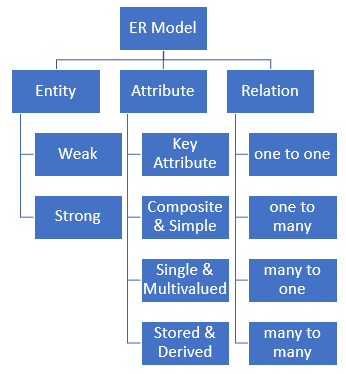
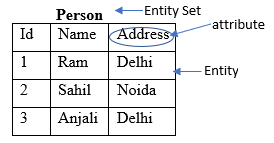
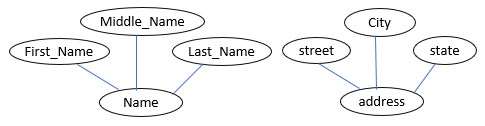
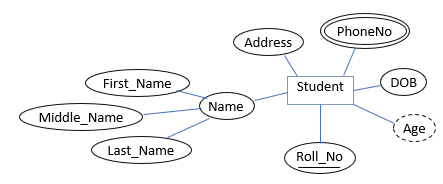
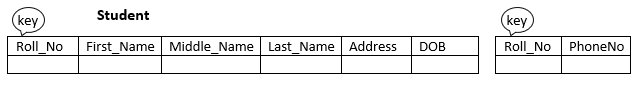
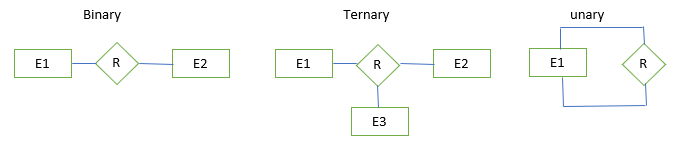
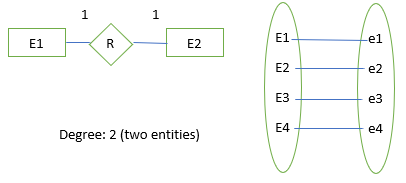
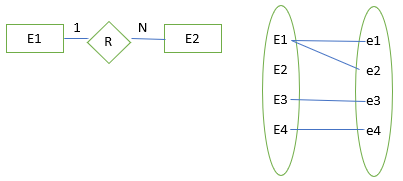
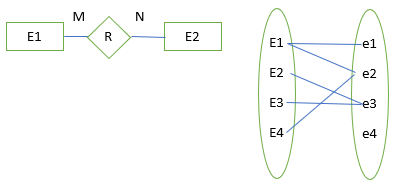
Introduction To Dbms

Data: any raw facts or figures that can be recorded or stored.  
   
 Information: when data are processed, organized to make meaningful or usable.   
   
 Database: collection of related data.  
   
 Types of database:  
   
 1. **Traditional Database:** it contains text and number.   
 Explore more about traditional Database:  
 a. [dbms-vs-traditional-file-system](https://www.includehelp.com/dbms/dbms-vs-traditional-file-system.aspx)  
 b. [traditional\_db](http://www.jhigh.co.uk/Intermediate2/Database/5traditional_db.html)  
   
 2. **Multimedia Database:** it is a collection of related multimedia data. Multimedia data include images, songs, videos etc.  
 Explore more about Multimedia Database:  
 [dbms-multimedia-database](https://www.geeksforgeeks.org/dbms-multimedia-database/)  
   
 3. **Geographic Information System (GIS):** It designed to capture, store or manipulate the geographical data.   
 Explore more about GIS:  
 [Geographic\_information\_system](https://en.wikipedia.org/wiki/Geographic_information_system)  
   
 4. **Data Warehouse:** it stores large amount of historical data.   
 Explore more about Data Warehouse Database:  
 [difference-between-a-database-and-a-data-warehouse](https://www.guru99.com/database-vs-data-warehouse.html/)  
   
 5. Others: [Types-of-databases](https://www.tutorialspoint.com/Types-of-databases)  
   
   
 Database Management System (DBMS):   
 It is a program for create and manipulate database. It provides systematic way to create, retrieve, update and delete data.   
   
 **Database system = database + DBMS**  
   
 Database basics: [Introduction to DBMS](https://www.youtube.com/watch?v=cKRnbP6dZwc&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=2)  
   
 Advantages and Disadvantages of DBMS:   
 a. [dbms-vs-file-system](https://beginnersbook.com/2015/04/dbms-vs-file-system/)  
 b. Video: [Advantages of DBMS over File System](https://www.youtube.com/watch?v=CqI0X8fNn2I&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=3)  
   
 OLAP vs OLTP:   
   
 1. [oltp-vs-olap](https://www.guru99.com/oltp-vs-olap.html)  
 2. [dbms-olap-vs-oltp](https://www.geeksforgeeks.org/dbms-olap-vs-oltp/)  
 3. **Video on OLAP vs OLTP:** [OLAP vs OLTP](https://www.youtube.com/watch?v=kFQRrgHeiOo&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=4)

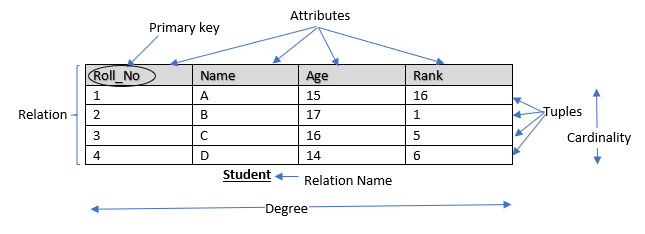
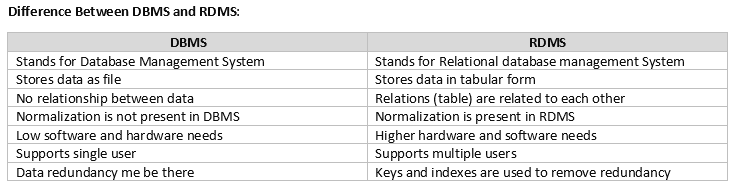
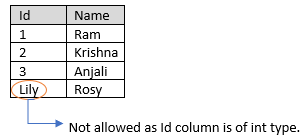
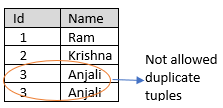
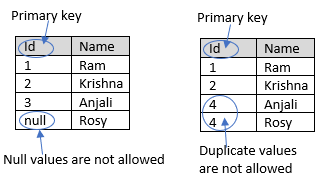
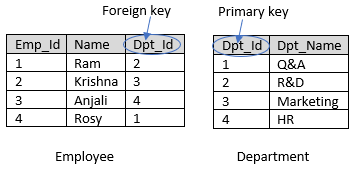
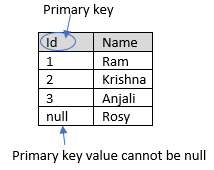
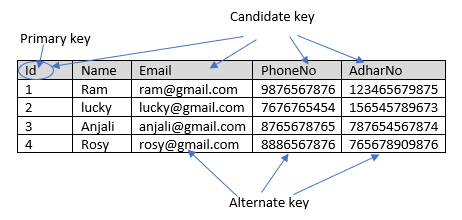
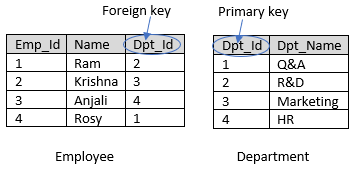
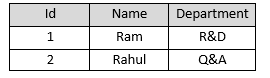
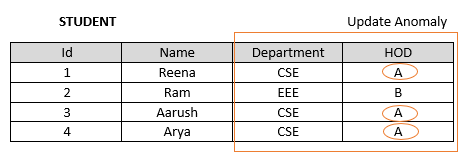
|  |  |
| --- | --- |
| **OLAP** | **OLTP** |
| Stands for Online Analytical Processing. | Stands for Online Transaction Processing. |
| It uses the Data Warehouse. | It uses Traditional DBMS. |
| Mostly select operations | Insert, Update, Delete data from the database. |
| Response time is seconds to minutes. | Response time is in milliseconds. |
| Mainly Read operations are performed | Allows Read/Write operations. |
| It is subject oriented | It is application oriented. |

Database Architecture:   
 1. [dbms-architecture](https://www.javatpoint.com/dbms-architecture)  
 2. [concepts-of-database-architecture](https://medium.com/oceanize-geeks/concepts-of-database-architecture-dfdc558a93e4)  
   
 Database Model/ Degree of Data Abstraction:   
 It determines the logical structure of a database and in which manner data can be stored, organized and manipulated.  
   
 Database model in detail: [data-modelling-conceptual-logical](https://www.guru99.com/data-modelling-conceptual-logical.html)  
   
 **1. High Level Model/Conceptual Model:**   
 This defines What the system contains. In this we use some diagrams (e.g. ER diagram). It defines the scope of information and the main purpose is for communication and agreement of context.   
   
 **2. Logical/representational/Implementation Model:**   
 This defines the How the system should be implemented. Relations (Tables) are used at this level. This model mainly used by Programmers. Main purpose is for communication and agreement of business logic.  
   
 **3. Low Level/ Physical level Model:**   
 This defines the How we access the data. It is the physical representation of the database. Physical model has the lowest level of the abstraction.

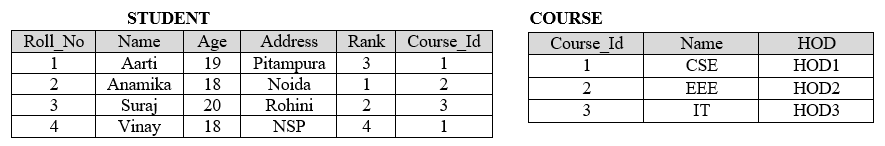
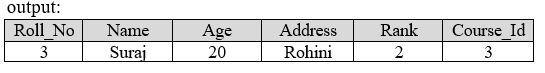
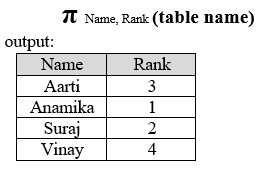
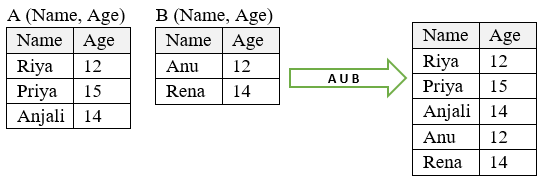
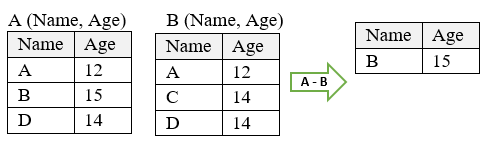
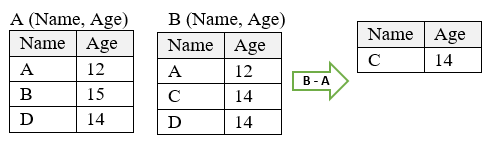
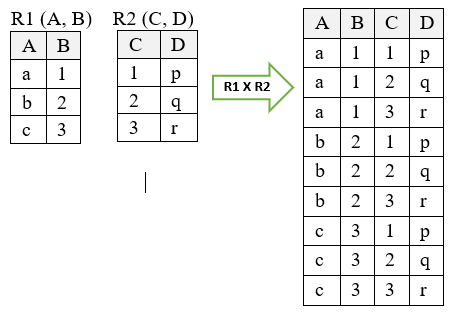
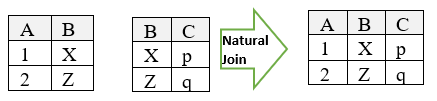
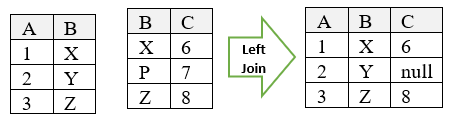
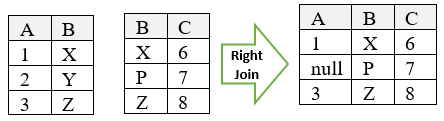
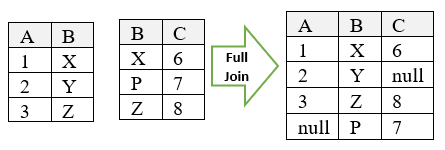
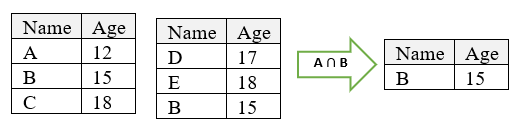
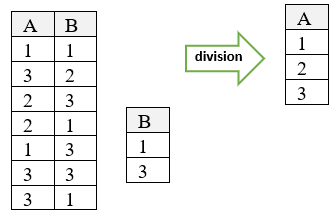
# Entity Relationship Model

