

مقدمة لقواعد البيانات

الفصل السادس لغة SQL

Introduction

- SQL (Structured Query Language) is one of the major reasons for the success of relational DB.
- It is a standard language for relational DB.
- It enables the migration among relational DBMS easily.
- It became a standard for most of the commercial relational DBMSs.
- SQL provides a high level declarative language enables the users to only specify what he needs and leaving the details to the system.

DDL and Constraints

Create Table (إنشاء الجداول):

Create Table Student(

ID	Integer,
Name	Varchar(30),
Grade1	Integer,
Grade2	Integer,
Grade3	Integer,
Total	Integer,
Primary Key (ID));	

DDL – Constraints (cont)

■ Data types (أنواع البيانات):

» Numeric (بيانات رقمية):

Integer – Long – Byte – Single – Double - Decimal

» String (بيانات نصية):

CHAR - CHAR(n) - VARCHAR(n)

» Date and time (تاريخ و وقت):

DATE - TIME

■ NOT NULL option (القيم المطلوبة):

Name Varchar(30) NOT NULL,

DDL (continued)

- **DROP TABLE Student;**
(Remove Student from the database (حذف جدول))
- **DROP TABLE Student CASCADE;**
(Remove Student and any related tables (حذف جدول))
- **ALTER TABLE Student ADD Dept VARCHAR(15);**
(Add new field Dept to student table (تعديل جدول بإضافة حقل))
- **ALTER TABLE Student DROP Grade3 CASCADE;**
(Remove field Grade3 from student table (تعديل جدول بحذف حقل))

Insert, Delete and Update in SQL

Insert (إضافة بيانات)

Examples:

- INSERT INTO Student VALUES
(12345, "Ahmed Saad", 50, 60, 70, 0, Null);

Note:

The values of the attributes in the insert command must satisfy all constraints.

Delete (حذف بيانات)

1. DELETE FROM Student
WHERE ID = 12345;
2. DELETE FROM Student;
(Delete all tuples of Student)

Insert, Delete and Update in SQL (cont)

Update (تعديل بيانات)

1. UPDATE Student
SET total = Grade1 + Grade2 + Grade3
WHERE ID = 12345;
2. UPDATE Student
SET Grade1 = Grade1 * 1.1
WHERE Grade1 < 70;

Basic Queries in SQL

Select-From-Where structure (الاستفسارات و عرض البيانات)

SELECT <attribute list>
FROM <table list>
WHERE <condition>;

■ Example:

Retrieve the student ID and Total for Ahmed Saad.

SELECT ID,Total
FROM Student
WHERE NAME= 'Ahmed Saad';

Joining Tables (ربط الجداول)

- Join operations are used to join tables.

- We can join tables by using:

- JOIN (INNER JOIN)

(يستخدم عندما يكون المفتاح الأجنبي يختلف عن مسمى المفتاح الأساسي)

- NATURAL JOIN

(يستخدم عندما يكون المفتاح الأجنبي و المفتاح الأساسي لهم نفس المسمى)

Joining Tables (cont)

JOIN operation

Ex:

For this schema:

STUDENT (ID, Name , Dept)

COURSE (Course_id, Title , Instructor , Credit_H)

REGIST (S_ID , Course_id , Semester , Grade)

Retrieve student name, course_id and grade in all courses.

```
SELECT Name , Course_id , Grade
FROM (STUDENT JOIN REGIST ON ID =S_ID);
```

Joining Tables (cont)

NATURAL JOIN operation

Ex:

For this schema:

STUDENT (ID, Name , Dept)

COURSE (Course_id, Title , Instructor , Credit_H)

REGIST (S_ID , Course_id , Semester , Grade)

Retrieve student name, course_id, Title, Credit Hour and grade in all courses.

```
SELECT Name , Course_id , Title , Credit_H , Grade
FROM   (STUDENT JOIN REGIST ON ID =S_ID)
       NATURAL JOIN COURSE;
```

SQL commands (cont)

Unspecified WHERE and use of (*)

1. SELECT ID
FROM Student;
2. SELECT *
FROM Student
WHERE Grade1 < 60;

SQL commands (cont)

- Arithmetic operators: + - * /

Example:

```
SELECT      NAME, 1.10 * Total
FROM        Student
WHERE       Grade1 > 60 and Grade2 > 60 and Grade3 > 60;
```

Example:

Retrieve all students have total grade between 200 and 250.

```
SELECT      *
FROM        Student
WHERE       Total BETWEEN 200 AND 250;
```

SQL commands (cont)

Ordering (الفرز أو الترتيب)

- ORDER BY is used to order the values.
- The default order is in ascending order.
- The keyword ASC (Ascending) and DESC (Descending) can be used.

Example:

```
SELECT      ID , NAME , TOTAL
FROM        Student
WHERE       Total > 250
ORDER BY    NAME;
```

Aggregation Functions

الدوال التجميعية

Ex:

Retrieve the number of students, maximum total, minimum total and average total for each department.

```
SELECT      Dept,  
            Count(*) AS Count,  
            Max(Total) AS Max_T,  
            Min(Total) AS Min_T,  
            AVG(Total) AS AVG_T  
FROM        Student  
GROUP BY    DEPT;
```