A database is a tool that stores data and lets you create, read, update, and delete the data in some manner.

* Model reality – changes to the miniworld are reflected in the database.
* Logically coherent – random assortment of data not really a database
* Comprehensive – needs to include all the relevant information (think of a customer order system with only some of the orders being included in the database).
* Structured and organized to facilitate efficient retrieval of relevant information (imagine a dictionary that was not alphabetically organized)
* Built for a specific purpose and for an intended group of users and some set of applications those users are interested in (does a university class scheduling database care about the color and type of chairs in the classroom?)
* CRUD
* **Create** Store new pieces of information.
* **Read** Retrieving data previously stored.
* **Update** Alter the attributes of records in the database.
* **Delete** Remove records that are no longer needed.
* Consistency
* Consistency is the ability for a database to return the exact same results each time the data is requested (assuming the underlying data hasn’t changed).
* Consistency is not only related to the reliability inherent in the database, but also to how a specific database is designed.
* Validity
* Consistency means that different parts of the database don’t hold contradictory views of the same information, but says nothing about the correctness of that data.
* Validity is the extent to which data’s accuracy is checked before being stored in a database.
* Error prevention
* Error prevention is the extent to which a database prevents inadvertent or logically inconsistent actions .
* Easy Error Correction

Ideally, errors are prevented. However, if errors do occur, a database should have features that allow us to quickly and easily correct those errors.

* Speed
* Speed refers to how quickly a database can create, read, update, and delete records while under normal operating loads.
* Just about ANY database, even poorly designed, can be fast enough for a handful of users. However, poor design choices become more and more apparent as the amount of data grows and the number of users increase.
* Atomic Transactions
* Atomic transactions are a series of actions that are conceptually treated as one action.
* Either ALL the actions get successfully performed or NONE of them do.
* Persistence and Backup Capability
* Persistence is the capability for data to continue to exist after it is created. A good database must ensure the availability of data that has been saved.
* Most databases are intrinsically good at persisting data – that is their primary function.
* Portability
* Portability is the degree to which users who are no co-located with the database can easily access the data in a database.
* Most databases provide access through a local network or via the web.
* Security
* Security is the degree to which a database only provides authorized access to data it contains.
* Security and portability are often at odds – the more portable a database, the potentially less secure.
* Security also refers to the granularity of access provided. A good database will allow different levels of access to different sets of data. Access that is too restrictive can be *less* secure.
* Sharing
* Sharing is the degree to which a database accommodates multiple users accessing the same data.
* Although most aspects of sharing are more a property of the network than the database, good database design can facilitate sharing by allowing for the smallest amounts of data to be locked during updates.

Flat Files – just a plain text file, one record per line with values separated by commas: Adv.: Easy to understand, Can edit in simple text editor, good for storing text-based sequential data and data that does not change often. Cons: Not easy to update (have to rewrite entire file) and ad hoc searches not easy

Spreadsheets