Chapter 2 : Database

Data and information: Data is defined as the raw facts and figures. It could be any numbers, alphabets or any combination of it.

When data are processed using a database program or software, they are converted to the meaningful result, called information.

Data dictionary: A data dictionary is a file which contains meta-data that is data about data. It also called information system catalogue. It keeps all the data information about the database system such as location, size of the database, tables, records, fields, user information, recovery system, etc.

Field and record: A column in a table is called field and it contains specific piece of information within a record.

A row in a table is called record and it contains information about person, event, etc. Another name of record is tuple.

Domains and Tuples

In data management and database analysis, a data domain refers to all the values which a data element may contain. The rule for determining the domain boundary may be as simple as a data type with an enumerated list of values.

Tuple is the collection of information about the attributes of table for single instance. In simple this also can be called as a 'row' in a Table.

Relational Database Management System (RDBMS)

The database system which stores and displays data in tabular format of rows and columns, like spreadsheet, is known as Relational Database Management System.

Different between Database and DBMS

Database: It is a collection of related information about a subject organized in a useful manner that provides a base or foundation for procedure, such as retrieving information, drawing conclusion and make decision.

DBMS: A DBMS is a set of programs that manages the database files. It allows accessing the files, updating the records and retrieving data as requested.

The technique for designing a database using top-down methods is to write a main database parts that names all the major storage and retrievals it will need. Later, the programming team looks at the requirements of each of those database components and the process is repeated.

The top-down method starts from the general and moves to the specific. Basically, you start with a general idea of what is needed for the system and then ask the end-users what data they need to store. The analyst will then work with the users to determine what data should be kept in the database. Using the top-down method requires that the analyst has a detailed understanding of the system. The top-down method also can have shortcomings. In some cases, top-down design can lead to unsatisfactory results because the analyst and end-users can miss something that is important and is necessary for the system.

Advantages of the database being centralized

1. easier to organize, edit, update and back-up the data
2. less data duplication - data is only entered once but can be accessed by many users
3. data integrity - because data is stored once different data is no longer held in different databases in various departments around the organization

Advantages of database

1. Sharing data
2. Reduced data redundancy
3. Data backup and recovery
4. Inconsistency avoided
5. Data integrity
6. Data security
7. Data independence
8. Multiple user interfaces
9. Process complex query

Different between centralize and distributed database system

Centralized database system	**Distributed database system**
Simple type	Complex type
Located on particular location	Located in many geographical locations.
Consists of only one server	Contains servers in several locations
Suitable for small organizations	Suitable for large organizations
Less chance of data lost	More chances of data hacking, lost
Maintenance is easy and security is high	Maintenance is not easy and security is low
Failure of system makes whole system down	Failure of one server does not make the whole system down
There is no feature of load balancing	There is feature of load balancing
Data traffic rate is high	Data traffic rate is low
Cost of centralized database system is low	Cost of distributed database system is high

Different database models

1. **Hierarchical database model:** this is one of the oldest type of database models. In this model data is represented in the form of records. Each record has multiple fields. All records are arranged in database as tree like structure. The relationship between the records is called parent child relationship in which any child record relates to only a single parent type record.
2. **Network database model:** it replaced hierarchical network database model due to some limitations on the model. Suppose, if an employee relates to two departments, then the hierarchical database model cannot able to arrange records in proper place. So network, database model was emerged to arranged non-hierarchical database. The structure of database is more like graph rather than tree structure.
3. **Relational database model**: in this model, the data is organized into tables which contain multiple rows ad columns. These tables are called relations. A row in a table represents a relationship among a set of values. Since a table is a collection of such relationships, it is generally referred to the mathematical term relation, from which the relational database model derives its name.
4. **Entity-Relationship database model**: this model is based on perception of a real world that contains a collection of basic objects, called entities and of relationship among these objects and characteristics of an entity. It shows relationship between different entities.

Key: A key allows us to identify a set of attributes that is distinguishing entities from each other.

