Path Wala

11 Relational Databases

In This Chapter

11.1 Introduction
11.2 Purpose of DBMS
11.3 Relational Database Model
11.4 The Relational Model Terminology
11.5 Brief History of MySQL
11.6 MySQL Database System
11.7 Starting MySQL
11.8 MySQL and SQL

11.1 INTRODUCTION

A database system is basically a computer based record keeping system. The collection of data, usually referred to as the *database*, contains information about one particular enterprise. It maintains any information that may be necessary to the decision-making processes involved in the management of that organization. A database may also be defined as a collection of interrelated data stored together to serve multiple applications ; the data is such stored so that it is independent of programs which use the data ; a common and controlled approach is used in adding new data and in modifying and retrieving existing data within the database. The data is structured so as to provide a foundation for future application development. The intention of a database is that the same collection of data should serve as many applications as possible. Hence, a database is often conceived of as the repository of information needed for running certain functions in a corporation or organization. Such a database would permit not only the retrieval of data but also the continuous modification of data needed for control of operations. It may be possible to search the database to obtain answers to queries or information for planning purposes.

DATABASE

A **database** may be defined as a collection of interrelated data stored together to serve multiple applications.

Path Wala

11.2 PURPOSE OF DBMS

A database system should be a repository of the data needed for an organization's data processing. That data should be accurate, private, and protected from damage. It should be organized so that diverse applications with different data requirements can employ the data. Different application programmers and different end users will have different views of the data which must be derived from a common overall data structure. Their methods of accessing or searching the data will differ.

DBMS

A **DBMS (Data Base Management System)** refers to a software that is responsible for storing, maintaining and utilizing databases. A database along with a DBMS is referred to as a **database system**.

The ways in which end users want to utilize existing data will constantly change, and in some cases demands for new uses of the data will arise rapidly and urgently. The extent to which these demands can be satisfied determines the overall value of the database system.

In a typical file-processing system, permanent records are stored in various files. A number of different application programs are written to extract records from and add records to the appropriate files. But this scheme has a number of major limitations and disadvantages, such as *data redundancy* (duplication of data), *data inconsistency* (multiple copies of same data but not matching with one another), *unsharable data, unstandardized data, insecure data, incorrect data* etc. A *database management system* is answer to all these problems as it provides a centralized control of the data, thereby minimizing these problems in the database.

DATA REDUNDANCY

Duplication of data is known as **Data Redundancy**.

DATA INCONSISTENCY

Mismatched multiple copies of same data, is known as **Data Inconsistency**.

11.3 RELATIONAL DATABASE MODEL

In relational data model, the data is organized into tables (*i.e.*, rows and columns). These tables are called *relations*. A row in a table represents a *relationship* among a set of values. Since a table is a collection of such relationships, it is generally referred to using the mathematical term *relation*, from which the relational data model derives its name.

Let us see how a sample database can be represented in relational form. The sample database being shown here has three tables (*relations*) : *Suppliers, Items, Shipments*.

Suppliers (SuppNo, Supp_Name, Status, City)

Items (ItemNo, Item_Name, Price)

Shipments (SuppNo, ItemNo, Qty_supplied) [coloured fields represent *primary keys*[1]]

Each of these three tables closely resembles a conventional sequential file, with rows of the table corresponding to records of the file and columns corresponding to fields of the records. Each of these tables is actually a special case of the construct known in mathematics as a *relation*. (More about *relations* and their characteristics, we shall learn under section 11.4.)

Rows of *relations* are generally referred to as *tuples* and the columns are usually referred to as *attributes*.

1. Primary-key is the key-field that can uniquely identify a row in a relation.

Path Wala

Suppliers

SuppNo	Supp_Name	Status	City
S1	Britannia	10	Delhi
S2	New Bakers	30	Mumbai
S3	Mother Dairy	10	Delhi
S4	Cookz	50	Bangalore
S5	Haldiram	40	Jaipur

Items

ItemNo	Item_Name	Price
I1	Milk	15.00
I2	Cake	5.00
I3	Bread	9.00
I4	Milk Bread	14.00
I5	Plain Biscuit	6.00
I6	Cream Biscuit	10.00
I7	Ice Cream	16.00
I8	Cold Drink	8.00
I9	Namkeen	15.00

Shipments

SuppNo	ItemNo	Qty_supplied
S1	I2	10
S1	I3	20
S1	I6	20
S2	I4	20
S2	I5	10
S3	I1	10
S3	I7	10
S4	I8	30
S5	I9	30

Figure 11.1 Sample Database in Relational Form.

Notice, that here each supplier has a unique supplier number, exactly one name, status value and location. Likewise, we assume that each item has a unique *item number, name, price* ; and also that, at any given time, no more than one shipment exists for a given supplier/item combination.

Components of a Table

Let us quickly have a look at various components of a table.

These are :

Byte A byte is a group of eight bits and is used to store a character.

Data Item (Field) A data item is the smallest unit of named data. It may consist of any number of bits or bytes. A data item represents one type of information and is often referred to as a **field** or data element.

Record A record is a named collection of data items which represents a complete unit of information.

Table A table is a named collection of all occurrences of a given type of logical record.

Path Wala

Figure 11.2 explains all the above mentioned terms.

Fields are the columns containing one type of information.

Records are the rows representing one complete unit of information (logically related information). *A logical record* contains contents of all the fields.

