Chap3:Relational language and Relational Model

Relational Model

- represents the database as a collection of relations. A relation is nothing but a table of values. Every row in the table represents a collection of related data values. These rows in the table denote a real-world entity or relationship.
- The table name and column names are helpful to interpret the meaning of values in each row. The data are represented as a set of relations.
- In the relational model, data are stored as tables. However, the physical storage of the data is independent of the way the data are logically organized.
- Some popular Relational Database management systems are:

  -DB2 and Informix Dynamic Server - IBM

  -Oracle and RDB – Oracle

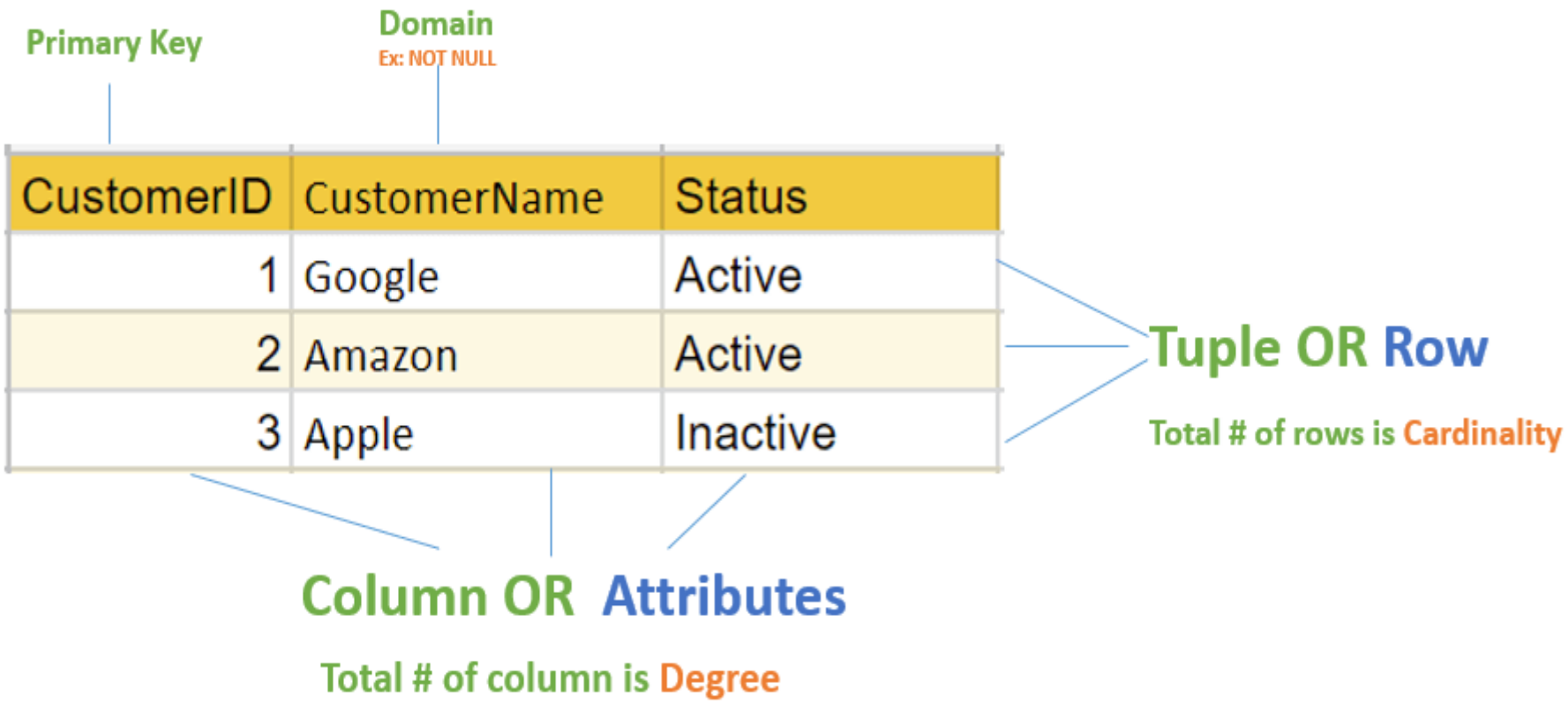
  -SQL Server and Access - Microsoft

Relational Model Concepts

- **Attribute:** Each column in a Table. Attributes are the properties which define a relation. e.g., SID, Name, etc.
- **Tables** – In the Relational model the, relations are saved in the table format. It is stored along with its entities. A table has two properties rows and columns. Rows represent records and columns represent attributes.
- **Tuple** – It is nothing but a single row of a table, which contains a single record.
- **Relation Schema:** A relation schema represents the name of the relation with its attributes.

