Your [**continued donations**](http://wikimediafoundation.org/wiki/Fundraising) keep Wikipedia running!

# Database normalization

### From Wikipedia, the free encyclopedia

Jump to: [navigation](http://en.wikipedia.org/wiki/Database_normalization#column-one), [search](http://en.wikipedia.org/wiki/Database_normalization#searchInput)

**This article or section is in need of attention from an expert on the subject**.

Please help recruit one, or [improve this page yourself](http://en.wikipedia.org/w/index.php?title=Database_normalization&action=edit) if you can. See [discussion page](http://en.wikipedia.org/wiki/Talk:Database_normalization) for details.

This article or section may be **confusing or unclear** for some readers,

and should be edited to rectify this.

Please improve the article, or discuss the issue on the [talk page](http://en.wikipedia.org/wiki/Talk:Database_normalization).

In [relational database](http://en.wikipedia.org/wiki/Relational_database) theory, **normalization** is the process of restructuring the [logical data model](http://en.wikipedia.org/wiki/Logical_data_model) of a database to eliminate [redundancy](http://en.wikipedia.org/wiki/Redundancy_(information_theory)), organize [data](http://en.wikipedia.org/wiki/Data) efficiently, reduce repeating data and to reduce the potential for anomalies during [data operations](http://en.wikipedia.org/wiki/CRUD_(acronym)). Data normalization also may improve data consistency and simplify future extension of the logical data model. The formal classifications used for describing a relational database's level of normalization are called [**normal forms**](http://en.wikipedia.org/wiki/Normal_form) (abbrev. **NF**).

| Contents  * [1 Description](http://en.wikipedia.org/wiki/Database_normalization#Description) * [2 History](http://en.wikipedia.org/wiki/Database_normalization#History) * [3 Normal forms](http://en.wikipedia.org/wiki/Database_normalization#Normal_forms)   + [3.1 First normal form](http://en.wikipedia.org/wiki/Database_normalization#First_normal_form)   + [3.2 Second normal form](http://en.wikipedia.org/wiki/Database_normalization#Second_normal_form)   + [3.3 Third normal form](http://en.wikipedia.org/wiki/Database_normalization#Third_normal_form)   + [3.4 Boyce-Codd normal form](http://en.wikipedia.org/wiki/Database_normalization#Boyce-Codd_normal_form)   + [3.5 Fourth normal form](http://en.wikipedia.org/wiki/Database_normalization#Fourth_normal_form)   + [3.6 Fifth normal form](http://en.wikipedia.org/wiki/Database_normalization#Fifth_normal_form)   + [3.7 Domain/key normal form](http://en.wikipedia.org/wiki/Database_normalization#Domain.2Fkey_normal_form)   + [3.8 Sixth normal form](http://en.wikipedia.org/wiki/Database_normalization#Sixth_normal_form) * [4 Denormalization](http://en.wikipedia.org/wiki/Database_normalization#Denormalization) * [5 Non-first normal form (NF²)](http://en.wikipedia.org/wiki/Database_normalization#Non-first_normal_form_.28NF.C2.B2.29) * [6 Further reading](http://en.wikipedia.org/wiki/Database_normalization#Further_reading) * [7 References](http://en.wikipedia.org/wiki/Database_normalization#References) * [8 External links](http://en.wikipedia.org/wiki/Database_normalization#External_links) * [9 See also](http://en.wikipedia.org/wiki/Database_normalization#See_also) |
| --- |

## **[**[**edit**](http://en.wikipedia.org/w/index.php?title=Database_normalization&action=edit&section=1)**]** **Description**

A non-normalized database can suffer from data anomalies:

* A non-normalized database may store data representing a particular referent in multiple locations. An update to such data in some but not all of those locations results in an **update anomaly**, yielding inconsistent data. A normalized database prevents such an anomaly by storing such data (i.e. data other than [primary keys](http://en.wikipedia.org/wiki/Primary_key)) in only one location.
* A non-normalized database may have inappropriate dependencies, i.e. relationships between data with no functional dependencies. Adding data to such a database may require first adding the unrelated dependency. A normalized database prevents such **insertion anomalies** by ensuring that database relations mirror functional dependencies.
* Similarly, such dependencies in non-normalized databases can hinder deletion. That is, deleting data from such databases may require deleting data from the inappropriate dependency. A normalized database prevents such **deletion anomalies** by ensuring that all records are uniquely identifiable and contain no extraneous information.

