SQL Basics

Structured Query Language:

→ SQL is a language used for interacting with Relational Database Management Systems (RDBMS)

SQL is not programming language, but we can use SQL to provide instructions to a RDBMS. So, it can be programming language.

RDBMS is a software application that we can use to create and maintain a relation database.

In order to interact with RDBMS, we can't just talk in english, we need to use special language called SQL.

Therefore, if we want to ask a Relational DBMS to do something for us,

like store a piece of info, create a table, update info, we can ask RDBMS to do that using SQL.

→ You can use SQL to get or tell the RDBMS to do things for you.

→ Create, set, view, update & delete data

→ Create, manage databases

→ Design & create databases tables

→ Perform administration tasks (security, user management, import/export, etc)

→ SQL implementations may vary between the systems.

SQL has a overall formal specification, which defines, how SQL needs to be used, and different commands can be used. But all the RDBMSs are going to implement SQL differently.

∴ we can write SQL code on one RDBMS, it may not work on other.

For most part, everything should be the same, we need to change little things.

Therefore, SQL is used on all of the major RDBMS, but it is used slightly differently

→ The concepts are the same, but the implementation may vary.

SQL is actually a hybrid language, it's basically 4 types of languages in one,

1) Data Query language (DQL)
   → Used to query the database for information
   → Get information that is already stored there.

2) Data Definition language (DDL)
   → Used for defining database schemas (Schema is basically the overall layout of the database) like what tables are there in it, what columns etc)

3) Data Control language (DCL)
   → It's used for controlling access to the data in the database.
   So, we could define a bunch of users for the database, and allow this one user to write the table, others to read and some other user to delete/update info

→ User & permissions Management.

4) Data Manipulation Language (DML)
Used for inserting, updating, and deleting data from the database.

Queries

A query is a set of instructions given to the RDBMS (written in SQL) that tell the RDBMS what information you want it to retrieve for you.

- → There may be tons of Data in a database
- → often hidden in a complex schema
- → Goal is to only get the piece of information or data you need.

∴ We're telling the RDBMS what we want, and then it only gives us back that information.

MySQL is RBBMS, or its a software application that which we can use to maintain and create and do all that sort of things with databases

∴ when we install MySQL on our computer, we can actually set up what's called a MySQL database server, ie a server where MySQL is running and then we can write SQL in sorts of queries and instructions in order to create and do stuff with database.

Creating Tables

SQL command- Create table.

Different datatypes we can store inside our database

Basic Datatypes (Core).

1) INT - whole number

2) DECIMAL (M, N) -

M - Total no. of digits that we want to store for this number

N - No. of digits that you want to store after the decimal point

3) VARCHAR (1) → String of text length 1

Variable character → How many characters we want this to be able to store.

To store string of text.

2. C++
3. DSA

4) BLOB

Binary large object.

It is a structure that can store large amounts of binary data.

(Generally to store, img)

5) DATE

'YYYY - MM - DD'

6) TIMESTAMP

('YYYY - MM - DD' 'HH:MM:SS')

Try to use upper case letters for commands in SQL, to differentiate it from other text we write.

Any command that we write in SQL is always going to end with semi colon.

To create a table

1) Command - Create table (table name)

then 'column name with datatype.

Codes

```
CREATE TABLE student (
    student_id INT, PRIMARY KEY,
    name VARCHAR(20),
    major VARCHAR(20)
);
```

we can also write at the end of columns.

```
PRIMARY KEY (student_id)
```

name of the column

→ Command DESCRIBE – describes everything about the table

```
DESCRIBE student
```

→ To delete a table – Use

```
DROP TABLE student;
```

→ To add another column

```
ALTER TABLE student ADD gpa DECIMAL(3,2);
```

student → name of the table
ADD → adds a column
gpa → name of the column
DECIMAL(3,2) → Datatype

→ To drop a specific column.

```
ALTER TABLE student DROP COLUMN gpa;
```

Inserting data

Inserting data into database tables.

→ In order to insert a piece of information into a table, we have to type

'INSERT INTO'

→ INSERT INTO student VALUES ();

student ↓ name of the table

VALUES () ↓ enter values.

First we need to insert values in student_Id, name, then major.

For string use ' '

```
SELECT * FROM student;
```

This is going to give us all the information from the student table.

```
INSERT INTO student VALUES (1, 'John',
                            'Biology');
```

If we have a student who don't have major,

NOT NULL and UNIQUE are two good ways to control the data that gets stored on the table.

A primary key is an attribute / column that is both NOT NULL and Unique.

DEFAULT

⇒ We use this when a student doesn't give his major.

Ex :- major VARCHAR(20) DEFAULT 'undecided',

↓

This when we doesn't give any value to major, it says undecided.

AUTO_INCREMENT

Student_id INT AUTO_INCREMENT,

↓

This automatically increases student_id.

Update & Delete

Updating and deleting rows from the database table.

Ex = Let's we change the official name for Biology major as 'Bio'.

To do so,

Code

```
UPDATE student
SET major = 'Bio'
WHERE major = 'Biology';
```

Update student table

Set major as Bio, but not for all the students

where major is Biology

Set as Bio.

→ We can also update with different conditions.

```
UPDATE student
SET major = 'Bio'
WHERE student_id = 4;
```

This means, to update student table by setting major as Bio for student_id with 4.

We can also update in this way !

```
UPDATE student
SET major = 'Biochemistry'
WHERE major = 'BIO' OR major = 'Chemistry';
```

↓

setting major as Biochemistry for the students who have their major as either Bio or chemistry. We are combining two majors.

You can either select specific rows or groups of rows with that WHERE or you can just do it to all of them by dropping the WHERE.

Delete Rows

1)
```
DELETE FROM student
WHERE student_id = 5;
```

2) Also we can use

```
WHERE name = 'TOM' AND
      major = 'undecided';
```

Just

```
DELETE FROM student
```

it deletes everything from the table.

Basic Queries

Query is just a block of SQL that's designed to ask the DBMS for a particular piece of information.

Ex. I want to just grab students who meet certain condition or student with certain major from trillions of students.

SELECT keyword is going to tell the RDBMS that we want to get some information from it.

```
SELECT *
```

↓

means to grab all the information

To get a specific column, use

```
SELECT major
```

↓

name of the column

Or

```
SELECT name, major
FROM student;
```

ORDER BY

sets in alphabetical order.

```
ORDER BY name;
```

By default, this is in ascending order, we can say DESC ~~and~~ for descending.

```
SELECT name, major
FROM student
ORDER BY name DESC;
```

↓

name of the column

ASC for ascending.

```
ORDER BY major, student_id;
```

Therefore, here it is order by major first, if any of them have major same, it is ordered by student_id then.

→ LIMIT 2;

If we want only specific no. of rows back from the table, then we'll get that

Col

```
SELECT *
FROM student
ORDER BY student_id DESC
LIMIT 2;
```

2 rows.

Code

```
SELECT *                    (* - all)
  FROM student
where major = 'Biology';
```

* <7 — not equal to'

Instead of '=' we can use

<= }
>= }
< }
> }

By using this in where, we can just do filtering.

Code

```
SELECT *
FROM student
WHERE name IN ('Claire', 'Kate', 'Mike');
```

This means
select all from the student where
the name is in these group of values.
So If the name is Claire, Kate or Mike,
then it's going to select that.

we can also combine,

```
WHERE major IN ('Biology', 'Chemistry')
      AND student_id > 2;
```