Benha University
Faculty of Science
Dept. of Mathematics

Time: Two Hours
First Semester 2014-2015
Date : 20/12/2014

Databases (MC357) for Third Level Students (Computer Science)

جامعة بنها – كلية العلوم – قسم الرياضيات

المستوي الثالث (علوم حاسب)

يوم الامتحان: السبت

تاريخ الامتحان: ٢٠ / ١٢ / ٢٠١٤ م

المادة : قواعد بيانات (٣٥٧ رس)

الممتحن: د/ مصعب عبد الحميد محمد حسان

مدرس بقسم الرياضيات بكلية العلوم

الاسئلة و نموذج الإجابة

ورقة كاملة

Benha University
Faculty of Science
Dept. of Mathematics

Time: Two Hours
First Semester 2014-2015
Date : 20/12/2014

Databases (MC357) for Third Level Students (Computer Science)

Answer the following questions:

Question 1. (12 marks)

A- Define database management system (DBMS) in details. (6 marks)

B- Discuss the main characteristics of the database approach and how it differs from traditional file systems. (6 marks)

Question 2. (12 marks)

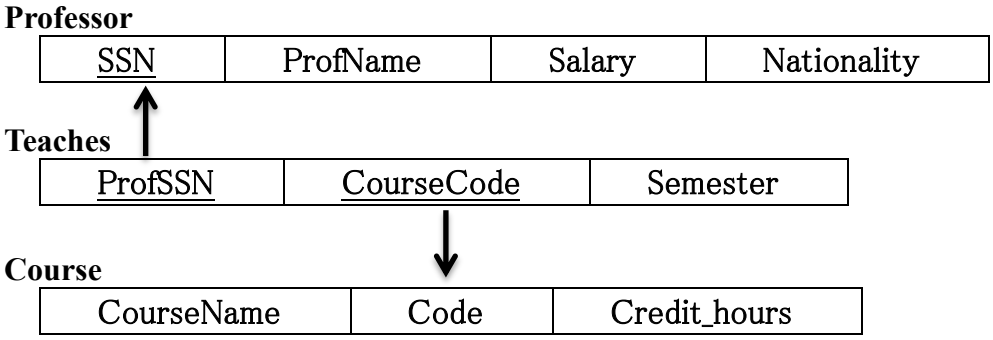
A- Draw a simplified database system environment and explain it. (4 marks)

B- Define data model, entity, attribute, relationship, and complex attribute. (4 marks)

C- Draw a simple ER diagram based on the following: An employee has a basic information such as SSN, Name, Birth Date, Address, Phone, and Age. (4 marks)

Question 3. (12 marks)

Consider the following relational database schema



Specify the following queries using the relational algebra:

A- Display a list of all "Egyptian" Professors data. (1 mark)

B- Display a list of professor name and Nationality for professors who got salary more than or equal 1000. (2 marks)

C- Display a list of professor name and code of courses he taught in "summer" semester. (3 marks)

D- Display a list of "Egyptian" Professor names and name of courses he taught in "summer" semester. (3 marks)

E- Display a list of professor who taught the course "CS355" and also taught the course "CS212". (3 marks)

Benha University
Faculty of Science
Dept. of Mathematics

Time: Two Hours
First Semester 2014-2015
Date : 20/12/2014

Databases (MC357) for Third Level Students (Computer Science)

Question 4. (12 marks)

Consider the employee, project, and work_on tables as follows.

employee (emp_no, emp_fname, emp_lname, dept_no)
project (project_no, project_name, budget)
works_on (emp_no, project_no, job, enter_date)

Write the SQL queries of the following:

A- Get the last and first names of all employees whose address is Cairo. (1 mark)

B- Set the task of employee number 18316, who works on project 2, to be "Manager". (1 mark)

C- Calculate the sum of all budgets of all projects. (1 mark)

D- Delete all managers in the works_on table. (1 mark)

E- Using stored procedure, calculate the average of all project budgets and compares this value with the budget of project 2. If the latter value (the budget of project 2) is smaller than the calculated value (the average of all project budgets), the budget of project 2 will be increased by the value of the parameter extra_budget which given to procedure. Also print a message to tell us if the budget of project 2 is changed or unchanged. (4 marks)

F- Using triggers, create the Audit_Budget(project_no, budget_old, budget_new) table, which stores all modifications of the budget column of the project table. Note that budget_old is the value of budget before updating and budget_new is the value of budget after updating. (4 marks)

Best Wishes
Dr. Mosab Abd El-Hameed

Answer of Question 1

A- A *database management system (DBMS)* is a collection of programs that enables users to create and maintain a database. The DBMS is a general-purpose software system that facilitates the processes of *defining*, *constructing*, *manipulating*, and *sharing* databases among various users and applications.

