

CSG1207/CSI5135: Systems and Database Design

Workshop 04 - Solutions

Standard Disclaimer

Many questions you encounter in this and other workshops have more than one solution which is valid and correct. There are often numerous ways to model data, numerous ways to achieve the same results in an SQL query, etc.

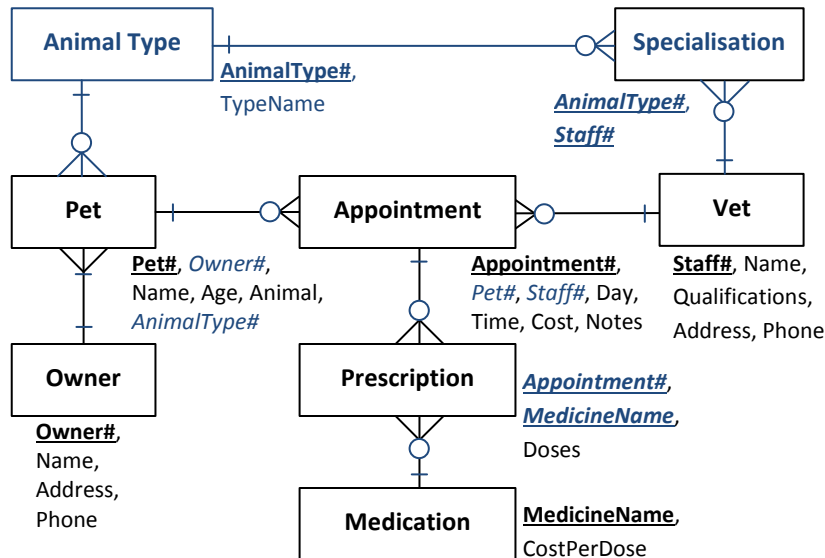
The solutions provided here are NOT the only correct answers to the questions. If you have arrived at solution to a workshop task that differs substantially from what is provided here and would like feedback on your solution, please contact your tutor.

Answers to review questions are not given in these solutions. If you are not confident in your answer to a review question, it is your responsibility to review the relevant unit content or conduct other research until you can answer the question.

Task 1

Answers to review questions are not given. See the Standard Disclaimer for more information.

Task 2



Given the ER diagram above, do the following:

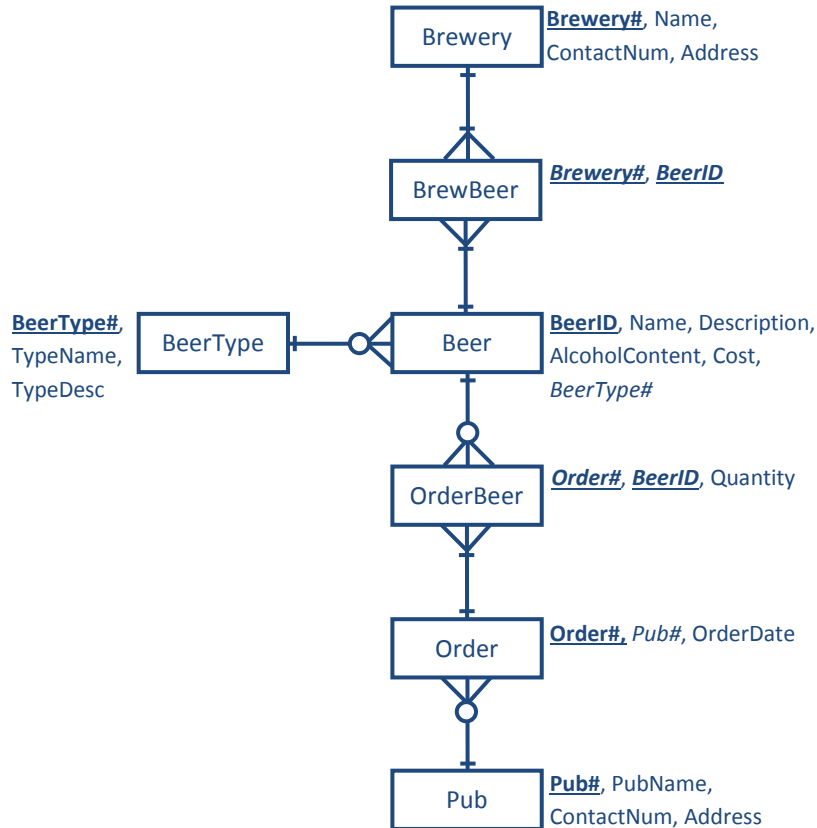
- A. Add cardinality to all relationships in the diagram, stating any assumptions you make
 - Owners must have at least one pet
 - Appointments may not result in a prescription
 - There may be medications that have never been prescribed
- B. Convert it to a physical diagram
 - See diagram
- C. List a possible create and drop order for the diagram
 - Create: Owner, Pet, Medication, Vet, Appointment, Prescription
 - Drop: Prescription, Appointment, Vet, Medication, Pet, Owner
 - Many more possible. Owner must be created before Pet, Vet and Pet must be created before appointment, Medication and Appointment must be created before Prescription.
- D. Add an entity named AnimalType which contains a list of animal types (Cat, Dog, etc), which will have relationships to the Pet entity and the Vet entity. A pet can only be one animal type, but a vet may specialise in many animal types (and there may be multiple vets who specialise in a specific animal)
 - See diagram. Animal attribute in Pet remains based on assumption that it can contain a more detailed description (breed of cat or dog, etc)
 - Assumption that some vets have no specialisation in any animal type

