**SQL THEORY:**

Types of programming

1. Procedural(imperative)
2. Object-oriented
3. Declarative
4. Functional

SQL is a declarative (nonprocedural) language

**Database creation**

**DDL – Data Definition Language:**

A syntax

A set of statements that allow the user to define or modify data structures and objects, such as tables.

1. The CREATE statement:

Used for creating entire database and database objects as tables

CREATE TABLE table\_name (column\_name data\_type);

1. The ALTER statement:h

Used when altering existing objects.

ADD  
REMOVE  
RENAME

ADD COLUMN date\_of\_purchase DATE;

1. The DROP statement:

Used to delete the a database object

DROP TABLE customers;

1. The RENAME statement:

Allows you to rename an object

RENAME TABLE customers TO customer\_data;

1. The TRUNCATE statement:

Instead of deleting an entire table through DROP, we can also remove it’s data and continue to have the table as an object as an object in the database

TRUNCATE TABLE customers;

**SQL Keywords:**

KEYWORDS in SQL cannot be variable names.

Keywords = reserved words

Keywords can’t be used when naming objects.

**DML – Data Manipulation Language**

Its statement allow us to manipulate the data in the tables of a database.

1. The SELECT statement:

Used to retrieve data from database objects, like tables.

SELECT \* FROM sales;

Why SELECT ?

A table with 2 million rows of data

It can be helpful if you could extract only a portion of the table that satisfies given criteria.

1. The INSERT statement:

Used to insert data into tables

INSERT …INTO….VALUES

INSERT INTO sales (purchase\_number, data\_of\_purchase) values (1, ‘2017-10-11’);

1. The UPDATE statement:

Allows you to renew existing data of your tables

UPDATE sales SET date\_of\_purchase = “2017-12-12”

WHERE purchase\_number = 1;

1. The DELETE statement

Functions similarly to the TRUNCATE statement

TRUNCATE vs DELETE

TRUNCATE allows us to remove all the records contained in a table.

Vs

With DELETE, you can specify precisely what you would like to removed.

DELETE FROM sales; = TRUNCATE TABLE sales;

Above command, will do the same for both DELETE and TRUNCATE.

DELETE FROM sales WHERE purchase\_numer =1;

Tis above command, will delete only that particular record which purchase\_no is 1.

**DCL - Data Control Language**

The Data Control Language is a syntax containing only two statements – GRANT and REVOKE.

The GRANT and REVOKE statements:

Allow us to manage the rights users have in a database.

1. The GRANT statement:

Gives (or grants) certain permissions to users.

One can grant a specific type of permission, like complete or partial

eg

GRANT type\_of\_permission ON database\_name.table\_name TO ‘username@’localhost’

These rights will be assigned to a person who has a username registered at the Local server (‘localhost’ : IP 127.0.0.1)

Eg

CREATE USER ‘logeshwari’@’localhost’ IDENTIFIED BY ‘password’

Localhost – IP of the host should mention

Password – password should mention

Eg.

GRANT SELECT ON sales.customers TO ‘logeshwari’@’localhost’

In this type of permission user can only use SELECT command only can see customers table from the sales.

GRANT ALL ON Sales.\* to ‘logeshwari’@’localhost’;

In this type of permission user can access all the tables from sales database also can use all the commands.

**Data Control Language**

Data administrators:

People who have complete rights to a database

They can grant access to users and can revoke it.

The REVOKE clause

Used to revoke permissions and privileges of database users.

The exact opposite of GRANT

REVOKE type\_of\_permission ON database\_name.table\_name FROM ‘username’@’localhost’

Eg.s

REVOKE SELECT ON sales.customers FROM ‘logeshwari’@’localhost’;

**TCL - Transaction Control Language**

Not every change you make to a database is saved automatically

1. The COMMIT statement

Saves the transaction in the database

Changes cannot be undone

Eg.

If we want to change the last name of the 4th customer id in customers table. Only administrator can do this changes.

UPDATE customers

SET last\_name = ‘new\_name’

WHERE customer\_id = 4

COMMIT;

Unless we can’t do commit the changes will not saved.

1. The ROLLBACK Clause

Allows you to take a step back

The last changes made will not count

Revers to the last non-committed state.

Eg . ROLLBACK;

**RELATIONAL DATABASE ESSENTIALS:**

Mail goal:

Organize huge amounts of data that can be quickly retrieved

Compact

Well-structured

Efficient

To manage database efficiency we will use RDBMS.

**Databases vs. Spreadsheets**

Data tables, databases and Excel spreadsheets are different things.

Both can contain a large amount of tabular data

Both can existing data to make calculations.

