Data Dependence

An application program that deals with data stored externally to it (such as in a file or a database) includes in its source code some structural definition of that data. The extent to which that program is exposed to changes made to that external source is called data dependence. A program is exposed, in the sense meant here, if some change to the external source invalidates the program and thus necessitates changes to its source code ("unproductive maintenance"). By "change to the external source" we normally mean structural changes of any kind—we assume application programs are immune to mere changes in the data content of the external source, such as addition and deletion of records and updates made to existing records.

Data independence, insofar as it is achievable, requires not only features toward that end in the database language used by the application program. It also requires the application program to make judicious use of those features. For a simple example, consider the following two SQL statements:

S1. INSERT Assignment VALUES ('s01', 'c4', 1, 80);

S2. INSERT Assignment(StudentId, CourseCode, AssignmentNumber, Mark)
VALUES ('s01', 'c4', 1, 80);

If the order of the columns in the Assignment table is changed—such that, say, the CourseCode column comes first, then statement S1 becomes incorrect whereas statement S2 still works properly. That's because in S1 the value 's01' is assigned to the first column, whichever column that is; in S2 the value 's01' is assigned to the StudentId column, wherever that column appears in the table. The DBMS supports data independence here by making S2 possible in its database language. The application that uses statement 1 instead fails to take advantage of that particular language feature and is thus data dependent.

Note that in relational theory there is no ordering at all to the attributes of a relation. In a database language that faithfully adheres to this theory, statements of the form of of S1 wouldn't be possible. Instead, each attribute value would have to be explicitly tied to its attribute name. For example, in the language **Tutorial D** the tuple to be inserted would be specified like this:

TUPLE { StudentId 's01', CourseCode 'c4', AssignmentNumber 1, Mark 80 }

or, equivalently, this:

TUPLE { CourseCode 'c4', StudentId 's01', Mark 80, AssignmentNumber 1 }

Of course there are many other ways in which applications can be exposed to data dependence, and consequently many other ways in which the DBMS can help them to avoid it.