

# **COMPUTER ORGANIZATION AND ARCHITECTURE**

## **Lecture 11**

### **Advanced Data Manipulation Language**

Dr Victoria Mukami

## **INTRODUCTION**

The last lecture introduced the data manipulation language. Specifically, the INSERT and UPDATE statements were introduced. For this lecture, we introduce the SELECT and the VIEW statement. For the SELECT statement, we will focus on the WHERE, statement. We will also learn about the CREATE VIEW statement that will be used to generate views.

### **Learning objectives**

By the end of this topic, you should be able to:

1. Understand the SELECT statement when querying data
2. Use the SELECT statement in combination with the WHERE clause
3. Create a table VIEW

## **OVERVIEW**

Previously we learnt the INSERT and UPDATE statements. These statements were used to insert data into the tables that were created during lecture 8. One of the core objectives of a database is to allow the database administrator or other users to query data. There are various reasons why one might want to query data. Key among them is to be able to make various decisions within an organization. For instance, the Bustani registrar might want to know how many students are expected to graduate in the next graduation ceremony. The human resource manager might want to know how many instructors have completed a higher degree. All these are queries that can be done to a database and will return values. The SELECT statement is key in providing query results. We review the SELECT statement next.

## **INTRODUCTION**

Throughout lectures 8, 9, and 10, we have referred to the tables that were used to create the Employee, Instructor and Department table. The table details are listed in Tables 1-3 as shown. We will refer to this data to run our SELECT statements.

Table 1: Employee Data Table

Emp_Num	FName	LName	Gender	DOB	MStatus
EMP001	Agnes	Njue	Female	12/09/1997	Single
EMP002	Will	Omondi	Male	24/08/1989	Married
EMP003	Carol	Njagi	Female	09/01/1993	Single
EMP004	Mary	Munene	Female	06/08/1998	Married
EMP005	Cathy	Resley	Male	16/04/1989	Married
EMP006	Abdul	Muita	Male	30/07/1988	Single
EMP007	Steve	Kimuri	Male	02/03/1999	Married
EMP008	Ruth	Kimuli	Female	07/04/1996	Single

Table 2: Instructor Data Table

ID	Emp_Num	Dep_ID	Post
1	EMP001	CIT	Lecturer
2	EMP002	MTH	Chairman
3	EMP003	LAW	Dean
4	EMP004	CIT	Lecturer
5	EMP005	BUS	Lecturer
6	EMP006	EDU	Senior Lecturer
7	EMP007	MTH	Lecturer
8	EMP008	PSY	Senior Lecturer

Table 3: Department Data Table

Dep_ID	Dep_Name	School
CIT	Computing and Information Technology	SPAS
MTH	Statistics and Mathematics	SPAS
BUS	Business	Business
EDU	Education	Education
PSY	Psychology	Humanities
LAW	Law	Law

## SELECT STATEMENT

The SELECT statement is used to query data within a database [2]. The SELECT statement is a complex statement that has several clauses that can be used within the statement. We will only focus on two of these clauses, the WHERE, and GROUP BY clauses. The SELECT syntax is as follows

```
SELECT fieldname1  
  
FROM tablename;
```

If you remember when we introduced SQL we showed a SELECT statement that uses the wildcard (\*) symbol. The wildcard symbol allows us to return all the columns. If we wanted to show all the records from the department table we would use the following SQL statement.

```
SELECT *  
  
FROM Department;
```

### Output

```
mysql> use Bustani  
Database changed  
mysql> SELECT *  
-> FROM Department;  
+-----+-----+-----+  
| Dep_ID | Dep_Name | School |  
+-----+-----+-----+  
| BUS    | Business | Business |  
| CIT    | Computing and Information Technology | SPAS |  
| EDU    | Education | Education |  
| LAW    | Law | Law |  
| MTH    | Statistics and Mathematics | SPAS |  
| PSY    | Psychology | Humanities |  
+-----+-----+-----+  
6 rows in set (0.02 sec)
```

As shown in the output all the table contents are listed in tabular form. Assume we wanted to show only the Employee Number, DOB and Gender from the Employee table, we would use the following

```
SELECT Emp_Num, Gender, DOB  
  
FROM Employee;
```

## Output

```
mysql> SELECT Emp_Num, Gender, DOB
-> FROM Employee;
+-----+-----+-----+
| Emp_Num | Gender | DOB      |
+-----+-----+-----+
| EMP001  | Female | 1997-09-12 |
| EMP002  | Male   | 1989-08-24 |
| EMP003  | Female | 1993-09-01 |
| EMP004  | Female | 1998-08-06 |
| EMP005  | Male   | 1989-04-16 |
| EMP006  | Male   | 1988-07-30 |
| EMP007  | Male   | 1999-03-02 |
| EMP008  | Female | 1996-04-07 |
+-----+-----+-----+
8 rows in set (0.02 sec)
```

