

Database Design



Introduction

We might have come across the word "**Database**" quite often. This term carries a high emphasis on its arms. More often, it is not just related to the developer's perspective but is quite often used with non-tech groups or communities. Technically, a database is more of a storage term used to denote the relationship with different forms of data that are coagulated in a single place. Thus, we can define a database as an organized collection of data, generally stored and accessed electronically through computer systems. This article is highly centric to the database design and its association with citable terms and methodologies was commonly taken into account. We'll be discussing those terms concerning database design to understand the bits and pieces. Let's talk about it straight away.

What is Database Design?

Database design can be generally defined as a collection of tasks or processes that enhance the designing, development, implementation, and maintenance of enterprise data management system. Designing a proper database reduces the maintenance cost thereby improving data consistency and the cost-effective measures are greatly influenced in terms of disk storage space. Therefore, there has to be a brilliant concept of designing a database. The designer should follow the constraints and decide how the elements correlate and what kind of data must be stored.

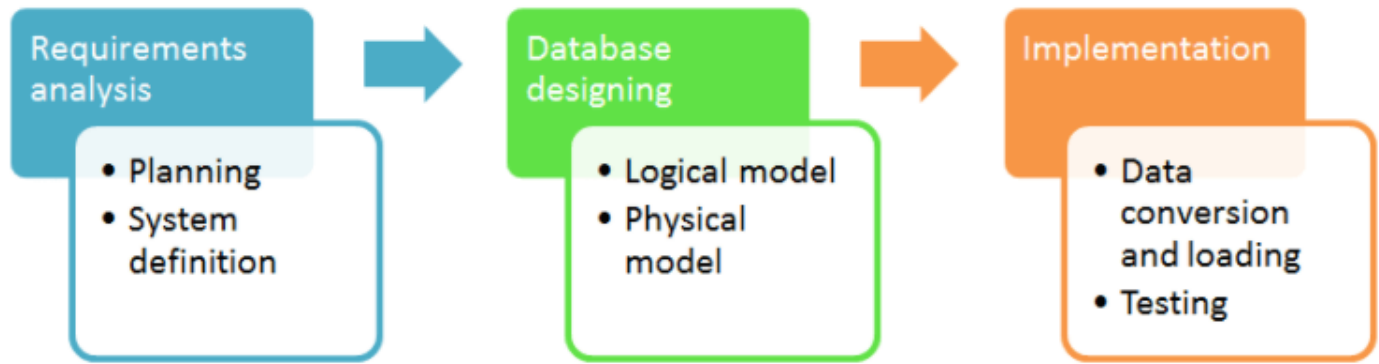
The main objectives behind database designing are to produce physical and logical design models of the proposed database system. To elaborate this, the logical model is primarily concentrated on the requirements of data and the considerations must be made in terms of monolithic considerations and hence the stored physical data must be stored independent of the physical conditions. On the other hand, the physical database design model includes a translation of the logical design model of the database by keep control of physical media using hardware resources and software systems such as Database Management System (DBMS).

Why is Database Design important?

The important consideration that can be taken into account while emphasizing the importance of database design can be explained in terms of the following points given below.

1. Database designs provide the blueprints of how the data is going to be stored in a system. A proper design of a database highly affects the overall performance of any application.
2. The designing principles defined for a database give a clear idea of the behavior of any application and how the requests are processed.
3. Another instance to emphasize the database design is that a proper database design meets all the requirements of users.
4. Lastly, the processing time of an application is greatly reduced if the constraints of designing a highly efficient database are properly implemented.

Life Cycle



Although, the life cycle of a database is not an important discussion that has to be taken forward in this article because we are focused on the database design. But, before jumping directly on the designing models constituting database design it is important to understand the overall workflow and life-cycle of the database.

Requirement Analysis

First of all, the planning has to be done on what are the basic requirements of the project under which the design of the database has to be taken forward. Thus, they can be defined as:-

Planning - This stage is concerned with planning the entire DDLC (Database Development Life Cycle). The strategic considerations are taken into account before proceeding.

System definition - This stage covers the boundaries and scopes of the proper database after planning.

Database Designing

The next step involves designing the database considering the user-based requirements and splitting them out into various models so that load or heavy dependencies on a single aspect are not imposed. Therefore, there has been some model-centric approach and that's where logical and physical models play a crucial role.

Physical Model - The physical model is concerned with the practices and implementations of the logical model.

