

SQL Course

Structured Query Language

- Much of the user benefits of relational database technology are offered by the powerful relational query languages.
- These languages exploit the relational theory to provide ad hoc query and data management.
- The standard language for defining, constructing and manipulating information in a relational database is called **SQL**(Structured Query Language).
- The **SQL**(pronounced **sequel**) has been standardised over the last ten years and is now the most widely used database query language.

What is SQL?

- SQL is the **standard** relational query language and is based on relational algebra.
- It is between a third and fourth generation language. (3GL/4GL)

SQL supports the:

- Creation and maintenance of database information.
- Rapid prototyping and testing as it makes information retrieval and Storage easy.
- Porting of database application programs onto different RDBMS implementations (with only minor alterations required) since SQL is a standard language.
- Control over security, concurrency and locking (of tables, rows and values).

SQL is not just an information retrieval language, it has 3 sub languages:

- **Data Definition Language:** allows definition of relations (tables), their content type and integrity rules which govern attribute usage and values.
- **Data Manipulation Language:** allows user to populate, manipulate, retrieve and delete information in the database.
- **Data Control Language:** allows users and database administrators to specify security checks and controls on the database.

The story of SQL includes the following milestones:

- **1970** Codd publishes relational paper.
- **1973/75** University Ingres and System R prototypes.
- **1980** Oracle product released.
- **1983** IBM release DB2 & SQL/DS.
- **1986** ANSI standard SQL.
- **1989** Updated ANSI & ISO standard for SQL.
- **1992** Latest SQL standard (ISO & ANSI).[Known as **SQL2**]

The benefits of the adoption of the SQL standard are:

- Reduced training costs as the language is common across all RDBMS implementations.
- Increases application longevity as the applications use the standard which evolves slowly over time and maintains backward compatibility.

- Increases application portability as all RDBMS vendors support the SQL standard.
- Provides a basis for inter system communication.
- Simplifies customer choice as comparison of different vendor offerings is easier.

Caveat: Although all RDBMS vendors implement the SQL standard, several dialects have evolved where vendors have added in extra features for their customer benefits. These dialects differ very slightly between each vendor RDBMS.

- The **SQL** syntax is designed to be simple and consistent in its construction
- The language supports "English like" statements: i.e. say what you want, from where you want it, and under what conditions
- Statements are always terminated with a semi-colon ;

Example

Consider a table called 'aircraft' which contains different aircraft names, models, number of club class seats, number of economy class seats and the call sign of each aircraft as shown in the diagram below.

<i>aircraft name</i>	<i>model</i>	<i>club seats</i>	<i>econ seats</i>	<i>call sign</i>
Eagle Flyer	ATR42	22	40	N410C
(NULL)	Boeing 707-320C	50	102	9J-AEB
(NULL)	Boeing 727-200	34	100	N7255U
Finians Dream	Boeing 737-200	22	96	DQ-FDM
(NULL)	Boeing 737-200	8	120	N301SW

To find out the call signs of the aircraft and the number of club class seats on each aircraft of model 'Boeing 737-200', the SQL query would be:

```
SELECT call_sign, no_club_seats
FROM aircraft
WHERE model = 'Boeing 737-200';
```

Although the central core of a database management system is its information storage and query abilities, many other commonly used facilities are usually provided by a Relational Database System. Here are just a few of the most widely used database tools:

- **Report Writers:** These allow the automatic production of printed formatted reports based on current database values.
- **Screen Generators Forms:** These allow the rapid development screens or forms to facilitate data entry or data querying.
- **Network Connectivity Tools:** These allow applications which use the relational database to reside on a different computer (perhaps in a different location) to use the relation database without the user being aware of the distribution over the network.
- **Support for Embedded 3rd Generation Languages:** This allows programs written in 3rd generation programming languages like PASCAL, C, etc. to make runtime queries to a relational database. They usually take the form of either a Library of functions which the programmer can invoke with his/her program or a means of actually specifying relational queries in the programming language which a special pre-compiler processes and translates into appropriate runtime calls to the database.

- **CASE dictionaries.**

Many relational Database vendors distinguish their products by the number, sophistication, and performance of such database tools.