Types:

1. **Super key**: It is a set of one or more attributes that allows identifying an entity uniquely. For example. Std-id attribute of an entity set student which is sufficient to distinguish one student entity from another. Thus, Std-id is a super key. Similarly {roll-no, name, class –section} is also a super key. Super key may contain extra attributes.
2. **Candidate key**: It is the minimal set of super key which can uniquely identify an entity .For examples, Std-id is a candidate key.
3. **Primary key**: This key is chosen by database administrator for identifying entity. The primary key can also be defined as the successful candidate key, if there is one candidate key then obviously the same will be declared as primary keys.
4. **Foreign key**: If the primary key of one entity set is used in another entity set for establishing the relationship among the entity set is known as foreign key.

Reference note on Database Model

A Database model defines the logical design and structure of a database and defines how data will be stored, accessed and updated in a database management system. While the **Relational Model** is the most widely used database model, there are other models too:

1. Hierarchical Model
2. Network Model
3. Entity-relationship Model
4. Relational Model

1. Hierarchical Model

This database model organizes data into a tree-like-structure, with a single root, to which all the other data is linked. The hierarchy starts from the **Root** data, and expands like a tree, adding child nodes to the parent nodes.

In this model, a child node will only have a single parent node.

This model efficiently describes many real-world relationships like index of a book, recipes etc.

In hierarchical model, data is organized into tree-like structure with one one-to-many relationship between two different types of data, for example, one department can have many courses, many professors and of-course many students.

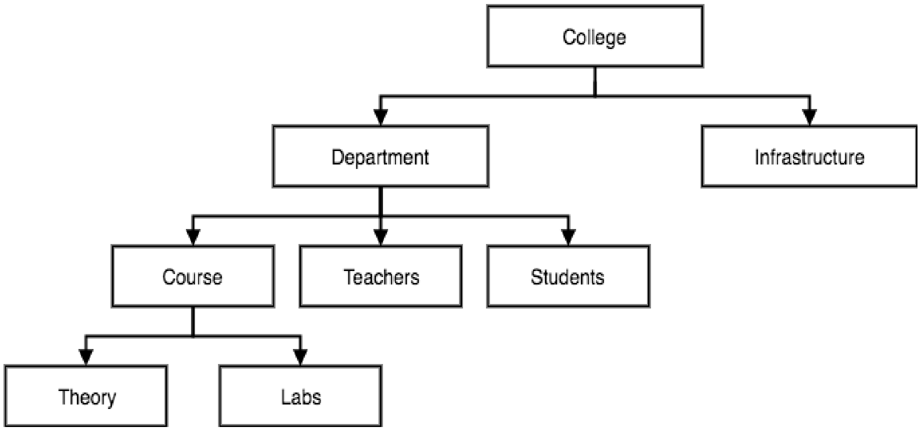


Fig. Hierarchical Model

2. Network Model

This is an extension of the Hierarchical model. In this model data is organised more like a graph, and are allowed to have more than one parent node.

In this database model data is more related as more relationships are established in this database model. Also, as the data is more related, hence accessing the data is also easier and fast. This database model was used to map many-to-many data relationships.

This was the most widely used database model, before Relational Model was introduced.

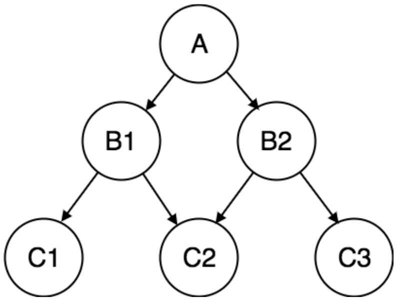


Fig. Network Model

3. Entity-Relationship Model

In this database model relationship are carried by dividing object of inserts into entity and it characteristics into attributes. Different entities are related using relationship.

ER model are a high level data model based on a perceptions of a real world that consist of collection of basic object called **entities**. An **entity** is a thing or object in the real world that is distinguishable from other object. Entities are describe in a database by a set of attribute .a relationship is an association among several entities. The set of all entities of the same type is called **an entity set**. Overall logical structure of a database can be expressed graphically by ER diagram. The basic components of this diagram are:

- Rectangle (represent entity sets).
- Ellipse (represent attributes).
- Diamonds (represent relationship set among entity sets).
- Lines (link attributes to entity sets and entity sets to relationship sets).

