interskill learning

Introduction to Db2 Relational Databases and SQL

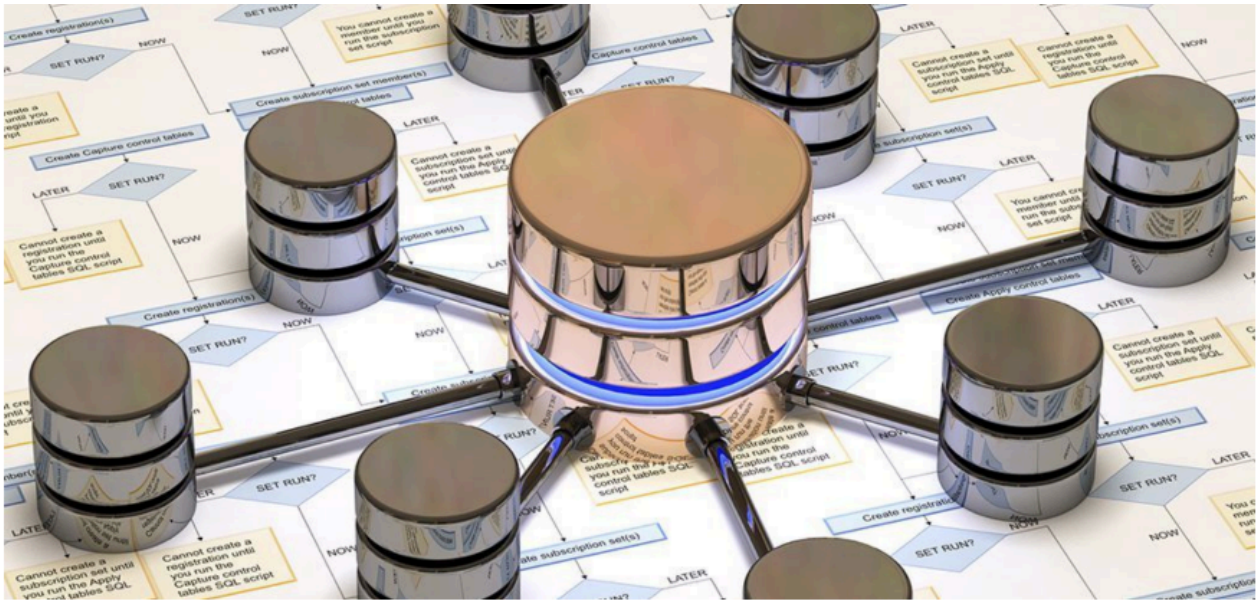
Objectives

Introduction to Db2 Relational Databases and SQL

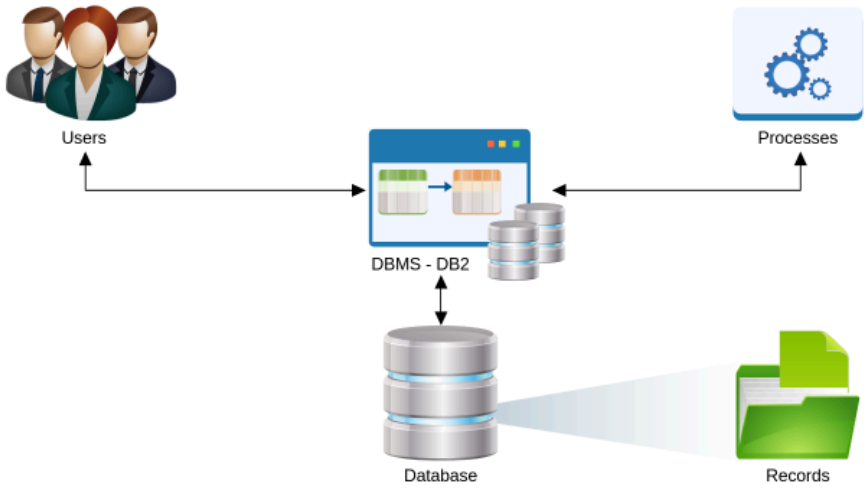
In this module you will look at the types of databases commonly used and focus on the workings of the relational database structure. You will also be introduced to SQL, which is the language used to interrogate data stored in a relational database management system (RDMS).

