Chapter-1: Database Concepts and SQL

Learning Objectives

At the end of this chapter the students will be able to understand:

- What is DBMS?
- What is relational database model?
- Relation
- Tuples
- SQL
- DDL
- DML
- Relational Algebra
- Selection
- Projection
- Union
- Cartesian Product

Introduction

Database is a collection of related information that is organized in such a way that supports for easy access, modify and maintain data. The contents of a database are obtained by combining data from all the different sources in an organization. Generally, the database is managed by some special software packages known as Database Management Systems (DBMSs). DBMSs are specially designed applications to create connection between user and program, and to store data in an organized manner. The purpose of DBMSs software is to allow the user to create, modify and administration of database. Examples of database management systems are: Ms-Access, MySQL, PostgreSQL, SQLite, Microsoft SQL Server, Oracle, SAP, dBASE, FoxPro, etc.

Relational data model

The relational data model is a database model based on first-order predicate logic(First Order Predicate Logic is one where the quantification is over simple variables), formulated and proposed by Edgar F. Codd. in 1969. The first-order predicate logic is a symbolised reasoning, in which statement is broken down into a subject and a predicate. The predicate modifies the properties of the subject, while in the first-order logic, a predicate can only refer to a single subject. In the relational data model, database is represented as collection of related tables. Each table is termed as relation and has its unique name in the relational data model. Tables are formed by using rows and columns. A row (horizontal subset) of a table represents a tuple or record, while column (vertical subset) of a table represents an attribute.

Relation

In database, a relation means a 'table', in which data are organized in the form of rows and columns. Therefore in database, relations are equivalent to tables.

For example

Relation: Student

Ad No	Name	Class	Section	Average
101	Anu	12	A	85
105	Balu	12	D	65
203	Leena	11	B	95
205	Madhu	10	B	75
305	Surpreeth	9	C	70
483	Usha	6	A	60

Domain

A domain is the original sets of atomic values used to model data. In data base management and database, a domain refers to all the possible unique values of a particular column.

For example:

i) The domain of gender column has a set of two possible values i.e, Male or Female.

ii) The domain of marital status has a set of four possible values i.e, Married, Unmarried, Widows and Divorced.

Therefore, a domain is a set of acceptable values of a particular column, which is based on various properties and data types. We will discuss data types later in this chapter.

Tuple

Horizontal subset/information in a table is called tuple. The tuple is also known as a 'record', which gives particular information of the relation (table).

For example:

i) In customer table, one row gives information about one customer only.

ii) In student table, one row gives information about one student only.

Key

Keys are an important part of a relational database and a vital part of the structure of a table. They help enforce integrity and help identify the relationship between tables. There are three main types of keys -

candidate keys, primary keys, foreign keys and alternate keys.

Primary Key: A column or set of columns that uniquely identifies a row within a table is called primary key.

Candidate Key: Candidate keys are set of fields (columns with unique values) in the relation that are eligible to act as a primary key.

Alternate Key: Out of the candidate keys, after selecting a key as primary key, the remaining keys are called alternate key.

Foreign Key: A foreign key is a field (or collection of fields) in one table that uniquely identifies a row of another table. In other words, a foreign key is a column or a combination of columns that is used to establish a link between two tables.

Degree

The degree is the number of attributes (columns) in a table.

Cardinality

Cardinality is number of rows (tuples) in a table.

Example:

Relation: Student

Ad No	Name	Class	Section	Average
101	Anu	12	A	85
105	Balu	12	D	65
203	Leena	11	B	95
205	Madhu	10	B	75
305	Surpreeth	9	C	70
483	Usha	6	A	60

Fields (Attributes/Columns):- AdNo, Name, Class, Section and Average.

Tuples (Rows/Records):

101	Anu	12	A	85

Domain: Possible values of section are ('A','B','C','D')

Degree: 5 (Number of columns).

Cardinality: 6 (Number of rows).

Candidate Key: In the above table, AdNo and Name has unique values. Therefore, AdNo and Name are candidate keys.

Primary Key: Out of the AdNo and Name, AdNo is the primary key.

Alternate Key: In the candidate key, AdNo is the primary key and the Name is the Alternate key.

Structured Query Language (SQL)

Structured Query Language (SQL) is a standard language used for accessing databases. This is a special purpose programming language used to create a table, manage data and mainipulate data.

SQL provides statements for a variety of tasks, including:

i) Querying data
ii) Inserting, updating, and deleting rows in a table
iii) Creating, replacing, altering, and dropping objects (tables)
iv) Controlling access to the database and its objects (tables)
v) Guaranteeing database consistency and integrity

SQL unifies all of the preceeding tasks in one consistent language.

