

Assignment 6 – Week 9

---

**This assignment is based on lecture 8 (chapter 16 & 17).**

- Submit your *own work* on time. No credit will be given if the assignment is submitted after the due date.
  - Note that the completed assignment should be submitted in .doc, .docx, .rtf or .pdf format only.
  - In MCQs, if you think that your answer needs more explanation to get credit then please write it down.
  - You are encouraged to discuss these questions in the Sakai forum.
- 

- (1) 3NF essentially identifies
- A. 1-\* relationships
  - B. \* - \* relationships
  - C. 1-1 relationships
  - D. None of the above

ANS: A

- (2) While checking our tables for normalization, if we find that they are not even in 2NF then we must have missed some
- A. 1-\* relationships
  - B. \* - \* relationships
  - C. 1-1 relationships
  - D. None of the above

ANS: B

- (3) How to identify parent and child entities in a relationship?

ANS: We can identify the Parent and child entities in a relationship based on the foreign key constraint in the child entity. This relation occurs due to the is and has relation between the parent and child entities. Foreign key is created on the child table, and which match exactly with the parent key of the Parent table. Parent table must have a primary or unique key to make that foreign key on the child table.

- (4) Solve review question 17.2/ 16.2 (a,b,c,d,g,i) from 5<sup>th</sup> /4<sup>th</sup> edition of the course text book.

ANS:

- a. Strong Entity Types: These entity types are not dependent on any other entity and always contains a primary key.
- b. Weak Entity Types: These entity types are dependent on other entity. These depends on the strong entity types.
- c. One to Many (1: \*) binary relationship types: One item of the entity from the first end is mapped(related) to many items of second entity end and each item of second entity is mapped(related) to first end entity.

- d. One-to-one (1:1) binary relationship types: one item of first end entity is mapped to exactly one item of second end entity and vice-versa.
- e. Many-to-many (\*: \*) binary relationship types: Each item from first end of the entity can be mapped to each item of the second end entity and vice-versa. To establish the relationship between these entities we need to create a third table/entity i.e., association table.
- f. Multi-valued attributes: These attributes have set of values for each entity.

(5) Discuss how the technique of normalization can be used to validate the relations derived from the conceptual data model. (17.3/16.3)

ANS:

Normalization is used to design the database such way that data can be manipulate/render at a maximum efficient way. It also ensures the logical database design that is derived from the conceptual design. It ensures the all the requirements of database for the target operation with minimal set of attributes. It also ensures that it should has low data redundancy with maximum stability. It should be in a 3<sup>rd</sup> normalization form (3NF) when logical design is derived from the conceptual design.

(6) Solve exercise 17.8/16.8 from the 5<sup>th</sup> /4<sup>th</sup> edition of the course text book. In the ERD, only those attributes are listed which are PK for that entity. You are required to add more attributes to the relations which will be logically applicable to that entity.

ANS:

**Customer** => (customerNo(PK), name, TelNo, DOB, street, city, state, zipCode)

**Employee** => (employeeNo(PK), firstName, lastName, address, TelNo, Email, SSN, DOB, position, sex, salary, dateStarted)

**Product** => (productNo(PK), productName, description)

**Order** => (orderNo(PK), date, street, city, state, zipCode, deliveryDate, status, customerNo(FK), employeeNo(FK))

**OrderDetail** => (orderDetailNo(PK), orderNo(FK), productNo(FK), price, quantity)

**PaymentMethod** => (pMethodNo(PK), pMedia)

**Invoice** => (invoiceNo(PK), invoiceDate, creditCardNo, expiryDate, methodNo, orderNo(FK), pMethodNo(FK))

**Shipment** => (shipmentNo(PK), quantity, status, orderNo, shipdate, productNo(FK), employeeNo(FK), sMethodNo(FK), orderDetailNo(FK))

**ShipmentMethod** => (sMethodNo(PK), sMedia)