ER Models:  
 It is a high-level conceptual data model. It is a design or blueprint of the database. It shows the relationship among entity sets that exist in real world. The database structure is showed as a diagram known as Entity Relationship diagram. It provides logical way to visualize data. It is easy to understand by non-technical user.  
   
 Components of ER Model:   
   
   
   
 **Entity:** An entity is a real-world object that can be recorded in database. An entity may be person, student, place, class etc.   
   
 ***Types of Entities:***   
 1. *Tangible:* Entities which exists physically. E.g. Pencil, book etc.  
 2. *Non-Tangible/ Intangible:* Entities which exists logically. E.g. Account etc.  
   
 **Entity Set:** set of similar type of entities that can share some attributes. It is represented by rectangle in ER diagram and as a table(relation) in relational model.  
   
   
   
 *Weak Entity:* any entity which is not having key (primary key) attribute.  
 weak_entity  
   
 *Strong Entity:* any entity which is having key attribute.   
 strong_entity  
   
 **Attributes:** attributes are the properties describe the entities. It is represented by oval in ER diagram and as a column in relation(table).  
   
 *Domain:* set of values for an attribute.  
   
 **Types of Attribute:**   
 1. **Simple and Composite attribute:**   
 *Simple attribute* cannot be divider furthered. It is represented by oval.  
 e.g. Roll\_No, Id, DOB, first name, city etc.  
   
 simple_attr  
   
 *Composite attribute* can be divided furthered. In this one oval can be connected by many ovals.  
 e.g. Name, Address etc.  
   
   
   
 2. **Single and Multivalued attribute:**  
 *Single attribute* can have only one value.   
 e.g. Age, Name, Roll No etc.  
   
 simple_attr  
   
 *Multivalued attribute* can have more than one value. In relational model separate table is used for multivalued attribute.  
 e.g. Phone No etc.  
   
 multivalued_attr  
   
 3. **Stored and Derived:**  
 *Stored: store* some information in database.  
 e.g. DOB  
   
 stored_attr  
   
 *Derived attribute* derive value from the stored value. These values can be computed at run time using stored value.  
 e.g. Age.  
   
 derived_attr  
   
 4. **Key Attribute:** It represents primary key (holds unique value for an attribute). It is represented by an oval with text underlined.  
   
 key_attr  
   
 5. **Composite Attribute:** Any attribute combination of composite and multivalued.  
 E.g. Address, Phone number.   
   
 ER diagram example:   
 *Entity Set:* Student   
 *Properties of student:* Roll No, Name, Age, DOB, Address, Phone No etc.   
   
   
   
 *Relation(table) derived from ER diagram:*   
   
   
   
 Important Links:   
 1. [database-management-system-er-model](https://www.geeksforgeeks.org/database-management-system-er-model/)  
 2. [dbms-er-model-concept](https://www.javatpoint.com/dbms-er-model-concept)  
 3. [dbms-enhanced-er-model](https://www.geeksforgeeks.org/dbms-enhanced-er-model/)  
 4. [minimization-er-diagram](https://www.geeksforgeeks.org/minimization-er-diagram/)  
 5. [er-model-generalization-specialization-aggregation](https://www.geeksforgeeks.org/dbms-er-model-generalization-specialization-aggregation/)  
 6. [recursive-relationships](https://www.geeksforgeeks.org/dbms-recursive-relationships/)  
   
 Videos:   
 1. Entity Relationship Diagram: [Entity Relationship Diagram](https://www.youtube.com/watch?v=iUApyIf2h-A&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=5)  
 2. Entity: [Entity set definition and types of entity relationship model diagram](https://www.youtube.com/watch?v=jbl5rWNP5hU&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=6)  
 3. Attributes: [types of attributes in ER diagram](https://www.youtube.com/watch?v=7mi_3zi66C8&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=7)  
   
 Relationship in ER diagram:   
 First read these links in the give sequence:   
 1. Relationship: [Relationship in ER diagram](https://www.youtube.com/watch?v=YvQtgE6R_q0&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=8)  
 2. Degree of relationship: [Degree of relationship](https://www.youtube.com/watch?v=JGOvqCFrldU&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=9)  
 3. Cardinality ratio/Mapping:  [Cardinality ratio/Mapping](https://www.youtube.com/watch?v=XUokY6sA3UE&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=10)  
 4. Participation Constraints: [Participation Constraints:](https://www.youtube.com/watch?v=sWZw69Wf5SQ&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=11)  
   
 **Relationship:** association between two or more entities.  
 **Relationship type:** a set of similar type of relationship. It is represented by diamond in ER diagram and as a separate table or column in relational model.  
   
 **Characteristics of relationship:**   
 1. **Degree:** how many entity sets are participating.  
 E.g. binary, ternary, unary, quaternary, N-ary etc.  
   
   
   
 2. **Cardinality Ratio:** maximum number of relationships in which an entity can participate.  
   
 3.  **Participation:**  minimum number of relationships in which an entity can participate.  
 Total Participation of an entity = 1  
 Partial Participation of an entity = 0   
   
 Participation of weak entity in identifying relationship must be total i.e. participation of weak entity is always true (1).  
   
 Note: If the entity is weak then its participation is total (1) but total participation of any entity doesn't make that entity weak.  
   
 Types of relationship:   
   
 **Mapping cardinalities:** shows the number of entities to which other entity can be related via relationship.   
   
 1. **One to One relationship:**   
 When entity of the first entity set can be related to the at most one entity of the other entity set and vice versa.  
   
   
   
 2. **One to Many relationships:**   
 When entity of the first entity set can be related to the many entities of the other entity set.  
   
   
   
 3. **Many to Many relationships:**   
 When many entities of the first entity set can be related to the many entities of the other entity set and vice versa.  
   
   
   
   
 Mapping from ER Model to Relational Model [mapping-from-er-model-to-relational-model](https://www.geeksforgeeks.org/mapping-from-er-model-to-relational-model/)

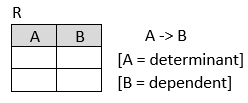
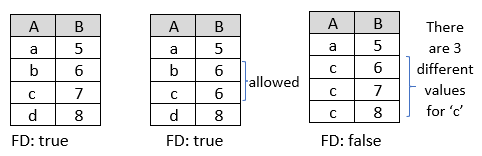
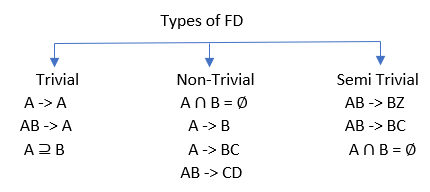
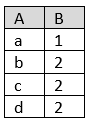
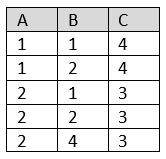
# Relational Model

**Relational Model:** represents the database as a collection of relations (tables).  
   
 **Relation:** it is the table which contains rows and columns.  
   
 **Tuple:**  one row or entity of a relation (table) which contains single record.  
   
 **Attribute:** each column in a relation (table). Attribute are the properties which defines a relation.  
   
 **Domain:** possible values an attribute can take in a relation.   
   
 **Relation Schema:** represents the name of the relation with its attribute.  
   
 **Degree:** total number of attributes (column) in a relation.  
   
 **Cardinality:** total number of tuples (rows) in a relation.  
   
   
   
 **Relational Model and CODD's rule:** [relational-model-introduction-and-codd-rules](https://www.geeksforgeeks.org/dbms-relational-model-introduction-and-codd-rules/)  
   
 **Relation Model:** [relational-model](https://www.geeksforgeeks.org/relational-model/)  
   
 Difference Between DBMS and RDMS:   
   
   
 *More differences between DBMS and RDMS:*   
 1. [difference-dbms-vs-rdbms](https://www.guru99.com/difference-dbms-vs-rdbms.html)  
 2. [difference-between-rdbms-and-dbms](https://www.geeksforgeeks.org/difference-between-rdbms-and-dbms/)  
   
 Constraints in Relational Model:   
 Constraints are the conditions which must hold for data present in database.   
 These constraints checked before any operation on database.  
 Purpose of Constraints is to maintain the data integrity during any operation like insert, update, delete. Constraints ensure the correctness of data in the database.  
   
 **Types of Constraints:**  
 1. Domain Constraint.  
 2. Tuple Uniqueness Constraint.  
 3. Key Integrity Constraint.  
 4. Referential Integrity Constraint.  
 5. Entity Integrity Constraint.  
   
 **1. Domain Constraint:**   
 Column of a relation allows only a same type of data to enter, based on its data type. The column doesn't accept the values of any other data type.   
   
 e.g. column - Id int (10)   
 Since Id column is of int type so, Id column allows only values of int type.  
   
   
   
 **2. Tuple Uniqueness Constraint:**  
 It specifies that each tuple in a relation must be identified uniquely.  
   
   
   
 **3. Key integrity Constraint:**   
 Primary key and foreign key are used to define relationships between tables.   
   
 **Primary Key:**   
 -only one primary key is allowed in a relation.  
 -All the values of primary key must be unique.  
 -value of primary key can't be null.  
 -Primary key uniquely identifies each record in a relation.  
   
   
   
 **4. Foreign key constraint (Referential Integrity Constraints):**  
 Referential Integrity Constraint enforced when foreign key references the primary key of a relation. It specifies that values taken by foreign key must present in the primary key of other relation.  
   
