Cairo University
Faculty of Computers and Information
IS221 Course
Database Systems 1

Section 2
SQL DML 1 (One table only)
(Insert, Update, Delete, Simple Select On One Table)

Part 1: Quick Introduction:

- Introduce to students SQL"Structured Query Language" is the standard query language to query relational DB.
- SQL {
  - DDL:Data Definition Language
  - DML: Data Manipulation Language
  - DCL: Data Control Language

Consider University Database with the following schema:

Student (SSN, Name, City, DOB, Major,GPA)
Course (CrsCode, Name)
Registered (SSN, CrsCode, Semester, Year)
Department (DCode, Name)

Part 2: Insert, Update and Delete tuples:

1 . Insert New Rows

```
INSERT INTO <tablename> [<column list>]
VALUES (<expression>{[,<expression>]})
```

Example 1:

- Add a new student record into " Student" relation with the following data values : ID= 201, name = "Adam", city = "Cairo", DOB = 1995, Major = "IT", GPA = 3.6

```
Insert into student (ssn, name, city, DOB, major,GPA)

Values (201, 'Adam', 'Cairo' , 1995, 'IT', 3.6)
```

Important notes:

- Primary key cannot be repeated nor assigned no value.
- To leave an attribute value blank, Nulls should be allowed for this attribute in the table design.
- Correct data types are important.
- If you want to change the order of the inserted columns or you will leave some attributes blank you should specify the columns
- You can specify values to be inserted without specifying the column names because SQL assumes that the order in which the values are entered is the same as the default sequence of the columns.

Example 2 (not filling all the attributes) :

```
insert into Student (ssn, name, major) Values ( 211, 'Neveen', 'IS')
```

Example 3(attribute list is not mentioned) – (primary key repetition):

```
insert into Course values ('IS221','Information Retrieval')
```

"Leaving PK blank or entry of wrong data types will lead to error at excution"

2. Update values in tuples:

UPDATE <tablename>
SET <column name>=<expression>[,<column name>=<expression>]
[WHERE <Condition>]

Example 1(update all tuples in the relation):

- Increase the GPA of all students by 5%

```
Update  Student
Set        GPA = GPA*1.05
```

Example 2 (Using where clause):

- Change the GPA of the student named Ayman to be 3.6

```
Update   Student
Set        GPA = 3.6
Where   Name = 'Ayman'
```

Example 3 (Using where clause and selecting the right table to be updated):

- Student Wafaa SSN=110 changed her course registeration from DB sysems1 course with code IS221 to DB systems 2 course with code IS312

```
Update registered
Set  CrsCode = 'IS312'
Where SSN = 110
And CrsCode = 'IS221'
```

Example 4 (Updating more than one column with where clause):

- Change the City and Major of Student whose SSN = 109 to Giza and IT

```
Update Student
Set City = 'Giza', Major = 'IT'
Where SSN = 109
```

3. Delete from Table :

DELETE
FROM <table name>
[WHERE <Condition>]

Example1 (Delete all data in a table) "Don't try it on real data"

- Clear the data on student-course registration

```
Delete
From Registered
```

Example 2 (using where clause):

- Remove the registration of student Ahmed whose SSN = 101 in Database Systems2 whose code = "IS312"

```
Delete
From Registered
Where SSN = 101
And CrsCode = 'IS312'
```

"Note that you cannot delete only a subset of the tuple"

Part 3: Simple Select Statements

- Introduce the SELECT- FROM- WHERE structure

```
SELECT <Columns list>
FROM <table list>
[WHERE <Condition>]
```

Example 1:

- Displays departments data

```
Select *
From Department
```

Example 2(selecting specific columns to display)

- Get all students’ names and majors

```
Select Name, Major
 From Student
```

Example 3 (using where clause)

- Get the codes of all courses registered in spring 2012

```
Select CrsCode
From registered
Where semester = 'Spring'
And Year = 2012
```

Example 4 (Using DISTINCT) :

- Get the codes of all courses registered in spring 2012 and eliminate repetitions.

```
Select DISTINCT (CrsCode)
From Registered
Where Semester = 'Spring'
And Year = 2012
```

Example 5 (Using Comparison Operators / Between)

Comparison Operators → (= , > , > =, < , < =, , < >)

- Get all students’ SSNs who registered in IS221 course between 2009 and 2011

```
Select SSN
From registered
Where CrsCode= 'IS221'
And year <= 2011 AND year >= 2009
```

Another Solution (Using Between)

```
Select SSN
From registered
Where CrsCode= 'IS221'
And year BETWEEN 2009 AND 2011
```

Another Solution (Using OR)

```
Select SSN
From registered
Where CrsCode= 'IS221'
And (year = 2009 OR year = 2010 OR year = 2011)
```

Example 6 (Using IN / NOT IN / OR):

- Get Students’ names and city for those who live in Fayoum, Alexandria or Aswan

```
Select name, city
From student
Where city IN ('Fayoum','Alexandria','Aswan')
```

Another solution (using OR)

```
Select name, city
From student
Where city = 'Fayoum' OR city = 'Alexandria' OR city = 'Aswan'
```

Example 7 (Using Like, %, _)

- Get course names starting with 'Data' - containing the word 'Data'… ending with the letter 's' - its second letter is 'r'
  - `Select Name    From course   Where  Name LIKE     'Data%'`
  - `Select Name    From course    Where  Name LIKE     '%Data%'`
  - `Select Name    From course    Where  Name LIKE     '%s'`
  - `Select Name    From course    Where  Name  Like    '_r%'`
  -

Example 8 (Using not null)

- Get all students' names with known cities

```
Select Name
From Student
Where city IS NOT NULL
```

OR

```
Select Name
From Student
Where Not city IS NULL
```