After completing this module, you should be able to identify:

- How a Relational Database is Structured
- The Tasks Performed by the Database Administrator (DBA)
- The Purpose and General Structure of SQL Statements



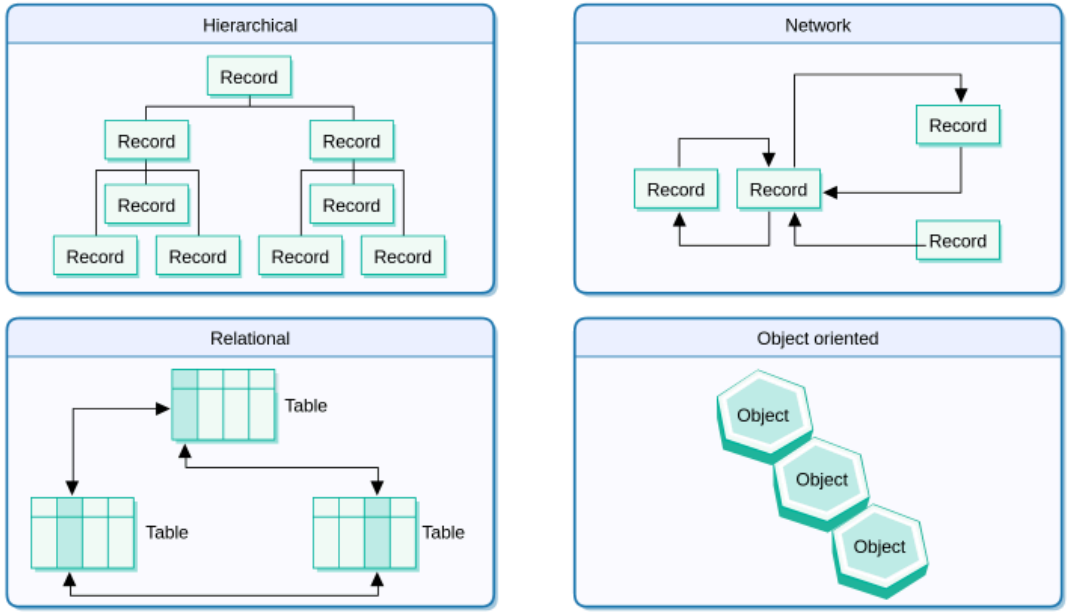
A database stores business data in an organized way for use by business systems. The database should provide a single, consistent view of the business data so that it can be centrally controlled and managed.



This diagram highlights the difference between a database and a database management system (DBMS).

A database is a collection of related data that is organized to enable easy retrieval and maintenance.

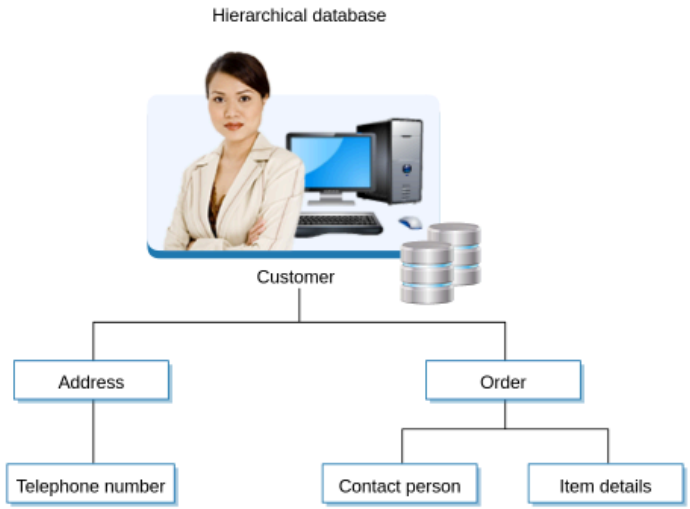
A DBMS is a set of software programs that manage the database and act as an interface to the data, insulating the user from the physical storage of the data and protecting the integrity of the database.