Advantages of using SQL:

1) **SQL is portable:** SQL is running in all servers, mainframes, PCs, laptops, and even mobile phones.
2) **High speed:** SQL queries can be used to retrieve large amounts of records from a database quickly and efficiently.
3) **Easy to learn and understand:** SQL generally consists of English statements and as such, it is very easy to learn and understand. Besides, it does not require much coding unlike in programming languages.
4) **SQL is used with any DBMS system with any vendor:** SQL is used by all the vendors who develop DBMS. It is also used to create databases, manage security for a database, etc. It can also be used for updating, retrieving and sharing data with users.
5) **SQL is used for relational databases:** SQL is widely used for relational databases.
6) **SQL acts as both programming language and interactive language:** SQL can do both the jobs of being a programming language as well as an interactive language at the same time.
7) **Client/Server language:** SQL is used for linking front end computers and back end databases. It provides client server architecture (Email, and the World Wide Web - all apply the client-server architecture).
8) **Supports object based programming:** SQL supports the latest object based programming and is highly flexible.

Types of SQL Statements

The SQL statements are categorized into different categories based upon the purpose. They are;

i) Data Definition Language (DDL) statement
ii) Data Manipulation Language (DML) statement
iii) Transaction Control Statement
iv) Session Control Statement
v) System Control Statement
vi) Embedded SQL Statement

Out of these six, we will be studying only the first two types in this course.

Data Definition Language (DDL) Statements

Data Definition Language (DDL) or Data Description Language (DDL) is a standard for commands that defines the different structures in a database. DDL statements are used to create structure of a table, modify the existing structure of the table and remove the existing table. Some of the DDL statements are CREATE TABLE, ALTER TABLE and DROP TABLE.

Data Manipulation Language (DML) Statements

Data Manipulation Language (DML) statements are used to access and manipulate data in existing tables. The manipulation includes inserting data into tables, deleting data from the tables, retrieving data and modifying the existing data. The common DML statements are SELECT, UPDATE, DELETE and INSERT.

Data Types

Each value manipulated by SQL Database has a data type. The data type of a value associates a fixed set of properties with the value. In SQL there are three main data types: Character, Number, and Date types.

Character

Character data types stores character (alphanumeric) data, which are words and free-form text. They are less restrictive than other data types and consequently have fewer properties. For example, character columns can store all alphanumeric values, but number columns can store only numeric values. Character data types are;

i) CHAR
ii) VARCHAR
iii) VARCHAR2

CHAR: CHAR should be used for storing fix length character strings. String values will be space/blank padded (The adding of meaningless data [usually blanks] to a unit of data to bring it up to some fixed size) before they are stored on the disk. If this type is used to store variable length strings, it will waste a lot of disk space (always allocate fixed memory) . If we declare data type as CHAR, then it will occupy space for

NULL values.

Format: CHAR(n)

Fixed-length character string having maximum length n.

VARCHAR: Varchar is a variable character string. If we declare data type as VARCHAR, then it will occupy space for NULL values. It can have maximum of 2000 characters.

Format: VARCHAR (n)

Variable-length character string having maximum length n.

VARCHAR2: VARCHAR2 is used to store variable length character strings. The string value's length will be stored on disk with the value itself. VARCHAR2 can store up to 4000 bytes of characters. Thus, the difference between VARCHAR and VARCHAR2 is that VARCHAR is ANSI standard but takes up space for variables, whereas the VARCHAR2 is used only in Oracle but makes more efficient use of space.

Format: VARCHAR2 (n)

Example:

CHAR(10) has fixed length, right padded with spaces.

VARCHAR(10) has fixed length, right padded with NULL

VARCHAR2(10) has variable length.

Name char (10): Suppose if we store Name is as "Ravi", then first four places of the ten characters are filled with Ravi and the remaining 6 spaces are also allocated to Name. Thus, the size of name is always ten.

Name varchar (10): Suppose if we store Name is as "Ravi", then first four places of the ten characters are filled with Ravi and the remaining 6 spaces are filled with NULL.

Name varchar2 (10): Suppose if we store Name is as "Ravi", then only first four places are filled with Ravi.