Table is a group of all the logically related records.

Primary key ↓ Fields

SalesmanNumber	First Name	Surname	Sales made	Target
EA-01	VINEET	ARORA	25000	29000
EA-05	SUNILA	LUNA	16000	20000
EA-12	KAPIL	KUMAR	23000	21000
NO-10	BHUPESH	ARORA	15000	15000
NO-11	SHILPI	SINGH	18000	16000
NO-32	SANDEEP	SETHI	51000	50000
SO-09	SUBHASH	KUMAR	22000	20000
SO-11	ANAND	SWAMI	25000	25000
SO-17	VENKET	SWAMI	19000	20000
WE-17	VIJAY	SHAH	33000	32000
WE-19	ANUJ	SHARMA	17000	25000
WE-21	VANDANA	PAHWA	18000	18000

Record

Requires 5 bytes

↑ See, primary key field has unique value for each row.

Data item

Requires 4 bytes

Figure 11.2 Components of a table.

In a table, each record is identified with a unique value *i.e.*, in a field of table, there are unique values for all records. This unique value identifies the record. For instance in Fig. 11.2, the field **SalesmanNumber** has unique values for all the records. So **SalesmanNumber** is the primary key here.

In a table, there may be more than one field having unique values. In such a case, only one of them is declared as *primary key*.

The relational data model is based on a collection of tables (relations). The user of the (relational) database system may query these tables, insert new tuples (record/row), delete tuples, and modify tuples. There are several languages for expressing these operations. Two such languages are relational query language and relational algebra.

Examples of common Relational Database Management Systems include ***Oracle database, Microsoft SQL Server, MYSQL, IBM DB2, IBM Informix, SAP Sybase Adaptive Server Enterprise, SAP Sybase IQ, Teradata, PostgreSQL, SQLite*** etc.

Path Wala

Out of these, ***MySQL, PostgreSQL*** and ***SQLite*** are Open Source implementations.

MySQL is a relational dbms that can run on virtually all platforms, including *Linux, Unix* and *Windows*. Popular for web-based applications and online publishing, MySQL is part of open-source enterprise stack LAMP (Linux, Apache, MySQL, PHP).

SQLite is a relational database management system. SQLite is not a client–server database engine, rather, it is embedded into the end program. It is arguably the most widely deployed database engine, as it is used today by several widespread browsers, operating systems, and embedded systems (such as mobile phones), among others.

PostgreSQL is a general purpose and object-relational database management system, the most advanced open source database system. PostgreSQL is free and open source software. Its source code is available under PostgreSQL license, a liberal open source license.

Common database management tools for mobile devices include ***SQL Anywhere, DB2 Everywhere, IBM Mobile Database, SQL Server Compact, SQL Server Compact, SQL Server Express, Oracle DatabaseLite, SQLite, SQLBase*** etc. Out of these *SQLite* is a public domain Open Source implementation.

There are some other database models namely *network data model, hierarchical data model* and *object oriented data model*. But covering these here is beyond the scope of the book.

11.4 THE RELATIONAL MODEL TERMINOLOGY

The relational model was propounded by *E.F. Codd* of the IBM and has since been acknowledged as a very important concept in DBMS (Data Base Management Systems) technology. Let us explore this model in details. Different terms used in the relational model are being discussed here.

Relation. In general, a relation is a table *i.e.*, data arranged in rows and columns. A relation has the following properties :

1. In any given column of a table, all items are of the same kind whereas items in different columns may not be of the same kind.
2. For a row, each column must have an atomic (indivisible) value and also for a row, a column cannot have more than one value.
3. All rows of a relation are distinct. That is, a relation does not contain two rows which are identical in every column. That is, each row of the relation can be uniquely identified by its contents.
4. There is no order maintained for rows inside a relation. That is, we cannot retrieve any thing by saying that from row number 5, column *name* is to be accessed.
5. The columns of a relation are assigned distinct names and the ordering of these columns is immaterial.

Domain. A domain is a pool of values from which the actual values appearing in a given column are drawn. For example, the values appearing in the **SuppNo** column of both the ***Suppliers*** table and the ***Shipments*** table are drawn from the underlying domain of all valid supplier numbers. (*See* Fig. 11.3).

NOTE

A crucial feature of relational data model is that associations between rows are represented solely by data values in columns drawn from a common domain.

Path Wala

A domain is said to be *atomic* if elements of the domain are considered to be indivisible units. For example, the set of integers is an atomic domain but the set of all sets of integers is a non-atomic domain.

Shipments

SuppNo	ItemNo	Qty_supplied
S1	I2	10
S1	I3	20
S1	I6	20
S2	I4	20
S2	I5	10
S3	I1	10
S3	I7	10
S4	I8	30
S5	I9	30

Suppliers

SuppNo	Supp_Name	Status	City
S1	Britannia	10	Delhi
S2	New Bakers	30	Mumbai
S3	Mother Dairy	10	Delhi
S4	Cookz	50	Bangalore
S5	Haldiram	40	Jaipur

Figure 11.3 SuppNo in Suppliers and Shipments draw values from same domain.

In the sample database, observe that relations ***Suppliers*** and ***Shipments*** have a common domain (supplier numbers domain) and so do ***Items*** and ***Shipments*** (Item numbers domain).