The current major database technologies are:

- Hierarchical
- Network
- Relational
- Object oriented

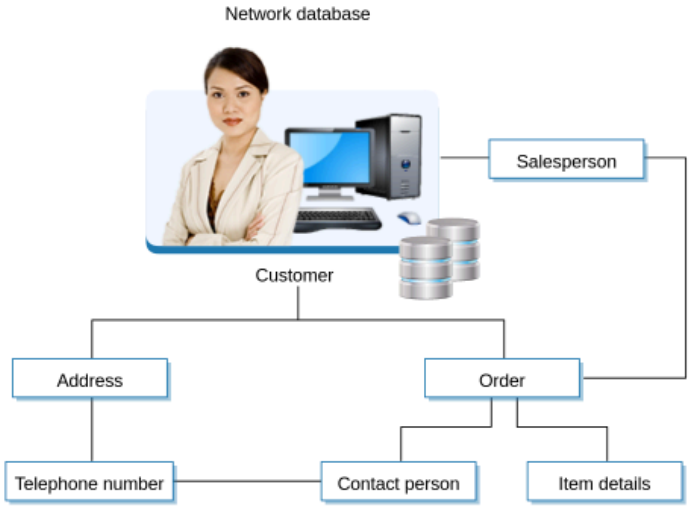
You will now take a closer look at these.



IBM's IMS DB is an example of a hierarchical database.

In this architecture, the relationships between data is defined in a definite hierarchical structure that reflects the business needs of the organization.

When the hierarchy has been defined and the data has been loaded, it is not easy to change the structure.



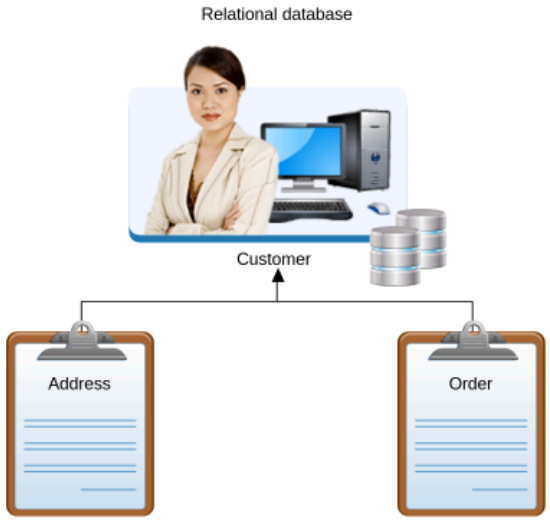
Network databases like IDMS cannot process relationships that are not explicitly included during database definition.

Each record in a network database usually has a combination of pointers associated with it, such as first or last records and next or prior pointers.

Object-oriented database

Object-oriented databases store data as objects. This corresponds to object-oriented design in programming where objects are independent and self-contained entities.

For performance reasons, the object model is often implemented as a relational model that is accessible through an object-relational interface.



A relational database consists of a collection of related tables that contain information that is necessary to the business. DBMS manages the database and acts as an interface between user and database. Queries used to extract or manipulate data can be run interactively or may be embedded in application programs written in PL/I, FORTRAN, assembly language, or REXX.

Relational Data and databases can also be accessed using object orientated languages such as OO COBOL, C++, Java, and web development languages such as ASP and PHP.

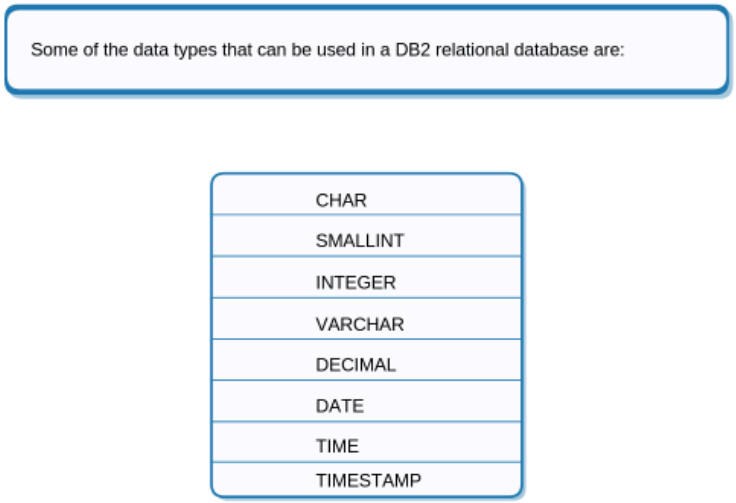
The programs are written by application programmers and users.