**Spreadsheets:**

An electronic ledger,

An electronic version of paper accounting worksheets

Different cells can contain calculations (functions and formulas)

Only 1 million rows max

Multiusers is the problem. Every user should make change separately.

Difficult to finding out who changed or deleted information incorrectly.

**Relational database**

Pre-set the type of data contained in a certain field

All calculations and operations are done after data retrieval

Can do calculations in ‘views’

Record of data != calculation

Greater than 1 million rows

Provide a stable structure, controlling access permissions and user restrictions.

No duplicates

**Database Terminology**

**Database Engineer:**

Plot the entire database system on a canvas using a visualization tool.

1. **Relational Schema:**

An Existing idea of how the database must be organized.

Arranging the table with proper structure:

Eg.

Table name:

Column names

1. **Database Creation:**

Use SQL to set up the database physically

1. **Database Manipulation:**

Allows you to use your dataset to extract business insights

If these above steps are good we can write queries and navigate in a database rather quickly.

Database design + Creation + Manipulation = database management

**Database Administration:**

This person is responsible for database. This person is maintaining the database with daily care.

**Relational Schema:**

These tables have a tabular form a relational schema can be applied to represent them.represent the concept database administrators must implement

Depict how a database is organized

Blueprints or a plan for a database

**Primary Key:**

A column (or a set of columns) whose value exists and is unique for every record in a table is called a primary key.

Each table can have one and only one primary key.

Primary key column should not be blank.

Primary key always mentioned in top of the table also underlined.

Not all tables will have a primary key.

Primary keys are the unique identifiers of a table.

In one table, cannot have 3 or 4 primary keys.

May be composed of a set of columns

For eg.

Id and date   
both has different values and unique values.

**Foreign Key:**

The primary key of one table (sales) the same column is the foreign key in another table (Sales).

**Relational Schemas: Unique Key & Null Values:**

|  |  |  |
| --- | --- | --- |
|  | Primary Key | Unique key |
| Null Values | No | Yes |
| Number of Keys | 1 | 0,1,2.. |
| Application to multiple columns | Yes | Yes |

**Relationships:**

Relationships tell us how much of the data from a foreign key field can be seen in the primary key column of the table the data is related to and vice versa.

**One-to-Many type of relationship**

One value from the customer\_id column under the “Customers” table can be found many times in the customer\_id column in the “sales” table.

**INSTALLATION AND CREATING DATABASE:**

Installed Workbench and SQL server:

Set up connection:

Creating the database:

**Syntax:**

CREATE DATABASE [If not exists] database\_name;

**Explanation:**

CREATE DATABASE:

Creates a database as an abstract unit

IF NOT EXISTS:

Verifies if a database with the same name exists already

* The brackets around mean the statement is optional (you could type or omit the statement)

Databse\_name:

Give a name that is short but at the time as related to the content of the data as possible

* The SQL code is not case sensitive
* In this element the quotes are optional (eg. Naming the database as sales. We can represent like Sales, SALES, sales, “sales” anything is fine)
* ; (Always ends with semicolon character)
* Semicolon functions as a statement terminator
* When code contains more than a single statement, ; is indispensable
* Will help to avoid errors sometimes
* Will improve the readability of the code.

Eg. CREATE DATABASE IF NOT EXISTS Sales;

CREATE SCHEMA IF NOR EXISTS Sales;

Sales database created.

Instead of database keyword, Schema also we can use.

**USE keyword:**

USE Sales;

This USE statement tells MySQL to use the named database as the default (current) database for subsequent statements

**DATA TYPES IN SQL:**

We must always specify the type of data that will be inserted in each column of the table

**STRING:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data type** | **Surname of a person** | **Length** | **Size** |
| **String** | **‘james’** | **5 symbols** | **5 ytes** |

Digits, symbols, or blank can also be used in the string format.

These will only convey text information

Eg. Address column: “No.25, Nehru Street, chengalpattu ”

Size:

Indicated the memory space used by a data type.

* Measured in bytes
* 1 byte ~ 1 symbol

Storage

The physical space in the computer drive’s memory, where the data is being saved or stored.

**STRING DATA TYPES:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| String\_data\_type |  | Storage | Examples | Length  (Symbols) | Size  (bytes) | Max Size  (bytes) | Notes |
| Character | CHAR | Fixed | CHAR(5)  Eg.’James’  ‘Bob’ | 5  3 | 5  5 | 255 | 50% faster |
| Variable Character | VARCHAR | Variable | VARCHAR(5)  Eg.’James’  ‘Bob’ | 5  3 | 5  3 | 65,535 | A lot more responsive to the data value inserted |
| ENUM(“enumerate”) | ENUM |  | ENUM(‘M’,’F’)  Eg. For gender column we can use to represent male and female |  |  |  | MYSQL will show an error If we attempt to insert any value different from “M” or “F” |

CHAR(5):

Five(5) represents the maximum number of symbols you are allowed to use in writing a value in this format.

