Web Application Programming

Week 6: Introduction to Server-Side Programming(Introduction to databases and SQL)

By Elubu Joseph - MSc.IS

Lecturer

Department of Information Technology

Kumi University

Email: josebulinda@gmail.com or jose@kumiuniversity.ac.ug

Agenda

1. Summary of previous lecture
2. Introduction to databases
3. Introduction to SQL

Summary of previous lecture

1. Introduction to Server-Side programing
2. Basics of server-side programming languages (e.g., PHP, Python, Node.js)
3. Installation of PHP through the server, Code editor, learnt some basics of php,

Introduction to Databases

Introduction to Databases

What is Data?

Data is nothing but facts and statistics stored or free flowing over a network, generally it's raw and unprocessed.

For example: When you visit any website, they might store your IP address, that is data, in return they might add a cookie in your browser, marking you that you visited the website, that is data, your name, it's data, your age, it's data.(studytonight.com, n.d).

Introduction to Databases+

Data becomes **information** when it is processed, turning it into something meaningful.

Like, based on the cookie data saved on user's browser, if a website can analyse that generally men of age 20-25 visit us more, that is information, derived from the data collected.

What is a Database?

Introduction to Databases+

A Database is a collection of related data organised in a way that data can be easily accessed, managed and updated. Database can be software based or hardware based, with one sole purpose, storing data.

During early computer days, data was collected and stored on tapes, which were mostly write-only, which means once data is stored on it, it can never be read again. They were slow and bulky, and soon computer scientists realised that they needed a better solution to this problem.

Larry Ellison, the co-founder of Oracle was amongst the first few, who realised the need for a software based Database Management System.

What is DBMS?

Database management system is a software that allows creation, definition and manipulation of database, allowing users to store, process and analyse data easily.

DBMS provides us with an interface or a tool, to perform various operations like creating database, storing data in it, updating data, creating tables in the database and a lot more.

DBMS also provides protection and security to the databases. It also maintains data consistency in case of multiple users.

Examples of DBMS

Here are some examples of popular DBMS used these days:

1. MySQL
2. Oracle
3. SQL Server
4. IBM DB2
5. PostgreSQL
6. Amazon SimpleDB (cloud based) etc.

Characteristics of Database Management System

A database management system has following characteristics:

1. **Data stored into Tables:** Data is never directly stored into the database. Data is stored into tables, created inside the database. DBMS also allows to have relationships between tables which makes the data more meaningful and connected. You can easily understand what type of data is stored where by looking at all the tables created in a database.
2. **Reduced Redundancy:** In the modern world hard drives are very cheap, but earlier when hard drives were too expensive, unnecessary repetition of data in database was a big problem. But DBMS follows Normalisation which divides the data in such a way that repetition is minimum. .(studytonight.com, n.d).

Characteristics of Database Management System

3. **Data Consistency:** On Live data, i.e. data that is being continuosly updated and added, maintaining the consistency of data can become a challenge. But DBMS handles it all by itself.
4. **Support Multiple user and Concurrent Access**: DBMS allows multiple users to work on it(update, insert, delete data) at the same time and still manages to maintain the data consistency.
5. **Query Language:** DBMS provides users with a simple Query language, using which data can be easily fetched, inserted, deleted and updated in a database.

Characteristics of Database Management System

6. Security: The DBMS also takes care of the security of data, protecting the data from un-authorised access. In a typical DBMS, we can create user accounts with different access permissions, using which we can easily secure our data by restricting user access.

DBMS supports transactions, which allows us to better handle and manage data integrity in real world applications where multi-threading is extensively used. .(studytonight.com, n.d).

Advantages of DBMS

1. Segregation of applicaion program.
2. Minimal data duplicacy or data redundancy.
3. Easy retrieval of data using the Query Language.
4. Reduced development time and maintainance need.
5. With Cloud Datacenters, we now have Database Management Systems capable of storing almost infinite data.
6. Seamless integration into the application programming languages which makes it very easier to add a database to almost any application or website.