Structure of a table	
023883	Accounts
344565	R & D
233198	Payroll
789786	Personnel

Data in a relational database is stored in tables. A table can be thought of as a flat file with rows instead of records and columns instead of fields.

Each column has a data type associated with it. The length of the column and the data type are part of the table definition. Every row in a table has the same structure as every other row. This means that all rows have the same number of columns, and columns in the same position have the same data type.

In fact, it is impossible to put rows with different columns in the same table because the column definitions or data types, and the positions of the columns in the table, are part of the table definition.

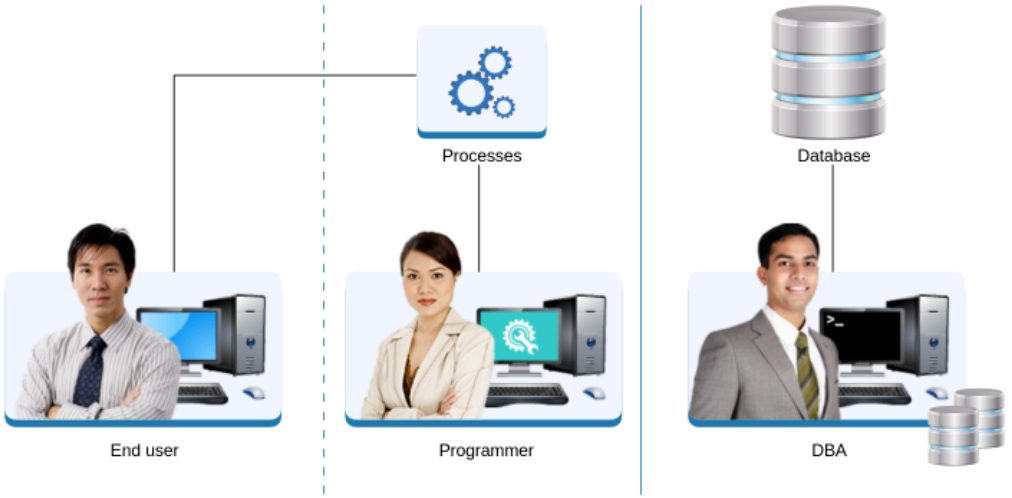


The column definitions, including data types, form part of the table definition. Some of the data types available for use in Db2 relational databases are shown above.

Data type	COBOL	PL/I
CHAR(n)	PIC X(n)	CHAR(n)
SMALLINT	PIC S9(4) COMP	FIXED BIN (15)
INTEGER	PIC S9(9) COMP	FIXED BIN (31)
VARCHAR(n)	01 VAR-NAME. 49 VAR-LEN PIC S9(4) USAGE BINARY. 49 VAR-TEXT PIC X(n).	CHAR(n) VAR
DECIMAL(p,s)	PIC S9(p-s)V9(s) COMP-3	FIXED DEC(p,s)
DATE	PIC X(10)	CHAR (10)
TIME	PIC X(8)	CHAR(8)
TIMESTAMP	PIC X(26)	CHAR(26)

These data types are defined and maintained by Db2, and their structure is independent of the environment or language accessing them.

In some cases, there is no exact match for a Db2 data type in a programming language so a structure or character is used. For example, the dates and times shown for COBOL and PL/I.

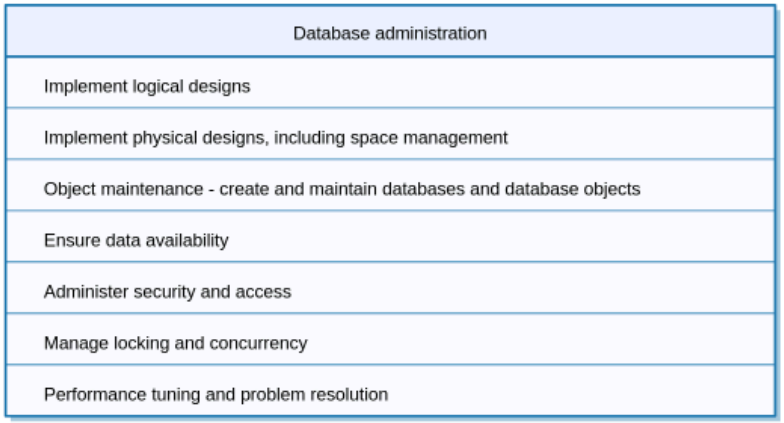


There are many security and maintenance functions to perform when dealing with a large and important system like Db2.