See that in the table ***Shipments*** the supplier **S1** is supplying **10** units of item **I2** and from the table ***Suppliers*** it can be said that a **Delhi** based supplier **Britannia** is supplying **10** units of item **I2**. That is how the association between rows (*S1, I2, 10* of ***Shipments*** and *S1, Britannia, 10, Delhi* of ***Suppliers***) can be represented.

Tuple. The rows of tables (relations) are generally referred to as *Tuples* (usually pronounced to rhyme with "couples").

Attributes. The columns of tables (relations) are generally referred to as *attributes*.

Degree. The number of attributes in a relation determine the *degree* of a relation. A relation having 3 attributes is said to be a relation of degree 3. Similarly, a relation having *n* attributes is said to be a relation of degree *n*. Relations of degree one are said to be *unary*, relations of degree two are *binary*, relations of degree three are *ternary*, ..., and relations of degree *n* are *n-ary*.

Cardinality. The number of tuples (rows) in a relation is called the *cardinality* of the relation ; *e.g.*, the cardinality of ***Shipments*** relation is 9. Similarly, the cardinality of ***Suppliers*** relation is 5.

TUPLES, ATTRIBUTES

The rows and columns of a relation are known as **Tuples** and **Attributes** respectively. The number of attributes in a relation is called **Degree** and the number of rows in a relation is known as **Cardinality**.

Path Wala

11.4.1 Views

A view is a kind of table whose contents are taken from other tables depending upon a condition. Views do not contain data of their own. The contents of a view are determined by carrying out the execution of the given query (the given condition).

> **VIEW**
>
> A **View** is a (virtual) table that does not really exist in its own right but is instead derived from one or more underlying base table(s).

Let us elaborate the concept.

The kinds of tables (relations) that you have come across with until now are called ***base tables***. These are the tables that actually contain data. There is another kind of tables, the views. *Views* are tables whose contents are derived from other tables (their *base tables*) everytime they are referred to. There is no stored file created for storing the contents of a view. Rather only the definition of a view is stored. Everytime a view is referred to, its contents are derived from its underlying base table(s) using its stored definition. Thus, a view is also referred to as a *virtual table*.

Following Fig. 11.4 explains the concept of a view. Views are like windows through which you view desired information that is actually stored in a base table.

Definition of the view

```
CREATE VIEW GoodItems AS        (name of the view)
SELECT * FROM Items             (name of the base table)
WHERE Price >12 ;               (the condition)
```

GoodItems (A Virtual Table based on **Items** table)

ItemNo	*Item_Name*	*Price*
I1	Milk	15.00
I4	Milk Bread	14.00
I7	Ice Cream	16.00
I9	Namkeen	15.00

Figure 11.4 The concept of Views.

A view can be used just like any other table. It can be queried, updated, inserted into, deleted from, and joined with other tables and views.

Views greatly extend the control you have over your data. They are an excellent way to give people access to some but not all of the information in a table. Even *Read-only* views can also be created which means that they can be queried, but they cannot be subjected to update commands. (Though there is a lot that can be discussed about views yet we are not going further in explaining views as it is beyond the scope of this chapter and the book.)

11.4.2 Structure of Relational Databases

Consider the ***Items*** table of our sample database. It has three attributes : *ItemNo, Item_Name,* and *Price*. For each attribute, there is a set of permitted values, called the *domain* of that attribute.

Path Wala

For the attribute *Item_Name*, for example, the domain is the set of all item names. Let *D2* be this set (of item names). *D1* be the set of item numbers, and let *D3* be set of prices. Now any row of ***Items*** must be a 3-type (*i.e.*, must have values for 3 attributes), say (*x1, x2, x3*) where *x1* is an item number, *x2* is the item name and *x3* is the *price* of the item. That means *x1* is in domain *D1*, *x2* is in domain *D2*, and *x3* is in domain *D3*. In general, the table ***Items*** will contain only a subset of the set of all possible rows. Therefore, ***Items*** is a subset of

$$D1 \times D2 \times D3$$

In general, a table having *n* attributes must be a subset of

$$D1 \times D2 \times \ldots\ldots \times D_{n-1} \times D_n$$

11.4.2A Keys

It is important to be able to specify how rows in a relation are distinguished conceptually, rows are distinct from one another, but from a database perspective the difference among them must be expressed in terms of their attributes. *Keys* come here for a rescue.

Primary Key

Within a given relation, a *set of one or more attributes* having values that are unique within the relation and thus are able to *uniquely identify that tuple*, is said to be the *primary key* of the relation.

Every relation does have a primary key. In our sample database, **SuppNo** is the primary key for ***Suppliers*** as it contains unique value for each tuple in the relation. Similarly, **ItemNo** is the primary key for ***Items*** and the combination of **SuppNo and ItemNo** is the primary key for the ***Shipments*** relation.

In some tables, combination of more than one attribute provides a unique value for each row. In such tables, the group of these attributes is declared as primary key. In such cases, the primary key consists of more than one attribute, it is called *composite- primary-key*.

PRIMARY KEY

A **Primary Key** is a set of one or more attributes that can uniquely identify tuples within the relation.

The primary key is non-redundant *i.e.* it does not have duplicate values in the same relation. The non-primary-key attributes of a table can be referred to as *non-key* attributes.

Candidate Key

Occasionally we may encounter a relation in which there is more than one attribute possessing the unique identification property.