The following table gives possible string data types used in different DBMS

Data type	Access	SQL Server	Oracle	My SQL	Postgre SQL
string (fixed)	N/A	Char	Char	Char	Char
string (variable)	Text (<256) Memo (65k+)	Varchar	Varchar Varchar2	Varchar	Varchar

Numeric data type: Numeric data types are mainly used to store number with or without fraction part. The numeric data types are:

1. NUMBER
2. DECIMAL
3. NUMERIC

4. INT
5. FLOAT

NUMBER: The Number data type stores fixed and floating-point numbers. The Number data type is used to store integers (negative, positive, floating) of up to 38 digits of precision.

The following numbers can be stored in a Number data type column:

- Positive numbers in the range 1 x 10-130 to 9.99...9 x 10125 with up to 38 significant digits.
- Negative numbers from -1 x 10-130 to 9.99...99 x 10125 with up to 38 significant digits.
- Zero

Format:

NUMBER (p, s)

Where;

- 'p' is the precision or the total number of significant decimal digits, where the most significant digit is the left-most nonzero digit and the least significant digit is the right-most known digit.
- 's' is the scale or the number of digits from the decimal point to the least significant digit.

DECIMAL and NUMERIC: Decimal and numeric data types have fixed precision and scale.

Format:

DECIMAL[(p[, s])] and NUMERIC[(p[, s])]

Square brackets ([ ]) are option.

where;

- 'p' is the precision or the total number of significant decimal digits, where the most significant digit is the left-most nonzero digit and the least significant digit is the right-most known digit.
- 's' is the scale or the number of digits from the decimal point to the least significant digit.

INT/INTEGER: The int data type is the integer data type in SQL. This used to store integer number (without any fraction part).

FLOAT: This data type is used to store number with fraction part(real numbers).

The following table gives possible numeric data types used in difference DBMS

Data type	Access	SQL Server	Oracle	My SQL	Postgre SQL
Integer	Number	Int	Number (integer)	Int Integer Numeric	Int Integer
Float	Number (single)	Float Real	Number	Float Decimal Numeric	Numeric

DATE

Date is used to store valid date values, which is ranging from January 1, 4712 BC to December 31, 9999 AD. The date formats are: YYYY-MM-DD or DD/MON/YY or YYYY/MM/DD or MM/DD/YY or DD-MON-YYYY.

Format: DATE

Relational Algebra

An algebra is a combination of set of operands and a set of operators. We can form algebraic expressions by applying operators to operands. Relational algebra consists of a set of operations that take one or two relations as input and produces a new relation as output. Operators map values are taken from the domain and put it into other domain values. If domain is produced from more than one relation, then we get relational algebra.

Operation in relational algebra:

1. Selection
2. Projection
3. Union
4. Cartesian Product

Selection

Selection in relational algebra returns those tuples(records) in a relation that fulfil a condition(Produce table containing subset of rows).

Syntax:

? condition (relation)

Example: The table S (for STUDENT).

Relation: Student

AdNo	Name	Class	Section	Average
101	Anu	12	A	85
105	Balu	12	D	65
203	Leena	11	B	95
205	Madhu	10	B	75
305	Surpreeth	9	C	70
483	Usha	6	A	60

? class=12 (S)

Output:

AdNo	Name	Class	Section	Average
101	Anu	12	A	85
105	Balu	12	D	65

Projection

Projection in relational algebra returns those columns in a relation that given in the attribute list (Produce table containing subset of columns).

Syntax:

π attribute list(relation)

Example:

π Adno,Name (S)

Output:

AdNo	Name
101	Anu
105	Balu
203	Leena
205	Madhu
305	Surpreeth
483	Usha

Union

The union operator is used to combine two or more tables. Each table within the UNION should have the same number of columns, similar data types and also the columns must be in the same order. In the union operation, duplicate records will be automatically removed from the resultant table.

For example:

Table: Student 1

Roll no	Name
11	Kumar
22	Mohan
33	Rohit

Table: Student2

Roll no	Name
22	Mohan
11	Rahul
77	Kavita

Query is

σ (Students 1)

Union

σ (Students 2)

Or

π rollno, Name (Students 1)

Union

π rollno, Name (Students 2)

Resultant table is:

