Four sub-sets

* + DML – Data Manipulation Language:
    - SELECT, INSERT, UPDATE, and DELETE
  + DDL – Data Definition Language
    - CREATE, ALTER, DROP and TRUNCATE
  + DCL – Data Control Language
    - GRANT, REVOKE, and DENY
  + TCL – Transaction Control Language
    - COMMIT, ROLLBACK, SAVEPOINT

Primary Key column

* + Uniquely identifies each row within a table and must be filled in
  + Could be a single column or combination of columns
  + Primary key is often an artificial “surrogate” key

Foreign key column

* + A foreign key is a column on the **child table** that references a unique identifier on the **parent table**
* Tables – hold the data for instances of entities
* Rows – each row holds the information for one thing
* Columns – each column holds one attribute

**First Normal Form**

**Every column must contain ONE value from the applicable domain.**

* + No repeated columns or groups of columns
  + Each column should provide a different piece of information
* 2NF and 3NF involve resolving functional dependencies
* From mathematics:
* Y = F(X)
* Y is a function of X. For any given value for X, there is one value for Y. As we change X, Y changes as well.
* Every non-key attribute in a table should be functionally dependent on the primary key. That is, the primary key represents an entity and the attributes provide information about that entity.

A Table is in **Second Normal Form** if:

* + It is in **First Normal Form**
  + Every non key attribute is functionally dependent on the entire primary key (and not just part of a composite primary key).

If the primary key consists of only one column then the table automatically meets 2NF.

A table is in **Third Normal Form** if:

* + It is in **Second** **Normal Form**
  + There are **no transitive dependencies.** i.e. every non key attribute is functionally dependent on the primary key and not functionally dependent on any other column .

Online Transaction Processing (OLTP) deals with the day to day work and transactions.

* + OLTP databases tend to be highly normalized

Online Analytical Processing (OLAP) deals with data analysis.

* + OLAP databases tend to be somewhat denormalized

**Indexes**

* An index makes the query fast”
* An index is a distinct structure in the database that is built using the **create index** statement
* Creating an index does not change table data
* Similar to an index at the end of a book
* Constantly changing
* Easier to keep index sorted rather than the table data

**“What is a clustered index and non-clustered index?**

* + The data in the data blocks is stored in the order of the clustered index.
  + The data blocks are considered to be the leaves of the clustered index tree.
  + Because a table can store data in only one order, a table can have only one clustered index. (So choose wisely.)

Non clustered data is separate thing pointing to the data.  Like the back of book

Clustered index is like a phone book that reorganizes the actual data in a way that’s easy to use

Non clustered is pointing to the data

Clustered organizes the actual daya

* Choosing the clustered index: If you usually want your data to come out sorted in one particular way, then that is a good candidate for the clustered index.
* SQL Server, MySQL, SYBASE, Informix, DB2 all have this important clustered vs. non-clustered distinction. A clustered index helps keep the table better organized.
* Oracle CAN create an “INDEX ORGANIZED TABLE” in which the data on the data blocks is stored in the order of the index. But for Oracle having every table as an index-organized table is simply not recommended.
  + Non-clustered indexes (or secondary indexes) can help a query find the block where the data resides, but the data on the block is not sorted by the index.