In our sample database, there are two candidate keys **SuppNo** and **Supp_Name** in the ***Suppliers*** relation. Both of these attributes contain unique values for each tuple. Similarly, in ***Items***, **ItemNo** and **Item_Name** are candidate keys.

CANDIDATE KEYS

All attribute combinations inside a relation that can serve as primary key are **Candidate Keys** as they are candidates for the primary key position.

In the case of two or more candidate keys, the database analyst decides one of them as the primary key for the relation.

Alternate Key

In case of two or more candidate keys, only one of them serves as the primary key. The rest of them are *alternates* only.

In ***Suppliers*** table, **Supp_Name** is the alternate key and in ***Items*** table **Item_Name** is the alternate key.

ALTERNATE KEY

A candidate key that is not the primary key is called an **Alternate Key.**

Path Wala

Foreign Key

A foreign key is used to represent the relationship between two tables. A **foreign key** is a non-key attribute (or a group of non-key attributes) whose value is derived from the *primary-key* of another table. Or in other words, a non-key attribute of a table, which is the primary-key of some other table is known as **foreign-key**.

The table in which this non-key attribute *i.e.*, the foreign-key attribute exists, is called a **Foreign table** or **Detail table**, and the table that defines the *Primary-key*, which the *foreign-key* of *detail-table* refers to, is called **Primary table** or **Master table**.

FOREIGN KEY

A non-key attribute, whose values are derived from the primary key of some other table, is known as **Foreign Key** in its current table.

Referential Integrity

A referential integrity is a system of rules that a DBMS uses to ensure that relationships between records in related tables are valid, and that users don't accidentally delete or change related data.

REFERENTIAL INTEGRITY

Referential integrity is a system of rules that a DBMS uses to ensure that relationships between records in related tables are valid, and that users don't accidentally delete or change related data.

You can set referential integrity when all of the following conditions are met :

- The matching field from the primary table is a primary key or has a unique index.
- The related fields have the same data type.
- Both tables belong to the same database. If the tables are linked tables, they must be of same DBMS format, and you must open the database in which they are stored to set referential integrity. Referential integrity can't be enforced for linked tables from databases in other formats.

When referential integrity is enforced, you must observe the following rules :

- You can't enter a value in the foreign key field of the related table that doesn't exist in the primary key of the primary table. However, you can enter a Null value in the foreign key, specifying that the records are unrelated. For example, you can't have an order that is assigned to a customer that doesn't exist, but you can have an order that is assigned to no one by entering a **Null** value in the CustomerID field.
- You can't delete a record from a primary table if matching records exist in a related table. For example, you can't delete an employee record from the Employees table if there are orders assigned to the employee in the Orders table.
- You can't change a primary key value in the primary table, if that record has related records. For example, you can't change an employee's ID in the Employees table if there are orders assigned to that employee in the Orders table.

11.5 BRIEF HISTORY OF MYSQL

MySQL is a freely available open source *Relational Database Management System* (RDBMS) that uses Structured Query Language (SQL). It is down-loadable from site www.mysql.org In a MySQL database, information is stored in *Tables*. A single MySQL database can contain many tables at once and store thousands of individual records. MySQL provides you with a rich set of features that support a secure environment for storing, maintaining, and accessing data. MySQL is a fast, reliable, scalable alternative to many of the commercial RDBMSs available today.

Path Wala

MySQL was created and is supported by *MySQL AB*, a company based in Sweden (www.mysql.com). This company is now a subsidiary of *Sun Microsystems*, which holds the copyright to most of the codebase. On April 20th, 2009 *Oracle Corp.*, which develops and sells the proprietary *Oracle database*, announced a deal to acquire *Sun Microsystems*.

The chief inventor of MySQL was *Michael Widenius* (a.k.a. *Monty*). MySQL has been named after Monty's daughter *My*. The logo of MySQL, the dolphin, is named as *"Sakila"*.

11.6 MYSQL DATABASE SYSTEM

The key role of a database management system is information management. A database server is the key to solving the problems of information management. In general, a server must reliably manage a large amount of data in a multi-user environment so that many users can concurrently access the same data. A database server must also prevent unauthorized access and provide efficient solutions for failure recovery.

MySQL database system refers to the combination of a MySQL server instance and a MySQL database. MySQL operates using client/server architecture in which the server runs on the machine containing the databases and clients connect to the server over a network. The server operating systems is usually a *Linux* (like *Redhat 9.0* etc.) or *Windows* operating system. Typically MySQL is supported on *Windows XP, Windows Server 2003, Red Hat Fedora Linux*, and *Debian Linux*, and others. As with any other client/server application, MySQL is a multi-user database system, meaning several users can access the database simultaneously.

Here :

- **The server** (MySQL server) listens for client requests coming in over the network and accesses database contents according to those requests and provides that to the clients.
- **Clients** are programs that connect to the database server and issue queries in a pre-specified format. MySQL is compatible with the standards based SQL (Structured Query Language). The client program may contact the server programmatically (meaning a program call the server during execution) or manually. For example, when you are issuing commands over a session to a MySQL server, you are issuing the requests to the server by typing commands at your command prompt manually. On the other hand, if you have input some data (say your credit card information on the Internet towards purchase of some goods) in a form, and the form is processed by using a server side program, then the MySQL server is contacted programmatically. This is often the case in credit card approvals, member subscriptions etc.

