

INFO 6210- Data Management and Database Design

Final Project

Title – Creating learning materials to teach SQL

Abstract:

The main aim of creating this learning material is to give basic idea for beginners about the fundamentals of SQL. This learning material will cover all the queries and its related syntax to retrieve, insert, update and delete a value from the database. At the end of this learning material, a list of assessment questions is attached based on this learning material.

What is SQL?

SQL is Structured Query Language. SQL is language which is used to communicate with the databases. SQL is the standard language for relational database management systems (RDBMS). SQL can be used to perform tasks such as query , insert , update and modify data into the database.

This materials will teach you the basics of SQL syntax that are used to create a database, insert , update , modify the database and combine the multiple tables with the help Joins.

Creating an SQL database:

A database is a collection of structured information stored in an organized manner to make it easily accessible for manipulation.

CREATE DATABASE *database_name*;

The above syntax is used to create a database.

For example, to create a database named statistics, the syntax should be

CREATE DATABASE *Statistics*;

First, let's create a table inside the Statistics database. Let's say the name of the table be Match and the Match table contains the following fields inside the table.

CREATE TABLE *Player* (

ID, Name, Matches, Runs, Wickets);

When you execute the following the above syntax, you will get the below table a output. You can see the output that it has 5 columns and no values in Rows.

ID	Name	Matches	Runs	Wickets

INSERT statement:

Now, we the table let's insert the data with the help of INSERT statement.

INSERT INTO *table_name* (Column1, Column2,)

VALUES (value1, value2, ..)

INSERT into *player* **VALUES** (1, "Ben", 150, 1254, 78)

INSERT into *player* **VALUES** (2, "Mark", 233, 3025, 3)

INSERT into *player* **VALUES** (3, "Curran", 78, 7520, 157)

INSERT into *player* **VALUES** (4, "James", 350, 2600, 350)

You can see the output below when you execute the above statement.

ID	Name	Matches	Runs	Wickets
1	Ben	150	1254	78
2	Mark	233	3025	3
3	Curran	78	7520	157
4	James	350	2600	350

Also, in the INSERT INTO you can specify which columns you want to insert data.

INSERT INTO *player* (ID, Name, Matches)

VALUES (5, Andrew, 298)

When you execute the above statement, the updated table looks like

ID	Name	Matches	Runs	Wickets
1	Ben	150	1254	78
2	Mark	233	3025	3
3	Curran	78	7520	157
4	James	350	2600	350
5	Andrew	298	NULL	NULL

SELECT Statement:

The SQL SELECT statement is used to select a particular value from the table.

SYNTAX: **SELECT * FROM** *table_name*;

SELECT * FROM *player*;

When you execute this statement, you will get every value from the table Player

ID	Name	Matches	Runs	Wickets
1	Ben	150	1254	78
2	Mark	233	3025	3
3	Curran	78	7520	157
4	James	350	2600	350

SELECT DISTINCT statement:

The SELECT DISTINCT statement is used to select unique values of the columns from the table.

SYNTAX : **SELECT DISTINCT** *column1 , column2,..*

FROM *table_name*

SELECT Columns:

In SELECT statement, you can choose the specific column to extract.

For example ,

SELECT *ID, Name, Matches*
FROM *player*;

ID	Name	Matches
1	Ben	150
2	Mark	233
3	Curran	78
4	James	350

This statement returns the specified columns and all the rows from the table.

COUNTING Rows:

Number of rows in the table can be found with the help of Count Function.

SYNTAX: **SELECT COUNT(*) FROM** *table_name* ;

SELECT COUNT(*) FROM *Player*;

	COUNT(*)
1	4

WHERE clause:

The WHERE clause is used to extract the records that fulfil the specific conditions.

SYNTAX: **SELECT * FROM** *table_name*

WHERE *condition*;

SELECT * FROM *player*

WHERE *ID= 3*;

It gives only the row where ID= 3

ID	Name	Matches	Runs	Wickets
3	Curran	78	7520	157

SQL AND operator:

AND operator are used to extract records with more than one condition It filters records that satisfy all the conditions.

