Comments on Treatment of Constraints in Block 1

Some of the text on constraints in Block 1 can give rise to confusion for people who are very new to the subject. These comments are offered in the interests of clarification. Text from Block 1 is shown in black.

Imposing limits on the types of operation that can be applied is one example of imposing constraints on the data held in a database. (p.20)

> It is true that a constraint has the effect of disallowing certain specific update operations but a constraint isn't specified in such terms. E.g., CHECK HireDate > BirthDate has the effect of prohibiting the recording of an employee being hired on or before their birthday, but the wording suggests that a constraint has the effect of prohibiting certain types of operation, such as INSERT, DELETE, UPDATE.

Most complex databases have many rules, usually known as constraints, that define the representation and values permitted for valid data being entered into the database or that describe valid changes that can be made to existing data. (p.24)

> A constraint doesn't define representation, nor does it "describe valid changes".

… things like user access controls, constraints, backup and recovery mechanisms are all tools that can be used to protect data assets. (p.51)

> Including constraints here is likely to cause confusion. It is not usual to think of consistency (with constraints) as being involved in asset protection.

There should be a constraint imposed in the database that ensures this inconsistency cannot arise. (p.61)

> Yes, that's exactly the kind of thing a constraint is for. (We would normally say imposed on rather than imposed in.)

The database designer has implemented a constraint to ensure this inconsistency is not permitted within the database. (p.61)

> Yes, that's good too, though it is really the DBMS that implements constraints—designers just define (or *declare*) them.

Failing to identify dependencies between data items and protect that dependency with appropriate constraints can lead to the entry of data that is inconsistent, or can allow changes to data that should not be permitted. It is preferable to have the database enforce the constraints consistently for all the user applications rather than depend on the individual applications to be programmed to check the constraints. (p.61)

> This is good. It clearly shows update operations as being affected by the existence of constraints but not explicitly involved in their declaration. For example, the constraint CHECK HireDate > BirthDate does not mean, "When the HireDate is altered, make sure it's greater than the BirthDate", for that would allow the BirthDate to be altered to be greater than the HireDate!

A **logical schema** is the central component in this architecture. It defines the logical properties of data in a database, being concerned with a representation of the data and associated constraints that are independent of how it is stored in files. (p.69)

> Well, yes, but the whole of the LS is independent of "how it is stored in files", not just the constraints.

Constraint management (p.75)

This DBMS function is concerned with the definition of constraints and, since constraints are properties of data, they are included in a schema in the same way as described for data definition. Indeed, constraint definition could be considered as just one aspect of data definition, but we have described it separately to emphasise the importance of constraints as a special characteristic of data in a database.

This is a bit fuzzy. The first sentence is trying to say that it's a responsibility of the DBMS to permit constraints to be declared, and such declarations are considered to be part of the logical schema. Instead of saying, rather loosely, that a constraint is a property of the data, we would such that it is a *condition* that the data must always *satisfy*. A similar comment applies to "a special characteristic of data".

After a constraint has been defined, a DBMS must ensure that data in a database never violates the constraint – that is, a constraint is enforced automatically. Constraints are checked whenever a request to the DBMS would result in changes to the data that might affect the constraint. Note that many DBMSs allow constraint enforcement to be switched off either temporarily or permanently – this control should be exercised with great caution.

This is good. "Never violates" means the same as "always satisfies". The second sentence could be replaced by this: "Notionally, every constraint is checked whenever the database is updated. In practice, of course, we expect the DBMS to check just the constraints that reference the parts of the database affected by the update."

ACTIVITY 3.1 (p.78)

Examining the DBMS system tables

Shows that constraint definitions in SQL do not mention update operations (apart perhaps from in triggered actions such as ON DELETE CASCADE).