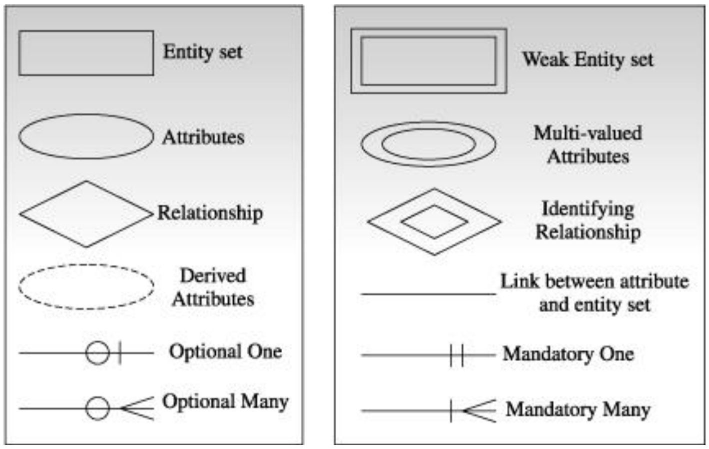


Fig. Symbol used in ER- diagram

E-R Models are defined to represent the relationships into pictorial form to make it easier for different stakeholders to understand.

This model is good to design a database, which can then be turned into tables in relational model (explained below).

Let's take an example, If we have to design a School Database, then **Student** will be an **entity** with **attributes** name, age, address etc. As **Address** is generally complex, it can be another **entity** with **attributes** street name, pin code, city etc, and there will be a relationship between them.

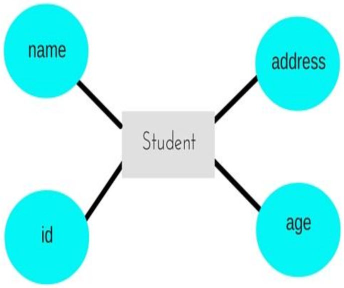
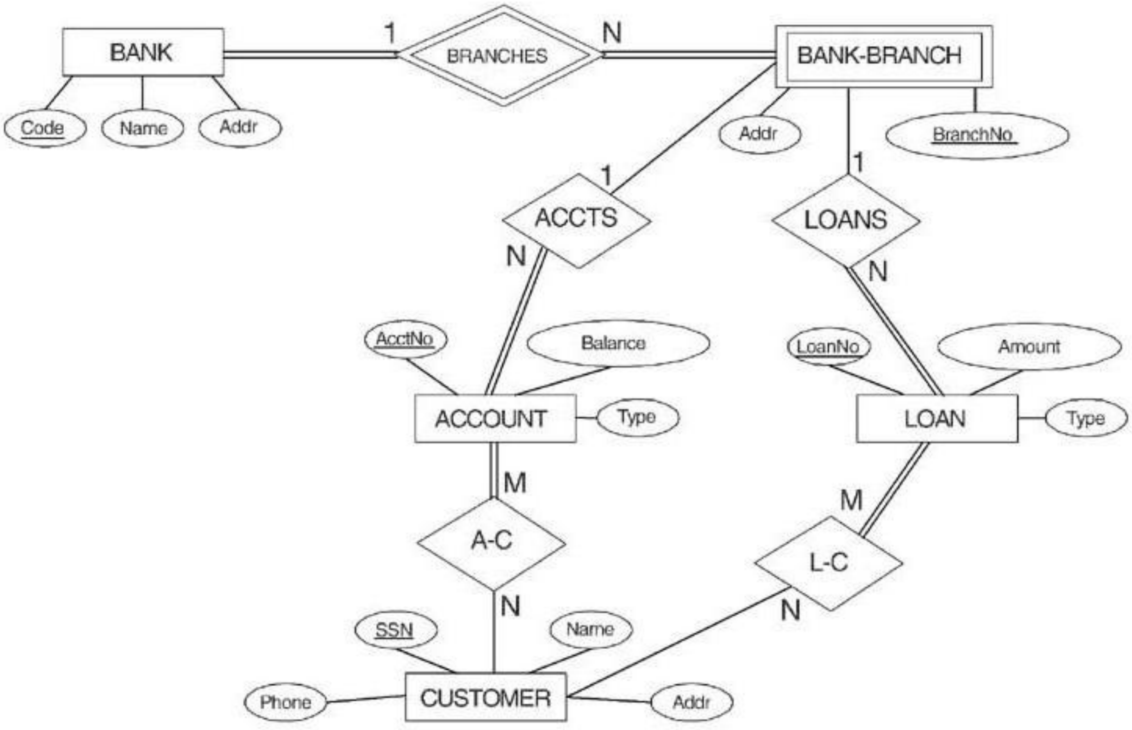