   
   
 **5. Entity Integrity Constraint:**   
 Entity Integrity Constraint specifies that primary key Value can't be null as primary key is used to identify unique tuple in relation and if primary key allows null value then we are not able to identify unique tuple.   
   
   
   
 Keys in Relational Model:   
   
   
   
 **Super key:** it is a set of one or more attributes to uniquely identify tuples in a relation or we can say that it is a group of single or multiple keys which identifies tuples (rows) in a relation(table).   
   
 **Candidate key:** the minimal set of attributes which can uniquely identify a tuple, or we can say that a super key with no repeated attribute. It is a subset of Super Key or minimal superkey.  
   
 **Primary key:** a column or group of columns that uniquely identifies tuples in a relation. The primary key value cannot be null. Primary key can be selected from one of the candidate keys and becomes the identifying key of a relation. It is a subset of Candidate key.  
   
 **Alternate Key:** All the candidate keys other than primary key. It is also known as secondary key.  
   
 **Composite Key:** A key that consists of more than one attribute to uniquely identify tuple in a relation. Composite key may or may not be a part of foreign key.  
   
 **Foreign Key:** when attribute of one relation references the primary key of another table. It is used to create a relationship with other relation (table).   
   
   
   
 Learn keys in detail with examples: [dbms-keys](https://www.guru99.com/dbms-keys.html)  
   
 Anomalies in Relational model:   
   
 There is different type of anomalies that can occur in poorly planned, unnormalized databases. These anomalies can occur when the constraints are missing, or the normalization level is incorrect.   
   
 *Types of Anomalies:* Insert Anomaly, Update Anomaly, Delete Anomaly  
   
 **Insert Anomaly:** this anomaly occurs when attributes cannot be inserted into the database without the presence of another attribute. This mainly occurs due to inserting inconsistent data into a relation.   
   
   
   
 We can't add a new department unless we have at least one employee maps to the department.  
 e.g. if we want to add a new department i.e. Sales then we can't add that department until there is at least one employee maps to the Sales Department.  
   
 **Update Anomaly:** this anomaly occurs when one or more instances of duplicate data is updated, but not all.  
   
   
   
 Suppose HOD of CSE department has been changed, now current HOD of CSE department is 'New\_HOD\_Name'. So, for updating a single entity we have to update all the instance of HOD of CSE department.  
   
 **Delete Anomaly:** this occurs when certain attributes are lost because of the deletion of other attributes.   
 Suppose student Ram leaves the department EEE then all the information about department EEE is lost.  
   
 Video on these Anomalies: [Anomalies in Relational Model](https://www.youtube.com/watch?v=K5P-2-oWXqs)

# Relational Algebra

First learn about procedural and non-procedural query language: [procedural and non-procedural query language](https://www.youtube.com/watch?v=uZogMFWBGSg&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=86)  
   
 Relational Algebra:   
 Relational Algebra is a procedural query language which take relation as input and gives relation as output.  
 It uses various operation to perform this action. It is performed recursively on a relation and intermediate results are also considered as relations.  
   
 Video on Relational Algebra: [Relational Algebra](https://www.youtube.com/watch?v=AepLj_C4ywM&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=87)  
   
 Theory:  
 [database-management-system-relational-algebra](https://www.geeksforgeeks.org/database-management-system-relational-algebra/)  
 [relational-algebra-dbms](https://www.guru99.com/relational-algebra-dbms.html)  
 [basic-operators-in-relational-algebra](https://www.geeksforgeeks.org/basic-operators-in-relational-algebra-2/)  
 [extended-operators-in-relational-algebra](https://www.geeksforgeeks.org/extended-operators-in-relational-algebra/)  
 [dbms-row-oriented-vs-column-oriented-data-stores](https://www.geeksforgeeks.org/dbms-row-oriented-vs-column-oriented-data-stores/)  
   
 Types of operations in relational algebra:   
   
 1. **Basic/fundamental operations -** Select, project, Union, Set Difference, Cartesian product, Rename.  
 2. **Derived operations -** Join (Natural, Left, Right, Full), Intersection, Division.  
   
   
   
   
 1. Basic/Fundamental operations:   
   
 **a. Select (σ):**   
 It selects tuples that satisfy the given predicate/condition from a relation.  
 It is Unary operator which can take one table at a time.   
   
 **Syntax:**  select_algebra   
   
  **e.g.**  Find the details of Student who live in Rohini  
   
 select_algebra_eg   
   
    
   
 Video on Select operation:   
 1. [Select operation1](https://www.youtube.com/watch?v=z0dzM7Dadi8&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=88)  
 2. [Select operation2](https://www.youtube.com/watch?v=PJkHEdG8E-A&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=89)  
   
 **b. Project (π):**   
 It selects the desired columns from a relation.  
 It is Unary operator which can take one table at a time.   
   
 **Syntax:**  project_algebra   
   
  **e.g.** Find all student name along with their rank  
   
    
   
   
 1. Video on Project Operation: [Project Operation](https://www.youtube.com/watch?v=Cob3hPUdrJk&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=90)  
   
 2. Select and Project operations examples: [Select and Project operations](https://www.youtube.com/watch?v=iOJlmwXzDB4&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=91)  
   
 **c. Union (U):**  
 It works on multiple tables. It includes all the tuples that are in table A or in B. Union helps to remove duplicate tuples.   
   
 **Syntax:**  A U B  
   
    
   
 Video on Union: [Union](https://www.youtube.com/watch?v=c-FqvuGIi-8&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=92)  
   
 **d. Set difference (-):**  
 If A - B, then the result of this includes all the tuples that are in A but not in B.  
 If B - A, then the result of this includes all the tuples that are in B but not in A.  
   
    
   
    
   
 Video on Set difference: [Set difference](https://www.youtube.com/watch?v=c-FqvuGIi-8&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=92)  
   
 **e. Cartesian product (X):**  
 It is helpful to merge columns from two relations.   
   
    
   
 Video : [Cartesian product](https://www.youtube.com/watch?v=VJp27dmRxXs&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=93)  
 Problems: [Problems in Cartesian product](https://www.youtube.com/watch?v=CHwLkUlq-vE&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=94)  
   
 **f. Rename (ρ):**  
 Result of relational algebra is relation but without any name so, rename operations helps in that. Rename operation helps to rename the output relation.   
   
 2. Derived operations:   
   
 **a. Natural Join:**  
 Video on Natural Join [Natural Join](https://www.youtube.com/watch?v=PHPHkqHAu9Y&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=101)  
 A Natural Join is a join operation in which 2 tables are joined based on all common columns.   
   
    
   
 **b. Left Outer Join:**  
 It returns all records from the left relation and matched records from the right relation and if there is no match then the result is null from the right side.  
   
    
   
 **c. Right Outer Join:**  
 It returns all records from the right table and matched records from the left table and if there is no match then the result is null from the left side.  
   
    
   
 **d. Full Outer Join:**  
 It returns all matching records from both tables whether the other matches or not.  
   
    
   
 Outer Join or Full Outer join: [Outer Join or Full outer join](https://www.youtube.com/watch?v=ldos7yYMjs8&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=104)  
   
 Join or Inner Join: [Join or Inner Join](https://www.youtube.com/watch?v=lJv614Y0lFA&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=102)  
   
 Join or Inner Join with 'ON' operation: [Join or Inner Join with 'ON' operation](https://www.youtube.com/watch?v=z996Uv9MNaQ&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=103)  
   
 **e. Intersection (∩):**  
 Records that are present in both the relation A and B will only present in the set obtained by intersection of A and B (i.e. common tuple in both relation).   
   
    
   
 **f. Division:**   
    
   
 **Extended operators in Relational Algebra:** [Extended operators in Relational Algebra](https://www.geeksforgeeks.org/extended-operators-in-relational-algebra/)

# Functional Dependencies

Functional Dependency (FD):   
 It is a relationship that exists when one attribute uniquely determines another attribute.   
   
   
   
 If A and B are attributes of relation R then functional dependency between the attributes is represented by A->B, specifies that B is functional dependent on A where A is determinant set and B is a dependent attribute.  
   
 **E.g. FD: A -> B**  
   
   
   
 **Note:** If there is FD: A -> B in a relation R, shows that for the given value of A, we can search the value of B.  
   
   
   
 Que1. Find out the correct functional dependency  
   
   
   
 **A -> B** (FD holds as there is distinct values for each value a, b, c, d)   
 **B -> A** (FD doesn't hold as there are three different values when B = 2 i.e. 2 determines b, c, d)  
   
 Que 2: Find out correct functional dependency  
   
   
   
 **A -> B**   X (there are two different values (B = 1,2) when A = 1 and three different values (B= 1, 2, 4) when A = 2)  
 **B -> C**   X (there are two different values (C= 4, 3) when B = 1 and two different values (C = 4, 3) when B = 2)  
 **B -> A**   X (there are two different values (A = 1, 2) when B = 1 and three different values (A= 1, 2) when B = 2)  
 **C -> B**   X (there are two different values (B = 1, 2) when C = 4 and three different values (B= 1, 2, 4) when C = 3)  
 **C -> A**   FD holds (there are distinct values of A for each value of C)  
 **A -> C**   FD holds   
   
 Theory:   
 1. [functional-dependency-and-attribute-closure](https://www.geeksforgeeks.org/functional-dependency-and-attribute-closure/)  
 2. [functional-dependency](https://www.guru99.com/dbms-functional-dependency.html)  
   
 Video:   
 1. [functional-dependency](https://www.youtube.com/watch?v=wez3fXrjBAE&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=12)  
 2. [practice problem on functional-dependency](https://www.youtube.com/watch?v=y8XuGhEdslM&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=13)  
 3. [practice problem on functional-dependency2](https://www.youtube.com/watch?v=d37Ct1gHmsU&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=14)  
   
 Inference rules used in FD:   
   
 **1. Reflexive:**    
 A -> B if B ⊆ A  
   
 **2. Transitive:**   
 A -> B and B -> C then A -> C  
   
 **3. Decomposition:**    
 If A -> BC, then A -> B and A -> C  
   
 **4. Augmentation:**   
 If A -> B, then AC -> BC  
   
 **5. Union:**    
 If A -> B and A -> C, then A -> BC  
   
 **6. Composition:**    
 If A -> B and C -> D, then AC -> BC  
   
 **7. Pseudo transitivity:**   
 If X -> Y and WY -> Z, then WX -> Z   
   
 Armstrong's Axioms in FD: [Armstrong's Axioms](https://www.geeksforgeeks.org/dbms-armstrongs-axioms-functional-dependency/)  
   
 Armstrong's axioms are a set of axioms or inference rules. These are used to infer all the functional dependencies on relational database.   
   
 Video on Armstrong's axioms: [Armstrong's Axioms](https://www.youtube.com/watch?v=vs65S6Nku5g&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=15)  
   
 Closure sets/ Attribute closure:  
 Attribute Closure of an attribute can be defined as set of attributes which we can identified from it.  
   
