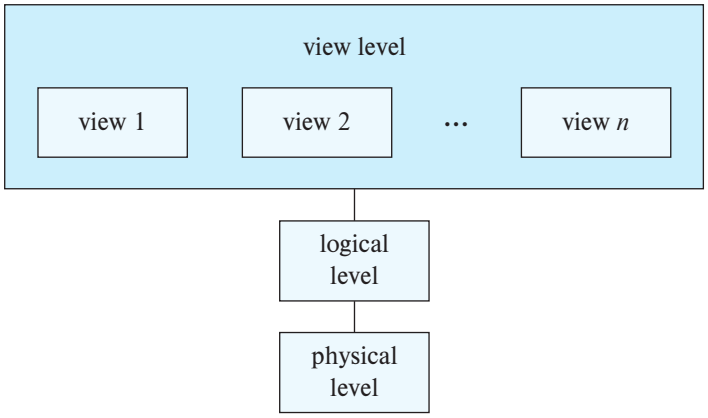

Figure 1.2 The three levels of data abstraction.

database-application developers. The database system allows application developers to store and retrieve data using the abstractions of the data model, and converts the abstract operations into operations on the low-level implementation.

An analogy to the concept of data types in programming languages may clarify the distinction among levels of abstraction. Many high-level programming languages support the notion of a structured type. We may describe the type of a record abstractly as follows:[1]

```
type instructor = record
                    ID : char (5);
                    name : char (20);
                    dept_name : char (20);
                    salary : numeric (8,2);
                  end;
```

This code defines a new record type called *instructor* with four fields. Each field has a name and a type associated with it. For example, **char**(20) specifies a string with 20 characters, while **numeric**(8,2) specifies a number with 8 digits, two of which are to the right of the decimal point. A university organization may have several such record types, including:

- *department*, with fields *dept_name*, *building*, and *budget*.
- *course*, with fields *course_id*, *title*, *dept_name*, and *credits*.
- *student*, with fields *ID*, *name*, *dept_name*, and *tot_cred*.

[1]The actual type declaration depends on the language being used. C and C++ use **struct** declarations. Java does not have such a declaration, but a simple class can be defined to the same effect.