TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES VISAYAS
Capt. Sabi St., City of Talisay, Negros Occidental

College of Engineering Technology
Office of the Program Coordinator

LEARNING MODULE

COMP 312: Advanced DataBase

DEPARTMENT: COMPUTER ENGINEERING TECHNOLOGY

COMPILED BY:

JONATHAN L. MANDIA

2020

VISION

The Technological University of the Philippines shall be the premier state university with recognized excellence in engineering and technology at par with leading universities in the ASEAN region.

MISSION

The University shall provide higher and advanced vocational, technical, industrial, technological and professional education and training in industries and technology, and in practical arts leading to certificates, diplomas and degrees.

It shall provide progressive leadership in applied research, developmental studies in technical, industrial, and technological fields and production using indigenous materials; effect technology transfer in the countryside; and assist in the development of small-and-medium scale industries in identified growth center. (Reference: P.D. No. 1518, Section 2)

QUALITY POLICY

The Technological University of the Philippines shall commit to provide quality higher and advanced technological education; conduct relevant research and extension projects; continually improve its value to customers through enhancement of personnel competence and effective quality management system compliant to statutory and regulatory requirements; and adhere to its core values.

CORE VALUES

T - Transparent and participatory governance
U - Unity in the pursuit of TUP mission, goals, and objectives
P - Professionalism in the discharge of quality service
I - Integrity and commitment to maintain the good name of the University
A - Accountability for individual and organizational quality performance
N - Nationalism through tangible contribution to the rapid economic growth of the country
S - Shared responsibility, hard work, and resourcefulness in compliance to the mandates of the university

TABLE OF CONTENTS

COURSE DESCRIPTION

This subject aims to introduce with the students the role of computers in society. Topics includes evolution of computer systems; development of digital systems, hardware, operating systems and software. It also talks about the internet, overview of database, distributive; real time systems; careers and ethics of computer professionals with emphasis on the principles, underlying concepts and emerging technologies of computer technology.

COURSE OUTCOMES

- ✓ Identify the different database models.
- ✓ Define the hierarchical structure, chart, and path.
- ✓ Define XML and its structure.
- ✓ Familiarize with relational database concepts.
- ✓ Familiarize with SQL data definition commands.
- ✓ Identify the use of entity-relationship diagrams.
- ✓ Familiarize with MS SQL Server, MySQL, and Oracle.

GENERAL GUIDELINES/CLASS RULES

1. Make-up exams and quizzes will only be given with prior approval of the professor and under exceptional circumstances. For excused absences during the exam, the university policy will be followed.

2. Students are not allowed to leave the classroom once the class has started, unless extremely necessary. Students who leave the classroom without any valid reason will be marked absent.

3. Students are expected to comply strictly with the university's rule on dress code, class tardiness and attendance.

4. Cell phones or any e-gadgets must be turned off or put in a silent mode during class hours.

5. Late homework or projects will not be accepted. Students are expected to maintain complete honesty and integrity in their academic work. Acts of academic dishonesty, such as cheating, plagiarism, or inappropriately using the work of others to satisfy course requirements, will not be tolerated and may result in failure of the affected assignments and/or failure of this class.

GRADING SYSTEM

The student will be graded according to the following:

Average of examinations	- 50%
Average of weekly assessment	- 50%

Prelim Grade: (Average of Weekly Assessments from Week 1 to 4) X 0.70 + (MTE x 0.30)

Midterm Grade: (Average of Weekly Assessments from Week 5 to 10) X 0.70 + (MTE x .0.30)
End term Grade: (Average of Weekly Assessments from Week 11 to 14) X 0.70 + (ETE x .0.40)
Final Grade: (Prelim Grade + Midterm Grade + End term Grade) / 3

The passing grade for this course is 5.0.

LEARNING GUIDE

Week No.: __6-7__

TOPIC/S

- ✓ Insert Statement
- ✓ Update Statement
- ✓ Select Statement

EXPECTED COMPETENCES

Upon completing this Learning Module, you will be able to:

- ✓ Analyze and understand the structure and concepts of insert statement.
- ✓ Analyze and understand the structure and concepts of update statement.
- ✓ Analyze and understand the structure and concepts of select statement.

CONTENT/TECHNICAL INFORMATION

Select Statement

A SELECT statement retrieves zero or more rows from one or more database tables or database views. In most applications, SELECT is the most commonly used data manipulation language (DML) command. As SQL is a declarative programming language, SELECT queries specify a result set, but do not specify how to calculate it. The database translates the query into a "query plan" which may vary between executions, database versions and database software. This functionality is called the "query optimizer" as it is responsible for finding the best possible execution plan for the query, within applicable constraints.

Syntax:

```
Select column_name1, column_name2
From table_name
```

or

```
Select *
From table_name
```

Note: The (*) means it will get all column names or field name on a certain table. Check out your laboratory module on this activity.

Insert Statement

The SQL **INSERT INTO** Statement is used to add new rows of data to a table in the database.

Syntax:

There are two basic syntaxes of the INSERT INTO statement which are shown below.

```
INSERT INTO TABLE_NAME (column1, column2, column3,...columnN)
VALUES (value1, value2, value3,...valueN);
```

Here, column1, column2, column3,...columnN are the names of the columns in the table into which you want to insert the data.

You may not need to specify the column(s) name in the SQL query if you are adding values for all the columns of the table. But make sure the order of the values is in the same order as the columns in the table.

The **SQL INSERT INTO** syntax will be as follows

```
INSERT INTO TABLE_NAME VALUES (value1, value2, value3,...valueN);
```

Note: Check out your laboratory module on this activity.

Update Statement

An SQL UPDATE statement changes the data of one or more records in a table. Either all the rows can be updated, or a subset may be chosen using a condition.