 Closure set of an attribute: [Closure set of an attribute](https://www.youtube.com/watch?v=fT0QMtgqPrU&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=16)  
   
  **e.g. Find Closure   
 FD: A ->B  
 B -> D  
 C -> DE  
 CD -> AB**   
   
 Solution:

##### **(A)+ = {A, B, D} (B)+ = {B, D} (C)+ = {C, D, E, A, B} (D)+= {D} (E)+ = {E} (AB)+ = {A, B, D} (CD)+ = {C, D, E, A, B} (ABD)+ = {A, B, D}**

Equivalence of Functional Dependency:   
 Theory: [equivalence-of-functional-dependencies](https://www.geeksforgeeks.org/equivalence-of-functional-dependencies-sets/)  
   
 Video:   
 1. [equivalence-of-functional-dependencies](https://www.youtube.com/watch?v=NeITRksKLzs&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=17)  
 2. [practice problem on equivalence-of-functional-dependencies](https://www.youtube.com/watch?v=0XmHRycmrp0&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=18)  
   
 Suppose F and G are two functional dependencies for a relation R.   
 If F ⊆ G and G ⊆ F, then they are equal.   
   
 Que: R (ACDEH)  
  **F: A -> C, AC -> D, E -> AD, E -> H   
 G: A -> CD, E -> AH**    
   
 **Step 1:** F ⊆ G ?  
 Find closure of A, AC and E using production rule of G

##### **(A)+ = {ACD} (AC)+ = {ACD} (E)+ = {EAHCD}**

Therefore, F ⊆ G holds  
   
 **Step 2:** G ⊆ F ?  
 Find closure of A and E using production rule of F

##### **(A)**+ **= {ACD} (E)**+**= {EADHC}**

Therefore, G ⊆ F holds  
   
 **Conclusion:** F and G functional dependencies are equal.  
   
 Minimal Set of Functional Dependencies:  
   
 Canonical Cover: [canonical-cover](https://www.geeksforgeeks.org/canonical-cover/)  
   
 Video:   
 [canonical-cover irreducible](https://www.youtube.com/watch?v=o0GQQFu-5C0&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=19)  
 [practice problem on canonical-cover-irreducible](https://www.youtube.com/watch?v=_-F6QfdheEk&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=20)  
   
 **Steps to find minimal set:**   
 Split the FDs in such a way that RHS contain single attribute  
   
 **e.g. FD:** A -> BC then split them as A -> B and A -> C  
   
 Find the redundant FDs and remove them from the set.  
   
  **e.g. FD:** A -> B, B -> C, A -> C   
 then remove redundant productions (here A -> C, as this production can be derived from first two productions)  
 now productions are: A -> B and B -> C   
   
 Find the redundant attributes on the LHS and remove them  
   
 Que: R (A, C, D, E, H)   
  **FD: A -> C  
 AC -> D  
 E -> A  
 E -> D  
 E -> H**   
   
 Step 1. Ignore A -> C and find (A)**+**

##### **(A)+= {A}**

Therefore, A -> C is required   
   
 **Step 2.**  Ignore AC -> D and find (AC)**+**

##### **(AC)+ = {AC}**

Therefore, AC -> D is required   
   
 **Step 3.** Ignore E -> A and find (E)**+**

##### **(E)+= {EDH}**

Therefore, E -> A is required   
   
 **Step 4.** Ignore E -> D and find (E)**+**

##### **(E)+ = {EAHCD}**

Therefore, E -> D is redundant as it is equal to R  
 So, remove this redundant production   
   
  **Step 5.** Ignore E -> H and find (E)**+**

##### **(E)+ = {EAC}**

Therefore, E -> H is required   
   
 After removing redundant productions, new production set is:  
   
  **A -> C  
 AC -> D  
 E -> A  
 E -> H**  
   
 Step 6. Find redundant attributes on LHS  
 AC -> D  
   
 Find (AC)+

##### **(AC)+= {ACD}**

Now find closure of A and C separately

##### **(A)+ = {ACD} (C)+ = {C}**

[Note if (A)+ = (AC)**+** then C is redundant and if (C)+ = (AC)+ then A is redundant]  
   
 Here, (A)+ = (AC)**+** then C is redundant here  
   
 Final FD is after removing all redundant productions is:  
 **A -> C  
 A -> D  
 E -> A  
 E -> H**  
   
 Find keys:   
 Keys : [Keys](https://www.youtube.com/watch?v=8N2Z2WPQFcU&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=21)  
   
 How to find Super Key, Candidate key, Primary key:   
 [How to find Super Key, Candidate key, Primary key](https://www.youtube.com/watch?v=MhvAcuNvvNk&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=22)  
 [find number of Candidate key in a relation](https://www.youtube.com/watch?v=x_inLVXPlSU&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=23)  
 [Practice questions to find Super Key, Candidate key, Primary key](https://www.youtube.com/watch?v=33ek39nWkVA&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=24)  
   
 Que:   
 **R (A, B, C, D, E, F)  
 FD:   
 C -> F   
 E -> A   
 EC -> D  
 A -> B  
 Find candidate key?**  
   
 Solution:   
 Number of possible candidate key: **2n - 1**   
 Here, n = 6 (number of attributes in a relation)   
 26 - 1 = 63   
   
 If **(A)+ = R,** then A is a candidate key   
   
 Step 1. Find the attributes that are not present in RHS i.e. attribute which is not having incoming edge  
   
 So, attributes with no incoming edge are: C, E  
   
 Step 2. Find closure of attributes C and E

##### **(C)+ = {CF} (Not equal to R) (F)+ = {F} (Not equal to R)**

Now find closure of CE

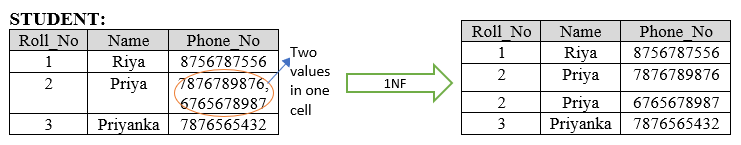
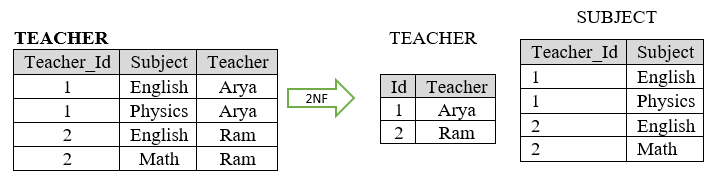
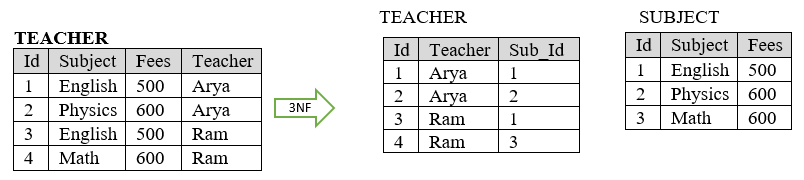
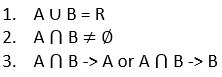
##### **(CE)+= {CEFADB}**

Since

##### **(CE)+ = R**

then CE is a candidate key.

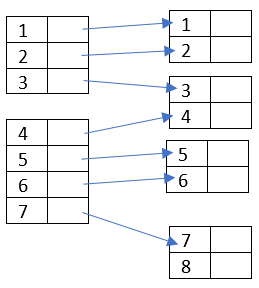
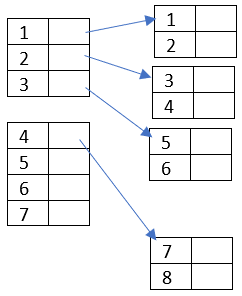
# Normalization

Prerequisite:   
 Anomalies in DBMS: [Insertion, deletion, updation Anomalies in DBMS](https://www.youtube.com/watch?v=K5P-2-oWXqs&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=25)  
   
 Normalization:   
 Normalization is a process of organizing the data in the database to avoid insert anomaly, update anomaly and deletion anomaly.   
 It is mainly used to avoid data redundancy in the database.   
 It mainly divides the larger table into the smaller table and link them using relationship.  
   
 *Theory:*   
 1. [database-normalization-introduction](https://www.geeksforgeeks.org/database-normalization-introduction/)  
 2. [database-normalization-normal-forms](https://www.geeksforgeeks.org/database-normalization-normal-forms/)  
   
 *Video on Normalization:* [Normalization](https://www.youtube.com/watch?v=px7HV91fx2I&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=26)  
   
 Types of Normal Forms (NF):   
 1. 1NF  
 2. 2NF  
 3. 3NF  
 4. BCNF  
   
 **1. First Normal Form (1NF):**  
 According to the 1NF, an attribute of a relation cannot hold multiple values i.e. only atomic values are allowed.   
   
   
   
 *Video:* [First Normal Form](https://www.youtube.com/watch?v=CedOasDoe-w&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=27)  
   
 **2. Second Normal Form (2NF):**   
 A relation is in 2NF if -  
 a. A relation must be in 1NF.  
 b. No partial dependency.   
   
 **Partial dependency:** when non-prime attribute is dependent on the proper subset of any candidate key of relation.   
   
 **Prime attributes:** those which are part of any candidate key.  
 **Non-prime attributes:** those which are not part of any candidate key.   
   
 *Problem:* one teacher can teach more than one subjects, then relation can have multiple tuples for the same teacher.  
   
   
   
 Normalize relation using 2NF: [Second Normal Form](https://www.youtube.com/watch?v=80CcB9_HSxU&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=28)  
   
 Reason for 2NF: [Reason for 2NF](https://www.youtube.com/watch?v=Wx5CzybyLXA&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=29)  
   
 Decompose a relation into 2NF: [Decompose a relation into 2NF](https://www.youtube.com/watch?v=yIN6k57OB3U&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=30)  
   
 **3. Third Normal Form (3NF):**   
 A relation is in 3NF if -  
 a. A relation must be in 2NF.  
 b. No transitive dependency.   
   
 If R is a relation and A -> B, then   
 Relation R is in 3NF if any of the following condition holds-  
 a. A is super key.  
 b. B is a prime attribute.   
   
   
   
 *Theory:* [normalization-normal-forms](https://www.geeksforgeeks.org/database-normalization-normal-forms/)  
   
 *Video:*  
 3NF [Third Normal Form](https://www.youtube.com/watch?v=aOZ5z7z9y-g&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=31)  
 3NF Decomposition [3NF Decomposition](https://www.youtube.com/watch?v=9H4aJqYyd9s&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=32)   
   
 **4. Boyce Codd Normal Form (BCNF):**   
 It is also referred as 3.5NF as it is advanced version of 3NF.   
 A relation is in BCNF if -  
 a. Relation is in 3NF  
 b. For every functional dependency A -> B, A should be the super key of the relation.   
   