- **Degree:** The total number of attributes within the relation is called the degree of the relation.
- **Cardinality:** Total number of rows present in the Table.
- **Column:** The column represents the set of values for a specific attribute.
- **Relation instance** – Relation instance is a finite set of tuples in the RDBMS system. Relation instances never have duplicate tuples.
- **Relation key** - Every row has one, two or multiple attributes, which is called relation key.
- **Attribute domain** – Every attribute has some pre-defined value and scope which is known as attribute domain

Table also called Relation



Keys in Relational Model

- Keys are the attributes that helps to distinguish the tuple in a relation.

Types:

1. Super key
2. Candidate key
3. Primary key
4. Alternate Key
5. Foreign Key
6. Composite Key

Example

STUDENT

STUD_NO	STUD_NAME	STUD_PHONE	STUD_STATE	STUD_COUNTRY	STUD_AGE
1	RAM	9716271721	Haryana	India	20
2	RAM	9898291281	Punjab	India	19
3	SUJIT	7898291981	Rajsthan	India	18
4	SURESH		Punjab	India	21

Table 1

STUDENT_COURSE

STUD_NO	COURSE_NO	COURSE_NAME
1	C1	DBMS
2	C2	Computer Networks
1	C2	Computer Networks

Table 2

Candidate Key: The minimal set of attribute which can uniquely identify a tuple is known as candidate key. For Example, STUD_NO in STUDENT relation.

- The value of Candidate Key is unique and non-null for every tuple.
- There can be more than one candidate key in a relation. For Example, STUD_NO is candidate key for relation STUDENT.
- The candidate key can be simple (having only one attribute) or composite as well. For Example, {STUD_NO, COURSE_NO} is a composite candidate key for relation STUDENT_COURSE.
- **Note –** In Sql Server a unique constraint that has a nullable column, **allows** the value '**null**' in that column **only once**. That's why STUD_PHONE attribute as candidate here, but can not be 'null' values in primary key attribute.

- **Super Key:** The set of attributes which can uniquely identify a tuple is known as Super Key. For Example, STUD_NO, (STUD_NO, STUD_NAME) etc.
- Adding zero or more attributes to candidate key generates super key.
- A candidate key is a super key but vice versa is not true.

Primary Key: There can be more than one candidate key in relation out of which one can be chosen as the primary key. For Example, STUD_NO, as well as STUD_PHONE both, are candidate keys for relation STUDENT but STUD_NO can be chosen as the primary key (only one out of many candidate keys).

Alternate Key: The candidate key other than the primary key is called an alternate key.

For Example, STUD_NO, as well as STUD_PHONE both, are candidate keys for relation STUDENT but STUD_PHONE will be alternate key (only one out of many candidate keys).

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

Contd..

- The referenced attribute of the referenced relation should be the primary key for it. For Example, STUD_NO in STUDENT_COURSE is a foreign key to STUD_NO in STUDENT relation.
- It may be worth noting that unlike, Primary Key of any given relation, Foreign Key can be NULL as well as may contain duplicate tuples i.e. it need not follow uniqueness constraint.
- For Example, STUD_NO in STUDENT_COURSE relation is not unique. It has been repeated for the first and third tuple. However, the STUD_NO in STUDENT relation is a primary key and it needs to be always unique and it cannot be null.

Composite key: A key that has more than one attributes is known as composite key. It is also known as compound key.

Note: Any key such as super key, primary key, candidate key etc. can be called composite key if it has more than one attributes.

Operations in Relational Model

Four basic update operations performed on relational database model are: Insert, update, delete and select.

- Insert is used to insert data into the relation
- Delete is used to delete tuples from the table.
- Modify allows you to change the values of some attributes in existing tuples.
- Select allows you to choose a specific range of data.
- Whenever one of these operations are applied, integrity constraints specified on the relational database schema must never be violated.

- **Insert Operation:**The insert operation gives values of the attribute for a new tuple which should be inserted into a relation.