## DISTINCT Clause

The select statement has a DISTINCT clause that allows for unique values to be returned. The syntax is as follows

```
SELECT DISTINCT columnname
FROM Tablename;
```

If we selected all the gender records, we would get the following data.

```
mysql> SELECT Gender
-> FROM Employee;
+-----+
| Gender |
+-----+
| Female |
| Male   |
| Female |
| Female |
| Male   |
| Male   |
| Male   |
| Female |
+-----+
8 rows in set (0.00 sec)
```

Now let's use the DISTINCT clause using the following statement

```
SELECT DISTINCT Gender
FROM Employee;
```

## Output

```
mysql> SELECT DISTINCT Gender
-> FROM Employee;
+-----+
| Gender |
+-----+
| Female |
| Male   |
+-----+
2 rows in set (0.01 sec)
```

As seen the DISTINCT clause only shows the unique values from the Gender column.

## NATURAL JOIN

The Natural Join is a clause used together with the SELECT statement used to connect two tables and return the query results. The tables require a relationship that had previously been created. The following statement will load Emp\_Num, FName, LName and Dep\_ID from two tables, the employee and instructor tables.

```
SELECT Emp_Num, FName, LName, Dep_ID
FROM Employee NATURAL JOIN Instructor;
```

## Output

```
mysql> SELECT Emp_Num, FName, LName, Dep_ID
-> FROM Employee NATURAL JOIN Instructor;
+-----+-----+-----+-----+
| Emp_Num | FName | LName | Dep_ID |
+-----+-----+-----+-----+
| EMP001  | Agnes | Njue  | CIT    |
| EMP002  | Will  | Omondi | MTH    |
| EMP003  | Carol | Njagi | LAW    |
| EMP004  | Mary  | Munene | CIT    |
| EMP005  | Cathy | Resley | BUS    |
| EMP006  | Abdul | Muita | EDU    |
| EMP007  | Steve | Kimuri | MTH    |
| EMP008  | Ruth  | Kimuli | PSY    |
+-----+-----+-----+-----+
8 rows in set (0.02 sec)
```

As seen the output returns values from both tables. Let us now try a NATURAL JOIN with three tables.

```
SELECT Emp_Num, FName, LName, Dep_ID, Post, School
FROM Employee NATURAL JOIN Instructor NATURAL JOIN Department;
```

## Output

```
mysql> SELECT Emp_Num, FName, LName, Dep_ID, Post, School
-> FROM Employee NATURAL JOIN Instructor NATURAL JOIN Department;
+-----+-----+-----+-----+-----+-----+
| Emp_Num | FName | LName | Dep_ID | Post      | School  |
+-----+-----+-----+-----+-----+-----+
| EMP005 | Cathy | Resley | BUS    | Lecturer | Business|
| EMP001 | Agnes | Njue   | CIT    | Lecturer | SPAS    |
| EMP004 | Mary  | Munene | CIT    | Lecturer | SPAS    |
| EMP006 | Abdul | Muita  | EDU    | Senior Lecturer | Education|
| EMP003 | Carol | Njagi  | LAW    | Dean     | Law     |
| EMP002 | Will  | Omondi | MTH    | Chairman | SPAS    |
| EMP007 | Steve | Kimuri | MTH    | Lecturer | SPAS    |
| EMP008 | Ruth  | Kimuli | PSY    | Senior Lecturer | Humanities|
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

As shown in the output the results are from three tables. Additional joins exist, however, these will be introduced in the advanced database class.

## WHERE Clause

The next clause we review is the WHERE clause. While the SELECT statement is used to query records, the WHERE clause used in combination with the SELECT clause is used to filter out the records based on a specific condition. The syntax is as follows

```
SELECT fieldname
FROM tablename
WHERE condition;
```

The condition could be criteria that you are only interested in. For instance, assume that you would like to list all the male employees at Bustani College. The following statement would be used

```
SELECT EMP_Num, FName, Gender
FROM Employee
WHERE Gender = "Male";
```

## Output

```
mysql> SELECT Emp_Num, FName, Gender
-> FROM Employee
-> WHERE Gender="Male";
+-----+-----+-----+
| Emp_Num | FName | Gender |
+-----+-----+-----+
| EMP002 | Will  | Male   |
| EMP005 | Cathy | Male   |
| EMP006 | Abdul | Male   |
| EMP007 | Steve | Male   |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

As shown only the Male employees are listed within the results. Double quotes are required around conditions that are character datatypes but not required for integers, float or decimal numbers. Several mathematical formats can be used with WHERE such as =, >=, <=, BETWEEN among others. These are reviewed during the advanced database class.

