**Database design**

From Wikipedia, the free encyclopedia

[Jump to navigation](https://en.wikipedia.org/wiki/Database_design#mw-head) [Jump to search](https://en.wikipedia.org/wiki/Database_design#p-search)

**Database design** is the organisation of data according to a [database model](https://en.wikipedia.org/wiki/Database_model). The designer determines what data must be stored and how the data elements interrelate. With this information, they can begin to fit the data to the database model.[[1]](https://en.wikipedia.org/wiki/Database_design#cite_note-Teorey,_T.J._2009-1)

Database design involves classifying data and identifying interrelationships. This theoretical representation of the data is called an [*ontology*](https://en.wikipedia.org/wiki/Ontology_(information_science)). The ontology is the theory behind the database's design.



**Contents**

* [1 Determining data to be stored](https://en.wikipedia.org/wiki/Database_design#Determining_data_to_be_stored)
* [2 Determining data relationships](https://en.wikipedia.org/wiki/Database_design#Determining_data_relationships)
* [3 Logically structuring data](https://en.wikipedia.org/wiki/Database_design#Logically_structuring_data)
* [4 ER diagram (entity-relationship model)](https://en.wikipedia.org/wiki/Database_design#ER_diagram_(entity-relationship_model))
* [5 A design process suggestion for Microsoft Access[3]](https://en.wikipedia.org/wiki/Database_design#A_design_process_suggestion_for_Microsoft_Access[3])
* [6 Normalization](https://en.wikipedia.org/wiki/Database_design#Normalization)
  + [6.1 Conceptual schema](https://en.wikipedia.org/wiki/Database_design#Conceptual_schema)
  + [6.2 Schema refinement](https://en.wikipedia.org/wiki/Database_design#Schema_refinement)
  + [6.3 Physical design](https://en.wikipedia.org/wiki/Database_design#Physical_design)
* [7 See also](https://en.wikipedia.org/wiki/Database_design#See_also)
* [8 References](https://en.wikipedia.org/wiki/Database_design#References)
* [9 Further reading](https://en.wikipedia.org/wiki/Database_design#Further_reading)
* [10 External links](https://en.wikipedia.org/wiki/Database_design#External_links)

**Determining data to be stored**

In a majority of cases, a person who is doing the design of a database is a person with expertise in the area of database design, rather than expertise in the domain from which the data to be stored is drawn e.g. financial information, biological information etc. Therefore, the data to be stored in the database must be determined in cooperation with a person who does have expertise in that domain, and who is aware of what data must be stored within the system.

This process is one which is generally considered part of [requirements analysis](https://en.wikipedia.org/wiki/Requirements_analysis), and requires skill on the part of the database designer to elicit the needed information from those with the [domain knowledge](https://en.wikipedia.org/wiki/Domain_knowledge). This is because those with the necessary domain knowledge frequently cannot express clearly what their system requirements for the database are as they are unaccustomed to thinking in terms of the discrete data elements which must be stored. Data to be stored can be determined by Requirement Specification.[[2]](https://en.wikipedia.org/wiki/Database_design#cite_note-2)

**Determining data relationships**

Once a database designer is aware of the data which is to be stored within the database, they must then determine where dependency is within the data. Sometimes when data is changed you can be changing other data that is not visible. For example, in a list of names and addresses, assuming a situation where multiple people can have the same address, but one person cannot have more than one address, the address is dependent upon the name. When provided a name and the list the address can be uniquely determined; however, the inverse does not hold - when given an address and the list, a name cannot be uniquely determined because multiple people can reside at an address. Because an address is determined by a name, an address is considered dependent on a name.

(NOTE: A common misconception is that the [relational model](https://en.wikipedia.org/wiki/Relational_model) is so called because of the stating of relationships between data elements therein. This is not true. The relational model is so named because it is based upon the mathematical structures known as [relations](https://en.wikipedia.org/wiki/Relation_(mathematics)).)