Some of the key features of MySQL are :

1. ***Speed.*** If the server hardware is optimal, MySQL runs very fast. It supports clustered servers for demanding applications.
2. ***Ease of use.*** MySQL is a high-performance, relatively simple database system. From the beginning, MySQL has typically been configured, monitored, and managed from the command line. However, several MySQL graphical interfaces are also available.
3. ***Cost.*** MySQL is available free of cost. MySQL is a "Open Source" database. MySQL is part of LAMP (Linux, Apache, MySQL, PHP / Perl / Python) environment, a fast growing open source enterprise software stack. More and more companies are using LAMP as an alternative to expensive proprietary software stacks because of its lower cost, reliability, and documentation.

Path Wala

4. *Query Language Support.* MySQL understands standards based SQL (Structured Query Language).
5. *Portability.* MySQL provides portablility as it has been tested with a broad range of different compilers and can work on many different platforms. It is fully multi-threaded using kernel threads. It can easily use multiple CPUs if they are available.
6. *Data Types.* MySQL provides many data types to support different types of data. It also supports fixed-length and variable-length records.
7. *Security.* MySQL offers a privilege and password system that is very flexible and secure, and that allows host-based verification. Passwords are secure because all password traffic is encrypted when you connect to a server.
8. *Scalability and Limits.* MySQL can handle large databases. Some real life MySQL databases contain 50 million records, some have up to 60,000 tables and about 5,000,000,000 rows.
9. *Connectivity.* Clients can connect to MySQL Server using several protocols:
10. *Localization.* The server can provide error messages to clients in many languages.
11. *Clients and Tools.* MySQL provides several client and utility programs. These include both command-line programs such as **mysqldump** and **mysqladmin**, and graphical programs such as **MySQL Administrator** and **MySQL Query Browser**. MySQL Server has built-in support for SQL statements to check, optimize, and repair tables.

11.7 STARTING MYSQL

Starting MySQL is similar to the way you start other applications in Windows platform. Make sure that MySQL Server is installed on your machines. Once it is installed, you need to click at sure that MySQL Server is installed on your machines. Once it is installed, you need to click at **Start → All Programs → MySQL → MySQL Server → MySQL Command Line Client** (see figure below)

It will start the MySQL client where you have to specify the password. Ask your teacher about the password for the MySQL installation in your lab.

Path Wala

Now you can start working in MySQL. When you are through with your work, you can quit from MySQL by typing **Quit** at the **mysql>** prompt (see fig below).

```
C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.1.33-community MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> quit
```

11.8 MYSQL AND SQL

In order to access data within the MySQL database, all programs and users must use, *Structured Query Language* (SQL). SQL is the set of commands that is recognised by nearly all RDBMSs.

Usage of SQL has become a standard for most of RDBMSs. Though application programs and *MySQL* tools often allow users to access database without directly using SQL, but these applications in turn must use SQL when executing the user's request.

The Structured Query Language (SQL) is a language that enables you to create and operate on relational databases, which are sets of related information stored in tables.

The database world is becoming increasingly integrated, and this led to a clamour for a standard language that can be used to operate in many different kinds of computer environment. The *SQL* (Structured Query Language) has proved to be a standard language as it allows users to learn one set of commands and use it to create, retrieve, alter, and transfer information regardless of whether they are working on a PC, a workstation, a mini, or a mainframe.

There are numerous versions of *SQL*. The original version was developed at IBM's *San Jose Research Laboratory* (now the *Almanden Research Center*). This language, originally called *Sequel*, was implemented as part of the System R project in early 1970s. The *Sequel* language has evolved since then, and its name has changed to *SQL*. In 1986, the *American National Standards Institute* (*ANSI*) published an *SQL* standard that was updated again many times. Prominent updations include updations in years 1992, 2003 and 2008.

SQL has clearly established itself as the standard relational database language. In this chapter, we present a briefed survey of *SQL*.

11.8.1 Processing Capabilities of SQL

The *SQL* has proved to be a language that can be used by both casual users as well as skilled programmers. It offers a variety of processing capabilities, simpler ones of which may be used by the former and the more complex by the latter class of users. The various processing capabilities of *SQL* are :

1. ***Data Definition Language*** (**DDL**). The *SQL* DDL provides commands for defining relation schemas, deleting relations, creating indexes, and modifying relation schemas.

Path Wala

2. ***Interactive Data Manipulation Language*** *(DML).* The *SQL* DML includes a query language based on both the relational algebra and the tuple relational calculus. It includes also commands to insert, delete, and modify tuples in the database.
3. ***Embedded Data Manipulation Language.*** The embedded form of *SQL* is designed for use within general-purpose programming languages such as PL/1, COBOL, Fortran, Pascal, and C, C++ etc.
4. ***View Definition.*** The *SQL* DDL also includes commands for defining views.
5. ***Authorization.*** The *SQL* DDL includes commands for specifying access rights to relations and views.
6. ***Integrity.*** The *SQL* provides (limited) forms of integrity checking. Future products and standards of *SQL* are likely to include enhanced features for integrity checking.
7. ***Transaction Control.*** *SQL* includes commands for specifying the beginning and ending of transactions along with commands to have a control over transaction processing.

11.8.2 Data Definition Language

DATA DICTIONARY

A **Data Dictionary** is a file that contains "metadata" *i.e.*, "data about data".