Normalized databases have a [design](http://en.wikipedia.org/wiki/Database_design) that reflects the true [dependencies](http://en.wikipedia.org/wiki/Functional_dependency) between tracked quantities, allowing quick updates to data with little risk of introducing inconsistencies. Instead of attempting to lump all information into one table, data is spread out logically into many tables. Normalizing the data is decomposing a single relation into a set of smaller relations which satisfy the constraints of the original relation. Redundancy can be solved by decomposing the tables. However certain new problems are caused by decomposition. Normalization helps us to make a conscious decision to avoid redundancy keeping the pros and cons in mind.

One can only describe a database as having a normal form if the relationships between quantities have been rigorously defined. It is possible to use [set theory](http://en.wikipedia.org/wiki/Set_theory) to express this knowledge once a problem domain has been fully understood, but most database designers model the relationships in terms of an "idealized schema". (The mathematical support came back into play in proofs regarding the process of transforming from one form to another.)

The transformation of conceptual model to computer representation format is known as Normalization.

## **[**[**edit**](http://en.wikipedia.org/w/index.php?title=Database_normalization&action=edit&section=2)**]** **History**

[Edgar F. Codd](http://en.wikipedia.org/wiki/Edgar_F._Codd) first proposed the process of normalization and what came to be known as the **1st normal form**:

| **“** | There is, in fact, a very simple elimination\* procedure which we shall call normalization. Through decomposition nonsimple domains are replaced by "*domains whose elements are atomic (nondecomposable) values.*" | **„** |
| --- | --- | --- |
| —Edgar F. Codd, A Relational Model of Data for Large Shared Data Banks[[1]](http://en.wikipedia.org/wiki/Database_normalization#_note-0) | | |

\* His term *eliminate* is misleading, as nothing is "lost" in normalization. He probably described *eliminate* in a mathematical sense to mean elimination of complexity.

In his paper, Codd used the term "nonsimple" domains to describe a heterogeneous data structure, but later researchers would refer to such a structure as an [abstract data type](http://en.wikipedia.org/wiki/Abstract_data_type).

## **[**[**edit**](http://en.wikipedia.org/w/index.php?title=Database_normalization&action=edit&section=3)**]** **Normal forms**

[Edgar F. Codd](http://en.wikipedia.org/wiki/Edgar_F._Codd) originally defined the first three normal forms. The first normal form requires that tables be made up of a primary key and a number of atomic fields, and the second and third deal with the relationship of non-key fields to the [primary key](http://en.wikipedia.org/wiki/Primary_key). These have been summarised as requiring that all non-key fields be dependent on "the key, the whole key and nothing but the key". In practice, most applications in 3NF are fully normalized. However, research has identified potential update anomalies in 3NF databases. [BCNF](http://en.wikipedia.org/wiki/Database_normalization#Boyce-Codd_normal_form) is a further refinement of 3NF that attempts to eliminate such anomalies.

The fourth and fifth normal forms (4NF and 5NF) deal specifically with the representation of many-many and one-many relationships. Sixth normal form (6NF) only applies to [temporal databases](http://en.wikipedia.org/wiki/Temporal_database).

