Chapter - 11

Structured Query Language

* Some MySQL SQL Elements

The ~~three~~ basic elements in MySQL are :—

i. Literals
ii. Datatypes
iii. Nulls.
iv. Comments.

i. Literals

Literals, in general, refer to a fixed data value. This fixed value may be of character type or numeric literal or of other data types.

ii. Data Types

Data types are means to identify the type of data and associated operations for handling it.

MySQL uses many different data types, divided into three categories :—

i. Numeric
ii. Date and time
iii. String types.

→ Numeric data types

i. Int

A normal-sized integer that can be signed or unsigned.

ii. Float (M, D)

A floating-point number that cannot be unsigned.

iii. Double (M, D)

A double precision floating-point number that cannot be unsigned.

iv. Decimal (M, D)

An unpacked floating point number that cannot be unsigned.

→ **Date and Time Types**

i. **Date**

A date in YYYY-MM-DD format, between 1000-01-01 and 9999-12-31.

ii. **Datetime**

A date and time combination in YYYY-MM-DD HH:MM:SS format, between 1000-01-01 00:00:00 and 9999-12-31 23:59:59.

iii. **Timestamp**

A timestamp between midnight, January 1, 1970 and sometime in 2038.

iv **Time**

stores the time in HH:MM:SS format.

→ **String / Text Types**

i. **Char**

A fixed-length string between 1 and 255 characters in length.

ii. **Varchar**

A variable-length string between 1 and 225 character in length.

iii. **Blob or Text**

A field with a maximum length of 65535 characters.

Difference between char and varchar datatypes :—

Char	Varchar
Fixed length fields have fixed lengths i.e. they occupy fixed number of bytes for every data element they store.	Variable length fields have varied field lengths i.e. field length is determined separately for every data element inside the field.

Null Values

If a column in a row has no value, then column is said to be null, or to contain a null. Nulls can appear in columns of any data type provided they are not restricted by not null.

Comments

A comment is a text that is not executed; it is only for documentation propose.
You can include a comment in a statement using either of these means :—

i. Begin the comment with /*.
ii. Begin the comment with -- (followed by a space).
iii. Begin the comment with #.

* SQL Command Syntax

The SQL provides a predefined set of commands that help us work on relational databases.

Keywords : Keywords are words that have a special meaning in SQL.

Commands : Commands are instructions given by you to a SQL database.

Clauses : Commands consist of one or more logically distinct parts called clauses.

Arguments : Arguments complete or modify the meaning of a clause.

* A statement in MySQL SQL is completed with a semicolon (;).

* Databases in MySQL

i. Creating Databases

The syntax is as follows:—

```
create database [if not exist] <database name>;
```

The if not exists clause, if used, will first test whether a database by the mentioned name already exists or not. If it does, then 'create database command' is simply ignored, otherwise a database with the mentioned name is created.

ii. Opening Databases

The syntax is as follows:—

```
Use <database name>;
```

iii. Removing databases

The syntax is as follows:—

drop database <database name>;

* Creating Tables

The syntax is as follows:—

Create table <table name>
(<column name> <data type> [(<size>)],
<column name> <data type> [(<size>)....]);

i. Data Integrity Through Constraints

Constraint ~~in general~~ is a condition or check applicable on a field or a set of fields.

Constraint, in general refers to a condition or a check that is applied to a column or set of columns in a table. The constraints applied to maintain data integrity are also known as integrity constraints.

The two basic types of constraints are:—
1. Column constraints
2. Table constraints.

→ Different Constraints

The constraints are as follows :—

i. Unique constraint
ii. Primary Key constraint
iii. default constraint
iv. check constraint
v. foreign Key constraint.

i. Unique Constraint

This constraint ensures that no two rows have the same value in the specified column(s).

For e.g.

```
Create table employee
    (ecode int NOT NULL UNIQUE,
     ename char(20) NOT NULL,
     sex char(1) NOT NULL);
```

ii. Primary Key Constraint

This constraint declares a column as the primary key of the table.