A database scheme is specified by a set of definitions which are expressed by a special language called a *data definition language* (DDL). The result of compilation of DDL statements is a set of tables which are stored in a special file called *data dictionary* (or *directory*).

Whenever data is read or modified in the database system, the *data directory* is consulted.

The DDL provides a set of definitions to specify the storage structure and access methods used by the database system.

An ideal DDL should perform the following functions :

1. It should identify the types of data division such as *data item, segment, record,* and *data-base file.*
2. It should give a unique name to each *data-item-type, record-type, file-type, database,* and *other data subdivision.*
3. It should specify the proper data types.
4. It should specify how the record types are related to make structures.
5. It may define the type of encoding the program uses in the data items (*binary, character, bit, string,* etc.). This should not be confused with the encoding employed in physical representation.
6. It may define the length of the data items.
7. It may define the range of values that a data-item can assume.
8. It may specify means of checking for errors in the data.
9. It may specify privacy locks for preventing unauthorized reading or modification of the data.
10. A logical data definition should not specify addressing, indexing, or searching techniques or specify the placement of data on the storage units, because these topics are in the domain of physical, not logical, organization.

The *SQL* commands covered later in this unit, include DDL commands also.

Path Wala

11.8.3 Classification of SQL Statements

SQL, technically speaking, is a data sublanguage. That is, it is a language used to interact with database. In other words, all SQL statements are instructions to the database only. And that is where it differs from general-purpose programming languages like 'C' or 'C++' or 'BASIC' etc.

SQL provides many different types of commands used for different purposes. SQL commands can be divided into following categories :

(*i*) Data Definition Language (DDL) Commands

(*ii*) Data Manipulation Language (DML) Commands

(*iii*) Transaction Control Language (TCL) Commands

(*iv*) Session Control Commands

(*v*) System Control Commands.

Out of these, we shall not be talking about session control commands and system control commands as their discussion is beyond the scope of this book.

11.8.3A DDL Commands

The Data Definition Language (DDL) commands, as the name suggests, allow you to perform tasks related to data definition. That is, through these commands, you can perform tasks like :

(*i*) ***Create, alter, and drop schema objects.*** This section of DDL commands is used to create, or define, or change, or delete objects such as a *table*, a *view*, an *index* etc. CREATE commands are used to create schema objects, ALTER commands are used to modify or change the definition of already existing schema objects and DROP commands are used to delete or remove schema objects. Some examples of such DDL statements are : CREATE TABLE, ALTER TABLE, DROP TABLE, CREATE INDEX, ALTER INDEX, DROP INDEX, RENAME TABLE, TRUNCATE etc.

(*ii*) **Grant and revoke privileges and roles.** This section of DDL Commands is used to grant or revoke permissions or privileges to work upon schema objects. For example, a user (say scott) who creates a table (say *table1*) becomes the owner of the table. The owner of the table can allow others to work on his/her table. This can be achieved by GRANTing privilege for the same to others. To revoke already granted privileges, REVOKE command is used. This section of DDL commands is also known as *Data Control Language* (DCL) Commands.

(*iii*) **Maintenance commands.** This section of DDL commands is used to analyse information on a table with an aim of maintaining it. Examples of table maintenance commands are : ANALYZE TABLE, CHECK TABLE, REPAIR TABLE, RESTORE TABLE etc.

11.8.3B DML Commands

By data manipulation, we mean :

- the *retrieval* of information stored in the database.
- the *insertion* of new information into the database.
- the *deletion* of information from the database.
- the *modification* of data stored in the database.

DML

A **Data Manipulation Language** (DML) is a language that enables users to access or manipulate data as organized by the appropriate data model.

The DMLs are basically of *two* types :

(*i*) *Procedural DMLs* require a user to specify *what* data is needed and *how* to get it.

(*ii*) *Non-procedural DMLs* require a user to specify *what* data is needed without specifying how to get it.

Path Wala

The Data Manipulation Language (DML) commands, as the name suggests, are used to manipulate data. That is, DML commands query and manipulate data in existing schema objects. For example, to insert a tuple in a table, a DML statement (INSERT INTO) is used ; to modify a tuple in a table, another DML statement (UPDATE) is used ; to delete a tuple in a table, the DML statement DELETE is used. Other examples of DML commands are : SELECT, LOCK TABLE etc.

11.8.3C TCL Commands

As already mentioned, a *transaction* is one complete unit of work *e.g., preparing report-card for a student* is a transaction, but it involves many steps like – *open student's table, read data from it, open marks' table, read data from it, calculate percentage and grade, write the computed result to result file* and finally *close these three tables* (student table, marks table and result table).

A transaction is successfully completed (known as COMMIT) if and only if all its constituent steps are successfully completed. To manage and control the transactions, the *transaction control commands* are used. These commands manage changes made by DML commands.

Some examples of TCL commands are :

COMMIT	it makes all the changes made by statements issued, permanent
ROLLBACK	it undoes all changes since the beginning of a transaction or since a savepoint
SAVEPOINT	it marks a point upto which all earlier statements have been successfully completed and if required – in case of failure – one may undo the changes *i.e.*, rollback up to this very point
SET TRANSACTION	it establishes properties for the current transactions.