Disadvantages of DBMS

1. It's Complexity
2. Except MySQL, which is open source, licensed DBMSs are generally costly.
3. They are large in size.

Introduction to SQL

Introduction to SQL

Structure Query Language(SQL) is a database query language used for storing and managing data in Relational DBMS. SQL was the first commercial language introduced for E.F Codd's Relational model of database.

Today almost all RDBMS(MySql, Oracle, Infomix, Sybase, MS Access) use SQL as the standard database query language. SQL is used to perform all types of data operations in RDBMS.

SQL Command

SQL defines following ways to manipulate data stored in an RDBMS.

1. DDL: Data Definition Language

This includes changes to the structure of the table like creation of table, altering table, deleting a table etc.

All DDL commands are auto-committed. That means it saves all the changes permanently in the database.

SQL Command

Command	Description
create	to create new table or database
alter	for alteration
truncate	delete data from table
drop	to drop a table
rename	to rename a table

SQL Command

SQL defines following ways to manipulate data stored in an RDBMS.

2. DML: Data Manipulation Language

DML commands are used for manipulating the data stored in the table and not the table itself.

DML commands are not auto-committed. It means changes are not permanent to database, they can be rolled back.

SQL Command

DML: Data Manipulation Language

Command	Description
insert	to insert a new row
update	to update existing row
delete	to delete a row
merge	merging two rows or two tables

SQL Command

SQL defines following ways to manipulate data stored in an RDBMS.

3. TCL: Transaction Control Language

These commands are to keep a check on other commands and their effect on the database.

These commands can annul changes made by other commands by rolling the data back to its original state. It can also make any temporary change permanent.

SQL Command

SQL defines following ways to manipulate data stored in an RDBMS.

Command	Description
commit	to permanently save
rollback	to undo change
savepoint	to save temporarily

SQL Command

SQL defines following ways to manipulate data stored in an RDBMS.

4. DCL: Data Control Language

Data control language are the commands to grant and take back authority from any database user.

Command	Description
grant	grant permission of right
revoke	take back permission.

SQL Command

SQL defines following ways to manipulate data stored in an RDBMS.

5. DQL: Data Query Language

Data query language is used to fetch data from tables based on conditions that we can easily apply.

Command	Description
select	retrieve records from one or more table

SQL: create command

create is a DDL SQL command used to create a table or a database in relational database management system.

Creating a Database

To create a database in RDBMS, create command is used. Following is the syntax,

```
CREATE DATABASE <DB_NAME>;
```

Example for creating Database

```
CREATE DATABASE Test;
```

The above command will create a database named Test, which will be an empty schema without any table.

Creating a Table

To create tables in this newly created database, we can again use the **create** command.

create command can also be used to create tables. Now when we create a table, we have to specify the details of the columns of the tables too. We can specify the names and datatypes of various columns in the create command itself.

Following is the syntax,

```
CREATE TABLE <TABLE_NAME>
(
    column_name1 datatype1,
    column_name2 datatype2,
    column_name3 datatype3,
    column_name4 datatype4
);
```

create table command will tell the database system to create a new table with the given table name and column information.

Example for creating Table

```
CREATE TABLE Student(
student_id INT, name
VARCHAR(100), age INT);
```

Creating a Table+

The above command will create a new table with name **Student** in the current database with 3 columns, namely **student_id**, name and age. Where the column **student_id** will only store integer, name will hold upto **100** characters and age will again store only integer value.

If you are currently not logged into your database in which you want to create the table then you can also add the database name along with table name, using a dot operator .

Creating a Table++

For example, if we have a database named **Test** and we want to create a table **Student** in it, then we can do so using the following query:

```
CREATE TABLE Test.Student( student_id INT, name VARCHAR(100), age INT);
```

Most commonly used datatypes for Table columns

Here we have listed some of the most commonly used datatypes used for columns in tables.

Creating a Table+++

Datatype	Use
INT	used for columns which will store integer values.
FLOAT	used for columns which will store float values.
DOUBLE	used for columns which will store float values.
VARCHAR	used for columns which will be used to store characters and integers, basically a string.
CHAR	used for columns which will store char values(single character).
DATE	used for columns which will store date values.
TEXT	used for columns which will store text which is generally long in length. For example, if you create a table for storing profile information of a social networking website, then for **about me** section you can have a column of type TEXT.