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive
4	Alibaba	Active

- **Update Operation: I**n the below-given relation table CustomerName= 'Apple' is updated from Inactive to Active.

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive
4	Alibaba	Active

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Active
4	Alibaba	Active

- **Delete Operation:** To specify deletion, a condition on the attributes of the relation selects the tuple to be deleted.

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Active
4	Alibaba	Active

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
4	Alibaba	Active

- In the above-given example, CustomerName= "Apple" is deleted from the table.
- The Delete operation could violate referential integrity if the tuple which is deleted is referenced by foreign keys from other tuples in the same database.

- **Select Operation**

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
4	Alibaba	Active

CustomerID	CustomerName	Status
2	Amazon	Active

In the above-given example, CustomerName="Amazon" is selected

Best Practices for creating a Relational Model

- Data need to be represented as a collection of relations
- Each relation should be depicted clearly in the table
- Rows should contain data about instances of an entity
- Columns must contain data about attributes of the entity
- Cells of the table should hold a single value
- Each column should be given a unique name
- No two rows can be identical
- The values of an attribute should be from the same domain

Advantages of using Relational model

- **Simplicity**: A relational data model is simpler than the hierarchical and network model.
- **Structural Independence**: The relational database is only concerned with data and not with a structure. This can improve the performance of the model.
- **Easy to use**: The relational model is easy as tables consisting of rows and columns is quite natural and simple to understand
- **Query capability**: It makes possible for a high-level query language like SQL to avoid complex database navigation.
- **Data independence**: The structure of a database can be changed without having to change any application.
- **Scalable**: Regarding a number of records, or rows, and the number of fields, a database should be enlarged to enhance its usability.

Disadvantages of using Relational model

- Few relational databases have limits on field lengths which can't be exceeded.
- Relational databases can sometimes become complex as the amount of data grows, and the relations between pieces of data become more complicated.
- Complex relational database systems may lead to isolated databases where the information cannot be shared from one system to another.

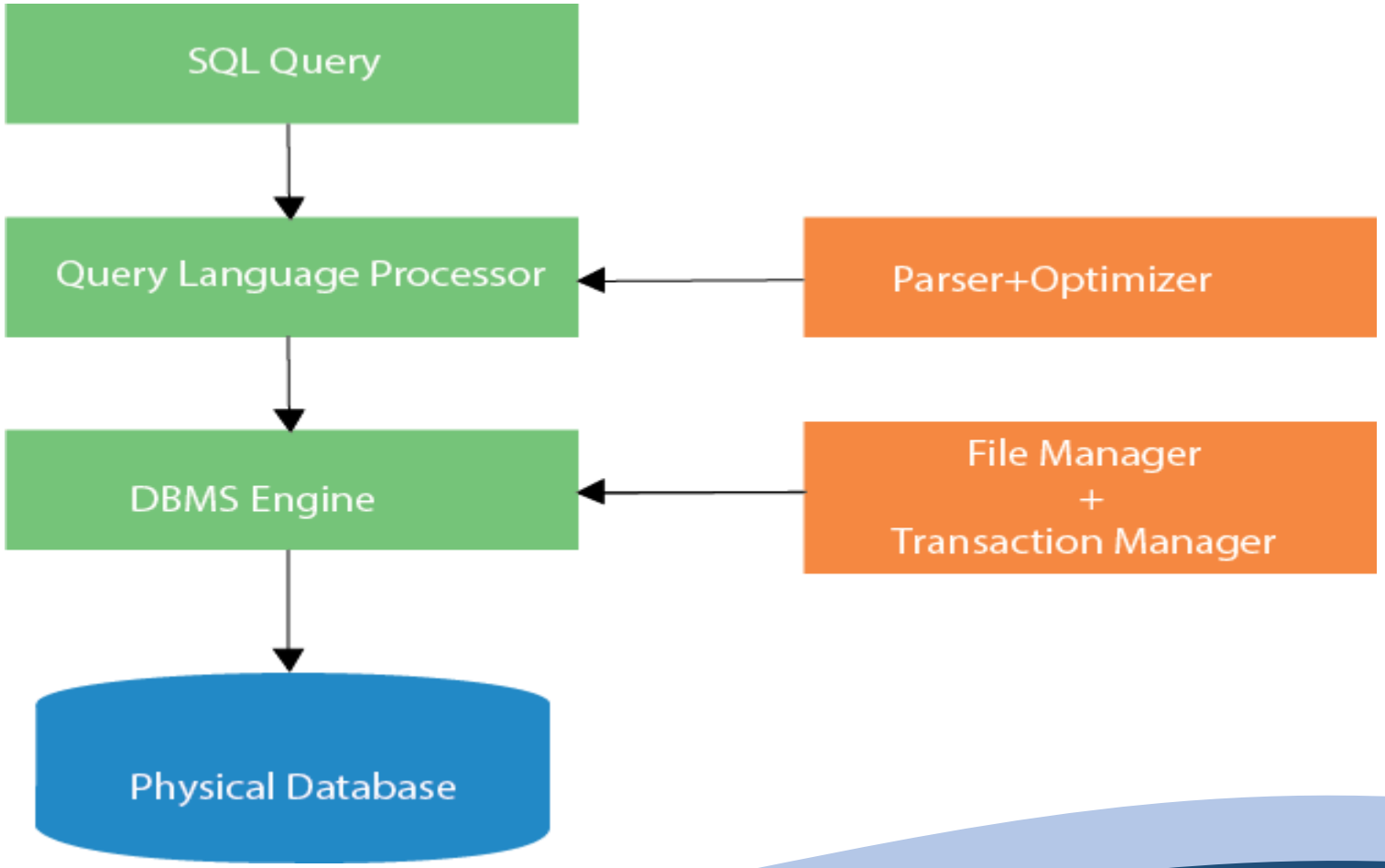
SQL(Structured Query Language)