Bytes are 5 even for bob because the storage is fixed.

Eg company short code kind of with all values are same number of characters then we can use CHAR (no of characters)

Eg. CHAR (5)

Eg. If we want to store passwords with max 10 characters. Also passwords will contains any char and symbol then we can use VARCHAR(10)

**Integers**

Numeric data types

Integer

Fixed-point

Floating-point

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Numeric data type | Size  (bytes) | Minimum value  (signed / unsigned) | Maximum value  (signed / unsigned) |  |
| TINYINT | 1 | -128  0 | 127  255 |  |
| SMALLINT | 2 | -32, 768  0 | 32,767  65,535 |  |
| MEDIUMINT | 3 | -8,388,608  0 | 8,388,607  16,777,215 |  |
| INT | 4 | -2,147,483,648  0 | 2,147,483,647  4,294,967,295 |  |
| BIGINT | 8 | -9,223,372,036,854,775,808  0 | 9,223,372,036,854,775,807  18,446,744,073,709,551,615 |  |

**Signed:**

If the encompassed range includes both positive and negative values

**Unsigned:**

If integers are allowed to be only positive.

**Why not just use BIGINT all the time?**

**e.g**

if we won’t need an integer smaller than 0 or greater than 100, TINYINT would do the job perfectly and not need more storage space per data point

A smaller integer type may increase the processing speed.

**Fixed and Floating-point Data types**

The main difference between the fixed- and the floating-point type is in the way the value is represented in the memory of the computer.

|  |  |  |
| --- | --- | --- |
| DECIMAL(5,3)  10.523624 | 10.524 SQL will show warning in this case | fixed |
| FLOAT(5,3)  10.523624 | 10.524 SQL will not show warning in this case | floating |

|  |  |  |  |
| --- | --- | --- | --- |
| **Floating-point data type** | **Size(bytes)** | **Precision** | **Max num of digits** |
| FLOAT | 4 | Single | 23 |
| DOUBLE | 8 | Double | 53 |

**Other Datatypes:**

**DATE:** Used to represent a date in the format YYYY-MM-DD

**DATETIME:** Represents the date shown on the calendar and the time shown on the clock

**TIMESTAMP:** Used for a well-defined, exact point in time.  
 Representing a moment in time as a number allows you to easily obtain the difference between two TIMESTAMP values

**Eg.** End time : ‘2018-07-25 10:30:00’ UTC TIMESTAMP

Start time : ‘2018-07-25 9:.00:00’ UTC TIMESTAMP

**BLOB : Binary Large Object**

Refers to a file of binary data – data with 1s and 0s

Involves saving files in a record

**CREATING SQL TABLE**

CREATE TABLE table\_name (column names );

Compulsory requirement: add at least one column

**Syntax:**

CREATE TABLE table\_name

(

Column\_1 data\_type constraints,

Column\_2 data\_type constraints,

…..

Column\_n data\_type constraints

);

**Using Databases and Tables:**

Whenever we would like to refer to an SQL object in our queries, we must specify the database to which it is applied.

**SQL Objects:**

SQL table

Views

Stored procedures

Functions

1. Set a default database

USE sales;

SELECT \* FROM customer;

1. Call a table from a certain database

Database\_object.sql\_object

**CONSTRAINT:**

PRIMARY KEY:

FOREIGN KEY:

A foreign key in SQL is defined through a foreign key constraint

The foreign key maintains the referential integrity within the database.

ON DELETE CASCADE:

If a specific value from the parent table’s primary key has been deleted, all the records from the child table referring to this value will be removed as well.

UNIQUE KEY:

Used whenever you would like to specify that you don’t want to see duplicate data in a given field.

Ensures that all values in a column (or a set of columns) are different.

UNIQUE KEY (email\_address)

Or

ALTER TABLE table\_name

ADD UNIQUE KEY (column\_name);

DEFAULT Constraint:

Helps us assign a particular default value to every row of column

A value different from the default can be stored in a field where the DEFAULT constraint has been applied, only if specifically indicated.

eg

company\_name VARCHAR(255) DEFAULT 'X',

NOT NULL constraint:

When you insert values in the table, you cannot leave the respective field empty

If you leave it empty, MYSQL will signal an error.