Fig. ER model of Student

Example of ER Diagram for Bank:-



4. Relational Model

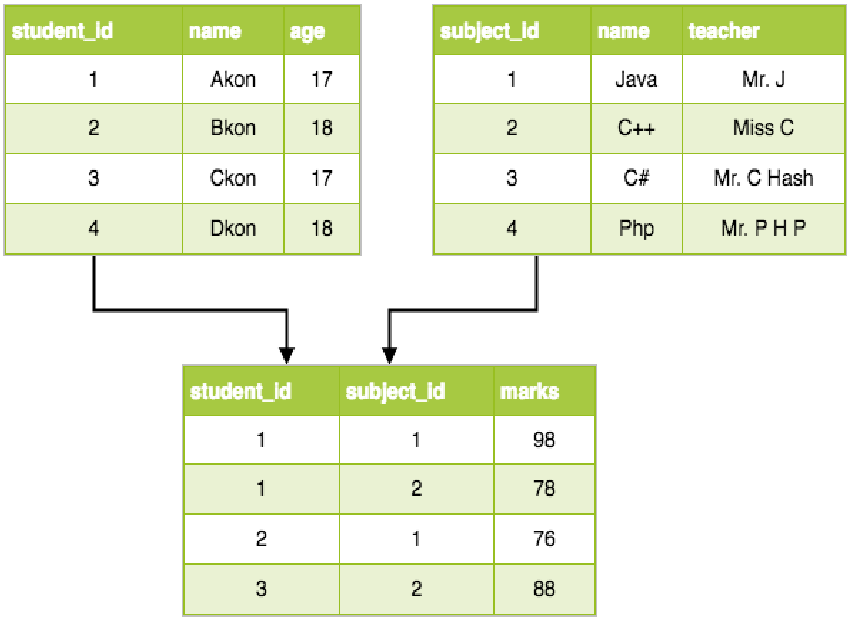
In this model, data is organized in two-dimensional **tables** and the relationship is maintained by storing a common field.

This model was introduced by E.F Codd in 1970, and since then it has been the most widely used database model, in fact, we can say the only database model used around the world.

The basic structure of data in the relational model is tables. All the information related to a particular type is stored in rows of that table.

Hence, tables are also known as **relations** in relational model.

In the coming tutorials we will learn how to design tables, normalize them to reduce data redundancy and how to use Structured Query language to access data from tables.



student_id	name	age
1	Akon	17
2	Bkon	18
3	Ckon	17
4	Dkon	18

subject_id	name	teacher
1	Java	Mr. J
2	C++	Miss C
3	C#	Mr. C Hash
4	Php	Mr. P H P

student_id	subject_id	marks
1	1	98
1	2	78
2	1	76
3	2	88

Fig. Relational Model

DBA and responsibilities of DBA

DBA is the most responsible person in an organization with sound knowledge of DBMS. He/she is the overall administrator of the system. He/she has the maximum amount of privileges (permission to access the database) for accessing the database, settings up system and defining the role of the employees which use the system.

