

يوم الامتحان: الثلاثاء

المستوي الثالث (حاسب)

تاريخ الامتحان: ١٠ / ١ / ٢٠١٧ م

المادة: قواعد بيانات (٣٥٧ رس)

الممتحن: د/ مصعب عبد الحميد محمد حسان

مدرس بقسم الرياضيات بكلية العلوم

الاسئلة و نموذج الإجابة

ورقة كاملة



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

**Answer the following questions:**

**Question 1. (8 marks)**

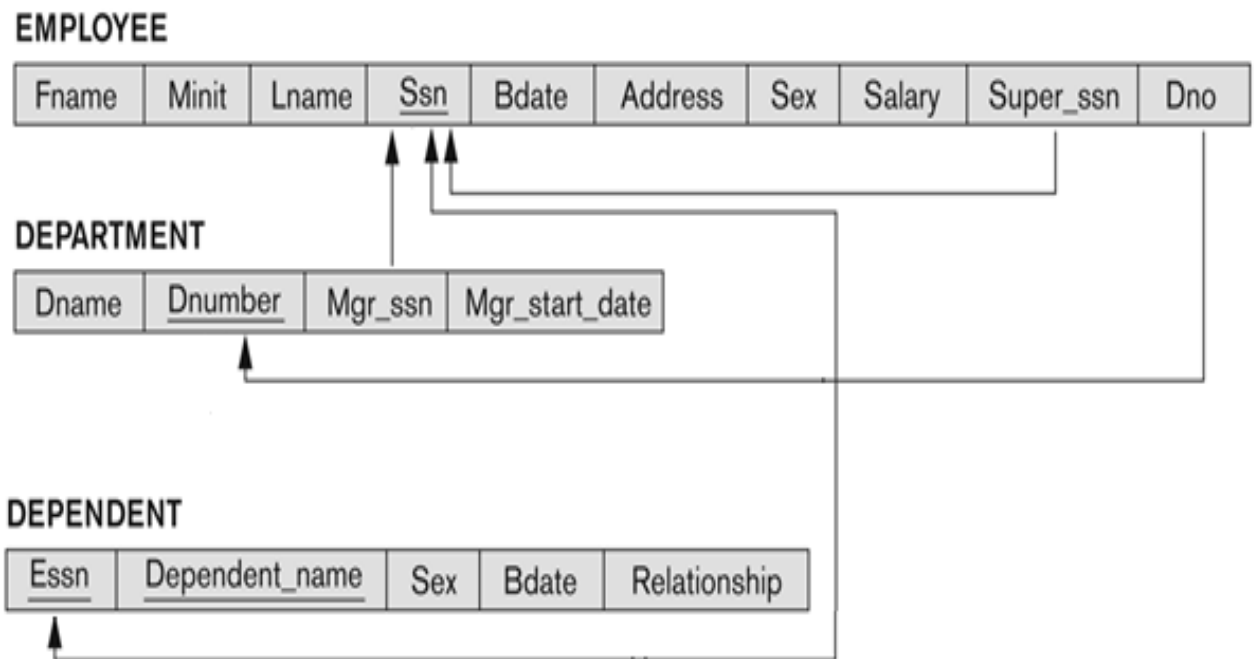
- A- Define database, data model, simple attribute, and multi\_valued attribute. (4 marks)
- B- Draw a simplified database system environment. (4 marks)

**Question 2. (14 marks)**

- A- Define relationship in details. (4 marks)
- B- Define database management system (DBMS) in details. (6 marks)
- C- Draw a simple ER diagram based on the following: a Department has a basic information such as Number, Name, locations, manager, and manager\_start\_date. (4 marks)

**Question 3. (12 marks)**

Consider the following relational database schema



Specify the following queries using the relational algebra:

- A- Select the employee tuples whose salary is greater than 1000. (1 mark)
- B- List each employee's first and last name. (1 marks)
- C- Retrieve SSN of all employees who either work in department 5 or directly supervise an employee who works in department 5. (3 marks)
- D- Retrieve the name of female employees and their dependents. (4 marks)
- E- Retrieve the name of the manager of each department. (3 marks)

**Question 4. (14 marks)**

**1- Consider the employee and the project tables as follows.**

**employee (emp\_no, emp\_fname, emp\_lname, address, salary ,dept\_no)  
project (project\_no, project\_name, cost)**

**Write the SQL queries of the following:**

**A- Get the first names of all employees whose salary is less than 2000.  
(1 mark)**

**B- Calculate the average of costs of all projects. (1 mark)**

**C- Using triggers, create the Employee\_Salary(emp\_no, salary\_old, salary\_new) table, which stores all modifications of the salary column of the employee table. Note that salary\_old is the value of salary before updating and salary\_new is the value of salary after updating. (5 marks)**

**2- Consider the student and the free\_positions tables as follows.**

**student (std\_no, std\_name, faculty, address)**

**free\_positions (faculty, free\_pos)**

**Using stored procedure and transaction, insert a new student in student table and reduce the free\_pos of the student faculty by one in free\_postions table and print a message to tell us that a new student is inserted. (7 marks)**

