Database design including integrity constraints-Part-II

Hi, we will learn about the importance of logical database design in this lecture.

Creating a database logical design is one of the first important steps in designing a database. There are four logical database models that can be used hierarchical, network, relational, or object-oriented. The design concepts should first start with a data model and those models have to b transformed to the physical considerations which will be specific to the DBMS, that is going to be selected.

What exactly is logical database design?

Logical modeling deals with gathering business requirements and converting those requirements into a model. The logical model revolves around the needs of the business, not the database, although the needs of the business are used to establish the needs of the database. Logical modeling involves gathering information about business processes, business entities (categories of data), and organizational units. After this information is gathered, diagrams and reports are produced including entity relationship diagrams, business process diagrams, and eventually process flow diagrams. The diagrams produced should show the processes and data that exist, as well as the relationships between business processes and data. Logical modeling should accurately render a visual representation of the activities and data relevant to a particular business.

Logical modeling affects not only the direction of database design, but also indirectly affects the performance and administration of an implemented database. When time is invested performing logical modeling, more options become available for planning the design of the physical database.

The diagrams and documentation generated during logical modeling is used to determine whether the requirements of the business have been completely gathered. Management, developers, and end users alike review these diagrams and documentation to determine if more work is required before physical modeling commences.

Logical Modeling deliverables

Typical deliverables of logical modeling include:-

An Entity Relationship Diagram is also referred to as an analysis ERD. The point of the initial ERD is to provide the development team with a picture of the different categories of data for the business, as well as how these categories of data are related to one another.

The **Business process model** illustrates all the parent and child processes that are performed by individuals within a company. The process model gives the development team an idea of how data moves within the organization. Because process models illustrate the activities of individuals in the company, the process model can be used to determine how a database application interface is design.

So the logical database design is the process of constructing a model of information used in tan enterprise based on a specific data model, but independent of a particular DBMS or other physical considerations.

Why a logical data model is required?

A **logical data model is required** before you can even begin to design a physical database. And the logical data model grows out of a conceptual data model. And any type of data model begins with the discipline of data modeling.

The first objective of conceptual data modeling is to **understand the requirements**. A data model, in and of itself, is of limited value. Of course, a data model delivers value by enhancing communication and understanding, and it can be argued that these are quite valuable. But the primary value of a data model is its ability to be used as a blueprint to build a physical database.

When databases are built from a well-designed data model the resulting structures provide increased value to the organization. The value derived from the data model exhibits itself in the form of minimized redundancy, maximized data integrity, increased stability, better data sharing, increased consistency, more timely access to data, and better

usability. These qualities are achieved because the data model clearly outlines the data resource requirements and relationships in a clear, concise manner. Building databases from a data model will result in a better database implementation because you will have a better understanding of the data to be stored in your databases.

Another benefit of data modeling is the ability to discover new uses for data. A data model can clarify data patterns and potential uses for data that would remain hidden without the data blueprint provided by the data model. Discovery of such patterns can change the way your business operates and can potentially lead to a competitive advantage and increased revenue for your organization.

Data modeling requires a different mindset than requirements gathering for application development and process-oriented tasks. It is important to think "what" is of interest instead of "how" tasks are accomplished. To transition to this alternate way of thinking, follow these three "rules":

- Don't think physical; think conceptual – do not concern yourself with physical storage issues and the constraints of any DBMS you may know. Instead, concern yourself with business issues and terms.
- Don't think process; think structure – how something is done, although important for application development, is not important for data modeling. The things that processes are being done to are what is important to data modeling.
- Don't think navigation; think relationship – the way that things are related to one another is important because relationships map the data model blueprint. The way in which relationships are traversed is unimportant to conceptual and logical data modeling.

E-R Diagrams- A Graphical format for data modeling

Data models are typically rendered in a graphical format using an entity-relationship diagram, or E/R diagram for short. An E/R diagram graphically depicts the entities and relationships of a data model. There are many popular data modeling tools on the market

from a variety of vendors. But do not confuse the tool as being more important than the process. Of what use is a good tool if you do not know how to deploy it?

A data model is built using many different components acting as abstractions of real world things. The simplest data model will consist of entities and relationships. As work on the data model progresses, additional detail and complexity is added. Let's examine the many different components of a data model and the terminology used for data modeling.

The first building block of the data model is the entity. An entity, at a very basic level, is something that exists and is capable of being described. It is a person, place, thing, concept, or event about which your organization maintains facts. For example: "STUDENT," "INSTRUCTOR," and "COURSE" are specific entities about which a college or university must be knowledgeable to perform its business.

Entities are comprised of **attributes**. An attribute is a characteristic of an entity. Every attribute does one of three things:

1. Describe – An attribute is descriptive if it does not identify or relate, but is used to depict or express a characteristic of an entity occurrence.
2. Identify – An attribute that identifies is a candidate key. If the values of an identifying attribute changes, it should identify a different entity occurrence. An attribute that identifies should be unchangeable and immutable.
3. Relate – An attribute that relates entities is a foreign key; the attribute refers to the primary key attribute of an occurrence of another (or the same) entity.

Each attribute is assigned a **domain** that defines the type of data, its size, and the valid values that can be assigned to the attribute. As a general rule of thumb, nouns tend to be entities and adjectives tend to be attributes. But, of course, this is not a hard and fast rule: be sure to apply of the business to determine which nouns and attributes are entities and which are attributes. Every attribute must either identify the entity occurrence, describe the entity occurrence, or relate the entity occurrence to another entity occurrence (in the same or another entity).

Relationships define how the different entities are associated with each other. Each relationship is named such that it describes the role played by an entity in its association with another (or perhaps the same) entity. A relationship is defined by the keys of the participating entities: the primary key in the parent entity and the foreign key in the dependent entity. Relationships are not just the "lines" that connect entities, but provide meaning to the data model and must be assigned useful names.

Keep in mind that as you create your data models, you are developing the lexicon of your organization's business. Much like a dictionary functions as the lexicon of words for a given language, the data model functions as the lexicon of business terms and their usage. Of course, this short introduction just scrapes the tip of the data modeling iceberg.

Bad logical database design results in bad physical database design, and generally results in poor database performance. So, if it is your responsibility to design a database from scratch, be sure you take the necessary time and effort to get the logical database design right. Once the logical design is right, then you also need to take the time to get the physical design right.

Review Questions

1. What do you mean by logical database design?
2. What are the deliverables for logical modeling?

References

http://www.microsoft-accesssolutions.co.uk

Date, C.J., Introduction to Database Systems (7th Edition) Addison Wesley, 2000

Leon, Alexis and Leon, Mathews, Database Management Systems, LeonTECHWorld.