CSCU9B3 Database Principles and Applications Autumn 2018

Tutorial 1 - Answers

1. Consider the following relations which might form part of a database for a hairdressing salon:

   Client(CNo,Name,Phone,FavouriteStylist)
   Stylist(SNo,Name,Phone)
   Treatment(TreatmentName,Price,Duration)
   Booking(CNo,SNo,Date,Time,TreatmentName)

   Indicate in each case all candidate keys, discussing any assumptions that you make. Choose a primary key for each relation. Identify any foreign keys.

2. The following Tutor and Student tables show tutors who are assigned to students. Each student's tutor is identified by the *tutorID* column of the Student table. The primary keys are underlined. Do these tables conform to the notions of *entity integrity* and *referential integrity?* State the reasons for your answers.

Tutor

tutorID	tutorName
21	Newman
34	Martin
56	Wright
78	Adams

Student

studentID	studentName	tutorID
990199	Young	56
990278	Fletcher	56
990445	Chung	45
Null	Cohen	21
990721	Kennedy	78

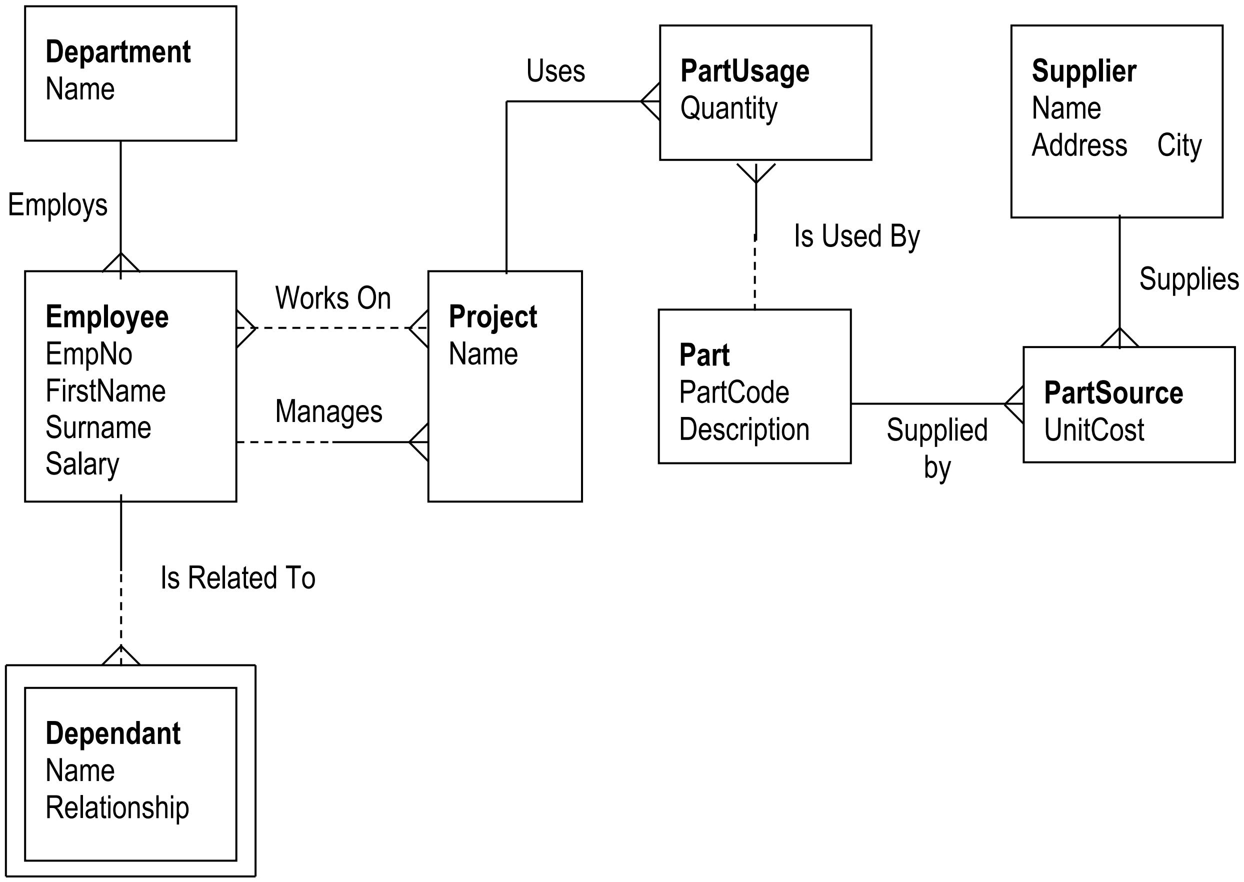
3. Draw an ER diagram representing the Tutor and Student example shown above.