 *Video:*   
 1. Boyce Codd Normal Form [Boyce Codd Normal Form](https://www.youtube.com/watch?v=cm1DYCQhFZw&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=33)  
 2. BCNF identification: [BCNF identification](https://www.youtube.com/watch?v=iuJtkzoNtbg&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=34)  
   
 4th and 5th Normal Form: [introduction-of-4th-and-5th-normal-form](https://www.geeksforgeeks.org/dbms-introduction-of-4th-and-5th-normal-form/)  
   
 How to find normal form?  
 1. [How to find normal form1](https://www.youtube.com/watch?v=mzxnbsmIRNw&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=35)  
 2. [How to find normal form2](https://www.youtube.com/watch?v=oeWSmSH5YeI&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=36)  
 3. [How to find normal form3](https://www.youtube.com/watch?v=hL9-RDzvO-0&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=37)  
   
 Lossless join Decomposition:   
 It is central in removing redundancy safely from the databases while preserving the original data. It can also be non-additive join decomposition.   
   
 A decomposition (R1, R2, ..., Rn) of a relation R is called a lossless decomposition for relation R if the natural join of R1, R2, ..., Rn produces exactly the relation R.   
   
 If R is a relation that split into A and B then, Decomposition is lossless if:   
   
   
 Theory:  
 1. [lossless-decomposition](https://www.geeksforgeeks.org/database-management-system-lossless-decomposition/)  
 2. [lossless-join-and-dependency-preserving-decomposition](https://www.geeksforgeeks.org/lossless-join-and-dependency-preserving-decomposition/)  
   
 Video: [lossless-join-decomposition](https://www.youtube.com/watch?v=M5OClnF_mlA&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=38)  
   
 Example: [practice-problem-on-lossless-join-decomposition](https://www.youtube.com/watch?v=3aQ0dnZHaRQ&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=39)  
   
 Dependency preserving decomposition:   
 If relation R is decomposed into relation A and B, then all dependencies (productions) of R either must be a part of A or B must be derivable from combination of FDs of A and B.  
 Functional dependency set = F   
   
   
   
 Theory: [dependency-preserving-decomposition](https://www.geeksforgeeks.org/data-base-dependency-preserving-decomposition/)  
 Video: [dependency-preserving-decomposition](https://www.youtube.com/watch?v=isOLwsdB4Ns&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=40)  
   
 How to Normalize a relation table: [How to Normalize a relation table](https://www.youtube.com/watch?v=I3Zwz4lKmDU&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=41)

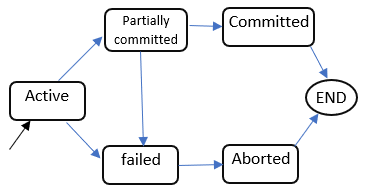
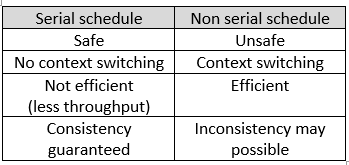
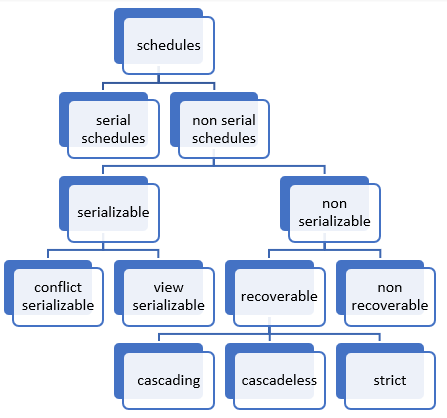
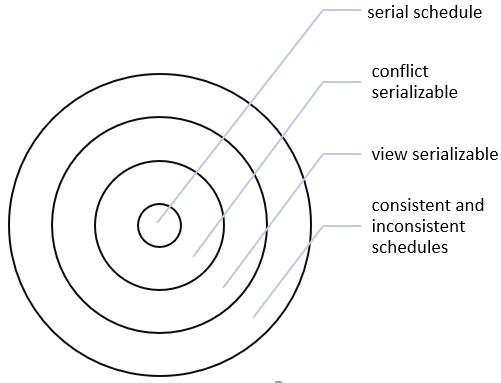
File Organization

In RDBMS, data is stored in the form of table but in actual this huge amount of data is stored in physical memory in the form of files.   
   
 File:   
 File is a collection of related information that can be recorded on secondary storage.   
   
 File Organization:   
 Files organization specifies that how file records are mapped on to disk block. It mainly refers to the logical relationship among various records.   
 Files of the fixed length records are easier to implement than the files of variable length records.   
   
 File structure in DBMS: [File structure](https://www.youtube.com/watch?v=RgYU5r9A5TU&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=42)  
   
 Objective of file organization:   
 1. Records can be selected as fast as possible.   
 2. Operations on the records (insert, update, delete) should be quick and duplicated records cannot be induced as a result of these operations.   
 3. Records should be stored efficiently so that the cost of storage is minimal.  
   
 Methods of organizing files:  
 1. Sorted file  
 2. Unsorted file  
   
 **1. Sorted file:** in this, records of a file are sorted according to some attribute so that access is very fast.   
 **2. Unsorted file:** in this, records are ordered randomly. In this, access is very slow as it requires the linear search.   
   
 Difference between sorted and unsorted files: [Difference between sorted and unsorted file](https://www.youtube.com/watch?v=D7pmQDnjcV4&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=43)  
   
 Types of file organization:   
 1. Sequential File Organization.  
 2. Heap File Organization.  
 3. Hash File Organization.  
 4. B+ tree File Organization.  
 5. Cluster File Organization.  
   
 Tabular difference between types of file organization: [Tabular difference between types of file organization](https://www.tutorialcup.com/dbms/file-organization.htm)  
   
 **1. Sequential File Organization:**  
 Records are stored and accessed in an order using a key field.   
 Searching is sequential throughout the file i.e. record by record.   
   
 Theory with examples:   
 [sequential-file-organization](https://www.javatpoint.com/dbms-sequential-file-organization)  
 [file-organization-set-1](https://www.geeksforgeeks.org/dbms-file-organization-set-1/)  
   
 **2. Heap File Organization:**   
 Records are stored at the end of the file but the address in the memory is random.  
 No sorting or ordering is required.   
   
 Theory:   
 [file-organization-set-1](https://www.geeksforgeeks.org/dbms-file-organization-set-1/)  
 [heap-file-organization](https://www.javatpoint.com/dbms-heap-file-organization)  
   
 **3. Hash File organization:**  
 Records are stored at the hash address generated. It uses the computation of hash function on some fields of the records and the output of the hash function determines the address of the disk block where the records are to be placed. The field on which hash function is calculated is known as hash field and if that field is acts as a key of the relation then it is known as hash key.   
   
 Theory:  
 [hashing-in-dbms](https://www.geeksforgeeks.org/hashing-in-dbms/)  
 [hash-file-organization](https://www.javatpoint.com/dbms-hash-file-organization)  
   
 **4. B+ Tree organization:**   
 Records are stored in tree like structure. B+ tree mainly used to store large amount of data that will not fit in main memory so for this mainly secondary memory is used to store the leaf nodes of the tree and internal nodes of the tree are stored in main memory. In B+ tree the leaf nodes store data and all other nodes (index nodes) are used as indexing.   
   
 Theory:   
 [file-organization-set-3](https://www.geeksforgeeks.org/dbms-file-organization-set-3/)  
 [b-plus-file-organization](https://www.javatpoint.com/dbms-b-plus-file-organization)  
   
 **5. Cluster file Organization:**   
 In this, frequently joined tables are clubbed into one file based on cluster key. This method reduces the searching cost for various records in different files.   
   
 Types of cluster file organization:   
 a. Indexed clusters.  
 b. Hash clusters.   
   
 **a. Indexed clusters:** records are grouped based on the cluster key and stored together.   
 **b. Hash clusters:** value of hash key is generated for the cluster key and store the records with the same hash key value.   
   
 Indexing in DBMS:   
   
 It is used to optimize the performance of a database by minimizing the number of disk accesses required when a query is processed.   
 Index is a type of data structure which is used to locate and access the data in a database table quickly.   
 Indexing in the database is like the what we see in books.   
   
 **Index table:**    
 it is a small table having only two columns in which first column contains a search key (which is a copy of the primary or candidate key of a table) and the second column contains pointers holding the reference of the disk block where that particular key value can be found.

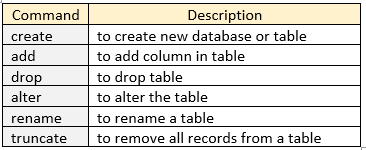
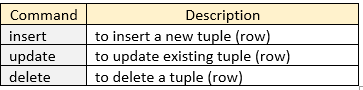
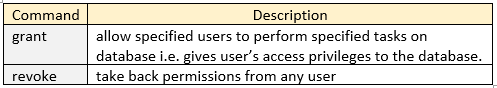
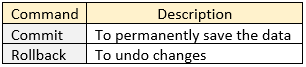
|  |  |
| --- | --- |
| **Search key** | **Data reference** |

Where search key is primary or candidate key of a table.   
   
 Video on Indexing : [Indexing](https://www.youtube.com/watch?v=SxHX1T53n_A&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=44)   
   
 Theory: [indexing-in-databases-set-1](https://www.geeksforgeeks.org/indexing-in-databases-set-1/)  
   
 Types of indexing:  
 1. Primary indexing.  
 2. Clustered indexing.  
 3. Secondary indexing.  
   
 Video on types of indexing: [types of indexing](https://www.youtube.com/watch?v=O-Mbn6VI1zc&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=45)  
   
 **1. Primary indexing:**  
 Primary index is defined on an ordered data file. The data file is ordered on a primary key (key field) of a table. In this, primary keys are stored in sorted order so that searching operation become fast and efficient.   
   
 Index can be of two types:   
 a. Dense index  
 b. Sparse index  
   
 **a. Dense index:** there is an index record for every search key value in the database therefore searching is faster. In this, number of records in the index table is equal to the number of records in the main table. It needs more space to store index record itself.   
   
   
   
 **b. Sparse index:** in this, index records are not created for every search key i.e. index records are created for few search keys. Instead of pointing to each record in the main table, the index points to the records in the main table in a gap.   
   
   
   
 Video on primary indexing: [primary indexing](https://www.youtube.com/watch?v=L-THExvsv0s&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=46)  
   
 Numerical on primary indexing: [primary indexing](https://www.youtube.com/watch?v=mbjE4WsWYCA&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=47)  
   
 **2. Cluster indexing:**   
 Cluster index is defined on sorted data file with two columns, one for non-key values and one for block pointer. The data file is ordered on a non-key field.  
 The index is created on non-key (key may be primary key or candidate key) columns which may not be unique for each record so for this we will group two or more columns to get the unique value and create index out of them.   
   
 Video on cluster index: [cluster index](https://www.youtube.com/watch?v=54gUz7QqE4o&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=48)  
   
 **3. Secondary indexing:**   
 Secondary index may be generated from a field which is a primary/candidate key or a non-key.   
   