The UPDATE statement has the following form:

```
UPDATE table_name SET column_name = value [, column_name = value
...] [WHERE condition]
```

For the UPDATE to be successful, the user must have data manipulation privileges (UPDATE privilege) on the table or column and the updated value must not conflict with all the applicable constraints (such as primary keys, unique indexes, CHECK constraints, and NOT NULL constraints).

In some databases, such as PostgreSQL, when a FROM clause is present, what essentially happens is that the target table is joined to the tables mentioned in the list, and each output row of the join represents an update operation for the target table. When using FROM, one should ensure that the join produces at most one output row for each row to be modified. In other words, a target row shouldn't join to more than one row from the other table(s). If it does, then only one of the join rows will be used to update the target row, but which one will be used is not readily predictable.

Because of this indeterminacy, referencing other tables only within sub-selects is safer, though often harder to read and slower than using a join.

Note: Check out your laboratory module on this activity.

PROGRESS CHECK

Activity 1: Essay

RUBRIC:

- Clarity	5
- Theme	5
- Organized Explanation	5
- Application	5
Total	**20**

Question:

- On your point of view, how the insert statement does is greatly affect the whole process on the given entities:

    a. Business software
    b. Game development
    c. Entertainment

- Explain this in five sentences each on paragraph form.

REFERENCE

Wikipedia (2020). Select (SQL).
Retrieved from
https://en.wikipedia.org/wiki/Select_(SQL)#:~:text=The%20SQL%20SELECT%20statement%20returns,manipulation%20language%20(DML)%20command.

Wikipedia (2020). Update (SQL).
Retrieved from
https://en.wikipedia.org/wiki/Update_(SQL)#:~:text=From%20Wikipedia%2C%20the%20free%20encyclopedia,column_name%20%3D%20value%20...%5D

LEARNING GUIDE

Week No.: __8__

TOPIC/S

- ✓ Delete Statement
- ✓ Aggregate Functions
- ✓ Sub-Query

EXPECTED COMPETENCIES

Upon completing this module the student can be able to:

- ✓ Understand the concepts of delete statement.
- ✓ Filter the necessary records on deleting a record.
- ✓ Use the different aggregate functions on record.
- ✓ Understand the concepts using sub-query.

CONTENT/TECHNICAL INFORMATION

Delete statement

The DELETE Statement in SQL is used to delete existing records from a table. We can delete a single record or multiple records depending on the condition we specify in the WHERE clause.

Basic Syntax:

```
DELETE FROM table_name WHERE some_condition;

table_name: name of the table
some_condition: condition to choose particular record.
```

Note: We can delete single as well as multiple records depending on the condition we provide in WHERE clause. If we omit the WHERE clause then all of the records will be deleted and the table will be empty.

Aggregate Functions

An aggregate function performs a calculation one or more values and returns a single value. The aggregate function is often used with the GROUP BY clause and HAVING clause of the SELECT statement.

The following table shows the SQL Server aggregate functions:

Aggregate function	Description
AVG	The AVG() aggregate function calculates the average of non-NULL values in a set.
CHECKSUM_AGG	The CHECKSUM_AGG() function calculates a checksum value based on a group of rows.
COUNT	The COUNT() aggregate function returns the number of rows in a group, including rows with NULL values.
COUNT_BIG	The COUNT_BIG() aggregate function returns the number of rows (with BIGINT data type) in a group, including rows with NULL values.
MAX	The MAX() aggregate function returns the highest value (maximum) in a set of non-NULL values.
MIN	The MIN() aggregate function returns the lowest value (minimum) in a set of non-NULL values.
STDEV	The STDEV() function returns the statistical standard deviation of all values provided in the expression based on a sample of the data population.
STDEVP	The STDEVP() function also returns the standard deviation for all values in the provided expression, but does so based on the entire data population.
SUM	The SUM() aggregate function returns the summation of all non-NULL values a set.
VAR	The VAR() function returns the statistical variance of values in an expression based on a sample of the specified population.
VARP	The VARP() function returns the statistical variance of values in an expression but does so based on the entire data population.

Sub-query

In SQL a Subquery can be simply defined as a query within another query. In other words we can say that a Subquery is a query that is embedded in WHERE clause of another SQL query.

Important rules for Subqueries:

- You can place the Subquery in a number of SQL clauses: WHERE clause, HAVING clause, FROM clause. Subqueries can be used with SELECT, UPDATE, INSERT, DELETE statements along with expression operator. It could be equality operator or comparison operator such as =, >, =, <= and Like operator.
- A subquery is a query within another query. The outer query is called as **main query** and inner query is called as **subquery**.
- The subquery generally executes first, and its output is used to complete the query condition for the main or outer query.
- Subquery must be enclosed in parentheses.
- Subqueries are on the right side of the comparison operator.
- ORDER BY command **cannot** be used in a Subquery. GROUPBY command can be used to perform same function as ORDER BY command.
- Use single-row operators with single row Subqueries. Use multiple-row operators with multiple-row Subqueries.

Syntax:

```
SELECT column_name

FROM table_name

WHERE column_name expression operator
    ( SELECT COLUMN_NAME  from TABLE_NAME   WHERE ... );
```

PROGRESS CHECK

Activity 1: Essay

RUBRIC:

- Clarity	5
- Theme	5
- Organized Explanation	5
- Application	5
Total	**20**

Questions:

1. Is it necessary to delete always a record? Yes/No. Support your answer.
2. Discuss the relevance of aggregate function in an application like MS Excel?

LIST OF REFERENCES:

GeeksforGeeks (2019). SQL Delete Statement.
Retrieved from
https://www.geeksforgeeks.org/sql-delete-statement/

GeeksforGeeks (2020). SQL Subquery.
Retrieved from
https://www.geeksforgeeks.org/sql-subquery/