- SQL stands for Structured Query Language. It is used for storing and managing data in relational database management system (RDMS).
- It is a standard language for Relational Database System. It enables a user to create, read, update and delete relational databases and tables.
- All the RDBMS like MySQL, Informix, Oracle, MS Access and SQL Server use SQL as their standard database language.
- SQL allows users to query the database in a number of ways, using English-like statements.

SQL process:

- When an SQL command is executing for any RDBMS, then the system figure out the best way to carry out the request and the SQL engine determines that how to interpret the task.
- In the process, various components are included. These components can be optimization Engine, Query engine, Query dispatcher, classic, etc.
- All the non-SQL queries are handled by the classic query engine, but SQL query engine won't handle logical files.

SQL Process


SQL Query
Query Language Processor
Parser+Optimizer
DBMS Engine
File Manager
+
Transaction Manager
Physical Database

Characteristics of SQL

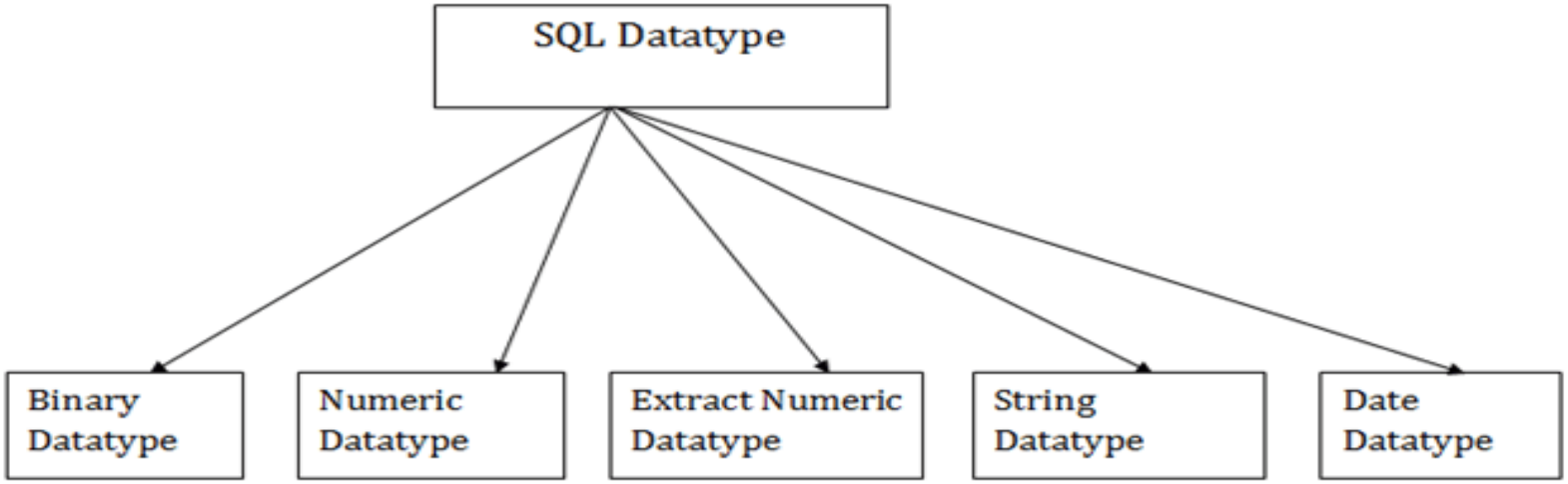
- SQL is easy to learn.
- SQL is used to access data from relational database management systems.
- SQL can execute queries against the database.
- SQL is used to describe the data.
- SQL is used to define the data in the database and manipulate it when needed.
- SQL is used to create and drop the database and table.
- SQL is used to create a view, stored procedure, function in a database.
- SQL allows users to set permissions on tables, procedures, and views.