 Video on secondary index: [secondary index](https://www.youtube.com/watch?v=SoYYwKXtCC0&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=49)  
   
 Numerical on secondary index: [Numerical on secondary index](https://www.youtube.com/watch?v=Tmbv15xiIPo&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=50)  
   
   
 Secondary Vs Primary Index: [Secondary Vs Primary Index](https://www.youtube.com/watch?v=6_94Fm_yNAM&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=51)  
   
 SQL queries on clustered and non-clustered indexes: [SQL queries on clustered and non-clustered indexes](https://www.geeksforgeeks.org/sql-queries-on-clustered-and-non-clustered-indexes/)

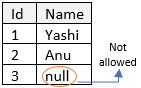
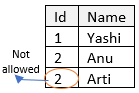
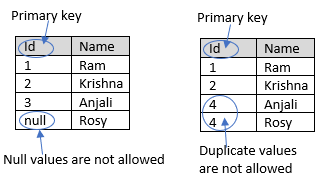
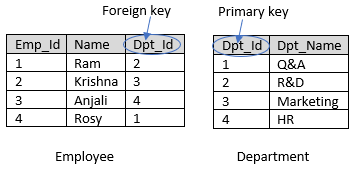
# Transactions And Concurrency Control

Transaction:   
 Set of instructions perform logical unit of work.   
 Instructions of transactions either run completely or not run at all.   
 These set of instructions are one atomic unit of work.   
 The main operations of a transaction are Read and Write.   
   
 **Read(A):** read operation reads the value of A from the database and stored it in a buffer in main memory.   
   
  **Write(A):** write operation writes the value back from buffer to the database.   
   
 We can do some operations on transactions:  
  **Commit:** it is used to save the work done permanently in database.  
  **Rollback:** it is used to undo the work done.   
   
 Transaction theory: [concurrency-control-introduction](https://www.geeksforgeeks.org/concurrency-control-introduction/)  
   
  **Transaction management video:**   
 1. [Transaction1](https://www.youtube.com/watch?v=a6KIAX5Aubg&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=52)  
 2. [Transaction2](https://www.youtube.com/watch?v=quz6gV7oDFI&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=53)  
   
 There are four important properties of transactions/ ACID properties:  
 1. Atomicity  
 2. Consistency.  
 3. Isolation.  
 4. Durability.   
   
  **1. Atomicity:**   
 Atomicity requires each transaction is all or nothing i.e. instructions of transaction either run completely or not run at all (rollback).  
   
  **2. Consistency:**   
 It ensures that database remains in consistent state before and after the transaction i.e. it ensures bringing the database from one consistent state to another consistent state.   
 Some integrity constraints are maintained so that database is consistent before and after the transaction.  
   
  **3. Isolation:**   
 Execution of one transaction is isolated from that of another transactions i.e. transactions executed isolate.   
 Although multiple transactions may execute concurrently, each transaction must be unaware of other concurrently executing transactions. Intermediate result of transaction must be hidden from other concurrently executed transactions.   
   
  **4. Durability:**  
 Once transaction completes successfully, the change made by the transaction should be permanent even if there are system failure.   
   
 ACID properties:   
 Theory:   
 [acid-properties](https://www.geeksforgeeks.org/acid-properties-in-dbms/)  
 [transaction-property](https://www.javatpoint.com/dbms-transaction-property)  
   
 Video: [ACID properties](https://www.youtube.com/watch?v=qMBQCDTVROc&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=54)  
   
 Transaction states:   
 Video [Transaction states](https://www.youtube.com/watch?v=914rYy0Djrg&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=55)  
   
   
   
  **1. Active:**  It is the first state of every transaction in which transaction is being executed.   
   
  **2. Partially committed:** when transaction executes its final operation but not committed i.e. value is still not saved in the database.   
   
  **3. Committed:** if transaction executes all its operation completely then it is said to be committed and all the effects are now permanently stored on the database.   
   
  **4. Failed:** A transaction considers failed when any one of the checks fails or if transaction is aborted while it is in active state.   
   
  **5. Aborted:** after the transaction has been rolled back and database restored to its state prior to the start of the transaction.   
 After aborting the transaction, the database recovery module will select one the two operations:   
 a. Restart the transaction.   
 b. Kill the transaction.   
   
 Concurrency control:  
 Concurrency control is the process for managing simultaneous operations without conflicting with each another.   
 Concurrency control is more important for proper functioning of system where two or more transactions require to access to the same data.   
   
 Need of concurrency control:  
 Since several transactions require to access the same data simultaneously then it may lead to problem of data integrity and inconsistency.  
   
 Advantage of concurrency control : [Advantage of concurrency control](https://www.youtube.com/watch?v=XrHzsxLbLh4&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=56)   
   
 Problems of concurrency control:  
 1. Dirty read.  
 2. Lost updates.  
 3. Unrepeatable read.  
 4. Phantom read.  
   
  **1. Dirty read:**  
 This problem occurs when second transaction selects a data which is updated by another transaction before committing.   
 Video on dirty read: [dirty read](https://www.youtube.com/watch?v=CPOfYF0Ya6E&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=57)  
   
 Theory: [dirty-read](https://www.geeksforgeeks.org/dbms-dirty-read-in-sql/)  
   
  **2. Lost updates:**   
 This problem occurs when multiple transactions select the same row and update the row based on the value selected.   
   
 Video on lost update: [lost update](https://www.youtube.com/watch?v=o7RUUgHS3ck&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=60)  
   
  **3. Unrepeatable read:**  
 This problem is also known as Inconsistent Retrievals Problem. This problem occurs when a transaction accesses data before and after another transaction finish working with such data.   
   
 Video on unrepeatable read: [unrepeatable read](https://www.youtube.com/watch?v=l0177OcTTdk&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=58)  
   
 **4. Phantom read:**   
 Data getting changed in current transaction by other transaction.   
   
 Video on Phantom read: [Phantom read](https://www.youtube.com/watch?v=-FyMthiA1kU&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=59)  
   
 Schedule:   
 Schedule describes the execution of the transactions running in the system i.e. order of execution of operations from various transactions.  
   
 Types of schedule:  
 1. Serial schedule.  
 2. Non serial schedule.  
   
  **1. Serial schedule:** Transaction can begin only after the completion of previous transaction i.e. one after another.  
  **2. Non serial schedule:** operations of concurrent transactions are interleaved.  
   
 Video on schedules: [schedules](https://www.youtube.com/watch?v=XCLA0cjk0o8&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=61)  
   
   
   
   
   
 Types of schedulers: [concurrency-control-types-of-schedules](https://www.geeksforgeeks.org/dbms-concurrency-control-types-of-schedules/)  
   
 Conflict serializability:   
 If two or more transactions in a non-serial schedule execute concurrently then it may lead to some conflicting operations.  
   
  **Conditions for conflicting operations:**  
 1. The operations belong to different transactions.  
 2. At least one of the operations is write operation.  
 3. If operations operate on same (shared) data.  
   
 If all the above conditions hold then operations are conflicting.  
   
 Video on conflict serializability: [conflict serializability](https://www.youtube.com/watch?v=vBo0UZl_VUM&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=62)  
   
 Practice question on conflict serializability:   
 1. [conflict serializability1](https://www.youtube.com/watch?v=QkROSmKbVFQ&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=63)  
 2. [conflict serializability2](https://www.youtube.com/watch?v=6feqtT3e-vA&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=64)  
   
 View serializable schedule:  
 If non serial schedule is view equivalent to some serial schedule.   
   
   
   
 Video on view serializability: [view serializability](https://www.youtube.com/watch?v=Ewzmv2jBlmM&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=65)  
   
 Practice problem on view serializability: [Practice problem on view serializability](https://www.youtube.com/watch?v=FJteasXARxg&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=66)  
   
 Theory: [how-to-test-two-schedule-are-view-equal-or-not](https://www.geeksforgeeks.org/dbms-how-to-test-two-schedule-are-view-equal-or-not-2/)  
   
 Recoverable schedules:   
 Recoverable schedules are those schedules which can be handle in between failure.   
 This ensures that transaction still has a chance to recover if the uncommitted transaction fails later.   
   
 Video on recoverable schedules: [recoverable schedules](https://www.youtube.com/watch?v=fCuEd3N9Nrs&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=67)  
   
 Cascading schedules:   
 The phenomenon in which single transaction failure which leads to a series of transaction rollbacks.  
   
 Cascadeless schedules:   
 These schedules avoid cascading rollbacks.   
   
 Video on Cascadeless schedules and cascading schedules: [Cascadeless schedules and cascading schedules](https://www.youtube.com/watch?v=qH2iYtuJEwQ&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=68)  
   
 Strict schedules:   
 A schedule is said to be strict if every value written by a transaction T is not read or changed by other transaction until first transaction T either commits or aborts.   
   
 Video on strict schedules: [strict schedules](https://www.youtube.com/watch?v=MvLHkPCjCtk&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=69)  
   
 Concurrency control technique:   
   
 Video: [Concurrency control technique](https://www.youtube.com/watch?v=fTRF3cr10RQ&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=70)  
   
 Time stamp ordering protocol:   
 This protocol uses a timestamp to serialize the execution of concurrent transactions. This protocol ensures that every conflicting read and write operations are executed in timestamp order.  
   
  **Timestamp:** it is a unique identifier used by DBMS to identify a transaction. Timestamp may be system time or logical count. Timestamp is usually assigned to transactions in the order in which they are entered in the system.   
   
 Theory:   
 [timestamp-ordering-protocols](https://www.geeksforgeeks.org/dbms-concurrency-control-protocols-timestamp-ordering-protocols/)  
 [timestamp-deadlock-prevention-schemes](https://www.geeksforgeeks.org/dbms-introduction-timestamp-deadlock-prevention-schemes/)  
   
 Video:   
 Introduction: [Time stamp ordering protocol](https://www.youtube.com/watch?v=k0Tuf2weFyA&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=71)  
 Protocol in detail: [Time stamp ordering protocol in detail](https://www.youtube.com/watch?v=PYgebFyWZwI&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=72)  
   
 Properties of time stamping protocol: [Properties of time stamping protocol](https://www.youtube.com/watch?v=LyTjtXtfBlM&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=73)  
   
  **Advantage of timestamp protocol:**  
 Eliminates the possibility of deadlock.  
   
 **Disadvantage of timestamp protocol:**  
 Starvation may be possible.  
   
 Thomas's write rule:  
 Theory: [concurrency-control-protocol-thomas-write-rule](https://www.geeksforgeeks.org/dbms-concurrency-control-protocol-thomas-write-rule/)  
 Video: [Thomas's write rule](https://www.youtube.com/watch?v=uLcEOs24Q3s&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=74)  
   
 Lock based protocol:   
 Video: [Lock based protocol](https://www.youtube.com/watch?v=44Uc6ohuOAk&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=75)  
   
 Properties of lock-based protocol: [Properties of lock-based protocol](https://www.youtube.com/watch?v=9aHA73fyt2g&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=76)  
   
 2 phase locking protocol (2PL): [2 phase locking protocol](https://www.youtube.com/watch?v=7gW5vi1fNq4&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=77)  
   
 Conservative state 2PL: [Conservative state 2PL](https://www.youtube.com/watch?v=VwZ58AvwYb0&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=78)

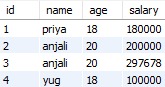
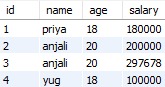
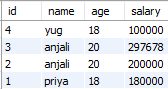
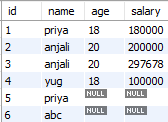
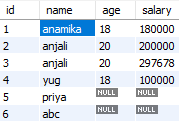
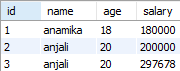
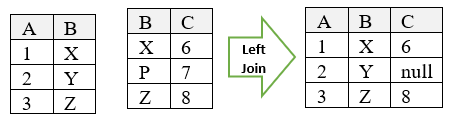
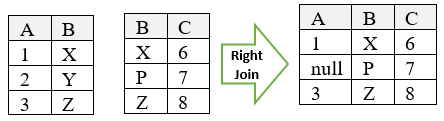
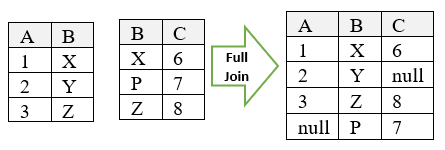
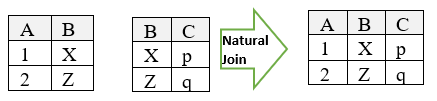
# Sql Introduction

SQL:  
 SQL Stands for Structured Query Language.   
 SQL is database query language which is used to create, manage and retrieve the relational database.   
   