### **[**[**edit**](http://en.wikipedia.org/w/index.php?title=Database_normalization&action=edit&section=4)**]** **First normal form**

*Main article: [First normal form](http://en.wikipedia.org/wiki/First_normal_form)*

**First normal form** (**1NF**) lays the groundwork for an organised database design:

* Ensure that each table has a primary key: minimal set of attributes which can uniquely identify a record.
* Eliminate repeating groups (categories of data which would seem to be required a different number of times on different records) by defining keyed and non-keyed attributes appropriately.
* Atomicity: Each attribute must contain a single value, not a set of values.

### **[**[**edit**](http://en.wikipedia.org/w/index.php?title=Database_normalization&action=edit&section=5)**]** **Second normal form**

*Main article: [Second normal form](http://en.wikipedia.org/wiki/Second_normal_form)*

**Second normal form** (**2NF**) If a table has a composite key, all attributes must be related to the whole key:

* The database must meet all the requirements of the first normal form.
* Data which is redundantly duplicated across multiple rows of a table is moved out to a separate table.

### **[**[**edit**](http://en.wikipedia.org/w/index.php?title=Database_normalization&action=edit&section=6)**]** **Third normal form**

*Main article: [Third normal form](http://en.wikipedia.org/wiki/Third_normal_form)*

**Third normal form** (**3NF**) requires that data stored in a table be dependent only on the primary key, and not on any other field in the table.

* The database must meet all the requirements of the second normal form.
* Any field which is dependent not only on the primary key but also on another field is moved out to a separate table.

### **[**[**edit**](http://en.wikipedia.org/w/index.php?title=Database_normalization&action=edit&section=7)**]** **Boyce-Codd normal form**

*Main article: [Boyce-Codd normal form](http://en.wikipedia.org/wiki/Boyce-Codd_normal_form)*

**Boyce-Codd normal form** (or **BCNF**) requires that there be no non-trivial functional dependencies of attributes on something other than a superset of a candidate key (called a [superkey](http://en.wikipedia.org/wiki/Superkey)).

### **[**[**edit**](http://en.wikipedia.org/w/index.php?title=Database_normalization&action=edit&section=8)**]** **Fourth normal form**

*Main article: [Fourth normal form](http://en.wikipedia.org/wiki/Fourth_normal_form)*

**Fourth normal form** (or **4NF**) requires that there be no non-trivial [multivalued dependencies](http://en.wikipedia.org/wiki/Multivalued_dependency) of attribute sets on something other than a superset of a candidate key. A table is said to be in 4NF [if and only if](http://en.wikipedia.org/wiki/If_and_only_if) it is in the BCNF and [multivalued dependencies](http://en.wikipedia.org/wiki/Multivalued_dependency) are [functional dependencies](http://en.wikipedia.org/wiki/Functional_dependency). The 4NF removes unwanted data structures: [multivalued dependencies](http://en.wikipedia.org/wiki/Multivalued_dependency).

### **[**[**edit**](http://en.wikipedia.org/w/index.php?title=Database_normalization&action=edit&section=9)**]** **Fifth normal form**

*Main article: [Fifth normal form](http://en.wikipedia.org/wiki/Fifth_normal_form)*

**Fifth normal form** (**5NF** and also **PJ/NF**) requires that there are no non-trivial join dependencies that do not follow from the key constraints. A table is said to be in the 5NF [if and only if](http://en.wikipedia.org/wiki/If_and_only_if) it is in 4NF and every join dependency in it is implied by the candidate keys.

### **[**[**edit**](http://en.wikipedia.org/w/index.php?title=Database_normalization&action=edit&section=10)**]** **Domain/key normal form**

*Main article: [Domain/key normal form](http://en.wikipedia.org/wiki/Domain/key_normal_form)*

**Domain/key normal form** (or **DKNF**) requires that the database contains no constraints other than domain constraints and key constraints.