Advantages of SQL

- **High speed**: Using the SQL queries, the user can quickly and efficiently retrieve a large amount of records from a database.
- **No coding needed**: In the standard SQL, it is very easy to manage the database system. It doesn't require a substantial amount of code to manage the database system.
- **Well defined standards**: Long established are used by the SQL databases that are being used by ISO and ANSI.
- **Portability**: SQL can be used in laptop, PCs, server and even some mobile phones.
- Int**eractive language**: SQL is a domain language used to communicate with the database. It is also used to receive answers to the complex questions in seconds.
- **Multiple data view**: Using the SQL language, the users can make different views of the database structure.

SQL Data type

- SQL Data type is used to define the values that a column can contain.
- Every column is required to have a name and data type in the database table.



- Binary Data types

Data Type	Description
binary	It has a maximum length of 8000 bytes. It contains fixed-length binary data.
varbinary	It has a maximum length of 8000 bytes. It contains variable-length binary data.
image	It has a maximum length of 2,147,483,647 bytes. It contains variable-length binary data.

- Approximate Numeric Data type :

Data type	From	To	Description
float	-1.79E + 308	1.79E + 308	It is used to specify a floating-point value e.g. 6.2, 2.9 etc.
real	-3.40e + 38	3.40E + 38	It specifies a single precision floating point number

- Exact Numeric Data type:

Data type	Description
int	It is used to specify an integer value.
smallint	It is used to specify small integer value.
bit	It has the number of bits to store.
decimal	It specifies a numeric value that can have a decimal number.
numeric	It is used to specify a numeric value.

- Character String Data type

Data type	Description
char	It has a maximum length of 8000 characters. It contains Fixed-length non-unicode characters.
varchar	It has a maximum length of 8000 characters. It contains variable-length non-unicode characters.
text	It has a maximum length of 2,147,483,647 characters. It contains variable-length non-unicode characters.

- Date and time Data types:

Datatype	Description
date	It is used to store the year, month, and days value.
time	It is used to store the hour, minute, and second values.
timestamp	It stores the year, month, day, hour, minute, and the second value.