 Relational Database: in relational database, data is stored as well as retrieved in the form of relations (table).  
   
 Terminologies used in Relation:  
 1. **Relation:** table.  
 2. **Attribute:** field or column that define a relation.  
 3. **Tuple:** record or row.  
 4. **Degree:** number of attributes in a relation.  
 5. **Cardinality:** number of tuples.   
   
 With the help of SQL, we can perform various operations:  
 1. Create new database.  
 2. Create new relations (tables) in a database.   
 3. Create stored procedures in a database.  
 4. Create views in a database.  
 5. Retrieve data from database.  
 6. Insert, Update, delete operations on data in a database.  
 7. Set permissions on relations(tables), stored procedures, views.  
   
 Note:  
 SQL keywords are not case sensitive.   
 Some database requires a semicolon at the end of each SQL statement.   
 Semicolon is used to separate each SQL statement in database so that multiple SQL statements can execute on same call (one click) to the server.   
   
 Sql categories:  
 1. DDL (Data Definition Language).  
 2. DML (Data Manipulation Language).  
 3. DQL (Data Query Language).  
 4. DCL (Data Control Language).  
 5. TCL (Transaction Control Language).  
   
 **1. DDL**   
 DDL stands for **Data Definition Language.**  
 DDL is used to define the structure of the database.   
 e.g.   
   
   
   
 **2. DML**   
 DML stands for **Data Manipulation Language.**  
 It is used to manipulate data in the table.   
 e.g.   
   
   
   
 **3. DQL**   
 DQL stands for **Data Query Language.**  
 It is used to retrieve the data from the relation.   
 e.g.   
   
 dql_query  
   
 **4. DCL**  
 DCL stands for **Data Control Language.**  
 It is used for authority purpose.   
 e.g.  
   
   
   
 **5. TCL**  
 TCL stands for **Transaction Control Language.**   
 These types of commands mainly used in transactions and concurrency control.  
 e.g.   
   
   
   
 SQL Categories in detail: [sql-ddl-dml-tcl-dcl](https://www.geeksforgeeks.org/sql-ddl-dml-tcl-dcl/)

# Sql Basics

SQL Datatypes:  
 Each column in a table is require having a name and data type.   
   
 Types of datatype:  
   
 **1. Exact numeric datatype:**   
 e.g. bigint, int, smallint, tinyint, bit, decimal, numeric, money, smallmoney.  
   
 **2. Approximate numeric datatype:**  
 e.g. float, real.  
   
 **3. Date and Time datatype:**  
 e.g. datetime, smalldatetime, date, time.  
   
 **4. Character strings datatype:**  
 e.g. char, varchar, varchar(max), text.  
   
 **5. Unicode character strings datatype:**  
 e.g. nchar, nvarchar, nvarchar(max), ntext.  
   
 **6. Binary datatype:**  
 e.g. binary, varbinary, vabinary(max), image.  
   
 **7. Misc datatype:**  
 e.g. timestamp, xml, json.   
   
 Datatype in sql: [sql-datatypes](https://www.geeksforgeeks.org/sql-datatypes/)  
   
 SQL Comments:   
 1. Single line comments.  
 2. Multi line comments.  
 3. Inline comments.  
   
 **1. Single line comments:**  
 Single line starting with '-' is a comment and will not be executed.  
 e.g. - - single line comment  
   
 **2. Multi line comments:**   
 We can comment out multiple line using '/\*' and '\*/'. Line starting with '/\*' is considered as starting point of comment and ends when '\*/' is encountered.  
 e.g. /\*multiline comment start  
 multiple comments  
 multiline comment ends\*/  
   
 3. Inline comments:  
 Comments can be stated in between the statements and are enclosed in between '/\*' and '\*/'.  
 e.g. create database demo /\*command for creating database\*/  
   
 Article on comments: [sql-comments](https://www.geeksforgeeks.org/sql-comments/)  
   
 SQL constraints:   
 Constraints are the rules or restrictions that we can apply on the type of data in a table.   
 Constraints can be specified at the time of table creation.  
   
 **Types of SQL constraints:**  
 1. Not Null  
 2. Unique.  
 3. Primary key.  
 4. Foreign key.  
 5. Check.  
 6. Default.  
 7. Index.  
   
   
 **1. Not Null constraint:**  
 This constraint ensures that a column cannot have a NULL value.   
   
 **Create table student (Id int (10), Name varchar (20) not null)**  
   
   
   
 **2. Unique constraint:**   
 Ensures that all the values in a column are unique i.e. duplicate values are not allowed.  
   
 **Create table student (Id int (10) unique, Name varchar (20))**  
   
   
   
 **3. Primary key:**  
 -It is a combination of not null and unique.  
 -only one primary key is allowed in a relation.  
 -All the values of primary key must be unique.  
 -value of primary key can't be null.  
 -Primary key uniquely identifies each record in a relation.  
   
   
   
 **4. Foreign key:**  
 Uniquely identifies tuples in another relation.  
 when attribute of one relation references the primary key of another table. It is used to create a relationship with other relation (table).   
   
   
   
 **5. Check:**   
 It ensures that value of column meets the condition.   
   
 **Create table student (Id int (10) primary key, Name varchar (20) not null, Age int (20) not null check (Age > 15))**  
   
 Therefore, Age column only allows those value to enter which are greater than 15  
   
 **6. Default:**   
 Sets the default value for a column when no value is specified.  
   
 **Create table user (Id int (10) not null, Name varchar (20), isActive tinyint (1) default '0'))**  
   
 **7. Index:**   
 It is used to create and retrieve data from the database very quickly.  
   
 SQL Constraint: [sql-constraints](https://www.geeksforgeeks.org/sql-constraints/)  
   
 SQL Creating Roles: [sql-creating-roles](https://www.geeksforgeeks.org/sql-creating-roles/)  
   
 SQL Indexing:   
 Indexes is used by the server to speed up the retrieval of tuples by using pointer. These Indexes are special lookup tables. Index mainly considered as a pointer to the data in table.   
   
 'create index' is used to create indexes in tables and duplicate values are allowed.   
 e.g.  **create index index\_name on table\_name (column1, column2, ...);**  
   
 'create unique index' is used to create unique indexes in tables and duplicate values are not allowed.   
 e.g. **create unique index index\_name on table\_name (column1, column2, ...);**  
   
 'drop index' is used to delete an index.   
 e.g.  **drop index index\_name on table\_name**  
   
 SQL index : [sql\_create\_index](https://www.w3schools.com/sql/sql_create_index.asp)  
   
 SQL Trigger: [sql-trigger-student-database](https://www.geeksforgeeks.org/sql-trigger-student-database/)  
   
 Introduction to NOSQL : [introduction-to-nosql](https://www.geeksforgeeks.org/introduction-to-nosql/)

Sql Clause

1. CREATE:   
 Create statement is used to create database or table.   
   
 **a. Database:**   
 **Syntax:** CREATE DATABASE DATABASE\_NAME   
   
 **e.g.** create database demo;  
   
 Create database: [sql\_create\_db](https://www.w3schools.com/sql/sql_create_db.asp)  
   
 **b. use database:** use statement is used to use the database.   
 **syntax:** USE DATABASE\_NAME  
   
 **e.g.** use demo;  
   
 **c. Table:**   
 Syntax: CREATE TABLE TABLE\_NAME (COLUMN\_NAME1 DATATYPE, COLUMN\_NAME2 DATATYPE...);  
   
 **e.g.** create table employee (id int (10) primary key, name varchar (20) not null, age int (20), salary int (100));  
   
 Create table: [sql\_create\_table](https://www.w3schools.com/sql/sql_create_table.asp)  
   
 2. SELECT:   
 Select statement is used to select data from a database.   
   
 **Syntax:**   
 SELECT \* FROM TABLE\_NAME;   
 OR  
 SELECT COLUMN\_NAME1, COLUMN\_NAME2, ... FROM TABLE\_NAME;  
   
 **e.g.** SELECT \* FROM demo.employee;   
 OR  
 SELECT \* FROM employee;  
   
 select_star_query  
   
 SELECT id, name FROM demo.employee;  
   
 select_attribute_query  
   
 Select clause: [sql\_select](https://www.w3schools.com/sql/sql_select.asp)  
   
 Select distinct: [sql\_distinct](https://www.w3schools.com/sql/sql_distinct.asp)  
   
 Select top: [sql\_top](https://www.w3schools.com/sql/sql_top.asp)  
   
 3. INSERT INTO:  
 Insert statement is used to insert new records in a table.  
   
 **Syntax:**    
 INSERT INTO TABLE\_NAME (COLUMN1, COLUMN2, ...) VALUES (VALUE1, VALUE2, ...);   
 OR  
 INSERT INTO TABLE\_NAME VALUES (VALUE1, VALUE2, ...); //this statement requires values to be inserted in the order or column  
   
 **e.g.**  
 insert into employee (id, name, age, salary) values (2, 'anjali', 20, 200000);   
 OR  
 insert into employee values (1,'priya',18,180000);   
   
 select \* from employee;  
   
 insert_query  
   
 Insert into clause: [sql\_insert](https://www.w3schools.com/sql/sql_insert.asp)  
   
 4. WHERE:   
 It is used to filter the records.  
   
 **Syntax:** SELECT \* FROM TABLE\_NAME WHERE CONDITION;  
   
 **e.g.**   
   
   
   
 select \* from employee where age = 18;  
   
 where_query  
   
 Where clause: [sql\_where](https://www.w3schools.com/sql/sql_where.asp)  
   
 Where clause with AND/OR/NOT: [sql\_and\_or.asp](https://www.w3schools.com/sql/sql_and_or.asp)  
   
