

Logical design database part ii essay examples

[Business](#), [Company](#)



\n[[toc title="Table of Contents"](#)]\n

\n \t

1. [Winfry Paul](#) \n \t
2. [Normalizing steps](#) \n \t
3. [Second Normal Form 2NF](#) \n \t
4. [And](#) \n \t
5. [Third Normal Form 3NF](#) \n \t
6. [Reference](#) \n

\n[/toc]\n \n

Winfry Paul

ACME Consulting Company.

Normalization is the practice of sorting out data in a database by building tables and establishing relationships between them in order to safeguard existing data from unintentional modification and attain flexibility.

Redundant data wastes disk space and create maintenance problems while inconsistent dependencies make data access difficulties due to missing or broken parts.

Normalized database will performs more efficiently in applications with more writer-intensive operations than the read load. Because of the smaller tables associated with normalized tables, the footprint is small and data are vertically divided. This enhances performance since they can easily fit into the buffer. Updating the database is usually faster since the data is located in a single place without duplication. Likewise the inserts are done in a single place without the need to duplicate hence faster performance.

According to when dealing with data that is not duplicated, there is less need to generate heavy duty queries and groups. The means of searching, sorting and cleaning indexes will be made faster with more and narrower tables that can fit on a single data page. More tables will also allow the utilization of segments to control physical posting of data.

When it comes to data modification, few indexes per table will lead to faster commands and reduced modification anomalies. Fully normalized databases containing indexes on the join columns that generally perform faster and the associated cost of finding rows in the data cache is extremely low.

However in some instances, full normalization of the data is not desirable. Denormalized databases are not efficient when dealing with heavy-write load. Denormalized database present data in same table hence do not require joins. For this reason, select commands are incredibly faster but because data is duplicated, updates and inserts become complex and very expensive.

Redundant data refers to data stored in more than one table. It requires vast resources to maintain them by updating and deleting. The aim of normalization is to create a clean database that is efficient for reporting and analyzing purposes.

The database community has established formal guidelines that are aimed at ensuring standard normalized databases. The normal forms are ranged from 1NF TO 3NF representing the lowest form through to the highest. The ACME consulting company will benefit from data normalization by mapping out all clear and advance data types to be stored. Further, normalizing from the existing ER model serves to sort out generalized and specialized data

sharing various attributes between them.

Normalizing the restaurant database in our ER model will save the disk space as well as eliminate the inconsistency dependency of data tables.

Normalizing steps

First Normal Form 1NF

This first form lays down the basic two guidelines for an organized database.

It includes;

Considering the restaurant customer meal table the above steps demonstrate the process of normalizing.

- Do away with duplicate columns from the same table.

Since one customer may opt to have two or more varieties of a meal such as an appetizer and the main course it is prudent to remove the fields, meal1, meal2, and create a table without repeated groups.

- Form separate tables for each group of correlated data and make out each row by using a distinctive set of primary keys.

Second Normal Form 2NF

This form concentrates on the concept of eliminating duplicate data in the following ways;

It is noted that there exist multiple meal values for each customer. Meals are not dependent on the primary key (CustomerID) so this relationship should be eliminated in the second normal form.

- Fulfill all the guidelines of 1NF to give;

- Delete subsets of data that are used in manifold rows of a table and put them in a separate table to yield; .

And

- Form relationships that link the two tables and their precursor through the use of foreign keys.

Third Normal Form 3NF

In this form the two rules implemented include;

- Satisfy all the requirement of 2NF

Since the table/location is functionally dependent on the Waiter attributes it is important to remove it from the Customer table and put it on the Waiters table.

- Delete columns that do not depend on the primary key.

Finally we will have the Waiters records table as;

Reference

Carlos Coronel, S. M. (2012). Database Systems: Design, Implementation, and Management. Cengage Learning.

Ralph Kimball, M. R. (2011). The Data Warehouse Lifecycle Toolkit. John Wiley & Sons.

Shouhong Wang, H. W. (2011). Database Design and Implementation. Universal-Publishers.