LET US REVISE

Check Point 11.1

1. Define the following terms :
   (*a*) relation (*b*) tuple
   (*c*) attribute (*d*) domain
   (*e*) primary key (*f*) candidate key
   (*g*) cartsian product (*h*) degree
2. What are views ? How are they useful ?
3. Define the following :
   (*i*) primary key (*ii*) candidate key
   (*iii*) alternate key (*iv*) foreign key.
4. What is an Alternate Key ?
5. What is the importance of a Primary Key in a table ? Explain with a suitable example.
6. What do you understand by the terms Primary Key and Degree of a relation in relational database ?
7. What do you understand by the terms Candidate Key and Cardinality of a relation in relational database ?

- *A collection of data is referred to as database and a database (management) system is basically a computer based record keeping system.*
- *Database systems help reduce data redundancy, data inconsistency and facilitate sharing of data, standardization of data and data security.*
- *A relational data model organizes the data into tables known as relations.*
- *A view is a virtual table derived from one or more underlying base tables.*
- *There is no stored file created for storing a view's contents, rather, only the view definition is stored.*
- *MySQL is a free, open-source Relational Database Management System.*
- *A MySQL database system consists of a MySQL server instance and a MySQL database.*
- *Some key features of MySQL are : portability, secure system, scalable and supports localization.*
- *SQL statements can be classified into : Data Definition Language (DDL), Data Manipulation Language (DML), Transaction Control Language (TCL), Session Control Language and System Control Language.*

Path Wala

- *The DDL commands are used to define or redefine schema objects.*
- *The DML commands are used to manipulate data in existing schema objects.*
- *The TCL commands are used to mange transactions.*
- *A transaction is one complete unit of work.*

Objective Type Questions — OTQs

Multiple Choice Questions

1. A relational database consists of a collection of
   (*a*) Tables (*b*) Fields (*c*) Records (*d*) Keys
2. A relational database consists of a collection of
   (*a*) Tuples (*b*) Attributes (*c*) Relations (*d*) Keys
3. A(n) _____ in a table represents a logical relationship among a set of values.
   (*a*) Attribute (*b*) Key (*c*) Tuple (*d*) Entry
4. The term _____ is used to refer to a record in a table.
   (*a*) Attribute (*b*) Tuple (*c*) Field (*d*) Instance
5. The term _____ is used to refer to a field in a table.
   (*a*) Attribute (*b*) Tuple (*c*) Row (*d*) Instance
6. A _____ is a property of the entire relation, which ensures through its value that each tuple is unique in a relation.
   (*a*) Rows (*b*) Key (*c*) Attribute (*d*) Fields
7. Which of the following attributes cannot be considered as a choice for primary key ?
   (*a*) Id (*b*) License number (*c*) Dept_id (*d*) Street
8. An attribute in a relation is a foreign key if it is the _____ key in any other relation.
   (*a*) Candidate (*b*) Primary (*c*) Super (*d*) Sub
9. Consider the table with structure as :

   `Student(ID, name, dept name, tot_cred)`

   In the above table, which attribute will form the primary key ?
   (*a*) Name (*b*) Dept (*c*) Total_credits (*d*) ID
10. Which of the following is not a legal sub-language of SQL ?
    (*a*) DDL (*b*) QAL (*c*) DML (*d*) TCL

Fill in the Blanks

1. Collection of logically related data tables is called a _____ .
2. The duplication of data is known as data _____ .
3. A pool of values wherefrom a field can draw values, is called _____ .
4. A row in a relation is called a _____ .
5. A column in a relation is called an _____ .
6. The number of attributes in a relation is called its _____ .
7. The number of tuples in a relation is called its _____ .

Path Wala

8. An attribute that can uniquely identify each tuple in a relation is called _____.
9. A non-key attribute derived from the primary key of some other relation is called _____.
10. A data model wherein data is arranged in tabular forms called relations and linked through common attributes of relations, is called _____ data model.

True/False Questions

1. A table in a relational database can store empty values.
2. A relation is a table having unordered non-atomic values.
3. A primary key can store empty values in it.
4. Common attribute of two tables is called a foreign key.
5. A common attribute of two tables is called a foreign key if it is the primary in one table.
6. Part of SQL which creates and defines tables and other database objects, is called DDL.
7. Part of SQL which manipulates data in tables, is called TCL.
8. Part of SQL which accesses and manipulates data in tables, is called DML.
9. Part of SQL which controls transactions, is called TCL
10. MySQL is the name of a customised query language used by Oracle.

NOTE : Answers for OTQs are given at the end of the book.

Solved Problems

1. *What is a database system ? What is its need ?*

Solution. A database is a collection of interrelated data and a database system is basically a computer based recordkeeping system.

A typical file processing system suffers from some major limitations like *data redundancy* (duplication of data), *data inconsistency, unsharable data, unstandardized data, insecure data, incorrect data* etc. On the other hand, a database system overcomes all these limitations and ensures continuous efficiency.

The advantages provided by a database system are :

(*i*) Reduced data redundancy
(*ii*) Controlled data inconsistency
(*iii*) Shared data
(*iv*) Standardized data
(*v*) Secured data
(*vi*) Integrated data.

Therefore, to have the systems with increased performance and efficiency, the database systems are preferred.