Defining a database involves specifying the data types, structures, and constraints of the data to be stored in the database. The database definition or descriptive information is also stored by the DBMS in the form of a database catalog or dictionary; it is called *meta-data*.

Constructing the database is the process of storing the data on some storage medium that is controlled by the DBMS.

Manipulating a database includes functions such as querying the database to retrieve specific data, updating the database to reflect changes in the miniworld, and generating reports from the data.

Sharing a database allows multiple users and programs to access the database simultaneously.

B- A number of characteristics distinguish the database approach from the much older approach of programming with files. In traditional *file processing*, each user defines and implements the files needed for a specific software application as part of programming the application. For example, one user, the *grade reporting office*, may keep files on students and their grades. Programs to print a student's transcript and to enter new grades are implemented as part of the application. A second user, the *accounting office*, may keep track of students' fees and their payments. Although both users are interested in data about students, each user maintains separate files—and programs to manipulate these files—because each requires some data not available from the other user's files. This redundancy in defining and storing data results in wasted storage space and in redundant efforts to maintain common up-to-date data.

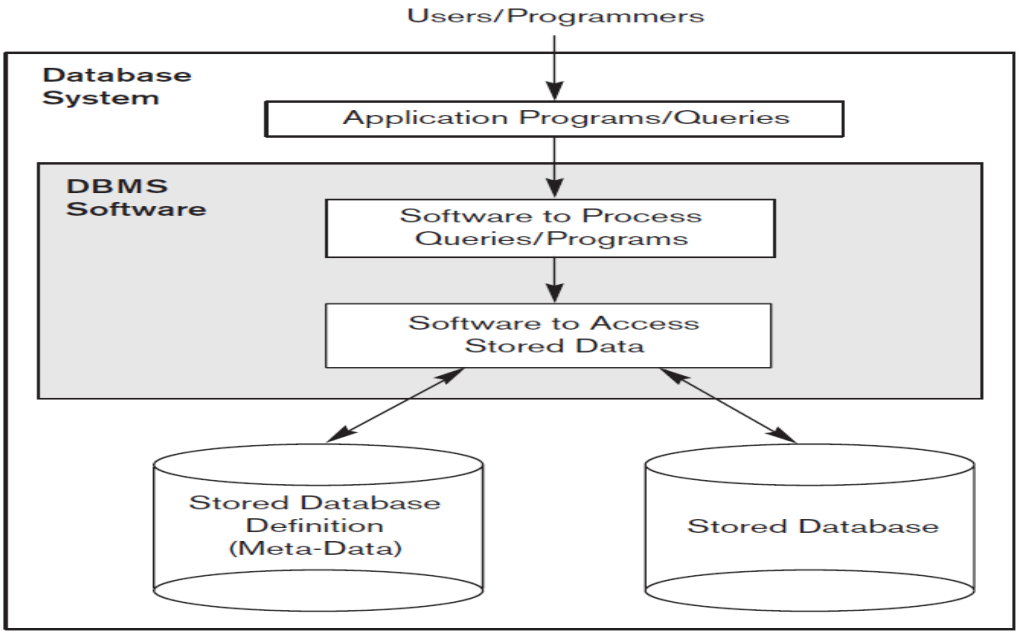
In the database approach, a single repository maintains data that is defined once and then accessed by various users. In file systems, each application is free to name data elements independently. In contrast, in a database, the names or labels of data are defined once,

and used repeatedly by queries, transactions, and applications. The main characteristics of the database approach versus the file-processing approach are the following:

1. Self-describing nature of a database system
2. Insulation between programs and data, and data abstraction
3. Support of multiple views of the data
4. Sharing of data and multiuser transaction processing
5. Controlling Redundancy
6. Restricting Unauthorized Access
7. Providing Storage Structures and Search Techniques for Efficient Query Processing
8. Providing Backup and Recovery
9. Providing Multiple User Interfaces

Answer of Question 2

A- we call the database and DBMS software together a database system. next Figure illustrates some of the concepts



B- *Data model* is a collection of concepts that can be used to describe the structure of a database.