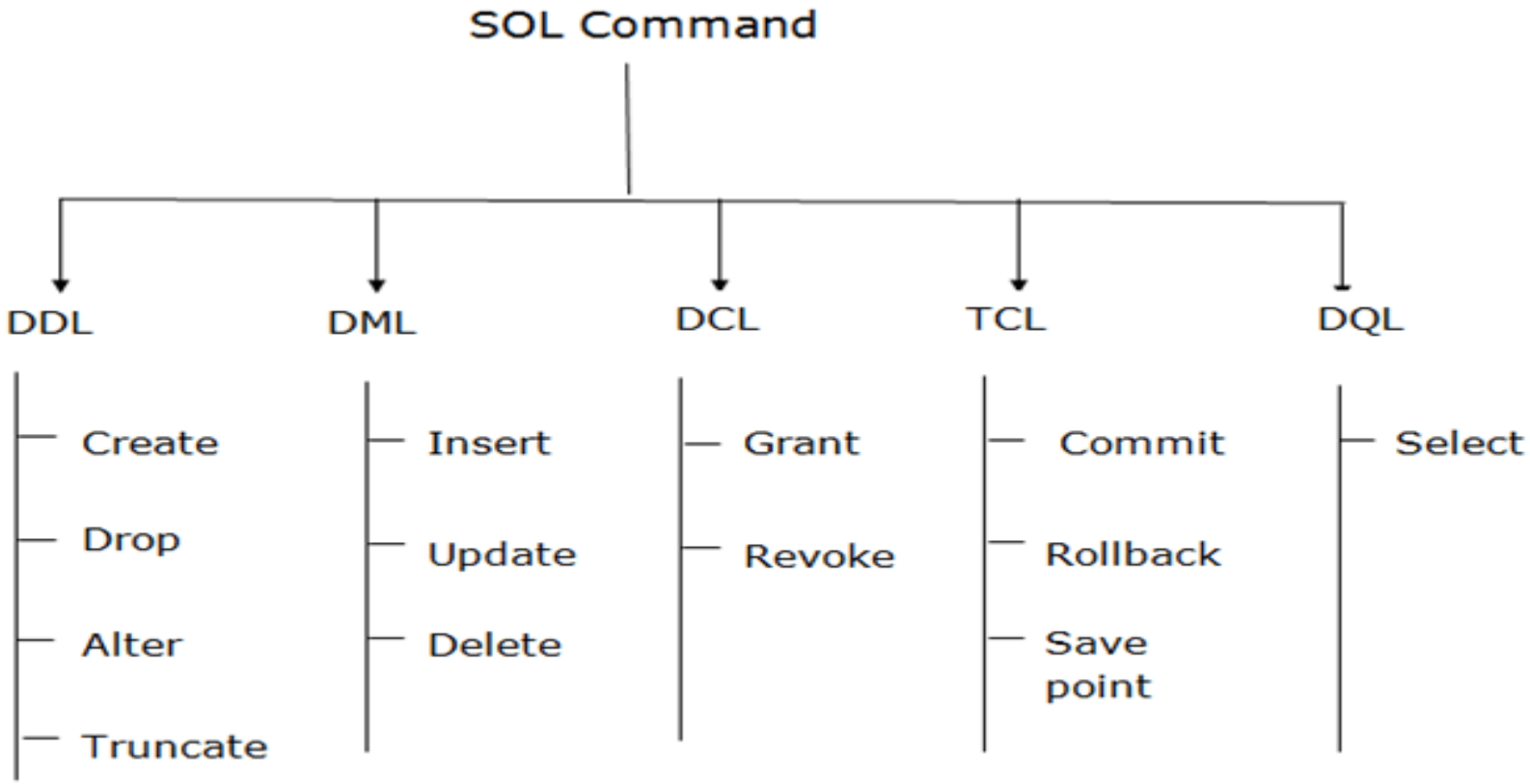
- **SQL Commands**

-SQL commands are instructions. It is used to communicate with the database. It is also used to perform specific tasks, functions, and queries of data.

-SQL can perform various tasks like create a table, add data to tables, drop the table, modify the table, set permission for users.

Types of SQL Commands

There are five types of SQL commands: DDL, DML, DCL, TCL, and DQL.



Data Definition Language (DDL)

- DDL changes the structure of the table like creating a table, deleting a table, altering a table, etc.
- All the command of DDL are auto-committed that means it permanently save all the changes in the database.
- Here are some commands that come under DDL:

  CREATE

  ALTER

  DROP

  TRUNCATE

Data Manipulation Language(DML)

- DML commands are used to modify the database. It is responsible for all form of changes in the database.
- The command of DML is not auto-committed that means it can't permanently save all the changes in the database. They can be rollback.
- Here are some commands that come under DML:

  INSERT

  UPDATE

  DELETE

Data Control Language(DCL)

- DCL commands are used to grant and take back authority from any database user.
- Here are some commands that come under DCL:

  Grant

  Revoke

Transaction Control Language (TCL)

- TCL commands can only use with DML commands like INSERT, DELETE and UPDATE only.
- These operations are automatically committed in the database that's why they cannot be used while creating tables or dropping them.
- Here are some commands that come under TCL:

  COMMIT

  ROLLBACK

  SAVEPOINT

Data Query Language(DQL)

- DQL is used to fetch the data from the database.
- It uses only one command:

  SELECT

Relational Query Language

A **Query language** is a language in which a use requests information from the database.