Responsibilities of DBA:

1. DBA defines data security, schemas, forms, reports, relationships and user privileges.
2. DBA has responsibility to install. Monitor and upgrade database server.
3. DBA provides different facilities for data retrieving and making reports as required.

4. DBA has responsibility to maintain database security, backup-recovery strategy, and documentation of data recovery.
5. DBA supervises all the activities in the system: addition, modification and deletion data from the database.

State and different types of data integrity

Mainly there are 3 types of data integrity constraints used in the database system. They are as:

1. Domain integrity constraints: it defines a set range of data values for given specific data field. And also determines whether null values are allowed or not in the data field.
2. Entity integrity constraints: it specify that all rows in a table have a unique identifier, known as the primary key value and it never be null i.e. blank.
3. Referential integrity constrains: it exists in a relationship between the two tables in a database. It ensures that the relationship between the primary keys in the master table and foreign key in child table are always maintained.

Normalization and normalization process

Normalization is a database design process in which complex database table is broken down into simple separate tables. It makes data model more flexible and easier to maintain. There are two goals of the normalization process: eliminating redundant data and ensuring data dependencies make sense.

For example: the table shown is our database without normalized. Here in table we can see that for the large records of this table, there would be multiple data row of same values especially in the country and city column. So, we can normalize the table by splitting it into two tables where one table only stores the location area of each person name and could be referenced by some unique id. Say Area code.

id	country	city	Name
15	Nepal	Kathmandu	Alex
16	India	Delhi	Martin
17	Nepal	Kathmandu	Melman
18	Japan	Tokyo	Gloria

The above table can be normalized in two tables as below:

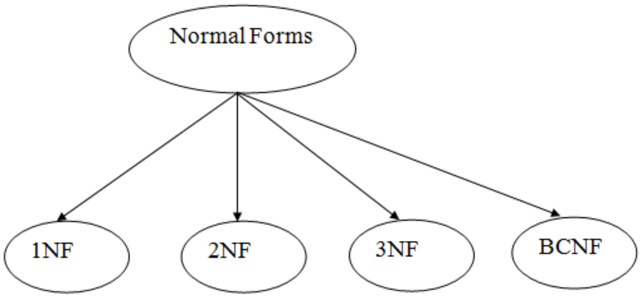
country	city	Area code
Nepal	Kathmandu	N1
India	Delhi	I1
Japan	Tokyo	J1

Id	Area code	Name
15	N1	Alex

16	I1	Martin
17	N1	Melman
18	J1	Gloria

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Types of Normal Form:-



Normal Form	Description
1NF	A relation is in 1NF if it contains an atomic value.
2NF	A relation will be in 2NF if it is in 1NF and all non-key attributes are fully functional dependent on the primary key.
3NF	A relation will be in 3NF if it is in 2NF and no transition dependency exists.
4NF	A relation will be in 4NF if it is in Boyce Codd normal form and has no multi-valued dependency.

5NF	A relation is in 5NF if it is in 4NF and not contains any join dependency and joining should be lossless.

1. First Normal Form (1NF)

- A relation will be 1NF if it contains an atomic value.
- It states that an attribute of a table cannot hold multiple values. It must hold only single-valued attribute.
- First normal form disallows the multi-valued attribute, composite attribute, and their combinations.

Example: Relation EMPLOYEE is not in 1NF because of multi-valued attribute EMP_PHONE.

EMPLOYEE table:

EMP_ID	EMP_NAME	EMP_PHONE	EMP_STATE
14	John	7272826385, 9064738238	UP
20	Harry	8574783832	Bihar
12	Sam	7390372389, 8589830302	Punjab

The decomposition of the EMPLOYEE table into 1NF has been shown below:

EMP_ID	EMP_NAME	EMP_PHONE	EMP_STATE
14	John	7272826385	UP
14	John	9064738238	UP
20	Harry	8574783832	Bihar
12	Sam	7390372389	Punjab
12	Sam	8589830302	Punjab

2. Second Normal Form (2NF)

- In the 2NF, relational must be in 1NF.
- In the second normal form, all non-key attributes are fully functional dependent on the primary key

Example: Let's assume, a school can store the data of teachers and the subjects they teach. In a school, a teacher can teach more than one subject.