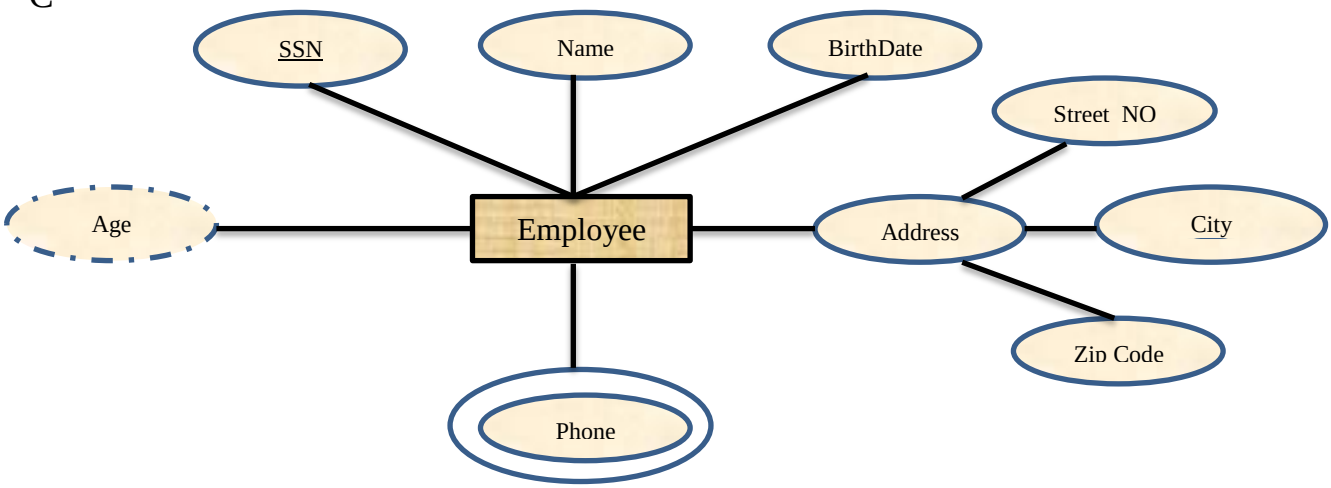
Entity is specific object or thing in the mini-world that is represented in the database. For example the EMPLOYEE John Smith, the Research DEPARTMENT, the ProductX PROJECT

Attribute is property used to describe an entity. For example an EMPLOYEE entity may have the attributes Name, SSN, Address, Sex, BirthDate.

A relationship relates two or more distinct entities with a specific meaning. For example, EMPLOYEE John Smith works on the ProductX PROJECT, or EMPLOYEE Franklin Wong manages the Research DEPARTMENT

complex attribute is multi-valued attribute (An entity may have multiple values for this attribute) and also is composite attribute (may be composed of several components)

C-



Answer of Question 3

A- $R_1 \leftarrow \sigma$ (Professor)
Nationality = "Egyptian"

B- $R_1 \leftarrow \sigma$ (Professor)
Salary >= 1000
$\Pi(R_1)$
ProfName, Nationality

C- $R_1 \leftarrow \sigma$ (Teaches)
Semester = "Summer"

$R_2 \leftarrow R_1 \lozenge\lozenge$ Professor
ProfSSN = SSN
$R_3 \leftarrow \Pi(R_2)$
ProfName, CourseCode

D- $R_1 \leftarrow \sigma$ (Professor)
Nationality = "Egyptian"

$R_2 \leftarrow \sigma$ (Teaches)
Semester = "Summer"

$R_3 \leftarrow R_1 \lozenge\lozenge R_2$
SSN = ProfSSN

$R_4 \leftarrow R_3 \lozenge\lozenge$ Course
CourseCode = code

$R_5 \leftarrow \Pi (R_4)$
ProfName, CourseName

E- $R_1 \leftarrow \sigma$ (Teaches)
CourseCode = "CS355"

$R_2 \leftarrow \Pi (R_1)$
ProfSNN

$R_3 \leftarrow \sigma$ (Teaches)
CourseCode = "CS212"

$R_4 \leftarrow \Pi (R_3)$
ProfSNN

$R_5 \leftarrow R_2 \cap R_4$

Answer of Question 4

A- SELECT emp_fname, emp_lname FROM employee WHERE address = 'cairo';

B- UPDATE works_on SET job = 'Manager' WHERE emp_no = 18316
AND project_no = 2;

C- SELECT SUM (budget) FROM project;

D- DELETE FROM works_on WHERE job = 'Manager';

E-

```
CREATE PROCEDURE PROC1 (@extra_budget MONEY)
AS
DECLARE @avg_budget MONEY
SELECT @avg_budget = AVG(budget) FROM project
IF (SELECT budget FROM project WHERE project_no=2) < @avg_budget
BEGIN
UPDATE project
SET budget = budget + @extra_budget WHERE project_no =2
PRINT 'Budget for project 2 increased by @extra_budget'
END
ELSE PRINT 'Budget for project 2 unchanged'
```

F-

```
CREATE TRIGGER modify_budget
ON project AFTER UPDATE
AS IF UPDATE(budget)
BEGIN
DECLARE @budget_old FLOAT
DECLARE @budget_new FLOAT
DECLARE @project_number CHAR(4)
SELECT @budget_old = (SELECT budget FROM deleted)
SELECT @budget_new = (SELECT budget FROM inserted)
SELECT @project_number = (SELECT project_no FROM deleted)
INSERT INTO audit_budget VALUES
@project_number,USER_NAME(),GETDATE(),@budget_old,
@budget_new)
END
```