### **[**[**edit**](http://en.wikipedia.org/w/index.php?title=Database_normalization&action=edit&section=11)**]** **Sixth normal form**

It has been suggested that this section be split into a new article entitled [*Sixth normal form*](http://en.wikipedia.org/wiki/Sixth_normal_form). ([Discuss](http://en.wikipedia.org/wiki/Talk:Database_normalization))

This normal form was, as of 2005, only recently proposed: the **sixth normal form** (**6NF**) was only defined when extending the relational model to take into account the [temporal](http://en.wikipedia.org/wiki/Time) [dimension](http://en.wikipedia.org/wiki/Dimension). Unfortunately, most current SQL technologies as of 2005 do not take into account this work, and most temporal extensions to SQL are not relational. See work by Date, Darwen and Lorentzos[[2]](http://en.wikipedia.org/wiki/Database_normalization#_note-DBDebunk) for a relational temporal extension, or see TSQL2 for a different approach.

**Tree structure data** implementation or **Logic-Based Database** [Hierarchical model](http://en.wikipedia.org/wiki/Hierarchical_model) using [relational model](http://en.wikipedia.org/wiki/Relational_model) needs to be considered in addition to **normalization** for a useful [database design](http://en.wikipedia.org/wiki/Database_design).

## **[**[**edit**](http://en.wikipedia.org/w/index.php?title=Database_normalization&action=edit&section=12)**]** **Denormalization**

*Main article: [Denormalization](http://en.wikipedia.org/wiki/Denormalization)*

Databases intended for [Online Transaction Processing (OLTP)](http://en.wikipedia.org/wiki/Online_transaction_processing) are typically more normalized than databases intended for [On Line Analytical Processing (OLAP)](http://en.wikipedia.org/wiki/OLAP). OLTP Applications are characterized by a high volume of small transactions such as updating a sales record at a super market checkout counter. The expectation is that each transaction will leave the database in a consistent state. By contrast, databases intended for OLAP operations are primarily "read only" databases. OLAP applications tend to extract historical data that has accumulated over a long period of time. For such databases, redundant or "denormalized" data may facilitate [Business Intelligence](http://en.wikipedia.org/wiki/Business_Intelligence) applications. Specifically, [dimensional tables](http://en.wikipedia.org/wiki/Dimension_table) in a [star schema](http://en.wikipedia.org/wiki/Star_schema) often contain denormalized data. The denormalized or redundant data must be carefully controlled during [ETL](http://en.wikipedia.org/wiki/ETL) processing, and users should not be permitted to see the data until it is in a consistent state. The normalized alternative to the star schema is the [snowflake schema](http://en.wikipedia.org/wiki/Snowflake_schema).

Denormalization is also used to improve performance on smaller computers as in computerized cash-registers. Since these use the data for look-up only (e.g. price lookups), no changes are to be made to the data and a swift response is crucial.

## **[**[**edit**](http://en.wikipedia.org/w/index.php?title=Database_normalization&action=edit&section=13)**]** **Non-first normal form (NF²)**

In recognition that denormalization can be deliberate and useful, the non-first normal form is a definition of database designs which do not conform to the first normal form, by allowing "sets and sets of sets to be attribute domains" (Schek 1982). This extension introduces hierarchies in relations.

Consider the following table:

Non-First Normal Form

| Person | Favorite Colors |
| --- | --- |
| Bob | blue, red |
| Jane | green, yellow, red |

Assume a person has several favorite colors. Obviously, favorite colors consist of a set of colors modeled by the given table.

To transform this NF² table into a 1NF an "unnest" operator is required which extends the relational algebra of the higher normal forms. The reverse operator is called "nest" which is not always the mathematical inverse of "unnest", although "unnest" is the mathematical inverse to "nest". Another constraint required is for the operators to be [bijective](http://en.wikipedia.org/wiki/Bijection), which is covered by the [Partitioned Normal Form](http://en.wikipedia.org/w/index.php?title=Partitioned_Normal_Form&action=edit) ([PNF](http://en.wikipedia.org/w/index.php?title=PNF&action=edit)).

