**Database:** Database is a collection of inter-related data which helps in efficient retrieval, insertion and deletion of data from database and organizes the data in the form of tables, views, schemas, reports etc.

**Database Management System:** The software which is used to manage database is called Database Management System (DBMS).

**Data Definition:** It helps in creation, modification and removal of definitions that define the organization of data in database.

**DDL COMMANDS:**

* [**CREATE**](https://www.geeksforgeeks.org/sql-create/) **– is used to create the database or its objects (like table, index, function, views, store procedure and triggers).**
* [**DROP**](https://www.geeksforgeeks.org/sql-drop-truncate/) **– is used to delete objects from the database.**
* [**ALTER**](https://www.geeksforgeeks.org/sql-alter-add-drop-modify/)**-is used to alter the structure of the database.**
* [**TRUNCATE**](https://www.geeksforgeeks.org/sql-drop-truncate/)**–is used to remove all records from a table, including all spaces allocated for the records are removed.**
* [**COMMENT**](https://www.geeksforgeeks.org/sql-comments/) **–is used to add comments to the data dictionary.**
* [**RENAME**](https://www.geeksforgeeks.org/sql-alter-rename/) **–is used to rename an object existing in the database.**

**Drop vs Delete vs Truncate:**

**TRUNCATE(no rollback)**

* TRUNCATE is a DDL command
* TRUNCATE is executed using a table lock and whole table is locked for remove all records.
* We cannot use Where clause with TRUNCATE.
* TRUNCATE removes all rows from a table.
* Minimal logging in transaction log, so it is performance wise faster.
* Truncate uses the less transaction space than Delete statement.
* Truncate cannot be used with indexed views/ foreign key referencing it.

**DELETE**

* DELETE is a DML command.
* DELETE is executed using a row lock, each row in the table is locked for deletion.
* We can use where clause with DELETE to filter & delete specific records.
* The DELETE command is used to remove rows from a table based on WHERE condition.
* It maintain the log, so it slower than TRUNCATE.
* The DELETE statement removes rows one at a time and records an entry in the transaction log for each deleted row.
* Delete uses the more transaction space than Truncate statement.
* Delete can be used with indexed views.

**DROP**

* The DROP command removes a table from the database.
* All the tables' rows, indexes and privileges will also be removed.
* No DML triggers will be fired.
* The operation cannot be rolled back.
* DROP and TRUNCATE are DDL commands, whereas DELETE is a DML command.
* DELETE operations can be rolled back (undone), while DROP and TRUNCATE operations cannot be rolled back.

**DML COMMANDS:**

[**SELECT**](https://www.geeksforgeeks.org/sql-select-clause/) **– is used to retrieve data from the a database.**

* [**INSERT**](https://www.geeksforgeeks.org/sql-insert-statement/) **– is used to insert data into a table.**
* [**UPDATE**](https://www.geeksforgeeks.org/sql-update-statement/) **– is used to update existing data within a table.**
* [**DELETE**](https://www.geeksforgeeks.org/sql-delete-statement/) **– is used to delete records from a database table.**

**DCL(Data Control Language) :**

* GRANT-gives user’s access privileges to database.
* REVOKE-withdraw user’s access privileges given by using the GRANT command.

**GRANT privileges\_names ON object TO user;**

GRANT SELECT, INSERT, DELETE, UPDATE ON Users TO 'Amit'@'localhost;

GRANT EXECUTE ON FUNCTION Calculatesalary TO 'Amit'@localhost';

**TCL(transaction Control Language) :** deals with the [transaction within the database](https://www.geeksforgeeks.org/sql-transactions/).

1. **COMMIT**– commits a Transaction.
2. [**ROLLBACK**](https://www.geeksforgeeks.org/sql-transactions/)– rollbacks a transaction in case of any error occurs.
3. **SAVEPOINT**–sets a savepoint within a transaction.
4. **SET TRANSACTION**–specify characteristics for the transaction.

**Phases of database design**

**1)** Conceptual Design: ER model is used for conceptual design of database.

**2)** Logical Design:. ER diagram produced is used to convert the data into Relational Model.

3) Physical Design:data in relational model is implemented using DBMS like Oracle, DB2.

**RELATIONAL MODEL:**

Key plays an important role in relational database; it is used for identifying unique rows from table. Every table has at least one key

**1) Super Key:** The set of attributes which can uniquely identify a tuple.

**2) Candidate Key:** The minimal set of attribute which can uniquely identify a tuple

**3) Primary Key:** There can be more than one candidate key in a relation out of which one can be a P\_key

**4)** **Alternate Key:** The candidate key other than primary key is called as alternate key

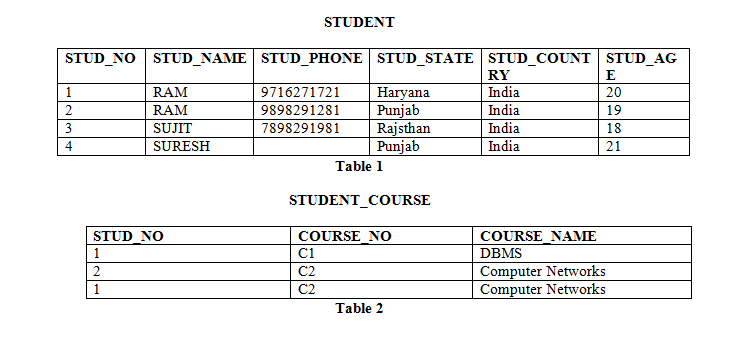
**5) Composite Key**: Any column(s) that can guarantee uniqueness is called a candidate key; however a composite key is a special type of candidate key that is only formed by a combination of two or more columns. This is a composite key, that is, a candidate key that guarantees uniqueness only when two or more columns are joined together.

**6) Foreign Key:** If an attribute can only take the values which are present as values of some other attribute, it will be foreign key to the attribute to which it refers. The relation which is being referenced is called referenced relation and corresponding attribute is called referenced attribute and corresponding attribute is called referencing attribute. Referenced attribute of referencing attribute should be unique key

**Summary:**

**Super key -> candidate key(minimal super key) -> primary key(one of the candidate key) -> alternate key(remaining candidate key) # composite key – special type of candidate key**

**Anomalies in Relational Model**



**Insertion anomaly:** If a tuple is inserted in referencing relation and referencing attribute value is not present in referenced attribute, it will not allow inserting in referencing relation. For Example, If we try to insert a record in STUDENT\_COURSE with STUD\_NO =7, it will not allow.

**Deletion and Updation anomaly:** If a tuple is deleted or updated from referenced relation and referenced attribute value is used by referencing attribute in referencing relation, it will not allow deleting the tuple from referenced relation. For Example, If we try to delete a record from STUDENT with STUD\_NO =1, it will not allow. To avoid this, following can be used in query:

* **ON DELETE/UPDATE SET NULL:** If a tuple is deleted or updated from referenced relation and referenced attribute value is used by referencing attribute in referencing relation, it will delete/update the tuple from referenced relation and set the value of referenced attribute to NULL.
* **ON DELETE/UPDATE CASCADE:** If a tuple is deleted or updated from referenced relation and referenced attribute value is used by referencing attribute in referencing relation, it will delete/update the tuple from referenced relation and referencing relation as well.