---

*Best Wishes  
Dr. Mosab Abd El-Hameed*

# Model Answer

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

### Answer of Question 1.

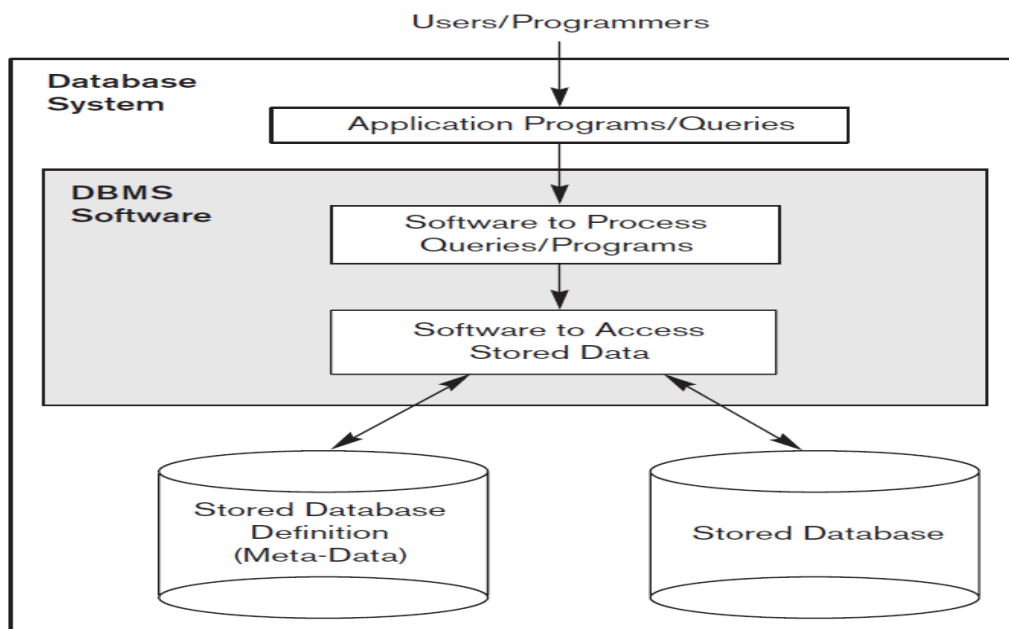
A- **A database** is a collection of related data. By data, we mean known facts that can be recorded and that have implicit meaning.

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

**Simple attribute:** Each entity has a single atomic value for the attribute. For example, SSN.

**Multi-valued attribute:** An entity may have multiple values for that attribute. For example, Color of a CAR

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



### Answer of Question 2.

A- A relationship relates two or more distinct entities with a specific meaning. For example, EMPLOYEE ahmed *works on* the ProductX PROJECT, or EMPLOYEE ali *manages* the Research EPARTMENT.

#### **Constraints on Relationship Types**

- 1- **One-to-one (1:1):** In a one-to-one relationship, each row in one database table is linked to 1 and only 1 other row in another table.
- 2- **One-to-many (1:N) or Many-to-one (N:1):** In a one-to-many relationship, each row in the related to table can be related to many rows in the relating table. This allows frequently used information to be saved only once in a table and referenced many times in all other tables.
- 3- **Many-to-many (M:N):** In a many-to-many relationship, one or more rows in a table can be related to one or many rows in another table.

**B- 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.**

c-



**Answer of Question 3.**

A-  $\square_{SALARY > 1000}$  (EMPLOYEE)

B-  $\square_{LNAME, FNAME}$  (EMPLOYEE)

C-  $DEP5\_EMPS \leftarrow \square_{DNO=5}$  (EMPLOYEE)

RESULT1  $\leftarrow \square_{SSN}$  (DEP5\_EMPS)

RESULT2(SSN)  $\leftarrow \square_{SUPERSSN}$  (DEP5\_EMPS)

RESULT  $\leftarrow RESULT1 \cup RESULT2$

**D- FEMALE\_EMPS** ← □<sub>SEX='F'</sub> (EMPLOYEE)  
**EMPNAMES** ← □<sub>FNAME, LNAME, SSN</sub> (FEMALE\_EMPS)  
**EMP\_DEPENDENTS** ← EMPNAMES x DEPENDENT  
**ACTUAL\_DEPS** ← □<sub>SSN=ESSN</sub> (EMP\_DEPENDENTS)  
**RESULT** ← □<sub>FNAME, LNAME, DEPENDENT\_NAME</sub> (ACTUAL\_DEPS)

**E- DEPT\_MGR** ← DEPARTMENT ◇◇<sub>MGRSSN=SSN</sub> EMPLOYEE

**Answer of Question 4.**

**1- A- SELECT emp\_fname FROM employee WHERE salary < 2000;**

**B- SELECT AVG (cost) FROM project;**

**C- CREATE TRIGGER modify\_salary ON employee AFTER UPDATE**

**AS IF UPDATE(salary)**

**BEGIN**

**DECLARE @salary\_old FLOAT**

**DECLARE @salary\_new FLOAT**

**DECLARE @employee\_number CHAR(4)**

**SELECT @salary\_old = (SELECT salary FROM deleted)**

**SELECT @salary\_new = (SELECT salary FROM inserted)**

**SELECT @employee\_number = (SELECT emp\_no FROM deleted)**

**INSERT INTO Employee\_Salary VALUES**

**@employee\_number, @salary\_old, @ salary\_new)**

**END**

**2- create proc InsertStudentTran (@MyStdId nvarchar(20),**  
**@MyStdName nvarchar(100),**  
**@MyFaculty nvarchar(150),**  
**@Myaddress nvarchar(150))**

**As**

**Begin transaction**

**Insert into student(std\_no, std\_name, faculty, address) values**

**(@MyStdId,@MyStdName, @MyFaculty,**

**@Myaddress);**

**if(@@ERROR <> 0)**

**rollback transaction**

**Update free\_positions set free\_pos = free\_pos -1 where faculty =**

**@MyFaculty;**

**if(@@ERROR <> 0)**

**rollback transaction**

**PRINT 'a new student is inserted'**

**if(@@ERROR <> 0)**

**rollback transaction**

**Commit transaction**