SELECT * FROM *player*

WHERE *runs> 2000 AND wickets >100*;

It gives records of people with runs greater than 2000 and wickets greater than 100

ID	Name	Matches	Runs	Wickets
3	Curran	78	7520	157
4	James	350	2600	350

SQL OR operator:

It filters records that satisfy at least any one of the given condition.

```
SELECT * FROM player  
WHERE runs > 2000 OR wickets > 100;
```

ID	Name	Matches	Runs	Wickets
2	Mark	233	3025	3
3	Curran	78	7520	157
4	James	350	2600	350

ORDER BY:

The ORDER BY statement is used to sort the result in either Ascending or descending order.

```
SELECT * FROM player  
ORDER BY name;
```

ID	Name	Matches	Runs	Wickets
1	Ben	150	1254	78
3	Curran	78	7520	157
4	James	350	2600	350
2	Mark	233	3025	3

We can see that the name of the players are arranged in ascending order.

Note: SQL ORDER BY will sort everything in ascending order unless it is mentioned.

```
SELECT * FROM player  
ORDER BY name DESC;
```

ID	Name	Matches	Runs	Wickets
2	Mark	233	3025	3
4	James	350	2600	350
3	Curran	78	7520	157
1	Ben	150	1254	78

The name of the players is arranged in descending order.

SQL UPDATE:

The UPDATE Statement is used to modify the existing records in a table.

UPDATE *table_name*

SET *Column1=Value1 , Column2=Value2*

WHERE *condition;*

The WHERE clause in update in UPDATE statement is very important because when the UPDATE statement executed without the WHERE statement will update the whole table.

UPDATE *player*

SET *Name = "Smith" , Matches = 189*

WHERE *ID= 4;*

ID	Name	Matches	Runs	Wickets
1	Ben	150	1254	78
2	Mark	233	3025	3
3	Curran	78	7520	157
4	Smith	189	2600	350

SQL DELETE:

DELETE statement is used to delete existing record from the table.

DELETE FROM *table_name*

WHERE *condition;*

The WHERE clause in DELETE statement is very important because when the DELETE statement is executed without the WHERE condition, the whole table will get deleted.

DELETE FROM *player*

WHERE *id=2;*

ID	Name	Matches	Runs	Wickets
1	Ben	150	1254	78
3	Curran	78	7520	157
4	James	350	2600	350

Here, we can see that the row with ID= 2 is deleted.

LIMIT statement:

LIMIT is used to specify the number record to return in the result.

```
SELECT * FROM player
```

```
LIMIT 2;
```

ID	Name	Matches	Runs	Wickets
1	Ben	150	1254	78
2	Mark	233	3025	3

Here we can see that LIMIT 2 returns only 2 rows in the result.

SQL BETWEEN operator:

The BETWEEN operator is used find the values within the given range from the table.

```
SELECT * FROM table_name
```

```
WHERE column_name BETWEEN value 1 AND value 2 ;
```

```
SELECT * FROM player
```

```
WHERE wickets BETWEEN 0 AND 100;
```

ID	Name	Matches	Runs	Wickets
1	Ben	150	1254	78
2	Mark	233	3025	3

The query returns only values that are BETWEEN 0 and 100 from the wickets column.

Assessment questions:

1. Write a SQL query to fetch all the "Name" of the customers from the CUSTOMER table in ascending order?
2. Write a SQL query to fetch all the columns from the CUSTOMER table?
3. With the help of SQL, fetch "FirstName", "LastName" , "State" from the table named Customer?
4. Write a SQL query to fetch all the columns where STATE= NY and COUNTRY = USA in the Customer table?
5. Write a SQL query to update CITY = "France" for ID = 10 in the Customer table?
6. Write a SQL query, to count all the distinct CITY in the Customer table?
7. Write a SQL query, to get all the information of the products with "PRICE" between 100 and 200 in the table named Products?
8. Write a SQL query to insert a new data in the customer table with values NAME= "AK", CITY = "Boston"?
9. How do you select all records of the products with ID=225 in the table named Products?
10. Write a SQL query to delete all the information with Country= "Japan" in the Products table?
11. How do you create a database named "Sales_Data"?
12. Write a SQL query, to fetch first 5 records in the Products table?
13. Write a SQL query to fetch all the columns where STATE= "CA" or COUNTRY = "Asia" in the Customer table?
14. Write a SQL query, to count distinct number "CITY" in the Customer table?

Contributions:

Self-Efforts: 50%

External Source: 50%

Citations:

<https://www.w3schools.com/sql/default.asp>

<https://www.linkedin.com/learning/sql-essential-training-3/creating-a-subselect?u=74653650>

Licence:

Copyright 2020 Kumaran Nehru Uthra. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.