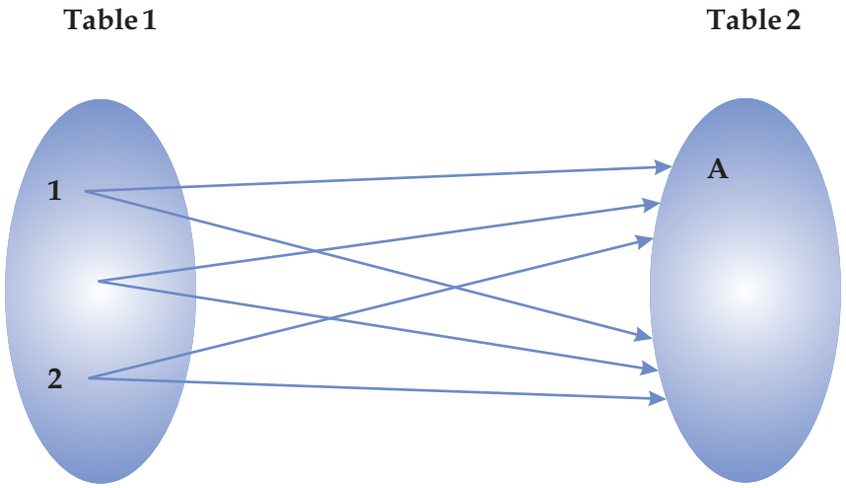
Roll no	Name
11	Kumar
22	Mohan
33	Rohit
11	Rahul
77	Kavita

In the above resultant table, student1 is copied as it is, but in student2, roll no 22 Mohan's information is same as student1. So, that is not copied in the resultant table again. Roll no 11 is same as student1, but name is different. So, that is copied in the resultant table. Roll no 77 is not in student1 table, so that is also copied in the resultant table.

Cartesian product

SQL joins are used to relate information in different tables. It combines fields from two or more tables by comparing values of common columns (join condition). A join condition is a part of the SQL query that retrieves rows from two or more tables. If join condition is ommitted or if it is invalid, then join operation will result in a Cartesian product. Cartesian product is a binary operation and is denoted by (x) Cartesian product returns a number of rows equal to number of rows in the first table multiply by number of rows in

the second table. At the same time, number of columns equal to number of columns in the first table added by number of columns in the second table.



For example:

Table 1: Product

Product_no	Product_name	Price
111	Computer	50000
222	Printer	10000
333	Scanner	12000
444	Modem	500

Table 2: Customer

Cust_no	Cust_name	City	Product_no
101	Kavitha	Delhi	333
201	Mohan	Mumbai	222
301	Rohan	Bangalore	111
401	Sahil	Mumbai	333
501	Rohita	Delhi	444

Query is

σ (Product, customer)

Product_no	Product_name	Price	Cust_no	Cust_name	City	Product_no
111	Computer	50000	101	Kavitha	Delhi	333
111	Computer	50000	201	Mohan	Mumbai	222
111	Computer	50000	301	Rohan	Bangalore	111
111	Computer	50000	401	Sahil	Mumbai	333
111	Computer	50000	501	Rohita	Delhi	444
222	Printer	10000	101	Kavitha	Delhi	333
222	Printer	10000	201	Mohan	Mumbai	222
222	Printer	10000	301	Rohan	Bangalore	111
222	Printer	10000	401	Sahil	Mumbai	333
222	Printer	10000	501	Rohita	Delhi	444
333	Scanner	12000	101	Kavitha	Delhi	333
333	Scanner	12000	201	Mohan	Mumbai	222
333	Scanner	12000	301	Rohan	Bangalore	111
333	Scanner	12000	401	Sahil	Mumbai	333
333	Scanner	12000	501	Rohita	Delhi	444
444	Modem	500	101	Kavitha	Delhi	333
444	Modem	500	201	Mohan	Mumbai	222
444	Modem	500	301	Rohan	Bangalore	111
444	Modem	500	401	Sahil	Mumbai	333
444	Modem	5003	501	Rohita	Delhi	444

Table 1:

Number of rows (cardinality) = 4

Number of columns (degree) = 3

Table 2:

Number of rows (cardinality) = 5

Number of columns (degree) = 4

Cartesian product:

Number of rows (cardinality) = 4*5 = 20

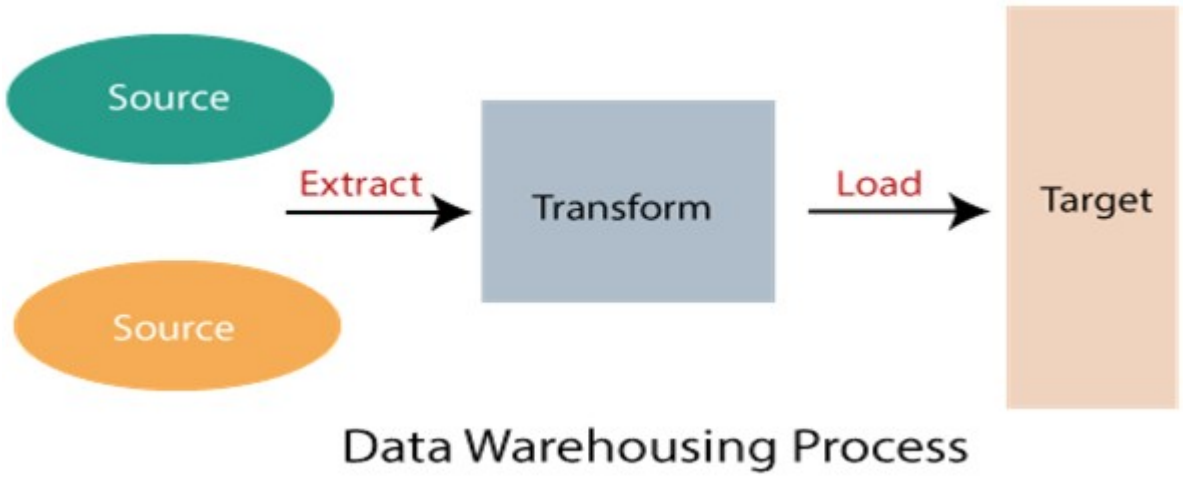
Number of columns (degree) = 3+4 = 7

Data Mining Vs Data Warehousing

Data warehouse refers to the process of compiling and organizing data into one common database, whereas **data mining** refers to the process of extracting useful data from the databases. The data mining process depends on the data compiled in the data warehousing phase to recognize meaningful patterns. A data warehousing is created to support management systems.

Data Warehouse:

A **Data Warehouse** refers to a place where data can be stored for useful mining. It is like a quick computer system with exceptionally huge data storage capacity. Data from the various organization's systems are copied to the Warehouse, where it can be fetched and conformed to delete errors. Here, advanced requests can be made against the warehouse storage of data.



Data Warehousing Process

Data warehouse combines data from numerous sources which ensure the data quality, accuracy, and consistency. Data warehouse boosts system execution by separating analytics processing from transnational databases. Data flows into a data warehouse from different databases. A data warehouse works by sorting out data into a pattern that depicts the format and types of data. Query tools examine the data tables using patterns.

Data warehouses and **databases** both are relative data systems, but both are made to serve different purposes. A data warehouse is built to store a huge amount of historical data and empowers fast requests over all the data, typically using **Online Analytical Processing** (OLAP). A

database is made to store current transactions and allow quick access to specific transactions for ongoing business processes, commonly known as **Online Transaction Processing** (OLTP).

Important Features of Data Warehouse

The Important features of Data Warehouse are given below:

1. Subject Oriented

A data warehouse is subject-oriented. It provides useful data about a subject instead of the company's ongoing operations, and these subjects can be customers, suppliers, marketing, product, promotion, etc. A data warehouse usually focuses on modeling and analysis of data that helps the business organization to make data-driven decisions.

2. Time-Variant:

The different data present in the data warehouse provides information for a specific period.

3. Integrated

A data warehouse is built by joining data from heterogeneous sources, such as social databases, level documents, etc.

4. Non- Volatile

It means, once data entered into the warehouse cannot be change.

Advantages of Data Warehouse:

- More accurate data access
- Improved productivity and performance
- Cost-efficient
- Consistent and quality data

Data Mining:

Data mining refers to the analysis of data. It is the computer-supported process of analyzing huge sets of data that have either been compiled by computer systems or have been downloaded into the computer. In the data mining process, the computer analyzes the data and extract useful information from it. It looks for hidden patterns within the data set and try to predict future behavior. Data mining is primarily used to discover and indicate relationships among the data sets.

Data mining aims to enable business organizations to view business behaviors, trends relationships that allow the business to make data-driven decisions. It is also known as knowledge Discover in Database (KDD). Data mining tools utilize AI, statistics, databases, and machine learning systems to discover the relationship between the data. Data mining tools can support business-related questions that traditionally time-consuming to resolve any issue.

Important features of Data Mining:

The important features of Data Mining are given below:

- It utilizes the Automated discovery of patterns.
- It predicts the expected results.
- It focuses on large data sets and databases
- It creates actionable information.

Advantages of Data Mining:

i. Market Analysis:

Data Mining can predict the market that helps the business to make the decision. For example, it predicts who is keen to purchase what type of products.

ii. Fraud detection:

Data Mining methods can help to find which cellular phone calls, insurance claims, credit, or debit card purchases are going to be fraudulent.

iii. Financial Market Analysis:

Data Mining techniques are widely used to help **Model Financial Market**

iv. Trend Analysis:

Analyzing the current existing trend in the marketplace is a strategic benefit because it helps in cost reduction and manufacturing process as per market demand.