TEACHER table

TEACHER_ID	SUBJECT	TEACHER_AGE
25	Chemistry	30
25	Biology	30
47	English	35
83	Math	38
83	Computer	38

In the given table, non-prime attribute TEACHER_AGE is dependent on TEACHER_ID which is a proper subset of a candidate key. That's why it violates the rule for 2NF.

To convert the given table into 2NF, we decompose it into two tables:

TEACHER_DETAIL table:

TEACHER_ID	TEACHER_AGE
25	30
47	35
83	38

TEACHER_SUBJECT table:

TEACHER_ID	SUBJECT

25	Chemistry
25	Biology
47	English
83	Math
83	Computer

3. Third Normal Form (3NF)

- A relation will be in 3NF if it is in 2NF and not contain any transitive partial dependency.
- 3NF is used to reduce the data duplication. It is also used to achieve the data integrity.
- If there is no transitive dependency for non-prime attributes, then the relation must be in third normal form.

A relation is in third normal form if it holds at least one of the following conditions for every non-trivial function dependency X → Y.

1. X is a super key.
2. Y is a prime attribute, i.e., each element of Y is part of some candidate key.

Example:

EMPLOYEE_DETAIL table:

EMP_ID	EMP_NAME	EMP_ZIP	EMP_STATE	EMP_CITY
222	Harry	201010	UP	Noida
333	Stephan	02228	US	Boston
444	Lan	60007	US	Chicago
555	Katharine	06389	UK	Norwich
666	John	462007	MP	Bhopal

Super key in the table above:

1. {EMP_ID}, {EMP_ID, EMP_NAME}, {EMP_ID, EMP_NAME, EMP_ZIP}....so on

Candidate key: {EMP_ID}

Non-prime attributes: In the given table, all attributes except EMP_ID are non-prime.

Here, EMP_STATE & EMP_CITY dependent on EMP_ZIP and EMP_ZIP dependent on EMP_ID. The non-prime attributes (EMP_STATE, EMP_CITY) transitively dependent on super key(EMP_ID). It violates the rule of third normal form.

That's why we need to move the EMP_CITY and EMP_STATE to the new <EMPLOYEE_ZIP> table, with EMP_ZIP as a Primary key.

EMPLOYEE table:

EMP_ID	EMP_NAME	EMP_ZIP
222	Harry	201010
333	Stephan	02228
444	Lan	60007
555	Katharine	06389
666	John	462007

EMPLOYEE_ZIP table:

EMP_ZIP	EMP_STATE	EMP_CITY
201010	UP	Noida
02228	US	Boston
60007	US	Chicago
06389	UK	Norwich
462007	MP	Bhopal

Data type in Microsoft access: In Ms-access, data type determines the kind of data that can be entered into the field. Ms –access uses the data type to ensure that right kind of data is entered in a field. The data types used in ms-access are explained below:

Data type	Description
Text	Short, alphanumeric values, such as a last name or a street address. Note, beginning in Access 2013, Text data types have been renamed to **Short Text**.
Number, Large Number	Numeric values, such as distances. Note that there is a separate data type for currency.
Currency	Monetary values.
Yes/No	Yes and No values and fields that contain only one of two values.
Date/Time	Date and Time values for the years 100 through 9999.
Rich Text	Text or combinations of text and numbers that can be formatted using color and font controls.
Calculated Field	Results of a calculation. The calculation must refer to other fields in the same table. You would use the Expression Builder to create the calculation. Note, Calculated fields were first introduced in Access 2010.
Attachment	Attached images, spreadsheet files, documents, charts, and other types of supported files to the records in your database, similar to attaching files to e-mail messages.
Hyperlink	Text or combinations of text and numbers stored as text and used as a hyperlink address.
Memo	Long blocks of text. A typical use of a Memo field would be a detailed product description. Note, beginning in Access 2013, Memo data types have been renamed to Long Text