4. A bakery uses a database system to record details about customers, products and orders. The system records customer details including the customer's name, address, and contact telephone number. A customer may place a number of orders, each of which requests various products. The system records the date on which each order was placed, the date the order is to be delivered, and the products requested. Each order is to be delivered to a unique customer, who may be different from the customer who placed that order (eg, a gift). Each product has a unique name and a unit price. Some products are made up of a combination of other products. For example, the "cocktail party selection" consists of 5 "cheese straws", 2 "sausage rolls", and 3 "vol au vents".

   Construct an Entity-Relationship (E-R) diagram to model the entities, attributes and relationships described above. Ensure that you show the participation and cardinality constraints that apply to each relationship. Give a brief explanation of what each entity is intended to represent.

5. Use the techniques described in the lectures to convert your ER model (developed for question 4 above) into a set of relations. Underline the primary key of each relation and clearly indicate any foreign keys.

6. Examine the ER diagram below and think about what it represents. Once you have a reasonable feel for this, use the techniques described in lectures/textbook to convert the ER model into a set of relations for use in a relational database. Underline your primary keys and clearly indicate any foreign keys.



Answers:

Q1. *Candidate keys are unique and irreducible identifiers for the entity.*

For *Client*, candidate key is (CNo)

For *Stylist*, candidate key is *SNo.*

For *Treatment* candidate key is *TreatmentName,* although might also sensibly be *(TreatmentName, Duration).*

For *Booking, candidate keys are (Cno, Date, Time) or (Sno, Date, Time) if it is assumed that a stylist can treat only one client at a time.* Either of these could be used as a primary key, or an artificial key (BookingNo) introduced instead.

FavouriteStylist is a FK in Client, and CNo, SNo and TreatmentName are all FKs in Booking.

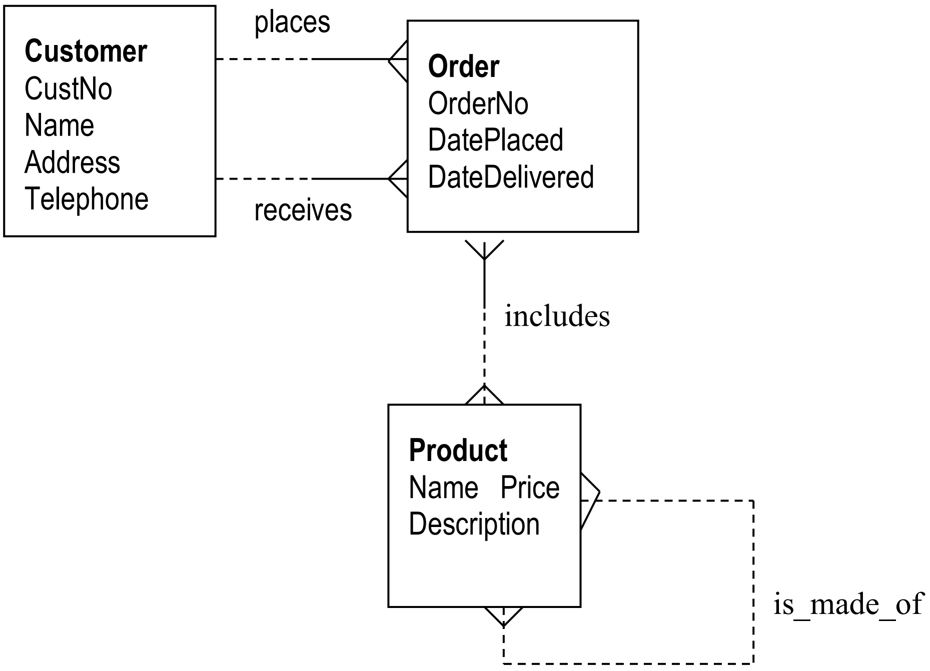
Q2. *Entity integrity: each entity is unique and has a non-null primary key. Referential integrity: every foreign key is a primary key in the other relation.*

They don't conform to either. The *Student* table has a null in the primary key column, violating entity integrity. The *Student* table has an entry 45 in the *tutorID* column (a foreign key column), which does not correspond to any existing value in the primary key (*tutorID*) column in the *Tutor* table, thus violating referential integrity.

