



Structure Query Language (SQL)

- Purpose and importance of SQL.
- How to retrieve data from database using SELECT and:
 - ◆ Use compound WHERE conditions.
 - ◆ Sort query results using ORDER BY.
 - ◆ Use aggregate functions.
 - ◆ Group data using GROUP BY and HAVING.
 - ◆ Use subqueries.
 - ◆ Join tables together.



Structure Query Language (SQL)

- How to update database using **INSERT**, **UPDATE** and **DELETE** statements.
- Data types supported by SQL-92.
- How to create and delete tables.



Objectives of SQL

- Ideally, database language should allow a user to:
 - ◆ create the database and relation structures;
 - ◆ perform insertion, modification, deletion of data from relations;
 - ◆ perform simple and complex queries.
- Must perform these tasks with minimal user effort and command structure and syntax must be easy to learn.
- It must be **portable**.



Objectives of SQL

- SQL is intended to satisfy these requirements
- SQL is a transform-oriented language with 2 major components:
 - ◆ A DDL for defining the database structure.
 - ◆ A DML for retrieving and updating data.
- SQL does not contain flow control commands. These must be implemented using a programming or job-control language, or interactively by the decisions of the user.



Objectives of SQL

- SQL is relatively easy to learn:
 - ◆ It is a non-procedural language - you specify ***what*** information you require, rather than ***how*** to get it.
 - ◆ It is essentially free-format.
 - ◆ Consists of standard English words:

```
CREATE TABLE staff(sno VARCHAR(5),  
                    lname VARCHAR(30),  
                    salary DECIMAL(7,2));
```

```
INSERT INTO staff  
VALUES ('SG16', 'Brown', 8300);
```

```
SELECT sno, lname, salary  
FROM staff  
WHERE salary > 10000;
```



Objectives of SQL

- Can be used by a range of users including DBAs, management personnel, application programmers and other types of end users.
- An ISO standard now exists for the SQL language, making it both the formal and ***de facto*** standard for relational databases.



History of SQL

- In 1974, D. Chamberlin (IBM) defined a language called 'Structured English Query Language' or **SEQUEL**.
- A revised version **SEQUEL/2** was defined in 1976 and subsequently changed to SQL for legal reasons.
- IBM subsequently produced a prototype DBMS called **System R**, based on SEQUEL/2 (eventually DB2).
- In the late 1970s, **ORACLE** was produced and was probably first commercial implementation of a relational DBMS based on SQL.



History of SQL

- In 1987, ANSI and ISO published an initial standard for SQL.
- In 1992, the first major revision to the ISO standard occurred, sometimes referred to as **SQL2** or **SQL/92**.
- **SQL Access Group** is attempting to define a set of enhancements that will support interoperability across disparate systems.



Writing SQL Commands

- SQL statement consists of ***reserved words*** and ***user-defined words***.
 - ◆ Reserved words are a fixed part of SQL and must be spelt exactly as required and cannot be split across lines.
 - ◆ User-defined words are made up by user and represent names of various database object such as relations, columns, views.
- Most components of an SQL statement are ***case insensitive***, except for literal character data.



Writing SQL Commands

- SQL is more readable if indentation and lineation are used:
 - ◆ Each clause in a statement should begin on a new line.
 - ◆ Start of a clause should line up with start of other clauses.
 - ◆ If clause has several parts, they should each appear on a separate line and be indented under start of clause.



SQL Statement Notation

- Use extended form of Backus Naur Form (BNF) notation:
 - ◆ Upper case letters represent reserved words.
 - ◆ Lower case letters represent user-defined words.
 - ◆ A vertical bar (|) indicates a *choice* among alternatives.
 - ◆ Curly braces { } indicate a *required element*.
 - ◆ Square brackets [] indicate a *optional element*.
 - ◆ An ellipsis (...) indicates optional repetition (0 or more times).



SELECT Statement

SELECT [DISTINCT | ALL]
{ * | column_expressions [AS new_name]] [...]}
FROM table_name [alias] [...]
[WHERE condition]
[GROUP BY column_list **[HAVING condition]**
[ORDER BY column_list

- **SELECT** Specifies which columns are to appear in output.
- **FROM** Specifies table or tables to be used.
- **WHERE** Filters rows subject to some condition.



SELECT Statement

- **GROUP BY** Forms groups of rows with same column value.
- **HAVING** Filters groups subject to some condition.
- **ORDER BY** Specifies the order of the output.

- Order of the clauses cannot be changed.
- Only SELECT and FROM are mandatory.