DDL, DML, DCL & TCL commands in SQL

- DDL is Data Definition Language
- DML is Data Manipulation Language
- DCL is Data Control Language
- TCL is Transaction Control Language

SQL commands list:

Language	**Command List**
DDL	CREATE DROP ALTER RENAME TRUNCATE

DML	SELECT
	INSERT
	UPDATE
	DELETE

DCL	GRANT REVOKE
TCL	START TRANSACTION COMMIT ROLLBACK

1. DDL(Data Definition Language):

DDL is used by the database designers and programmers to specify the content and structure of the table. It is used to define the physical characteristics of records. It includes commands that manipulate the structure of objects such as views, tables, and indexes, etc.

DDL allows you to create SQL statements to make operations with database data structures (schemas, tables etc.).These are SQL DDL commands list and examples:

- **CREATE**

CREATE statement is used to create a new database, table, index or stored procedure.

->**Create database** example: CREATE DATABASE explainjava;

->**Create table** example:

```
CREATE TABLE user (
  id INT(16) PRIMARY KEY AUTO_INCREMENT,
  name VARCHAR(255) NOT NULL
);
```

- **DROP**

DROP statement allows you to remove database, table, index or stored procedure.

Drop database example: DROP DATABASE explainjava;

Drop table example: DROP TABLE user;

- **ALTER**

ALTER is used to modify existing database data structures (database, table).

Alter table example: ALTER TABLE user ADD COLUMN lastname VARCHAR(255) NOT NULL;

- **RENAME**

RENAME command is used to rename SQL table.

Rename table example: RENAME TABLE user TO student;

- **TRUNCATE**

TRUNCATE operation is used to delete all table records.

Logically it’s the same as DELETE command.

Differences between DELETE and TRUNCATE commands are:

- TRUNCATE is really faster
- TRUNCATE cannot be rolled back
- TRUNCATE command does not invoke ON DELETE triggers

Example: TRUNCATE student;

2. DML (Data Manipulation Language):

DML is related with manipulation of records such as retrieval, sorting, display and deletion of records of data. It helps user to use query and display reports of the table. So it provides technique for processing the database.

DML is a Data Manipulation Language, it’s used to build SQL queries to manipulate (select, insert, update, delete etc.) data in the database.

This is DML commands list with examples:

- **SELECT**

SELECT query is used to retrieve a data from SQL tables.

Example: SELECT * FROM student;

- **INSERT**

INSERT command is used to add new rows into the database table.

Example: INSERT INTO student (name, lastname) VALUES ('Dmytro', 'Shvechikov');

- **UPDATE**

UPDATE statement modifies records into the table.

Example: UPDATE student SET name = 'Dima' WHERE lastname = 'Shvechikov';

- **DELETE**

DELETE query removes entries from the table.

Example: DELETE FROM student WHERE name = 'Dima';

3. DCL(Data Control Language):

It stands for Data Control Language. These commands are responsible for access restrictions inside of the database.

Let’s take a look at DCL statements definitions.

- **GRANT**

GRANT command gives permissions to SQL user account.

For example, I want to grant all privileges to ‘explainjava’ database for user ‘dmytro@localhost’.

CREATE USER 'dmytro'@'localhost' IDENTIFIED BY '123';

Then I can grant all privileges using GRANT statement:

GRANT ALL PRIVILEGES ON explainjava.* TO 'dmytro'@'localhost';

and we have to save changes using FLUSH command: `FLUSH PRIVILEGES;`

- **REVOKE**

REVOKE statement is used to remove privileges from user accounts.

Example: REVOKE ALL PRIVILEGES ON explainjava.* FROM 'dmytro'@'localhost';

and save changes: FLUSH PRIVILEGES;

4. TCL(Transaction Control Language) :

TCL is a Transaction Control Language.

These commands are used to manage transactions in SQL databases.