## **[**[**edit**](http://en.wikipedia.org/w/index.php?title=Database_normalization&action=edit&section=14)**]** **Further reading**

* [Litt's Tips: Normalization](http://www.troubleshooters.com/littstip/ltnorm.html)
* Date, C. J., & Lorentzos, N., & Darwen, H. (2002). [*Temporal Data & the Relational Model*](http://www.elsevier.com/wps/product/cws_home/680662) (1st ed.). Morgan Kaufmann. [ISBN 1-55860-855-9](http://en.wikipedia.org/w/index.php?title=Special:Booksources&isbn=1558608559).
* Date, C. J. (1999), [*An Introduction to Database Systems*](http://www.aw-bc.com/catalog/academic/product/0,1144,0321197844,00.html) (8th ed.). Addison-Wesley Longman. [ISBN 0-321-19784-4](http://en.wikipedia.org/w/index.php?title=Special:Booksources&isbn=0321197844).
* Kent, W. (1983) [*A Simple Guide to Five Normal Forms in Relational Database Theory*](http://www.bkent.net/Doc/simple5.htm), Communications of the ACM, vol. 26, pp. 120-125
* Date, C.J., & Darwen, H., & Pascal, F. *[Database Debunkings](http://www.dbdebunk.com/)*
* H.-J. Schek, P.Pistor Data Structures for an Integrated Data Base Management and Information Retrieval System

## **[**[**edit**](http://en.wikipedia.org/w/index.php?title=Database_normalization&action=edit&section=15)**]** **References**

1. [**^**](http://en.wikipedia.org/wiki/Database_normalization#_ref-0) [Codd, Edgar F.](http://en.wikipedia.org/wiki/Edgar_F._Codd) (June 1970). "[A Relational Model of Data for Large Shared Data Banks](http://www.acm.org/classics/nov95/toc.html)". [*Communications of the ACM*](http://en.wikipedia.org/wiki/Communications_of_the_ACM) **13** (6): 377-387.
2. [**^**](http://en.wikipedia.org/wiki/Database_normalization#_ref-DBDebunk_0) [DBDebunk](http://www.dbdebunk.com/page/page/621935.htm).

## **[**[**edit**](http://en.wikipedia.org/w/index.php?title=Database_normalization&action=edit&section=16)**]** **External links**

* [Database Normalization Basics](http://databases.about.com/od/specificproducts/a/normalization.htm) by Mike Chapple (About.com)
* [Database Normalization Intro](http://www.databasejournal.com/sqletc/article.php/1428511)
* [An Introduction to Database Normalization](http://dev.mysql.com/tech-resources/articles/intro-to-normalization.html) by Mike Hillyer.
* [Normalization](http://www.utexas.edu/its/windows/database/datamodeling/rm/rm7.html) by ITS, University of Texas.
* [Rules of Data Normalization](http://www.datamodel.org/NormalizationRules.html) by Data Model.org
* [A tutorial on the first 3 normal forms](http://phlonx.com/resources/nf3/) by Fred Coulson
* [Free PDF poster available](http://www.marcrettig.com/poster/) by Marc Rettig
* [Description of the database normalization basics](http://support.microsoft.com/kb/283878) by Microsoft

## **[**[**edit**](http://en.wikipedia.org/w/index.php?title=Database_normalization&action=edit&section=17)**]** **See also**

* [Aspect (functional perspectives orthogonal to core concern)](http://en.wikipedia.org/wiki/Aspect_(computer_science))
* [Core Concern (the intrinsic and inherent core functionality of a class)](http://en.wikipedia.org/w/index.php?title=Core_Concern&action=edit)
* [Cross-cutting Concern (common functionality orthogonal to core concern)](http://en.wikipedia.org/wiki/Cross-cutting_concern)
* [Inheritance semantics](http://en.wikipedia.org/wiki/Inheritance_semantics)
* [Functional normalization (object-oriented programming)](http://en.wikipedia.org/wiki/Functional_normalization_(object-oriented_programming))
* [Orthogonalization (mathematical normalization)](http://en.wikipedia.org/wiki/Orthogonalization)
* [Refactoring (restructuring software)](http://en.wikipedia.org/wiki/Refactoring)