End users usually have an organized view of data, which is provided by applications. Applications are made by programmers, who write programs to present and manipulate Db2 data in a logical way.

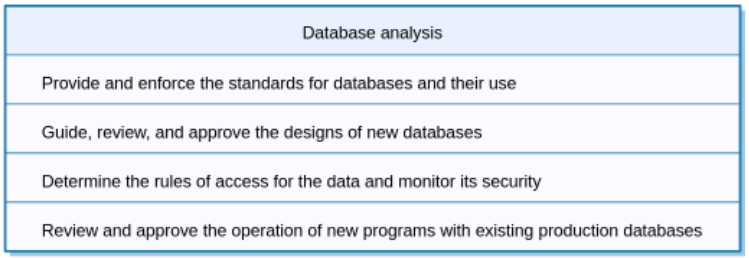
In most large organizations, the data and the database itself is managed by specialists called database administrators (DBA).

Many of the functions that are discussed in this course are only allowed to be performed on an Enterprise database by a DBA. We will therefore show some concepts and activities using the

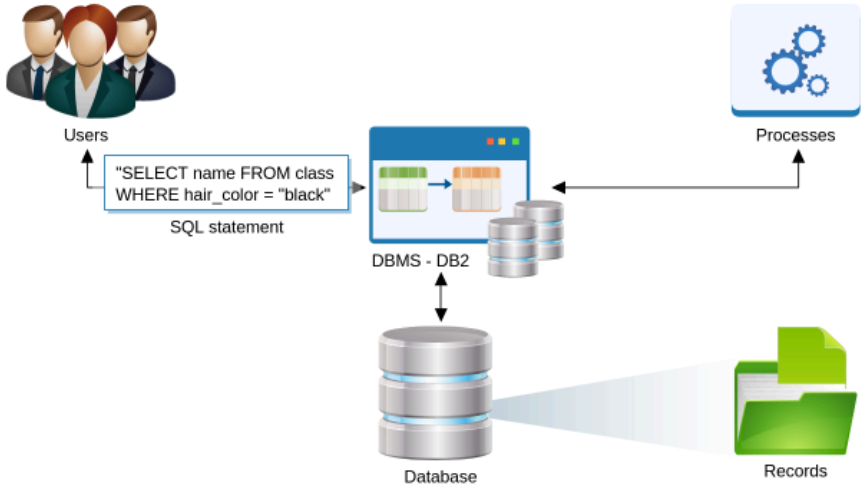


The DBA must administer the database system to ensure its ongoing operation and the security of the data.

Listed above are some of the tasks involved in the implementation, administration, and support of databases.



The DBA must also provide expert analysis and make recommendations relating to performance, which help programmers and end users access and use data in the most efficient way.