Logical Model - This stage is primarily concerned with developing a model based on the proposed requirements. The entire model is designed on paper without any implementation or adopting DBMS considerations.

Implementation

The last step covers the implementation methods and checking out the behavior that matches our requirements. It is ensured with continuous integration testing of the database with different data sets and conversion of data into machine understandable language. The manipulation of data is primarily focused on these steps where queries are made to run and check if the application is designed satisfactorily or not.

Data conversion and loading - This section is used to import and convert data from the old to the new system.

Testing - This stage is concerned with error identification in the newly implemented system. Testing is a crucial step because it checks the database directly and compares the requirement specifications.

Database Design Process

The process of designing a database carries various conceptual approaches that are needed to be kept in mind. An ideal and well-structured database design must be able to:

1. Save disk space by eliminating redundant data.
2. Maintains data integrity and accuracy.
3. Provides data access in useful ways.
4. Comparing Logical and Physical data models.

Logical

A logical data model generally describes the data in as many details as possible, without having to be concerned about the physical implementations in the database. Features of logical data model might include:

1. All the entities and relationships amongst them.
2. Each entity has well-specified attributes.

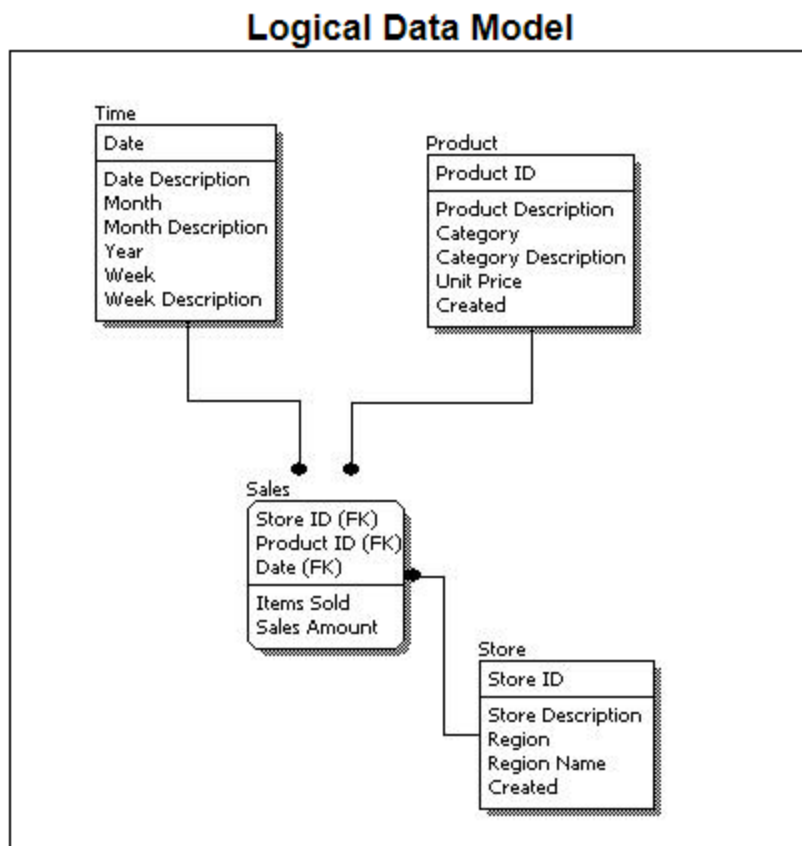
3. The primary key for each entity is specified.
4. Foreign keys which are used to identify a relationship between different entities are specified.
5. Normalization occurs at this level.

A logical model can be designed using the following approach:

1. Specify all the entities with primary keys.
2. Specify concurrent relationships between different entities.
3. Figure out each entity attributes
4. Resolve many-to-many relationships.
5. Carry out the process of normalization.

Also, one important factor after following the above approach is to critically examine the design based on requirement gathering. If the above steps are strictly followed, there are chances of creating a highly efficient database design that follows the native approach.

To understand these points, see the image below to get a clear picture.



If we compare the logical data model as shown in the figure above with some sample data in the diagram, we can come up with facts that in a conceptual data model there are no presence of a primary key whereas a logical data model has primary keys for all of its attributes. Also, logical data model the cover relationship between different entities and carries room for foreign keys to establish relationships among them.