| **Topics in** [**database management systems**](http://en.wikipedia.org/wiki/Database_management_system) **(DBMS)** ( [view](http://en.wikipedia.org/wiki/Template:Databases) • [talk](http://en.wikipedia.org/wiki/Template_talk:Databases) • [edit](http://en.wikipedia.org/w/index.php?title=Template:Databases&action=edit) ) |
| --- |
| *Concepts*  [Database](http://en.wikipedia.org/wiki/Database) | [Database model](http://en.wikipedia.org/wiki/Database_model) | [Relational database](http://en.wikipedia.org/wiki/Relational_database) | [Relational model](http://en.wikipedia.org/wiki/Relational_model) | [Relational algebra](http://en.wikipedia.org/wiki/Relational_algebra) | [Primary key](http://en.wikipedia.org/wiki/Primary_key) - [Foreign key](http://en.wikipedia.org/wiki/Foreign_key) - [Surrogate key](http://en.wikipedia.org/wiki/Surrogate_key) - [Superkey](http://en.wikipedia.org/wiki/Superkey)  **Database normalization** | [Referential integrity](http://en.wikipedia.org/wiki/Referential_integrity) | [Relational DBMS](http://en.wikipedia.org/wiki/Relational_database_management_system) | [Distributed DBMS](http://en.wikipedia.org/wiki/Distributed_database_management_system) | [ACID](http://en.wikipedia.org/wiki/ACID)   | *Objects*  [Trigger](http://en.wikipedia.org/wiki/Database_trigger) | [View](http://en.wikipedia.org/wiki/View_(database)) | [Table](http://en.wikipedia.org/wiki/Table_(database)) | [Cursor](http://en.wikipedia.org/wiki/Cursor_(databases)) | [Log](http://en.wikipedia.org/wiki/Transaction_log) | [Transaction](http://en.wikipedia.org/wiki/Database_transaction) | [Index](http://en.wikipedia.org/wiki/Index_(database)) | [Stored procedure](http://en.wikipedia.org/wiki/Stored_procedure) | [Partition](http://en.wikipedia.org/wiki/Partition_(database)) | *Topics in [SQL](http://en.wikipedia.org/wiki/SQL)*  [Select](http://en.wikipedia.org/wiki/Select_(SQL)) | [Insert](http://en.wikipedia.org/wiki/Insert_(SQL)) | [Update](http://en.wikipedia.org/wiki/Update_(SQL)) | [Merge](http://en.wikipedia.org/wiki/Merge_(SQL)) | [Delete](http://en.wikipedia.org/wiki/Delete_(SQL)) | [Join](http://en.wikipedia.org/wiki/Join_(SQL)) | [Union](http://en.wikipedia.org/wiki/Union_(SQL)) | [Create](http://en.wikipedia.org/wiki/Create_(SQL)) | [Drop](http://en.wikipedia.org/wiki/Drop_(SQL))  [Comparison of syntax](http://en.wikipedia.org/wiki/Comparison_of_SQL_syntax) | | --- | --- | |
| **Implementations of database management systems** |
| | *Types of implementations*  [Flat file](http://en.wikipedia.org/wiki/Flat_file_database) | [Deductive](http://en.wikipedia.org/wiki/Deductive_database) | [Dimensional](http://en.wikipedia.org/wiki/Dimensional_database) | [Hierarchical](http://en.wikipedia.org/wiki/Hierarchical_model) | [Object oriented](http://en.wikipedia.org/wiki/Object_database) | [Temporal](http://en.wikipedia.org/wiki/Temporal_database)  *Products*  [db4o](http://en.wikipedia.org/wiki/Db4o) | [dBASE](http://en.wikipedia.org/wiki/DBASE) | [Oracle](http://en.wikipedia.org/wiki/Oracle_database) | [Caché](http://en.wikipedia.org/wiki/Cach%C3%83%C2%A9_(software)) | [OpenLink Virtuoso](http://en.wikipedia.org/wiki/Virtuoso_Universal_Server) | [Sybase](http://en.wikipedia.org/wiki/Sybase_SQL_Server) | [Ingres](http://en.wikipedia.org/wiki/Ingres) | [MySQL](http://en.wikipedia.org/wiki/MySQL) | [SQLite](http://en.wikipedia.org/wiki/SQLite) | [Microsoft SQL Server](http://en.wikipedia.org/wiki/Microsoft_SQL_Server) | [PostgreSQL](http://en.wikipedia.org/wiki/PostgreSQL) | [InterBase](http://en.wikipedia.org/wiki/InterBase) | [Firebird](http://en.wikipedia.org/wiki/Firebird_(database_server)) | [DB2](http://en.wikipedia.org/wiki/IBM_DB2) | [Informix](http://en.wikipedia.org/wiki/Informix) | [Helix database](http://en.wikipedia.org/wiki/Helix_database) | [Teradata](http://en.wikipedia.org/wiki/Teradata) | [Comparison - relational](http://en.wikipedia.org/wiki/Comparison_of_relational_database_management_systems) | [Comparison - object-relational](http://en.wikipedia.org/wiki/Comparison_of_object-relational_database_management_systems) | *Components*  [Query language](http://en.wikipedia.org/wiki/Query_language) | [Query optimizer](http://en.wikipedia.org/wiki/Query_optimizer) | [Query plan](http://en.wikipedia.org/wiki/Query_plan) | [ODBC](http://en.wikipedia.org/wiki/Open_Database_Connectivity) | [JDBC](http://en.wikipedia.org/wiki/Java_Database_Connectivity)  *Lists*  [List of object-oriented database management systems](http://en.wikipedia.org/wiki/List_of_object-oriented_database_management_systems)  [List of relational database management systems](http://en.wikipedia.org/wiki/List_of_relational_database_management_systems)  [List of truly relational database management systems](http://en.wikipedia.org/wiki/List_of_truly_relational_database_management_systems) | | --- | --- | |