 5. ORDER BY:  
 It is used to sort the fetched data either in ascending or descending order based on one or more column.  
   
 **Syntax:   
 Ascending order:**   
 SELECT \* FROM TABLE\_NAME ORDER BY COLUMN\_NAME;   
 OR  
 SELECT \* FROM TABLE\_NAME ORDER BY COLUMN\_NAME ASC;   
   
 **Descending order:** SELECT \* FROM TABLE\_NAME ORDER BY COLUMN\_NAME DESC;   
   
 **e.g.**   
 in ascending order:   
 select \* from employee order by id;   
 OR  
 select \* from employee order by id asc;  
   
   
   
 select \* from employee order by id desc;  
   
   
   
 Order by clause: [sql\_orderby](https://www.w3schools.com/sql/sql_orderby.asp)  
   
 6. IS NULL : [sql\_null\_values](https://www.w3schools.com/sql/sql_null_values.asp)  
   
   
   
 **e.g.** SELECT \* FROM EMPLOYEE WHERE AGE IS NULL;  
   
 is_null_query  
   
 7. UPDATE:   
 It is used to modify the existing records in a table.   
   
 **e.g.** UPDATE EMPLOYEE SET NAME = 'ANAMIKA' WHERE ID = 1;  
   
   
   
 Update clause: [sql\_update](https://www.w3schools.com/sql/sql_update.asp)  
   
 8. DELETE:   
 It is used to delete the existing records in a table.   
   
 **e.g.** DELETE FROM EMPLOYEE WHERE ID=6; //tuple with id = 6 is deleted from employee table  
   
 delete clause: [sql\_delete](https://www.w3schools.com/sql/sql_delete.asp)  
   
   
 9. AGGREGATE FUNCTIONS:  
 Aggregate functions are used to perform calculations on multiple rows of a single column of a table and returns single value as output.  
 Aggregate functions are count, sum, avg, min, max.  
   
 Aggregate functions:  
 a. [aggregate-functions](https://www.geeksforgeeks.org/database-management-system-aggregate-functions/)  
 b. [sql\_count\_avg\_sum](https://www.w3schools.com/sql/sql_count_avg_sum.asp)  
 c. [sql\_min\_max](https://www.w3schools.com/sql/sql_min_max.asp)  
 d. [sql\_alias](https://www.w3schools.com/sql/sql_alias.asp)  
   
 **a. COUNT ():**  
 It returns the number of rows that satisfies the condition.   
   
 **e.g.** select count (\*) from employee //it returns the number of rows in a table  
   
 count_star  
   
 select count (\*) as Age\_equals\_18 from employee where age = 18;  
   
 count_star_condition //resultant column name renamed as Age\_equals\_18  
   
 **b. AVG ():**   
 It returns the average value of a numeric column.   
   
 **e.g.**   
 select avg(age) from employee;   
   
 avg  
   
 select avg(age) as average\_of\_age\_18 from employee where age = 18;  
   
 avg_condition  
   
 **c. SUM ():**  
 It returns the sum of numeric value.  
   
 **Eg.** select sum(age) as sum\_of\_age from employee  
   
 sum_query  
   
 **d. MAX ():**  
 It returns the maximum value of the selected column.  
   
 select max(salary) from employee  
   
 max_query  
   
 **Que:** find employee's detail having maximum salary:  
   
 **Nested query:** query inside another query.  
   
 SELECT \* FROM EMPLOYEE WHERE SALARY = (SELECT MAX(SALARY) FROM EMPLOYEE);  
   
 max_salary  
   
 **e. MIN ():**  
 It returns the minimum value of the selected column;  
   
 **e.g.** select min(salary) from employee;  
   
 10. LIKE:  
 It is used to search a particular pattern in a table. It is used with where clause. It uses some wildcard characters like %, \_ , $, \* etc.  
   
 Wild Card characters: [sql\_wildcard\_characters](https://www.w3schools.com/sql/sql_wildcards.asp)  
   
 **e.g.** SELECT \* FROM EMPLOYEE WHERE NAME LIKE '%AN%';  
   
 //it selects all the name having 'an' as substring.  
   
   
   
 Like operator: [sql\_like](https://www.w3schools.com/sql/sql_like.asp)  
   
 11. IN:  
 In operator is a shorthand for multiple 'or' conditions.  
   
 **e.g.**   
 with multiple OR conditions: SELECT \* FROM EMPLOYEE WHERE AGE=18 OR AGE= 20;  
 Now shorthand query for above query is: SELECT \* FROM EMPLOYEE WHERE AGE IN (18,20);  
   
 IN operator: [sql\_in](https://www.w3schools.com/sql/sql_in.asp)  
   
 12. BETWEEN:  
 It selects values in the given range.   
   
 **e.g.** SELECT \* FROM EMPLOYEE WHERE AGE BETWEEN 18 AND 20; //selects all employees with age between 18 and 20  
   
 between and not between: [sql\_between\_and\_not\_between](https://www.w3schools.com/sql/sql_between.asp)  
   
 13. ALIASES:  
 It is used to give temporary name to table or column using 'as' keyword.  
   
 **e.g.** SELECT NAME AS EMPLOYEE\_NAME FROM EMPLOYEE;  
   
 Aliases:   
 [sql\_alias\_w3school](https://www.w3schools.com/sql/sql_alias.asp)  
 [sql\_alias\_geeks](https://www.geeksforgeeks.org/sql-aliases/)  
   
 14. JOINS:   
 Joins: [sql\_join](https://www.w3schools.com/sql/sql_join.asp)  
   
 Inner, left, right, full join: [inner-left-right-and-full-joins](https://www.geeksforgeeks.org/sql-join-set-1-inner-left-right-and-full-joins/)  
   
 **a. Inner Join:** [sql\_join\_inner](https://www.w3schools.com/sql/sql_join_inner.asp)  
   
 Video on Join or Inner Join: [Join or Inner Join](https://www.youtube.com/watch?v=lJv614Y0lFA&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=102)  
   
 Join or Inner Join with 'ON' operation: [Join or Inner Join with 'On' operation](https://www.youtube.com/watch?v=z996Uv9MNaQ&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=103)  
   
 **b. Left Join:**   
 It returns all records from the left relation and matched records from the right relation and if there is no match then the result is null from the right side.  
   
    
   
 [sql\_join\_left](https://www.w3schools.com/sql/sql_join_left.asp)  
   
 **c. Right join:**  
 It returns all records from the right table and matched records from the left table and if there is no match then the result is null from the left side.  
   
    
   
 [sql\_join\_right](https://www.w3schools.com/sql/sql_join_right.asp)  
   
 **d. Full join:**   
 It returns all matching records from both tables whether the other matches or not.  
   
    
   
 [sql\_join\_full](https://www.w3schools.com/sql/sql_join_full.asp)  
   
 Video on Full Outer Join: [Full Outer Join](https://www.youtube.com/watch?v=ldos7yYMjs8&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=104)  
   
 **e. Self-join:**   
 [sql\_join\_self](https://www.w3schools.com/sql/sql_join_self.asp)  
 [sql-join-cartesian-join-self-join](https://www.geeksforgeeks.org/sql-join-cartesian-join-self-join/)  
   
 **f. Natural Join:**  
 Video on Natural Join [Natural Join](https://www.youtube.com/watch?v=PHPHkqHAu9Y&list=PLmXKhU9FNesR1rSES7oLdJaNFgmuj0SYV&index=101)  
 A Natural Join is a join operation in which 2 tables are joined based on all common columns.   
   
    
   
 15. UNION:  
 [sql\_union](https://www.w3schools.com/sql/sql_union.asp)  
 [sql-union-clause](https://www.geeksforgeeks.org/sql-union-clause/)  
   
 16. GROUP BY:   
 It is used to group the tuples that have same value with the help of some functions.   
   
 **e.g.** SELECT AGE FROM EMPLOYEE GROUP BY AGE;  
   
 [sql-group-by](https://www.geeksforgeeks.org/sql-group-by/)  
   
 17. HAVING:  
 [sql\_having](https://www.w3schools.com/sql/sql_having.asp)  
   
 18. DROP DATABASE:  
 It is used to drop an existing database.   
   
 **Syntax:** DROP DATABASE DATABASE\_NAME;  
   
 **e.g.** drop database demo;  
   
 Drop database: [sql\_drop\_db](https://www.w3schools.com/sql/sql_drop_db.asp)  
   
 19. DROP TABLE:  
 It is used to drop an existing table.   
   
 **Syntax:** DROP TABLE TABLE\_NAME;  
   
 **e.g.** drop table employee;  
   
 drop table: [sql\_drop\_table](https://www.w3schools.com/sql/sql_drop_table.asp)  
   
 20. TRUNCATE TABLE:  
 It is used to delete all the data inside the table but not the structure of table.   
   
 **Syntax:** TRUNCATE TABLE TABLE\_NAME;  
   
 **e.g.** truncate table employee;  
   
 truncate table: [sql\_drop\_table](https://www.w3schools.com/sql/sql_drop_table.asp)  
   
 21. ALTER TABLE:   
 It is used to add, delete and modify an existing column.  
   
 **Add column** in an existing table:   
 **Syntax:** ALTER TABLE TABLE\_NAME ADD COLUMN DATATYPE;  
 **e.g.** alter table employee add column department varchar (20) not null;  
   
 **modify column** in an existing table:  
 **syntax:** ALTER TABLE TABLE\_NAME MODIFY COLUMN COLUMN\_NAME DATATYPE;  
   
 **drop column** in an existing table:  
 **syntax:** ALTER TABLE TABLE\_NAME DROP COLUMN COLUMN\_NAME;  
   
 alter table: [sql\_alter](https://www.w3schools.com/sql/sql_alter.asp)  
   
 22. Constraints: [sql\_constraints](https://www.w3schools.com/sql/sql_constraints.asp)  
   
 23. View: [sql\_view](https://www.w3schools.com/sql/sql_view.asp)  
   
 24. Stored Procedures: [sql\_stored\_procedures](https://www.w3schools.com/sql/sql_stored_procedures.asp)  
   
 25. Sql Injection: [sql\_injection](https://www.w3schools.com/sql/sql_injection.asp)  
   
 26. MySQL Functions: [sql\_ref\_mysql](https://www.w3schools.com/sql/sql_ref_mysql.asp)  
   
   
   
 REFERENCES:  
 SQL Tutorial (w3school): [sql-tutorial\_w3school](https://www.w3schools.com/sql/default.asp)  
   
 SQL Tutorial (geeksforgeek): [sql-tutorial\_geeks](https://www.geeksforgeeks.org/sql-tutorial/)

# Sql Queries For Interview

[sql-interview-questions-answers](https://www.guru99.com/sql-interview-questions-answers.html)  
   
 [sql-queries-for-interview](https://artoftesting.com/interviewSection/sql-queries-for-interview.html)  
   
 [50-popular-sql-interview-questions](https://www.softwaretestinghelp.com/50-popular-sql-interview-questions-for-testers/)  
   
 [sql-interview-questions](https://www.geeksforgeeks.org/sql-interview-questions/)  
   
 [sql-query-questions-answers-for-practice](https://www.techbeamers.com/sql-query-questions-answers-for-practice/)