Physical

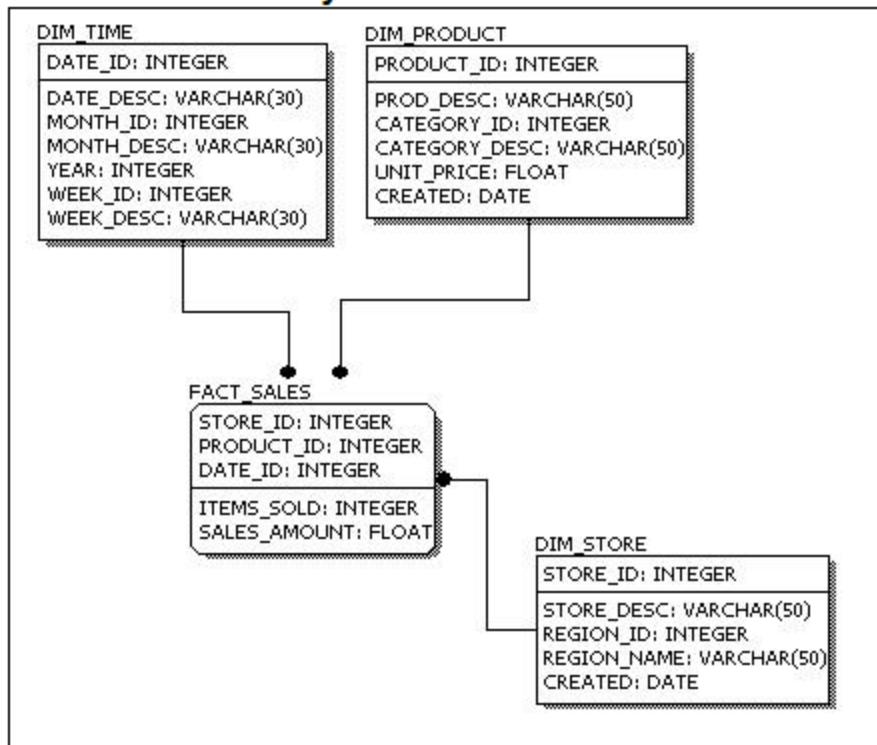
A Physical data mode generally represents how the approach or concept of designing the database. The main purpose of the physical data model is to show all the **structures** of the table including the **column name, column data type, constraints, keys(primary and foreign)**, and the relationship among tables. The following are the features of a physical data model:

1. Specifies all the columns and tables.
2. Specifies foreign keys that usually define the relationship between tables.
3. Based on user requirements, de-normalization might occur.
4. Since the physical consideration is taken into account so there will straightforward reasons for difference than a logical model.
5. Physical models might be different for different RDBMS. For example, the data type column may be different in MySQL and SQL Server.

While designing a physical data model, the following points should be taken into consideration:

1. Convert the entities into tables.
2. Convert the defined relationships into foreign keys.
3. Convert the data attributes into columns.
4. Modify the data model constraints based on physical requirements.

Physical Data Model



Comparing this physical data model with the logical with the previous logical model, we might conclude the differences that in a physical database entity names are considered table names and attributes are considered column names. Also, the data type of each column is defined in the physical model depending on the actual database used.

Glossary

Entity - An entity in the database can be defined as abstract data that we save in our database. For example, a customer, products.

Attributes - An attribute is a detailed form of data consisting of entities like length, name, price, etc.

Relationship - A relationship can be defined as the connection between two entities or figures. For example, a person can relate to multiple persons in a family.

Foreign key - It acts as a referral to the Primary Key of another table. A foreign key contains columns with values that exist only in the primary key column they refer to.

Primary key - A primary key is the pointer of records that is unique and not null and is used to uniquely identify attributes of a table.

Normalization - A flexible data model needs to follow certain rules. Applying these rules is called normalizing.

Summary

Database design is a method of identifying the gaps and opportunities of designing a proper utilization method. It is the main component of a system that gives a blueprint of the data and its behavior inside the system. A proper database design is always kept on priority due to the user requirements being kept excessively high and following up with the constraint practices of designing a database might only stand as a chance to gain the requested efficiency. Moreover, we also learned separately about the different design models that portray the ideal database design along with the limitless discussion on their properties and how to make use of them. Furthermore, we learned how the life-cycle of a database decides the design of the database and how to put the concept of design into the life-cycle methods so that efficient and highly sophisticated databases can be designed based on the user requirements.

[← Prev](#)[Next →](#)

For Videos Join Our Youtube Channel: [Join Now](#)

Feedback

- Send your Feedback to feedback@javatpoint.com

Help Others, Please Share



Learn Latest Tutorials