Information Management II

Sample Exam Answers

Instructions to Candidates:
Attempt **three** questions in total. **Question 1 is mandatory**. Answer **any two** questions from **Question 2, Question 3 and Question 4**

1. The ReallyGoodRestaurant has decided to open a small shop beside the restaurant to sell high quality artisan cooking ingredients. The restaurant uses several suppliers to provide good ingredients. Each supplier supplies an ingredient in only one unit size e.g. supplier A supplies sea salt in bags of 500 grams and supplier B supplies sea salt in 250 gram bags. Each supplier charges a specific price for an ingredient in that unit size. However different suppliers can provide the same ingredient in different unit sizes.

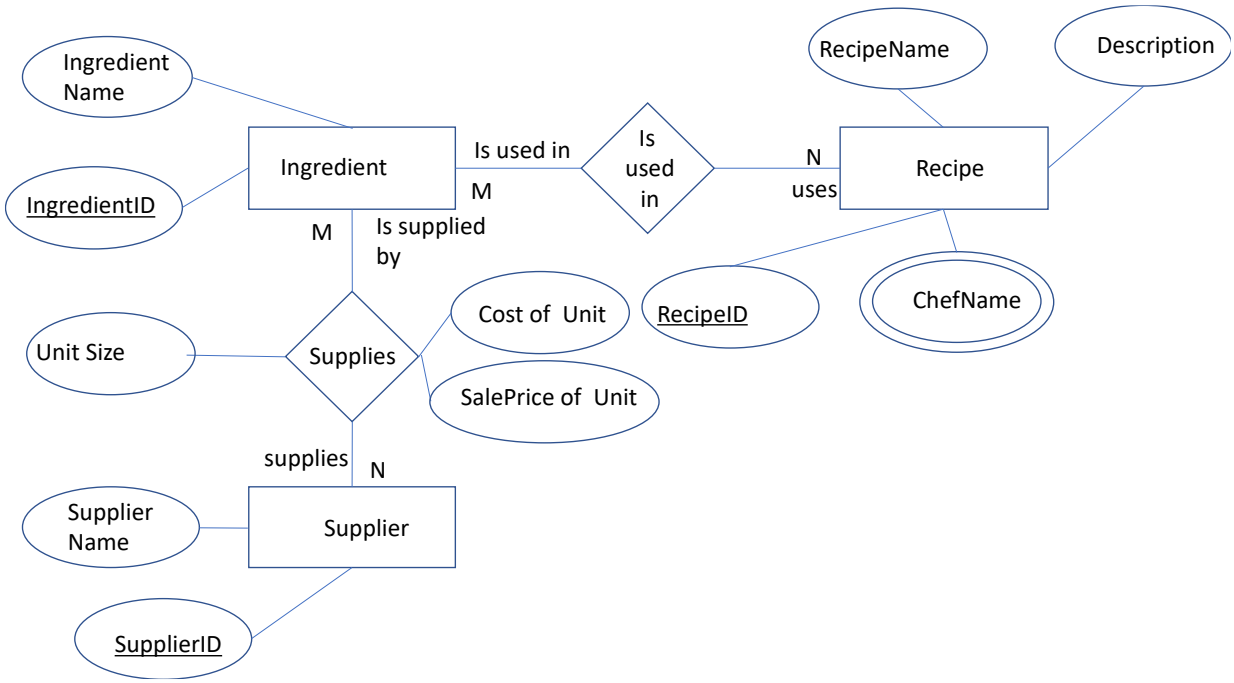
   Also to encourage the sale of these ingredients, the restaurant provides a text description of the recipes used in the restaurant and indicates which ingredients from each recipe can be bought from the shop. Each recipe has been designed by one or more chefs in the restaurant.

   The ReallyGoodRestaurant needs to maintain a database capable of storing information about ingredients, the suppliers of those ingredients, the unit size that each supplier supplies a particular ingredient, a text description of each recipe, the name(s) of the chef(s) who designed each recipe, an indication of which ingredients are used in which recipes, the cost (to the restaurant) of ingredients in a particular unit sizes from suppliers, and the price for which the shop sells each ingredient (in a particular unt size).

   (i) Develop an Entity Relational (ER) Model for the above database, stating any assumptions you make.

   [10 Marks]

Indicative ER Model solution as depicted below



   (ii) Using the appropriate mapping techniques, map the ER Model to a Relational Model and show the functional dependency between attributes within each table. In your answer, identify the Primary and Foreign keys for each relation.

[5 Marks]

Answer should identify Tables based on above ER Model. These could be as follows:

IngredientTable (IngredientID, IngredientName)
RecipeTable (RecipeID, RecipeName, RecipeDecription)
RecipeAuthors (*RecipeID*, ChefName)
SupplierTable (SupplierID, SupplierName)
SuppliesTable(*SupplierID*, *IngredientID*, UnitSize, CostPrice, SalesPrice)
UsedInTable (*IngredientID*, *RecipeID*)

(Primary keys are underlined, foreign keys are in italic. If attributes are both primary key(s) and foreign keys that are underlined & in italics).