**Logically structuring data**

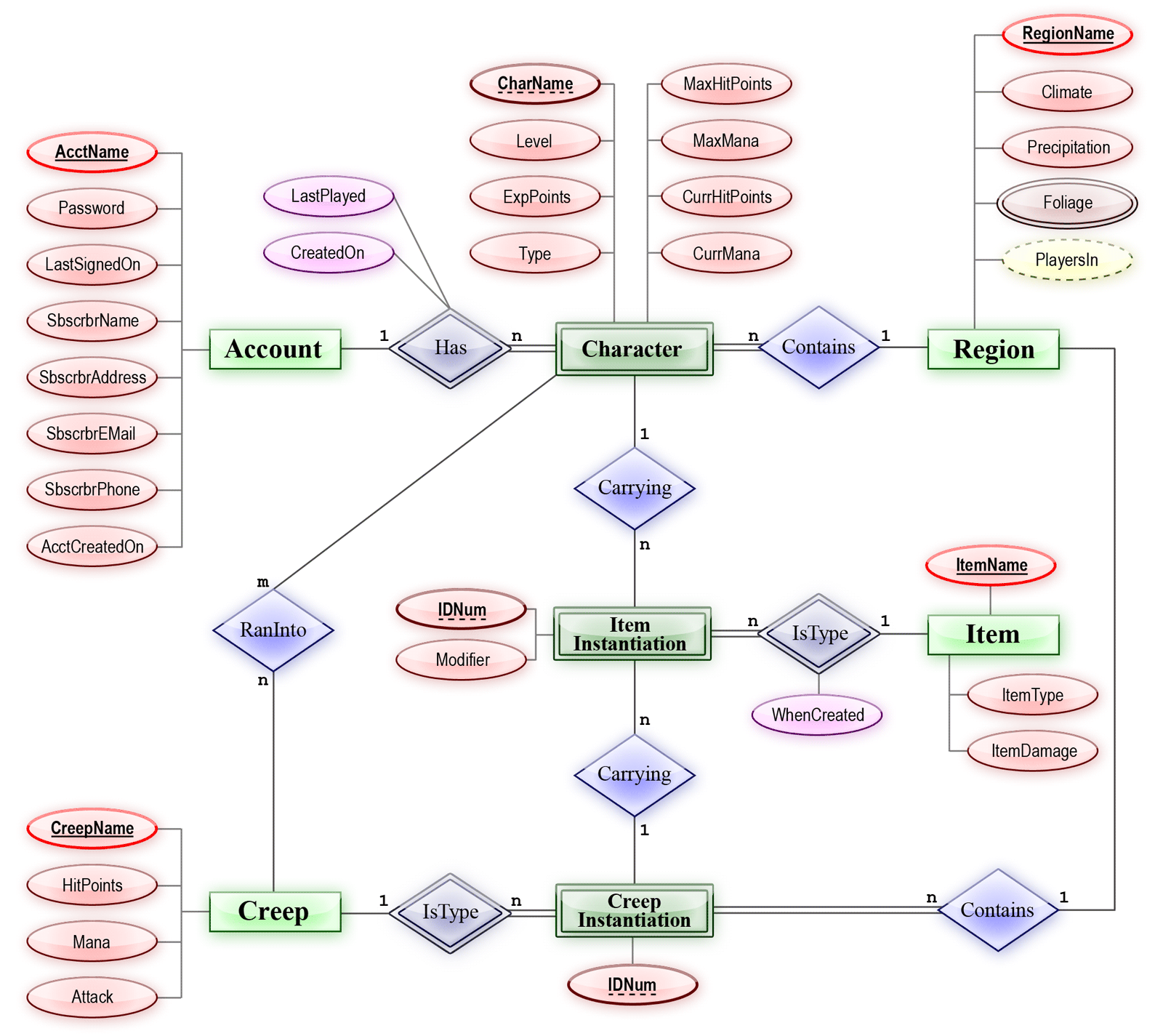
Main article: [Logical schema](https://en.wikipedia.org/wiki/Logical_schema)

Once the relationships and dependencies amongst the various pieces of information have been determined, it is possible to arrange the data into a logical structure which can then be mapped into the storage objects supported by the [database management system](https://en.wikipedia.org/wiki/Database_management_system). In the case of [relational databases](https://en.wikipedia.org/wiki/Relational_databases) the storage objects are [tables](https://en.wikipedia.org/wiki/Database_table) which store data in rows and columns. In an [Object database](https://en.wikipedia.org/wiki/Object_database) the storage objects correspond directly to the objects used by the [Object-oriented programming language](https://en.wikipedia.org/wiki/Object-oriented_programming_language) used to write the applications that will manage and access the data. The relationships may be defined as attributes of the object classes involved or as methods that operate on the object classes.