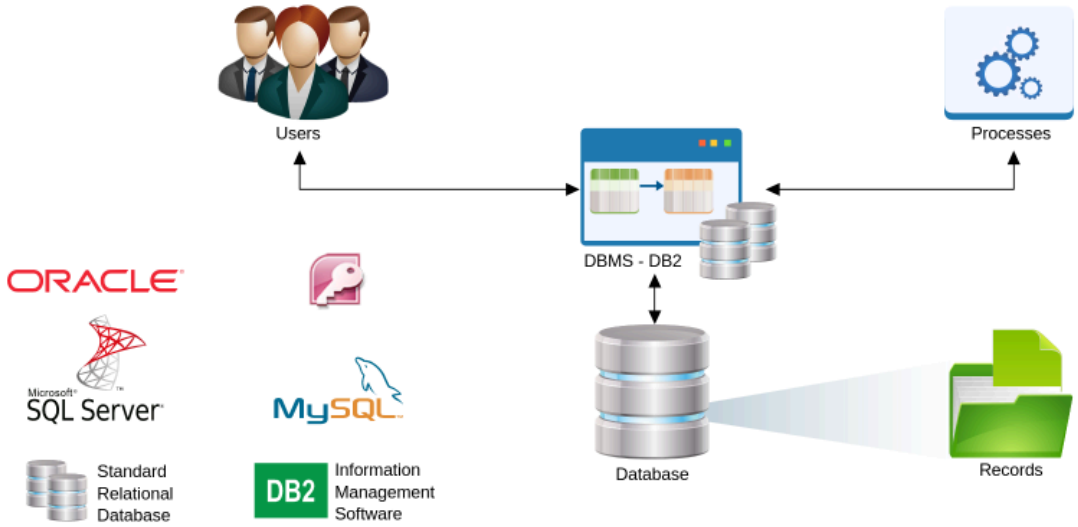


In relational databases like Db2, the user or external process does not access data directly; they send commands or requests to the DBMS - Db2 which then returns answers or results. The language used to construct or define these commands is called Structured Query Language (SQL).

SQL is designed to read like a sentence. For example, "Select names from employees where employment type is casual" would be written in SQL as:

```
SELECT name FROM employees WHERE emp_type = "casual"
```

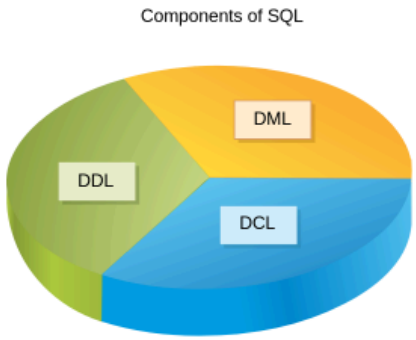
Click **Play** to see how SQL interfaces with a database.



SQL is a standardized language for relational databases and IBM has implemented the standard thoroughly. Most SQL written for Db2 can be run against other standard relational databases containing similarly defined data with little change.

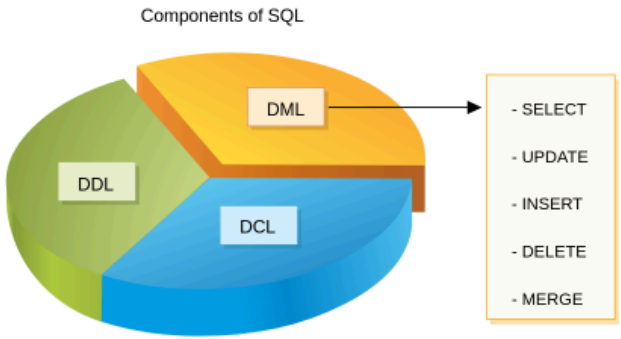
This standardization means SQL and database systems can be prototyped on personal databases, or moved from other environments and then moved to Db2 to take advantage of Db2 implementation for high-volume and system integrity.

Other databases that Db2 SQL can be run against are shown above.



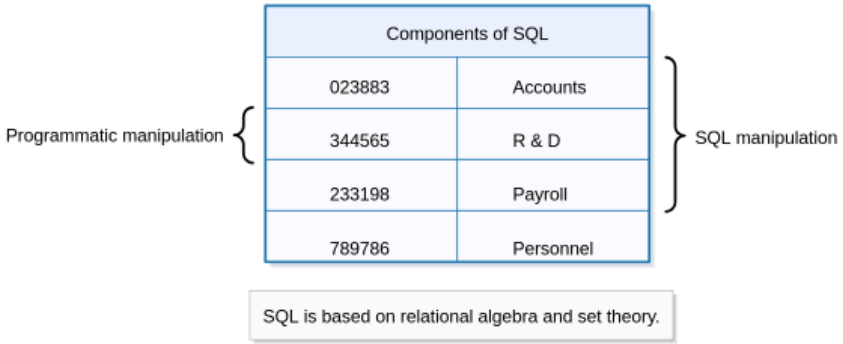
As the only interface to Db2, SQL must provide the ability to completely control the data and database. SQL consists of:

- Data Manipulation Language (DML), which enables data to be inserted, extracted, and so forth from tables
- Data Definition Language (DDL), which enables the database objects to be created
- Data Control Language (DCL), which enables authorities over data to be controlled



Most people are introduced to SQL when they use the DML component to access and update data in an existing database. DML generally consists of the following five statements:

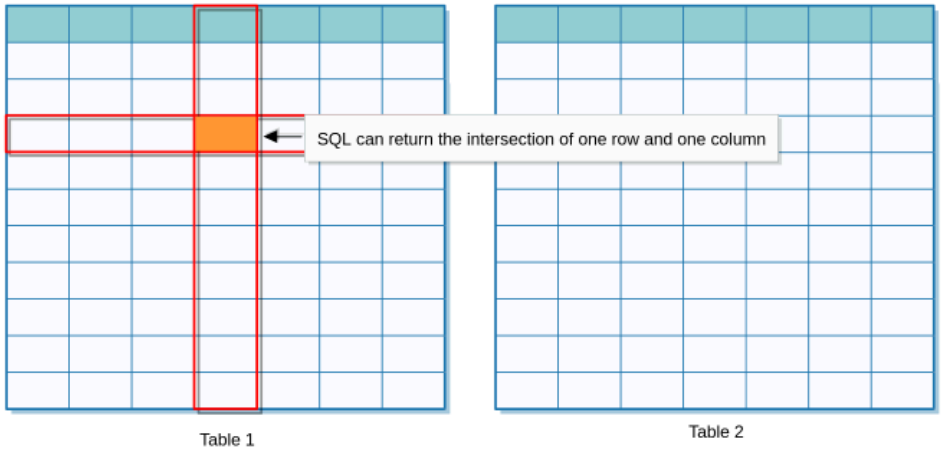
- SELECT
- UPDATE
- INSERT
- DELETE
- MERGE



It is important to remember that SQL is a set language, that is, a result set containing zero or any number of rows that can be manipulated.

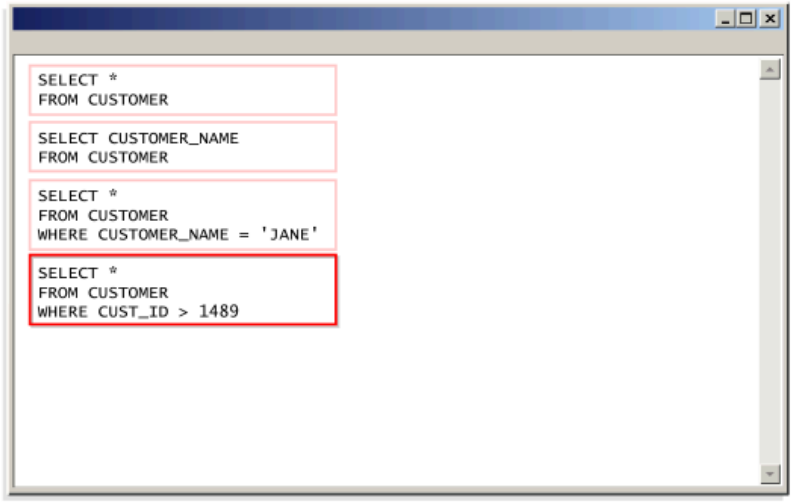
This is one of the major differences between SQL and conventional programming languages like COBOL or BASIC in which you usually deal with one record at a time.

In SQL, you deal with a set of rows.



The sets of data manipulated or returned by SQL can range from nothing at all in the simplest case to the intersection of one row and column, to a whole table, to the join or union of many tables.

Click **Play** to see a demonstration of this concept.



All columns are returned, selecting the rows of the table where the customer ID is greater than 1489 from the table titled CUSTOMER.

The examples shown here represent simple SQL code used to obtain specific database data.

Mouse-over the code to discover what it will do.

Summary

Introduction to Db2 Relational Databases and SQL

In this module you looked at the types of databases commonly used and focused on the workings of the relational database structure. You were also introduced to SQL, which is the language used to interrogate data stored in a relational database management system (RDMS).

You should now be able to identify:

- How a Relational Database is Structured
- The Tasks Performed by the Database Administrator (DBA)
- The Purpose and General Structure of SQL Statements