Retrieved from "<http://en.wikipedia.org/wiki/Database_normalization>"

[Categories](http://en.wikipedia.org/wiki/Special:Categories): [Pages needing expert attention](http://en.wikipedia.org/wiki/Category:Pages_needing_expert_attention) | [Wikipedia articles needing clarification](http://en.wikipedia.org/wiki/Category:Wikipedia_articles_needing_clarification) | [Articles to be split](http://en.wikipedia.org/wiki/Category:Articles_to_be_split) | [Databases](http://en.wikipedia.org/wiki/Category:Databases) | [Data modeling](http://en.wikipedia.org/wiki/Category:Data_modeling) | [Database constraints](http://en.wikipedia.org/wiki/Category:Database_constraints) | [Database normalization](http://en.wikipedia.org/wiki/Category:Database_normalization)

Views

* [Article](http://en.wikipedia.org/wiki/Database_normalization)
* [Discussion](http://en.wikipedia.org/wiki/Talk:Database_normalization)
* [Edit this page](http://en.wikipedia.org/w/index.php?title=Database_normalization&action=edit)
* [History](http://en.wikipedia.org/w/index.php?title=Database_normalization&action=history)

Personal tools

* [Sign in / create account](http://en.wikipedia.org/w/index.php?title=Special:Userlogin&returnto=Database_normalization)

