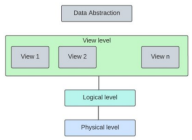
# Data Abstraction, Instances and Schema

09 August 2023 09:47

**Data Abstraction**

Developers hide complexity of data structures so that users can perform simple queries.
The different levels of data abstraction:
- Physical Level
- Logical level
- View level




Physical Level :
• It is the lowest level of data abstraction
• It describes how the data is actually stored in the computer

Logical Level :
• What kind of data is stored in the database and how they're related
• Describes the entire database in terms of a small number of relatively simple structures.

View Level :
• This is the highest level of abstraction
• Users interact with this layer
• Even though the logical level uses simpler structures, complexity remains because of the variety of information stored in a large database. Many users of the database system do not need all this information; instead, they need to access only a part of the database.
• At this level, different users might see different parts of the database, depending on their needs and permissions

Physical data independence:
- Modify the physical schema without changing the conceptual or logical schema
- Done for optimization purposes
- If it wasn't there changes to the storage system or hardware upgrades could potentially break existing queries, applications, and interfaces.

Logical data independence:
- Modify the logical schema without affecting the external schema i.e application program.
- The user view of the data would not be affected by any changes to the conceptual view of the data.
- If it wasn't there minor changes to the data model(adding or removing attributes, changing relationships b/w entities) could lead to widespread disruption and downtime for the applications,
  increasing the risk of introducing errors and inconsistencies.

Now let us understand this with the help of an example of a University database
- Let the university have the given Record types/ tables as shown

Now for the given database,
- Physical level:
  - specify how student records, course information, and department data are stored on the hard drive or in memory.
- Logical Level:
  - It would define entities like "Department," "Student," and "Course" and their attributes (e.g., department name, student ID, course code).
  - It also outlines the relationships between these entities, such as "Students belong to Departments" and "Courses are taken by Students."
- View Level:
  - A faculty member may have access to a view showing only the courses they are teaching and the students enrolled in those courses.
  - A student, on the other hand, may have a view displaying only the courses they are registered for and their respective grades.
  - These views provide a simplified and customized perspective of the database for different users.

**Instances and Schema**

The databases would have different type of schemas, partitioned according to the different levels of abstraction

• PHYSICAL SCHEMA : describes the database design at the physical level
- data storage format, file organization, and indexing methods.
- How data is stored on the hard disk, like using B-trees or hash indexes for efficient retrieval.
- Can usually be changed easily without affecting user programs.

• LOGICAL SCHEMA : describes database design at the logical level
- Structure and relationships between the tables.
- It is essential for application developers as they build software based on logical schema, so holds most significance.
- Defining the relationships between tables, such as the one-to-many relationship between students and courses, where one student can be enrolled in many courses, but each course can have multiple students.

• SUBSCHEMAS : there could be several schemas at the view level called subschemas, that describes different views
- Views allow users to see only relevant information and protect sensitive data.
- Creating a view that shows only the names of enrolled students and their corresponding course names, hiding other details like grades and personal information.