For e.g.

```
Create table employee
    (ecode int NOT NULL PRIMARY KEY,
     ename char(20) NOT NULL);
```

iii. Default Constraint

A default value can be specified for a column using the DEFAULT clause.
For e.g.

```
create table employee
(ecode int NOT NULL PRIMARY KEY,
 ename char(20) NOT NULL,
 grade char(2) DEFAULT "E1");
```

iv. Check Constraint

This constraint limits values that can be inserted into a column of a table.
For e.g.

```
create table employee
(ecode int NOT NULL PRIMARY KEY,
 ename char(20) NOT NULL,
 gross decimal CHECK (gross >2000));
```

v. Foreign Key

An example of syntax is as follows :—

```
Create table t1( id int primary key, name varchar(30),
                 marks int default 100);

Create table t2( id int, book_name varchar(30),
                 foreign key(id) references t1(id));
```

→ Viewing a Table structure

Syntax :—

DESC <table name> ;

* Inserting data in Tables

i. Insert into Command

Syntax is as followed :—

Insert into <table name> [<column list>]
values (<value>, <value>, ...) ;

→ Inserting NULL values

To insert value NULL in a specific column, you can type NULL without quotes.

→ Inserting dates

Dates are by default entered in 'YYYY-MM-DD' format.

* Making simple select Queries

i. Accessing database

The syntax is as follows :—

Use <database name> ;

ii. The SELECT Command

The syntax is as follows :—

Select <column name> [, <column name>, ...]
from <table name> ;

iii. Selecting all Columns

The syntax is as follows :—

Select * from ~~<database name>~~ <table name> ;

iv. Eliminating Redundant data (with keyword DISTINCT)

The syntax is as follows :—

Select DISTINCT <column name> from <table name> ;

v. Simple Calculations

The syntax is as follows :—

select <calculations>;

vi. Using Column Aliases

The columns that you select in a query can be given a different name i.e. column alias name for output purposes.
The syntax is as follows :—

Select <column name> as <column alias>;

vii. Handling Nulls

If you want to substitute null with a value in the output, you can use IFNULL() function.
The syntax is as follows :—

Select <column name> Ifnull(<column name>, value to be substituted);

viii. Selecting specific rows

To select specific rows we use WHERE clause.
The syntax is as follows :—

```
select <column name> [, <column name>, ...]
from <table name>
where <condition> ;
```

ix. Sorting Results - Order by clause

The order by clause is used as follows :—

```
select <column name> [, <column name>, ...]
from <table name>
[where <predicate>]
[order by <column name>] ;
```

* To display in descending order we use 'desc' in the end.

* More DML Commands

i. Modifying data with UPDATE Command

The syntax is as follows :—

update <table name> set <column name> = <new value>
where <column name> = <unique value of column which you want to change> ;

→ Updating multiple columns using update command

To update multiple columns, multiple column assignments can be specified with SET clause, separated by commas.

Syntax :—

Update <table name> set <column name> = <new value>, <column name> = <new value>, ...;

ii. Deleting data with delete Command

The delete statement takes the following general form :—

Delete from <table name>
[where <predicate>] ;

* More DDL Commands

i. Alter table command

→ Adding columns

Syntax :—

Alter table <table name> add <column name> <data type> <size> ;

→ Renaming column

Syntax :—

Alter table <table name> rename column <old column name> to <new column name> ;

→ Changing data type

Syntax :—

Alter table <table name> change <column name> <new data type>;

→ Dropping a column

Syntax :—

Alter table <table name> drop column <column name>;

ii. Addition or Removal of Constraints using Alter Command

→ Addition of Primary Key —

~~alt~~ Syntax :—

Alter table <table name> add ~~pri~~ primary key (<column name>);

→ Removal of Primary Key —

Syntax :—

Alter table <table name> drop primary key;

→ **Addition of default**

Syntax :—

alter table <table name> alter <column name> set default <value>;

→ **Removal of default**

Syntax :—

alter table <table name> alter <column name> drop default;

→ **Adding not null**

Syntax :—

alter table <table name> ~~alter <column name> drop default;~~ modify <column name> <column data type> not null;

→ **Removal of not null**

Syntax :—

alter table <table name> modify <column name> <column data type>;

→ **Addition of Unique Key**

Syntax :—

alter table <table name> add constraint <constraint name> unique (<column name>);

→ **Deletion of Unique Key**

Syntax :—

alter table <table name> drop constraint <constraint name>;

* [show create table <table name>;]

↑

This statement is used to see the constraint in the table.

→ **Addition of check**

Syntax :—

alter table <table name> add constraint <constraint name> check (<expression>);

→ **Removal of Check**

Syntax :—

alter table <table name> drop constraint <constraint name>;

→ **Addition of Foreign Key** —

Syntax :—

alter table <table name> add constraint <constraint name> foreign key (<column in which we want to implement foreign key>) references <table name> (<Primary key columns>);

→ **Removing Foreign Key** —

Syntax :—

⇒ alter table <table name> drop constraint <constraint name>;

⇒ alter table <table name> drop key <constraint name>;