Using INSERT SQL command

Data Manipulation Language (DML) statements are used for managing data in database. DML commands are not auto-committed. It means changes made by DML command are not permanent to database, it can be rolled back.

Talking about the Insert command, whenever we post a Tweet on Twitter, the text is stored in some table, and as we post a new tweet, a new record gets inserted in that table.

INSERT command

Insert command is used to insert data into a table. Following is its general syntax,

```
INSERT INTO table_name VALUES(data1, data2, ...)
```

Consider a table **student** with the following fields.

s_id	name	age

```
INSERT INTO student VALUES(101, 'Adam', 15);
```

s_id	name	age
101	Adam	15

INSERT command

Insert value into only specific columns

We can specify the column names along with the values to be inserted like this,

```
INSERT INTO student(id, name) values(102, 'Alex');
```

Insert NULL value to a column

```
INSERT INTO student(id, name) values(102, 'Alex');
```

Or.

```
INSERT INTO Student VALUES(102,'Alex', null);
```

S_id	S_Name	age
101	Adam	15
102	Alex	

Using UPDATE SQL command

Let's take an example of a real-world problem. These days, Facebook provides an option for Editing your status update, how do you think it works? Yes, using the Update SQL command.

Let's learn about the syntax and usage of the UPDATE command.

UPDATE command

UPDATE command is used to update any record of data in a table. Following is its general syntax

```
UPDATE table_name SET column_name = new_value WHERE some_condition;
```

WHERE is used to add a condition to any **SQL** query, we will soon study about it in detail.

UPDATE command

Lets take a sample table student,

tudent_id	name	age
101	Adam	15
102	Alex	
103	chris	14

```
UPDATE student SET age=18 WHERE student_id=102;
```

UPDATE command output

S_id	S_Name	age
101	Adam	15
102	Alex	18
103	chris	14

In the above statement, if we do not use the WHERE clause, then our update query will update age for all the columns of the table to 18

Using DELETE SQL command

DELETE command is used to delete data from a table.

Delete all Records from a Table

```
DELETE FROM student;
```

The above command will delete all the records from the table student.

Delete a particular Record from a Table

In our student table if we want to delete a single record, we can use the **WHERE** clause to provide a condition in our **DELETE** statement.

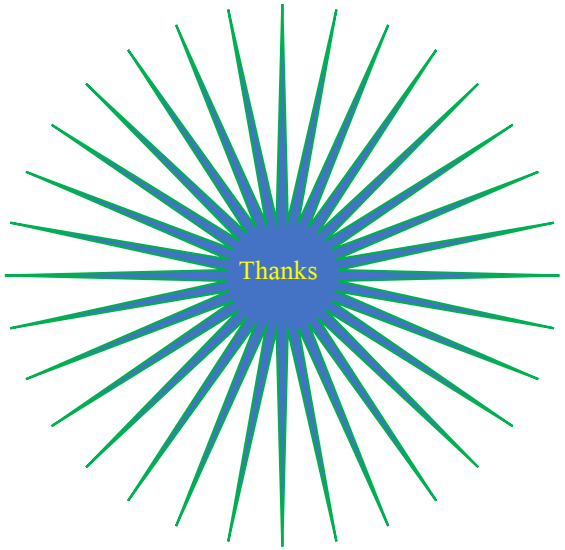
```
DELETE FROM student WHERE s_id=103;
```

Summary

Agenda

1. Summary of previous lecture
2. Introduction to databases(Definition, E.gs of DBMS, Advantages and Disavantages)
3. Introduction to SQL(Definition, E.gs of sql commands used to create database and manuplate its data)

Thank you for Listening



References

Include JavaScript in HTML. Studytonight.com. (n.d.). https://www.studytonight.com/javascript/include-javascript-in-html

JavaScript Data Types. OpenAI Knowledge Base. OpenAI. (2021, September 22)..