Task 3

You are required to come up with your own answer to this task. Ask your tutor if you want feedback on your answer.

Task 4

Find and correct any errors in the diagram, and make sure that it incorporates the details that you know. The result should be a well-structured physical ER diagram including cardinality. Once this is complete, list the table creation and dropping order for the database, and try to create a data dictionary for it.



Assumptions:

- All breweries produce at least one beer
- Each beer is produced in at least one brewery
- Each order consists of at least one beer
- Some beers may never have been ordered and some pubs may not have made an order

"Brewery" table (stores details about breweries)				
Column Name	Data Type & Length	Null	Constraints	Other
BreweryID	INT	NOT NULL	PK	IDENTITY
Name	VARCHAR(50)	NOT NULL		
ContactNum	VARCHAR(20)	NOT NULL		
Address	TEXT	NOT NULL		

“BeerType” table (stores details about beer types)				
Column Name	Data Type & Length	Null	Constraints	Other
BeerTypeID	INT	NOT NULL	PK	IDENTITY
TypeName	VARCHAR(20)	NOT NULL		
TypeDesc	TEXT	NULL		

“Pub” table (stores details about pubs)				
Column Name	Data Type & Length	Null	Constraints	Other
PubID	INT	NOT NULL	PK	IDENTITY
PubName	VARCHAR(20)	NOT NULL		
ContactNum	VARCHAR(20)	NOT NULL		
Address	TEXT	NOT NULL		

“Order” table (stores details about beer orders)				
Column Name	Data Type & Length	Null	Constraints	Other
OrderID	INT	NOT NULL	PK	IDENTITY
PubID	INT	NOT NULL	FK (Pub.PubID)	
OrderDate	DATETIME	NOT NULL		

“Beer” table (stores details about beers)				
Column Name	Data Type & Length	Null	Constraints	Other
BeerID	INT	NOT NULL	PK	IDENTITY
Name	VARCHAR(20)	NOT NULL		
Description	TEXT	NOT NULL		
AlcoholContent	NUMERIC(3, 1)	NOT NULL		(e.g. 5.3)
Cost	MONEY	NOT NULL		(per keg)
BeerTypeID	INT	NOT NULL	FK (BeerType. BeerTypeID)	

“BrewBeer” table (stores details about which beer is made in which brewery)				
Column Name	Data Type & Length	Null	Constraints	Other
BreweryID	INT	NOT NULL	PK, FK (Brewery.BreweryID)	
BeerID	INT	NOT NULL	PK, FK (Beer.BeerID)	

“OrderBeer” table (stores details about which beer is in which order)				
Column Name	Data Type & Length	Null	Constraints	Other
OrderID	INT	NOT NULL	PK, FK (Order.OrderID)	
BeerID	INT	NOT NULL	PK, FK (Beer.BeerID)	
Quantity	TINYINT	NOT NULL		

Task 5

You wish to create a database to store details of Pokémon. The database needs to keep track of Pokémon details, including the moves they can do and which Pokémon they evolve into.

Draw a physical ER diagram to model this, using the following guidelines:

- Pokémon details include number, name, description and type (Fire, Water, Earth, etc)
- Attack details include the name of the move and the amount of points it takes to use
- Pokémon can perform many moves, and each move can be done by multiple Pokémon
- Different Pokémon learn the same move at a different level – this should be included
- Moves are associated with a type, e.g. “Water Gun” is a Water type move
- You can assume a Pokémon can only evolve into one other Pokémon, and some do not evolve. This should be implemented as a self-referencing relationship

Assumptions:

- All Pokémon and moves have a type (even if it's a boring one like “Normal” or “Fighting”)
- There is at least one Pokémon and one move of each type