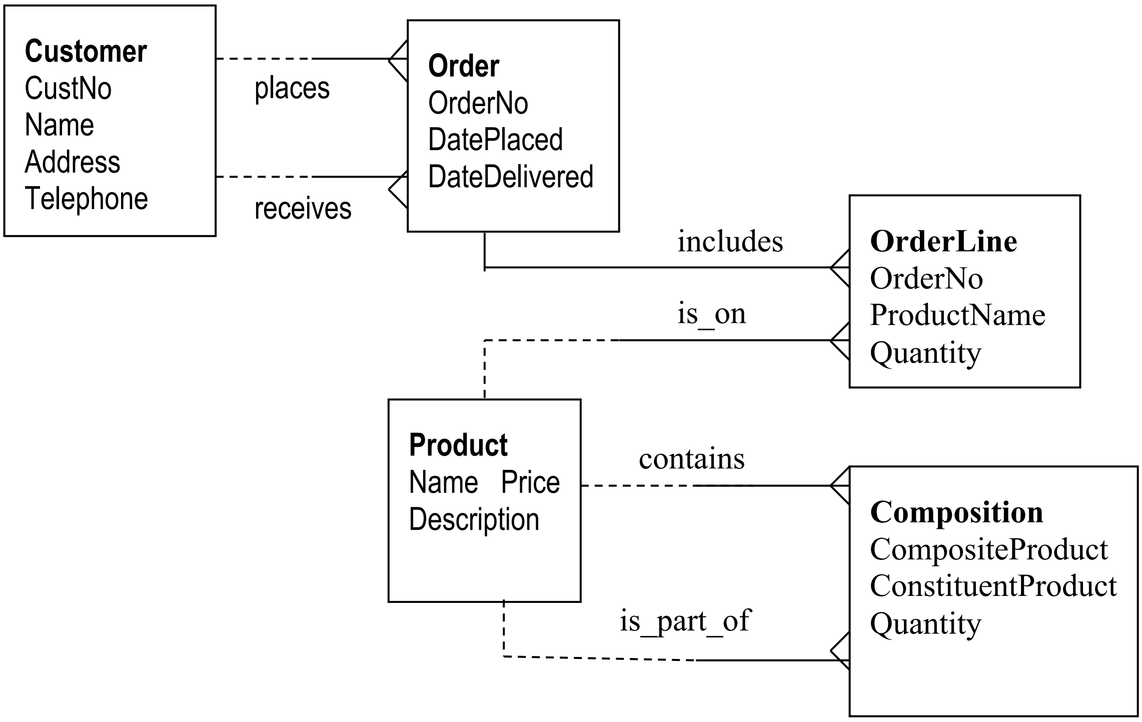
Q3. A one to many relationship (from the point of view of one tutor) and a one to one from the point of view of one student. Overall a one to many relationship from tutor to student. Cardinality chosen as per assumptions. For example, from the given data it seems that tutors need not be assigned to students, but that students must be assigned tutors.

Q4. The E-R diagram below represents one possible way of modelling the bakery situation. Other solutions are possible. In particular, you may disagree with the optionality that I have chosen for some of the relationships in the model. I've based my choices on the following assumptions:

- The database may hold details of customers who have never placed an order (but may have received one, for example).
- An order must include at least one product.
- There may be some products that have never been ordered (eg, new products).
- A product does not *have* to be made up of other products.
- A product does not *have* to be a constituent of some other product.
- An order is identified by an order number



Note that there are two many-to-many relationships in the E-R model. It is possible to replace these relationships by new entities using the techniques described in lectures. This is particularly appropriate when we wish to associate an attribute with the relationship (e.g. in both cases here a "quantity" attribute would be useful). The result will be something like this:



Q5. Using the second ER diagram as a starting point, the conversion to relations is very straightforward as there are no many-to-many relationships. First, for each entity we create a relation containing the attributes shown on the ER diagram:

Customer(CustNo, Name, Address, Telephone)

Order (OrderNo, DatePlaced, DateDelivered)

OrderLine (OrderNo, ProductName, Quantity)

Product(Name, Price, Description)

Composition (CompositeProduct, ConstituentProduct, Quantity)

Then we must represent the various one-to-many relationships by embedding foreign keys as appropriate. We must also identify the primary keys (underlined) and foreign keys (in italics.) This gives our final solution:

Customer (<u>CustNo</u>, Name, Address, Telephone)

Order (<u>OrderNo</u>, *CustPlacedBy*, *CustDeliveredTo*, DatePlaced, DateDelivered)

OrderLine (<u>*OrderNo*</u>, <u>*ProductName*</u>, Quantity)

Product (<u>Name</u>, Price, Description)

Composition (<u>*CompositeProduct*</u>, <u>*ConstituentProduct*</u>, Quantity)

Q6. The ER diagram presented appears to represent a small manufacturing company with employees organised in departments. The company keeps records of the dependents of employees. Work at the company is grouped by projects, and resources allocated to projects (employees, but also parts required). Suppliers supply parts. It’s possible for a part to be supplied by multiple different suppliers.

Department(Name)

Employee (EmpNo, FirstName, Surname, Salary, *DeptName*)

Dependent (Name, Relationship, *EmpNo*)

Project (Name, *Manager*)

WorksOn (*EmpNo, PName*)

Part (PartCode, Description)

PartSource (*PartCode*, *SupplierName*, UnitCost)

Supplier (Name, Address, City)

PartUsage (*PName*, *PartCode*, Quantity)