the database on the computer.
- Describes how the data is stored in the database.
Differences between Three Levels of ANSI-SPARC Architecture

External view 1

<table>
<thead>
<tr>
<th>sNo</th>
<th>fName</th>
<th>lName</th>
<th>age</th>
<th>salary</th>
</tr>
</thead>
</table>

Here you see mappings

Conceptual level

| staffNo | fName | lName | DOB | salary | branchNo |

And here as well

Internal level

```c
struct STAFF {
    int staffNo;
    int branchNo;
    char fName [15];
    char lName [15];
    struct date dateOfBirth;
    float salary;
    struct STAFF *next;
};
/* pointer to next Staff record */

index staffNo; index branchNo; /* define indexes for staff */
```
Data Independence

◆ **Physical Data Independence**
  – Refers to immunity of conceptual schema to changes in the internal schema.
  – Internal schema changes (e.g. using different file organizations, storage structures/devices).
  – Should not require change to conceptual or external schemas.

and logical data independence refers to the immunity of the external schema to changes in the conceptual schema
Data Independence and the ANSI-SPARC Three-Level Architecture

here's the primary way to think about the ANSI-SPARC model (three levels plus mappings between pairs of levels)