Types:

- Procedural Query Language
- Non-Procedural Language

❑In **Procedural language**, the user instructs the system to perform a sequence of operation on database to compute desired information. Example: Relational Algebra(RA).

❑In non Procedural language, the user describes the desired information without giving specific procedure for obtaining that information. Example: tuple relational calculus (TRC), Domain relational Calculus(DRC).

Relational Algebra (RA)

- Relational algebra is a **procedural** query language that works on relational model. The purpose of a query language is to retrieve data from database or perform various operations such as insert, update, delete on the data.

Types of operations in relational algebra

1. Basic Operations
2. Derived Operations

- **Basic/Fundamental Operations:**

1. Select (σ)
2. Project (∏)
3. Union (∪)
4. Set Difference (-)
5. Cartesian product (X)
6. Rename (ρ)

Derived Operations:

1. Natural Join (⋈)
2. Left, Right, Full outer join (⟕, ⟖, ⟗)
3. Intersection (∩)
4. Division (÷)

Schema:

branch(b_name, assets)

Depositor(cname,acc_no)

Loan(loan_no, b_name,amount)

Borrower(cname,loan_no)

Account (acc_no, balance)

Customer (cname,c_street,c_city)

Branch

B_name	Assets
New road	1500000
Bhaktapur	1000000
Lagankhel	1200000

Account

Acc_no	Balance
100	10000
101	15000
102	25000

Depositor

Cname	Acc_no
Geeta	102
Hari	100
Ram	101

Customer

Cname	C_Street	C_city
Ram	S1	Ktm
Geeta	S5	ktm
Hari	S2	Lalitpur
Rita	S3	Bhaktapur
John	S1	Bhaktapur

Loan

Loan_no	B_name	Amount
L100	Bhaktapur	100000
L101	Lalitpur	200000
L102	Bhaktapur	50000

Borrower

Cname	Loan_no
Hari	L102
Rita	L100
John	L101

Select Operation(σ)

- The select operation selects tuples that satisfy a given predicate.
- It is denoted by sigma (σ).
- Notation: σ p(r)
- **Where:**
- **σ** is used for selection prediction
  r is used for relation
  p is used as a propositional logic formula which may use connectors like: AND OR and NOT. These relational can use as relational operators like =, ≠, ≥, <, >, ≤.

1.Find the customers that live in bhaktapur.

RA expression:

σ c_city = “Bhaktapur”(Customer)

Output:

Customer

Cname	C_Street	C_city
Rita	S3	Bhaktapur
John	S1	Bhaktapur

Customer

Cname	C_Street	C_city
Ram	S1	Ktm
Geeta	S5	ktm
Hari	S2	Lalitpur
Rita	S3	Bhaktapur
John	S1	Bhaktapur

2. Find the customers who live in S5 street in Ktm.

σ c_city = “ktm” AND c_street= “S5” (Customer)

Output:

Customer

Cname	C_Street	C_city
Geeta	S5	ktm

- Find the loans that have amount greater than 50000.

σ amount>50000(Loan)

Output:

Loan

Loan_no	B_name	Amount
L100	Bhaktapur	100000
L101	Lalitpur	200000

Loan

Loan_no	B_name	Amount
L100	Bhaktapur	100000
L101	Lalitpur	200000
L102	Bhaktapur	50000

- Find the loans that have either loan less than 100000 or loan issued from lalitpur branch.

RA expression:

σ amount<100000 OR B_name="lalitpur" (Loan)

Output:

Loan

Loan_no	B_name	Amount
L101	Lalitpur	200000
L102	Bhaktapur	50000

Project Operation(∏)

- This operation shows the list of those attributes that we wish to appear in the result. Rest of the attributes are eliminated from the table.
- It is denoted by ∏.
- Notation: ∏ A1, A2, An (r)
- **Where**
- **A1**, **A2**, **A3** is used as an attribute name of relation **r**.
- **Degree**
  - Number of attributes in <attribute list>
- **Duplicate elimination**
  - Result of PROJECT operation is a set of distinct tuples

1.Find the customer name who have account at a bank.

∏ cname(Depositor)

2.Find the customer name who have loan at bank.

∏ cname(Borrower)

3.Find the customer name and city where they live.