Relationship: A relationship is an association among several entities and represents meaningful dependencies between them. It is represented by diamond. There are 3 types of relationship:

i. One to one

ii. One to many

iii. Many to one

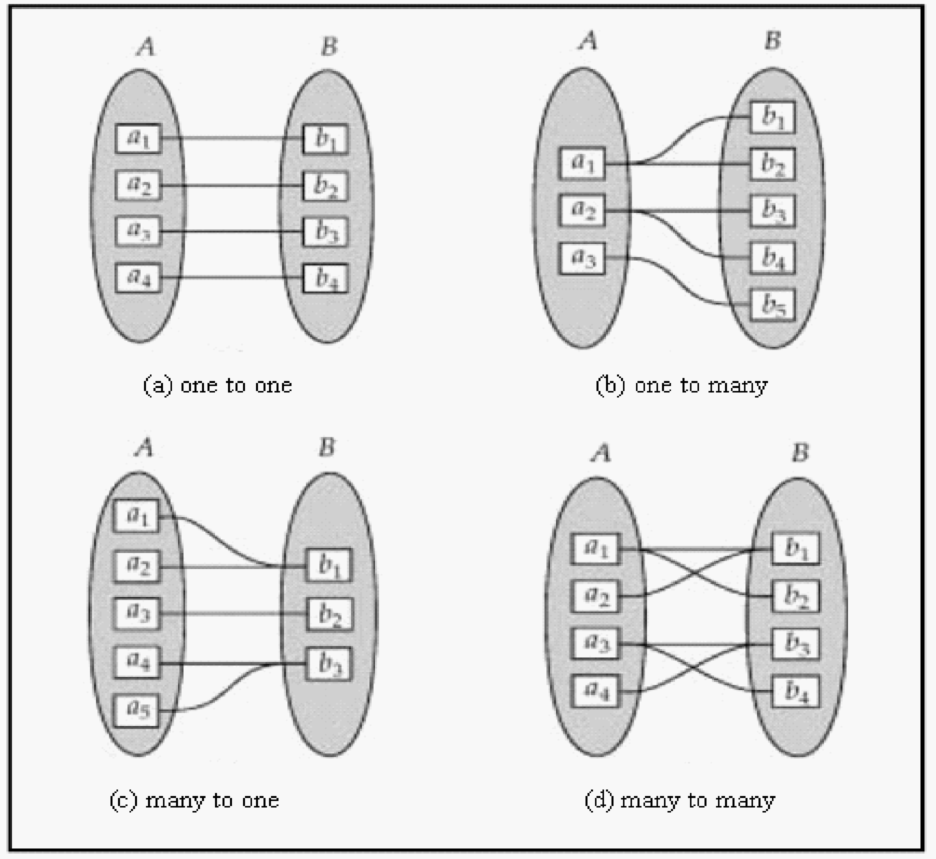
iv. Many to Many

i. **One to one**: An entity in A is associated with at most one entity in B and entity in B is associated with at most one entity in A.

ii. **One to many:** An entity in A is associated with zero or more entities in B but entity in B can be associated with at most one entity in A.

iii. **Many to one:** An entity in A is associated with at most one entity in B but an entity in B can be associated with zero or more entities in A.

iv. **Many to many:** An entity in A is associated with zero or more entities in B, and an entity in b is associated with zero or more entities in A.



Database security

The role of database security is to prevent unauthorized or accidental access to data. As the database environment has become more complex and more decentralized, management of data security and integrity has become more complex and time consuming job for data administrators.

Database security looks at the various threats to the database. It is the way of protecting the database from these and ways of controlling data access.

Threats to the database can be numerous. The threats can be accidental or accidental. In other case, security of the database and the entire system including the network, operating system , physical area where the database exist and the personal allows access must be consider.

The following threats must be addressed;

- Accidental losses
- Fraud and Theft

- Loss of privacy and confidentiality
- Loss of data integrity
- Loss of data availability

Database design must consider a security strategy address potential threat. Generally, a data security includes physical method of protection and use of data management software. Controlling the outside access and threats to the database is another form of security. These methods are;

- Backup and recovery
- Virus checking
- Authorization rules
- Firewalls
- Physical protection