SQL Database Environment

2.1

Data independence is important so changes to a lower architectural level won’t affect higher levels. This leads to a lesser risk of conflicts between these levels and makes it so that users won’t notice changes besides impacts on the overall performance

2.2

The ANSI-SPARC consists of a three-level architecture, divided into an external level, which is the view that the user interacts with, a conceptual level, which contains the logical structure, a description of the data that is stored and the relationship between the different kind of data, and an internal level, which represents the physical implementation and is responsible for how data is stored within the database.

This model firstly grants physical data independence, due to the fact the internal level is separated from the conceptual level and the external level, which means that physical changes to the internal level won’t affect the other two. It secondly grants logical data independence, because the external and the conceptual level are separated as well. Therefor, changes to the conceptual level won’t affect the external level.

2.3

A data model is used to represent data and make that data understandable for users and programmers alike. This makes it easier to write the database, but also easier to understand and to use it by the users. A data model consists of three components:

- A structural part that defines guidelines on how the database is setup, how data is stored and how it is constructed.
- A manipulative part that defines how interaction with the database takes place, what language is used to get, add or edit data and rules for changing or rewrite parts of the structure of the database.
- A set of integrity constraints, to make ensure the stored data’s integrity and accuracy.

The main types of a data model are:

- External data model, which is used for getting a view over the representation of each user and the organization. It is also called “Universe of Discourse” and falls within the category of record-based data models.
- Conceptual data model, which is used to give an overview of how specific parts of the database are related to each other and defines the structure, concepts and rules of the database. It is independent of the DBMS and falls within the category of object-based data models.
- Internal data model, which is the translation of the conceptual model. It is used to interact and exchange with the actual DBMS and falls within the category of physical data models.

2.4

Conceptual modelling is used like any form of prototyping to create a first and general overview over a much more complex entity and makes it easier to get a grasp of them. It helps to form a representation of the final entity and develop it over time. It additionally straightens and structures one's own thought process. Most important work is done when creating a concept or a scheme, rather than the actual coding, that benefits greatly from this preparatory work.

This is not only important when working alone, but even more crucial when working in a team or with end-users. It allows them an easier understanding of what is going on in the background and assists them in learning the system.

SQL The Relational Model

3.4

A relation is a set of tuples with the same attributes. These tuples represent the information of the referenced object. In general, a relation comes in the form of a table and it's containing rows and columns.

3.5

A candidate key is the number of fields of a superkey, that are at the very least necessary to uniquely identify any entry in a table. It is possible for a table to have several candidate keys.

A primary key is one of the candidate keys, but the one most fitting to uniquely identify any entry in a table.

This means the difference between a candidate key and a primary key is minor. The main difference is that a primary key should be chosen, so it is the most appropriate in fulfilling the role as the identifier (shortest, best distinguishable). Each candidate key can in theory be the primary key, but not each candidate key fits similar good.

A foreign key is a candidate key of another table, used to links these two tables together.

For example: We have a table with the columns studentNumber, studentName, degree and emailAddress. studentNumber and emailAddress are candidate keys, because both are unique for each entry in the table. It might be that studentName only occurs once too, but as soon as another student with the same name comes to the university, this won't work anymore as identifier, whereas degree is the same for several students anyway. studentNumber makes a better primary key, due to its easy pattern and similar length.

If we now have a second table with the column degree and lecturerName, we can use degree as primary key, due to there only being one degree of a kind but a lecturer theoretically being responsible for multiple degrees. Let's say we want to link degree of our first table with our second table, we can use degree (primary key) of the second table as foreign key in our first table.

3.7

A view is a dynamic and customizable processing of the underlying data. Its representation is virtual and derived of the database but does not necessarily match with the actual representation of the database.

A base relation is the part physically stored within a database, whereas view only corresponds to a specific part of the underlying data, necessary to perform a specific task. A view can be customized and restricted to the particular data needed.