∏ cname, c_city(Customer)

Depositor

Cname	Acc_no
Geeta	102
Hari	100
Ram	101

Borrower

Cname	Loan_no
Hari	L102
Rita	L100
John	L101

Customer

Cname	C_Street	C_city
Ram	S1	Ktm
Geeta	S5	ktm
Hari	S2	Lalitpur
Rita	S3	Bhaktapur
John	S1	Bhaktapur

Output:

1.

Depositor

Cname
Geeta
Hari
Ram

2.

Borrower

Cname
Hari
Rita
John

3.

Customer

Cname	C_city
Ram	Ktm
Geeta	ktm
Hari	Lalitpur
Rita	Bhaktapur
John	Bhaktapur

Project Operation with Selection

- Find the customer name who live in ktm city.

π cname(σ c_city="ktm"(customer))

Output:

Customer

Cname
Ram
Geeta

Customer

Cname	C_Street	C_city
Ram	S1	Ktm
Geeta	S5	ktm
Hari	S2	Lalitpur
Rita	S3	Bhaktapur
John	S1	Bhaktapur

- Find the customer name and customer street who live in S1 street.

π cname, c_street(σ c_street="S1"(customer))

OR

σ c_street="S1" (π cname, c_street((customer))

Output:

Customer

Cname	C_Street
Ram	S1
John	S1

Project Operation with Selection

- Find the customer name and customer street who live in S1 street or who live in lalitpur

π cname, c_street(σ c_street="S1" OR c_city="lalitpur"(customer))

- Find the customer name and customer street who live in S1 street and who live in lalitpur.

π cname, c_street(σ c_street="S1" AND c_city="lalitpur"(customer))

Output:

Customer

Cname	C_Street

Customer

Cname	C_Street	C_city
Ram	S1	Ktm
Geeta	S5	ktm
Hari	S2	Lalitpur
Rita	S3	Bhaktapur
John	S1	Bhaktapur

Output:

Customer

Cname	C_Street
Ram	S1
Hari	S2
John	S1

- Find the branch name whose assets is greater than or equals to 1200000.

π b_name(σ assets >= 1200000(Branch))

- Find the account no whose balance is between 12000 to 25000.

π acc_no(σ balance >= 12000 AND balance<=25000(Account))

- Find the loan details of loan whose loan amount is between 10000 to 100000

σ amount>=10000 AND amount<=100000(loan)

- Find the loan_no and amount whose loan is maintained at bhaktapur branch.

π loan_no, amount(σ branch = ‘’bhaktapur”(Loan))

Union Operation

- Suppose there are two tuples R and S. The union operation contains all the tuples that are either in R or S or both in R & S.
- It eliminates the duplicate tuples. It is denoted by ∪.

Notation: R ∪ S

- A union operation must hold the following condition:

  -R and S must have the attribute of the same number.

  -Duplicate tuples are eliminated automatically.

  - Data type of attributes must be same

- Example:

Find the customers name who have bank transactions.

π c_name(depositor) U π c_name(borrower)

Output:

Cname
Geeta
Hari
Ram
Rita
John

Borrower

Cname	Loan_no
Hari	L102
Rita	L100
John	L101

Depositor

Cname	Acc_no
Geeta	102
Hari	100
Ram	101

Intersection Operation

- Suppose there are two tuples R and S. The set intersection operation contains all tuples that are in both R & S.
- It is denoted by intersection ∩.

Notation: R ∩ S

- A union operation must hold the following condition:

  -R and S must have the attribute of the same number.

  -Duplicate tuples are eliminated automatically.

  - Data type of attributes must be same

- Example:

Find the customers name who have both account and loan at bank.

π c_name(depositor) ∩ π c_name(borrower)

Output:

Cname
Hari

Borrower

Cname	Loan_no
Hari	L102
Rita	L100
John	L101

Depositor

Cname	Acc_no
Geeta	102
Hari	100
Ram	101

Set Difference

- Suppose there are two tuples R and S. The set intersection operation contains all tuples that are in R but not in S.
- It is denoted by intersection minus (-).

Notation: R - S

Example

Find the Customers who have account but not loan at the bank.

π c_name(depositor) - π c_name(borrower)

Output:

Cname
Geeta
Ram

Borrower

Cname	Loan_no
Hari	L102
Rita	L100
John	L101

Depositor

Cname	Acc_no
Geeta	102
Hari	100
Ram	101