Navigation

* [Main Page](http://en.wikipedia.org/wiki/Main_Page)
* [Community Portal](http://en.wikipedia.org/wiki/Wikipedia:Community_Portal)
* [Featured articles](http://en.wikipedia.org/wiki/Wikipedia:Featured_articles)
* [Current events](http://en.wikipedia.org/wiki/Portal:Current_events)
* [Recent changes](http://en.wikipedia.org/wiki/Special:Recentchanges)
* [Random article](http://en.wikipedia.org/wiki/Special:Random)
* [Help](http://en.wikipedia.org/wiki/Help:Contents)
* [Contact Wikipedia](http://en.wikipedia.org/wiki/Wikipedia:Contact_us)
* [Donations](http://wikimediafoundation.org/wiki/Fundraising)

Search

Toolbox

* [What links here](http://en.wikipedia.org/wiki/Special:Whatlinkshere/Database_normalization)
* [Related changes](http://en.wikipedia.org/wiki/Special:Recentchangeslinked/Database_normalization)
* [Upload file](http://en.wikipedia.org/wiki/Special:Upload)
* [Special pages](http://en.wikipedia.org/wiki/Special:Specialpages)
* [Printable version](http://en.wikipedia.org/w/index.php?title=Database_normalization&printable=yes)
* [Permanent link](http://en.wikipedia.org/w/index.php?title=Database_normalization&oldid=88058464)
* [Cite this article](http://en.wikipedia.org/w/index.php?title=Special:Cite&page=Database_normalization&id=88058464)

In other languages

* [Deutsch](http://de.wikipedia.org/wiki/Normalisierung_(Datenbank))
* [Español](http://es.wikipedia.org/wiki/Normalizaci%C3%83%C2%B3n_de_una_base_de_datos)
* [Français](http://fr.wikipedia.org/wiki/Formes_normales)
* עברית
* [Italiano](http://it.wikipedia.org/wiki/Normalizzazione_del_database)
* [Nederlands](http://nl.wikipedia.org/wiki/Databasenormalisatie)
* [日本語](http://ja.wikipedia.org/wiki/%C3%A3%C6%92%C2%AA%C3%A3%C6%92%C2%AC%C3%A3%C6%92%C2%BC%C3%A3%E2%80%9A%C2%B7%C3%A3%C6%92%C2%A7%C3%A3%C6%92%C2%B3%C3%A3%EF%BF%BD%C2%AE%C3%A6%C2%AD%C2%A3%C3%A8%C2%A6%EF%BF%BD%C3%A5%C5%92%E2%80%93)
* [Polski](http://pl.wikipedia.org/wiki/Normalizacja_bazy_danych)
* [Русский](http://ru.wikipedia.org/wiki/%C3%90%EF%BF%BD%C3%90%C2%BE%C3%91%E2%82%AC%C3%90%C2%BC%C3%90%C2%B0%C3%90%C2%BB%C3%91%C5%92%C3%90%C2%BD%C3%90%C2%B0%C3%91%EF%BF%BD_%C3%91%E2%80%9E%C3%90%C2%BE%C3%91%E2%82%AC%C3%90%C2%BC%C3%90%C2%B0)
* [Slovenčina](http://sk.wikipedia.org/wiki/Normaliz%C3%83%C2%A1cia_(datab%C3%83%C2%A1zy))
* [Suomi](http://fi.wikipedia.org/wiki/Tietokannan_normalisointi)
* [Svenska](http://sv.wikipedia.org/wiki/Normalform_(databaser))
* [Dansk](http://da.wikipedia.org/wiki/Databasenormalisering)
* This page was last modified 20:59, 15 November 2006.
* All text is available under the terms of the [GNU Free Documentation License](http://en.wikipedia.org/wiki/Wikipedia:Text_of_the_GNU_Free_Documentation_License). (See [**Copyrights**](http://en.wikipedia.org/wiki/Wikipedia:Copyrights) for details.)  
  Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc.
* [Privacy policy](http://wikimediafoundation.org/wiki/Privacy_policy)
* [About Wikipedia](http://en.wikipedia.org/wiki/Wikipedia:About)
* [Disclaimers](http://en.wikipedia.org/wiki/Wikipedia:General_disclaimer)