What are the ACID properties of SQL transactions? If possible, explain each property with an illustration of an example.

In the context of transaction processing, the acronym *ACID* refers to the four key properties of a transaction: atomicity, consistency, isolation, and durability.

**Atomicity**

All changes to data are performed as if they are a single operation. That is, all the changes are performed, or none of them are.

For example, in an application that transfers funds from one account to another, the atomicity property ensures that, if a debit is made successfully from one account, the corresponding credit is made to the other account.

**Consistency**

Data is in a consistent state when a transaction starts and when it ends.

For example, in an application that transfers funds from one account to another, the consistency property ensures that the total value of funds in both the accounts is the same at the start and end of each transaction.

**Isolation**

The intermediate state of a transaction is invisible to other transactions. As a result, transactions that run concurrently appear to be serialized.

For example, in an application that transfers funds from one account to another, the isolation property ensures that another transaction sees the transferred funds in one account or the other, but not in both, nor in neither.

**Durability**

After a transaction successfully completes, changes to data persist and are not undone, even in the event of a system failure.

For example, in an application that transfers funds from one account to another, the durability property ensures that the changes made to each account will not be reversed.

Retrieved from <https://www.ibm.com/support/knowledgecenter/SSGMCP_5.4.0/product-overview/acid.html>

**Part 1**

**Explain an index in SQL**

Indexes are used to retrieve data from the database more quickly than otherwise. The users cannot see the indexes, they are just used to speed up searches/queries.

**Part 2**

**What are the different types of index? If possible, explain each type with an illustration.**

There are three types of index:

 Unique: if a column is uniquely indexed, no duplicate values will be allowed in the field. If a primary key is already defined, then a unique index is automatically applied.

 Clustered: this type of index will reorder the physical order of the table and searches based off the key values. Only one clustered index is allowed per table.

 Non-Clustered: a non-clustered index will not alter the physical form of the table. More than one non-clustered indexes are allowed per table.