جامــــعة المـــستقبل
AL MUSTAQBAL UNIVERSITY

قســــــم الامـــــــــــن
الــــــــــــــــــســيبرانـــــــــــــــــــي
DEPARTMENT OF CYBER SECURITY

SUBJECT:

DATABASE

CLASS:

SECOND

LECTURER:

M.SC. ALI HAIDER ALAZAM

M.SC. MOHAMMAD BAQER HALEEM

LECTURE: (3)

INTRODUCTION TO THE SQL LANGUAGE

History and Evolution

SQL (Structured Query Language) was originally developed at IBM in the 1970s. Since then, it has been standardized by ANSI and ISO.

Components of SQL

SQL is not just a query language; it is a multi-part language:

1- Data-Definition Language (DDL): For defining schemas, deleting relations, and modifying schemas.
2- Data-Manipulation Language (DML): For querying information and inserting/deleting/modifying tuples.
3- Integrity Constraints: Specifying rules for data consistency.
4- View Definition: Defining "virtual" tables.
5- Transaction Control: Defining the beginning and end of transactions.
6- Embedded & Dynamic SQL: How SQL interacts with programming languages (Java, Python, C++).
7- Authorization: Specifying access rights to users.

Data Definition Language (DDL) Detailed

1- Standard Data Types

- char(n): Fixed-length character string.
- varchar(n): Variable-length character string.
- int / integer: A finite set of integers.
- smallint: Small integer.
- numeric(p, d): Fixed-point number (Precision p, Scale d).
- real, double precision: Floating-point numbers.
- float(n): Floating-point number with precision of at least n digits.

2- Table Creation & Integrity Constraints

Creating a table involves defining the schema and the rules that govern the data.

```
create table department (
    dept_name   varchar(20),
    building    varchar(15),
    budget      numeric(12,2) check (budget > 0),
    primary key (dept_name)
);
```

Key Concepts explained:

- Primary Key: Attributes that must be unique and non-null.
- Foreign Key: Ensures that a value in one table exists in another (Referential Integrity).
- Not Null: Ensures an attribute cannot have a null value.

Basic Query Structure (Select - From - Where)

This is the heart of SQL. It is based on the Relational Algebra operations.

1- The Select Clause

The select clause is used to list the attributes desired in the query result.

Handling Duplicates: SQL relations are multisets (bags). To remove duplicates, use distinct.

Example: select distinct dept_name from instructor;

Arithmetic Expressions: You can perform calculations within the select.

select ID, name, salary * 1.1 from instructor;

2- The Where Clause

Used to filter the results based on specific conditions.

- Logical Connectives: and, or, not.
- Comparison Operators: <, <=, >, >=, =, <>.
- Between Operator: where salary between 60000 and 80000;

Join Operations: Combining Multiple Relations

One of the most powerful features of SQL.

Natural Join

The natural join operation matches tuples with the same values for all attributes that have the same name in both relation schemas.

Example: select name, course_id from instructor natural join teaches;

Risk of Natural Join: If two tables have columns with the same name that are not related (e.g., ID in both student and instructor), the natural join might return empty results.

Set Operations (Union, Intersect, Except)

SQL set operations automatically eliminate duplicates unless all is specified.

1- Union: Combines results.
2- Intersect: Returns common tuples.
3- Except: Returns tuples in the first result but not the second.

Aggregate Functions and Grouping

Aggregate functions operate on multiple values to return a single value.

1- Basic Functions

- count(*): Counts the number of rows.
- avg(attribute): Calculates the average.
- sum(attribute): Total sum.
- min and max: Extremes.

2- Group By Clause

Used to partition the relation into groups.

Example: Find the average salary of instructors in each department:

```
select dept_name, avg(salary) as avg_salary
from instructor
group by dept_name;
```

3- Having Clause

Used to apply conditions to the groups rather than individual tuples.

Condition: Find departments where the average salary is greater than $42,000.

```
select dept_name, avg(salary)
from instructor
group by dept_name
having avg(salary) > 42000;
```

Null Values and Three-Valued Logic

Null represents unknown or non-existent values.

- null + 5 = null
- Three-Valued Logic:
  - true and unknown = unknown
  - false and unknown = false
  - unknown or true = true
  - unknown or false = unknown

Nested Subqueries

A subquery is a select-from-where expression that is nested within another query.

- Set Membership: Using in and not in.
- Set Comparison: Using some and all.

Example: where salary > all (select salary from instructor where dept_name = 'Biology');

- Test for Empty Relations: Using exists and not exists.

Modification of the Database

1- Delete: delete from instructor where dept_name in (select dept_name from department where building = 'Watson');
2- Insert: insert into student values ('12345', 'Smith', 'Physics', 144);
3- Update: Using the case statement for conditional updates.

```
update instructor
set salary = case
    when salary <= 100000 then salary * 1.05
    else salary * 1.03
end;
```