2. *Can you think of disadvantages of using a database system ? What are these ?*

Solution. With the complex tasks to be performed by database systems, certain things may crop up that may be termed as disadvantages of using database system. These are :

1. Security may be compromised without good controls.
2. Integrity may be compromised without good controls.
3. Extra hardware may be required.
4. Performance overhead may be significant.
5. System is likely to be complex.

Path Wala

3. *A database is to contain information about persons and skills. At a particular time the following persons are represented in the database, and their skills are as indicated.*

Person	*Skills*
Anuj	*Programming*
Bhawna	*Operating and Programming*
Chetan	*Engineering and Programming*
David	*Operating and Engineering*

For each person, the database contains various personal details, such as address. For each skill it contains an identification of the appropriate basic training course, an associated job grade code, and other information. The database also contains the date each person attended each course, where applicable (the assumption is that attendance at the course is essential before the skill can be said to be acquired).

Sketch a relational view for this data.

Solution. Relational view

Person

Pname	*Address*	.....
Anuj	.....	.....
Bhawna	.....	.....
Chetan	.....	.....
David	.....	.....

Skill

Skill-Name	*Course*	*Jobcode*	.....
Programming	–	–	.....
Operating	–	–	.....
Engineering	–	–	.....

Perskill

Pname	*SkillName*	*Date*
Anuj	Programming	–
Bhawna	Operating	–
Bhawna	Programming	–
Chetan	Engineering	–
Chetan	Programming	–
David	Operating	–
David	Engineering	–

4. *What are views ? How are they useful ?*

Solution. A view is a virtual table that does not really exist in its own right but is instead derived from one or more underlying base table(s). The view is a kind of table whose contents are taken upon other tables depending upon a given query condition. No stored file is created to store the contents of a view rather its definition is stored only.

The usefulness of views lies in the fact that they provide an excellent way to give people access to some but not all of the information in a table.

5. *What is the purpose of using MySQL ?*

Solution. MySQL is a free, open source *Relational Database Management System* (RDBMS) that uses Structured Query Language (SQL). In a MySQL database, information is stored in *Tables*. A single MySQL database can contain many tables at once and store thousands of individual records.

Path Wala

6. *What do you understand by MySQL server instance ?*

Solution. MySQL server instance is created from background processes and applications. It stays in memory and listens for client requests coming in over the network and accesses database contents according to those requests and provides that to the clients.

7. *What do you understand by MySQL client ?*

Solution. **MySQL Clients** are programs that connect to the MySQL server and issue queries in a pre-specified format. MySQL is compatible with the standards based SQL (Structured Query Language). The client program may contact the server programmatically or manually.

8. *MySQL provides some command line programs and GUI programs. Name some of these.*

Solution. Command line programms : *mysqldump* and *mysqladmin*

GUI programs : *MySQL Administrator* and *MySQL QueryBrowser*

9. *What is SQL ? What are different categories of commands available in SQL ?*

Solution. In order to access data within the Oracle database, all programs and users must use, Structured Query Language (SQL). SQL is the set of commands that is recognised by nearly all RDBMSs.

SQL commands can be divided into following categories :

1. Data Definition Language (DDL) Commands.
2. Data Manipulation Language (DML) Commands.
3. Transaction Control Language (TCL) Commands.
4. Session Control Commands.
5. System Control Commands.

10. *Differentiate between DDL and DML commands.*

Solution. The Data Definition Language (DDL) commands, as the name suggests, allow you to perform tasks related to data definition. That is, through these commands, you can perform tasks like, create, alter and drop schema objects, grant and revoke privileges etc.

The Data Manipulation Language (DML) commands, as the name suggests, are used to manipulate data. That is, DML commands query and manipulate data in existing schema objects.

GLOSSARY

Alternate Key	A candidate key which is not primary key.
Attribute	A column of a relation.
Base Table	A table having an independent existence ; represented in storage by a distinct stored file.
Candidate Keys	Attribute combinations in a relation that can serve as a primary key.
Cardinality	Number of tuples in a relation.
Data Redundancy	Duplication of data.
Database	Collection of interrelated data.
Database System	A combination of a computer based recordkeeping system and a database.
Degree	Number of attributes in a relation.
Domain	A pool of values from which the actual values appearing in a given column are drawn.
Primary Key	Set of one or more attributes that can uniquely identify tuples within a relation.

Path Wala

Relation	Data arranged in rows and columns and having certain properties.
Tuple	A row of a relation.
View	A virtual table derived from one or more underlying base tables.
Data Dictionary	A file containing metadata i.e., data about data.
SQL	Structured Query Language A language to operate upon RDBMSs.

Assignments

1. Summarize the major differences between a relation and a traditional file.
2. Define the terms : (*i*) database (*ii*) SQL (*iii*) view.
3. What is data redudancy ? How does it impact a database ?
4. What is data inconsistency ? How does it impact a database ?
5. Define the terms : (*i*) tuple (*ii*) attribute (*iii*) domain (*iv*) degree (*v*) cardinality.
6. Define the terms : (*i*) primary key (*ii*) foreign key.
7. What is MySQL ? What are its functions and features ?
8. What is the role of database server in database management system ? Give the key features of MySQL.
9. What is the use of SQL in MySQL ?
10. How are SQL commands classified ?
11. What functions should be performed by ideal DDL ?
12. Differentiate between DDL and DML commands.
13. Name some commands used to assign/revoke priviledges from database users.
14. Name some table maintenance commands.
15. What is TCL part of SQL ?