The way this mapping is generally performed is such that each set of related data which depends upon a single object, whether real or abstract, is placed in a table. Relationships between these dependent objects is then stored as links between the various objects.

Each table may represent an implementation of either a logical object or a relationship joining one or more instances of one or more logical objects. Relationships between tables may then be stored as links connecting child tables with parents. Since complex logical relationships are themselves tables they will probably have links to more than one parent.

**ER diagram (entity-relationship model)**



A sample Entity-relationship diagram

Database designs also include ER ([entity-relationship model](https://en.wikipedia.org/wiki/Entity-relationship_model)) diagrams. An ER diagram is a diagram that helps to design databases in an efficient way.

Attributes in ER diagrams are usually modeled as an oval with the name of the attribute, linked to the entity or relationship that contains the attribute.

**A design process suggestion for Microsoft Access[[3]](https://en.wikipedia.org/wiki/Database_design" \l "cite_note-3)**

1. **Determine the purpose of the database** - This helps prepare for the remaining steps.
2. **Find and organize the information required** - Gather all of the types of information to record in the database, such as product name and order number.
3. **Divide the information into tables** - Divide information items into major entities or subjects, such as Products or Orders. Each subject then becomes a table.
4. **Turn information items into columns** - Decide what information needs to be stored in each table. Each item becomes a field, and is displayed as a column in the table. For example, an Employees table might include fields such as Last Name and Hire Date.
5. **Specify primary keys** - Choose each table's primary key. The primary key is a column, or a set of columns, that is used to uniquely identify each row. An example might be Product ID or Order ID.
6. **Set up the table relationships** - Look at each table and decide how the data in one table is related to the data in other tables. Add fields to tables or create new tables to clarify the relationships, as necessary.
7. **Refine the design** - Analyze the design for errors. Create tables and add a few records of sample data. Check if results come from the tables as expected. Make adjustments to the design, as needed.
8. **Apply the** [**normalization rules**](https://en.wikipedia.org/wiki/Database_normalization) - Apply the data normalization rules to see if tables are structured correctly. Make adjustments to the tables, as needed.

**Normalization**

Main article: [Database normalization](https://en.wikipedia.org/wiki/Database_normalization)

In the field of [relational database](https://en.wikipedia.org/wiki/Relational_database) design, *normalization* is a systematic way of ensuring that a database structure is suitable for general-purpose querying and free of certain undesirable characteristics—insertion, update, and deletion anomalies that could lead to loss of [data integrity](https://en.wikipedia.org/wiki/Data_integrity).

A standard piece of database design guidance is that the designer should create a fully normalized design; selective [denormalization](https://en.wikipedia.org/wiki/Denormalization) can subsequently be performed, but only for [performance](https://en.wikipedia.org/wiki/Computer_performance) reasons. However, some modeling disciplines, such as the [dimensional modeling](https://en.wikipedia.org/wiki/Dimensional_modeling) approach to [data warehouse](https://en.wikipedia.org/wiki/Data_warehouse) design, explicitly recommend non-normalized designs, i.e. designs that in large part do not adhere to 3NF. Normalization consists of normal forms that are 1NF,2NF,3NF,BOYCE-CODD NF (3.5NF),4NF and 5NF

**Conceptual schema**

Main article: [Conceptual schema](https://en.wikipedia.org/wiki/Conceptual_schema)

**Schema refinement**

Schema refinement of the database specifies that the data is normalized to reduce data insufficiency and conflicts.

**Physical design**

Main article: [Physical schema](https://en.wikipedia.org/wiki/Physical_schema)

The physical design of the database specifies the physical configuration of the database on the storage media. This includes detailed specification of [data elements](https://en.wikipedia.org/wiki/Data_element), [data types](https://en.wikipedia.org/wiki/Data_type), [indexing](https://en.wikipedia.org/wiki/Index_(database)) options and other parameters residing in the DBMS [data dictionary](https://en.wikipedia.org/wiki/Data_dictionary). It is the detailed design of a system that includes modules & the database's hardware & software specifications of the system.