

Reference: FUNDAMENTALS OF DATABASE SYSTEMS, 5th Edition, George J. Elmasri & Shampoon Chhabra, Prentice-Hall, Englewood Cliffs, New Jersey, 2003.
Database Query Language
Engineering Institute of Technology at Arlington

Database Query Languages

DBMS Languages

Once the design of a database is completed and a DBMS is chosen to implement the database, the first step is to specify conceptual and internal schemas for the database and any mappings between the two.

In many DBMSs where no strict separation of levels is maintained, one language, called the data definition language (DDL), is used by the DBA and by database designers to define both schemas.

In DBMSs where a clear separation is maintained between the conceptual and internal levels, the DDL is used to specify the conceptual schema only. Another language, the storage definition language (SDL), is used to specify the internal schema.

In most relational DBMSs today, there is no specific language that performs the role of SDL. Instead, the internal schema is specified by a combination of functions, parameters, and specifications related to storage.

For a true three-schema architecture, we would need a third language, the view definition language (VDL), to specify user views and their mappings to the conceptual schema, but in most DBMSs the DDL is used to define both conceptual and external schemas.

Once the database schemas are compiled and the database is populated with data, users must have some means to manipulate the database. Typical manipulations include retrieval, insertion, deletion, and modification of the data. The DBMS provides a set of operations or a language called the data manipulation language (DML) for these purposes.

In current DBMSs, the preceding types of languages are usually not considered distinct languages; rather, a comprehensive integrated language is used that includes constructs for conceptual schema definition, view definition, and data manipulation.

Storage definition is kept separate, since it is used for defining physical storage structures to fine-tune the performance of the database system, which is usually done by the DBA staff.

A typical example of a comprehensive database language is the SQL relational database language, which represents a combination of DDL, VDL, and DML, as well as statements for constraint specification, schema evolution, and other features.

The SDL was a component in early versions of SQL but has been removed from the language to keep it at the conceptual and external levels only.

There are *two main types of DMLs*.

A high-level or nonprocedural DML. Many DBMSs allow high-level DML statements *either* to be entered interactively from a display monitor or terminal *or* to be embedded in a general-purpose programming language.

A low-level or procedural DML must be embedded in a general-purpose programming language. This type of DML retrieves individual records or objects from the database and processes each separately. Therefore, it needs to use programming language constructs, such as looping, to retrieve and process each record from a set of records. Low-level DMLs are also called record-at-a-time DMLs because of this property.

DL/1, a DML designed for the hierarchical model, is a low-level DML that uses commands such as GET UNIQUE, GET NEXT, or GET NEXT WITHIN PARENT to navigate from record to record within a hierarchy of records in the database.

High-level DMLs, such as SQL, can specify and retrieve many records in a single DML statement; therefore, they are called set-at-a-time or set-oriented DMLs.

A query in a high-level DML often specifies which data to retrieve rather than how to retrieve it; therefore, such languages are also called declarative.

Whenever DML commands, are embedded in a general-purpose programming language, that language is called the host language and the DML is called the data sublanguage.

On the other hand, a high-level DML used in a standalone interactive manner is called a query language.

End users use a high-level query language to specify their requests, whereas programmers use the DML in its embedded form.

For naive users, there usually are user- interfaces for interacting with the database; these can also be used by casual users or others who do not want to learn the details of a high-level query language.