## ORDER BY Clause

The ORDER BY clause is used to sort data in either an ascending or descending manner. By default, ORDER BY will sort data in an ascending manner [1]. The syntax is as follows:

```
SELECT fieldname
FROM tablename
ORDER BY fieldname;
```

As seen in the code, the fieldname would be representative of what data you want to see and filter by. Assuming we wanted to view the Emp\_Num, FName and Gender from the Employee table while sorting based on FName, we would have the following syntax.

```
SELECT EMP_Num, FName, Gender
FROM Employee
ORDER BY FName;
```

## Output

```
mysql> SELECT EMP_Num, FName, Gender
-> FROM Employee
-> ORDER BY FName;
+-----+-----+-----+
| EMP_Num | FName | Gender |
+-----+-----+-----+
| EMP006  | Abdul | Male   |
| EMP001  | Agnes | Female |
| EMP003  | Carol | Female |
| EMP005  | Cathy | Male   |
| EMP004  | Mary  | Female |
| EMP008  | Ruth  | Female |
| EMP007  | Steve | Male   |
| EMP002  | Will  | Male   |
+-----+-----+-----+
8 rows in set (0.01 sec)

mysql> 
```

## VIEW Statement

The VIEW statement is used to create a virtual table that has columns and rows just like a normal table has. The VIEW statement is based on the SELECT statement and can contain a combination of columns and one to many tables [1]. The syntax for the VIEW statement is as follows [1]:

```
CREATE VIEW viewname AS
SELECT query;
```

Let us create a view from one of our previous SELECT statements. The select statement we will use is as follows

```
SELECT Emp_Num, Fname, LName, Dep_ID
FROM Employee NATURAL JOIN Instructor;
```

Using the SELECT query above we will then create a view called Bustani\_Employees in the following statement

```
CREATE VIEW Bustani_Employees AS
SELECT Emp_Num, Fname, LName, Dep_ID
FROM Employee NATURAL JOIN Instructor;
```

## Output

```
mysql> CREATE VIEW Bustani_Employees AS
-> SELECT Emp_Num, Fname, LName, Dep_ID
-> FROM Employee NATURAL JOIN Instructor;
Query OK, 0 rows affected (0.02 sec)
```

Once you create the view, you can display the results by using the following statement

```
SELECT * FROM Bustani_Employees;
```

## Output

```
mysql> SELECT * FROM Bustani_Employees;
+-----+-----+-----+-----+
| Emp_Num | Fname | LName | Dep_ID |
+-----+-----+-----+-----+
| EMP001 | Agnes | Njue  | CIT    |
| EMP002 | Will  | Omondi | MTH    |
| EMP003 | Carol | Njagi  | LAW    |
| EMP004 | Mary  | Munene | CIT    |
| EMP005 | Cathy | Resley | BUS    |
| EMP006 | Abdul | Muita  | EDU    |
| EMP007 | Steve | Kimuri | MTH    |
| EMP008 | Ruth  | Kimuli | PSY    |
+-----+-----+-----+-----+
8 rows in set (0.01 sec)
```

We can see all the views that have been created within the databases by using the following syntax.

```
SHOW FULL TABLES
WHERE table_type = 'VIEW';
```

Output

```
mysql> SHOW FULL TABLES
-> WHERE table_type = 'VIEW';
+-----+-----+
| Tables_in_bustani | Table_type |
+-----+-----+
| bustani_employees | VIEW      |
+-----+-----+
1 row in set (0.03 sec)

mysql> SHOW FULL TABLES;
+-----+-----+
| Tables_in_bustani | Table_type |
+-----+-----+
| bustani_employees | VIEW      |
| department        | BASE TABLE |
| employee           | BASE TABLE |
| instructor         | BASE TABLE |
| test               | BASE TABLE |
+-----+-----+
5 rows in set (0.00 sec)
```

As shown within the output there are two statements. The first statement shows only the VIEWS while the second shows all the tables but categorizes each as either a BASE TABLE or a VIEW. To delete a VIEW one would use the DROP command.

## SUMMARY

This lecture has been a review of the SELECT and VIEW statements. The SELECT statement was reviewed and the FROM, WHERE, DISTINCT AND NATURAL JOIN were explored. The FROM clause was used to pick the tables which the query would be displayed from. The WHERE clause was used to filter out the query results while DISTINCT was used to show only the unique values. The NATURAL JOIN was used to return values from tables that had a relationship, in our case all the tables were related. Finally, the CREATE VIEW statement was explored and a VIEW was created. The next lecture is an overview of database security and administration.

## DISCUSSION TOPIC

Following through with the Lands Office case study, we will go ahead and run some queries within our data. Use the tables you have created to come up with Surveyors

from a specific region, the various services offered by a specific agent and list all the surveyors in your database. Finally, create a VIEW of the surveyors from a specific region.

## **REFERENCES**

[1] Database systems: design, implementation, and management, Coronel, C., & Morris, S, Cengage Learning, 2019.

[2] Database Systems: A Practical Approach to Design, Implementation, and Management, Connolly, T., & Begg, C., Pearson, 2015.

[3] Fundamentals of database systems, Elmasri, R., & Navathe, S. B., Pearson Education Limited, 2016.