(iii) Give the SQL command(s) for the following queries:

(a). Retrieve a list of ingredients sold by the shop for the recipe 'Baked Lasagne'

[2 Marks]

Answer should give the following (or a similar SQL) as

```
SELECT IngredientName FROM IngredientTable, RecipeTable
  where RecipeTable.RecipeDecscription = 'BakedLasagne'
   AND RecipeTable.RecipeID = UsedInTable.RecipeID
   AND UsedInTable.IngredientID = IngredientTable.IngredientID);
```

(b). One of the existing suppliers (named 'Honest Henry') has a new ingredient (roasted chestnuts in units size of 400 grams) to be sold in the shop. The shop buys this ingredient €3 per 400 gram bag and the shop sells it for €8 per (400 gram bag). Give the SQL commands needed to enter this new ingredient.

[2 Marks]

Answer should first insert into Ingredient Table a new IngredientID and the IngredientName 'Roasted Chestnuts'. It should then Insert into the SuppliesTable the SupplierID for 'Honest Henry', IngredientID for HonestHenry, 400, 3, and 8

(c). Create a view of all the ingredients of supplier 'Honest Henry' with the unit size he supplies and the price he charges for that unit size.

[3 Marks]

Answer should give the SQL command

```
CREATE VIEW  HonestHenrysSuppliedIngredients  AS
SELECT  IngredientTable.IngredientName, SuppliesTable.UnitSize, suppliesTable.Cost
FROM SupplierTable, SuppliesTable, IngredientTable
Where
   SupplierTable.SupplierName = 'Honest Henry'
  AND SupplierTable.SupplierID = SuppliesTable.SupplierID
  AND SuppliesTable.IngredientID = IngredientTable.IngredientID;
```

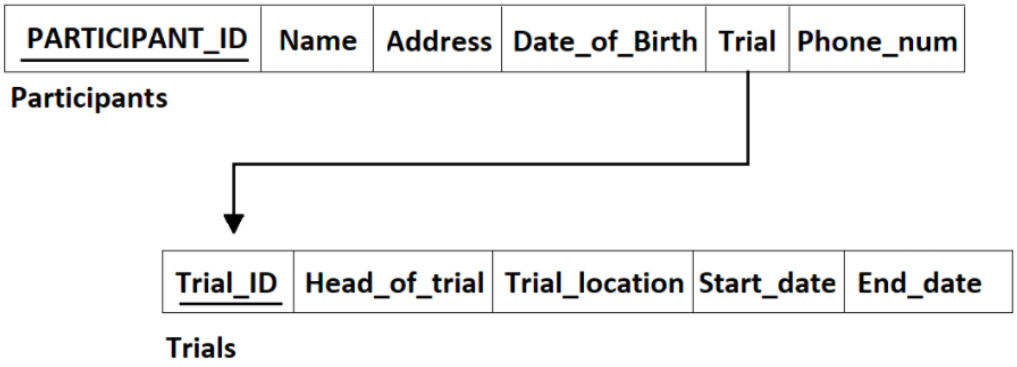
(d). Suppose the restaurant wishes to increase the prices for all ingredients it sells in the shop by 8%. Give the SQL command(s) required to make this happen.

[3 Marks]

Answer should give the appropriate UPDATE command for SuppliesTable.

UPDATE SuppliesTable SET SalesPrice = SalesPrice*1.08 ;

Q. 2 A database manager is designing a relational database to store information about research laboratory experiments involving Covid-19 vaccine trials. Below is part of the database design for information stored about vaccine trials:



When setting up human trials for experiments, some human participants will be added to a trial but can later become ineligible and need to be removed from the trial. Participants only take part in a single trial but this is not explicitly enforced within the schema. So, if someone is a participant in a trial for vaccine A, that is the only trial they will participate in.

(i) If a human participant is deleted from a trial, then all of his/her information should be deleted. Say what kind of constraint this applies to, security or integrity.

[3 Marks]

Answer should identify that Integrity Constraints apply here explain the reasoning for this answer.

(ii) Below is a list of potential operations that could be carried out on the database. For each of the below examples, say *whether or not* the operation violates integrity constraints and if so, the *kind of integrity constraint* that is violated and *why it is violated*:

(a) A new trial is started and information needs to be stored about that trial. Because we don't yet know the ID number of the trial at this point, an *insert* is carried out on the Trials table to insert the information about the new Trial, leaving Trial_ID null for now.

[3 Marks]

Answer should determine that Trial_ID is a primary key in the Trials table, therefore inserting information into the Trials table and leaving Trial_ID null is an entity constraint violation.

(b) Another trial begins, this time they do not yet have an estimated end date for the trial, so a tuple is inserted into the Trials table with a single attribute set to null, the End_date.

[3 Marks]

Answer should determine that No constraints are violated in this case.

(c) A trial is deleted from the Trials table with nothing removed from any other tables.

[3 Marks]

Answer should determine that this is likely to violate referential integrity since the patients of the trial are not deleted, they will reference a trial that no longer exists in the Trial table since Trial in Participants is a foreign key.

(iii) Provide the SQL command to add a constraint that enforces the start date of all human trials to be after 28th February 2019.

[5 Marks]

Answer: A complex constraint can be placed on the Trials table with a CHECK. In SQL the example constraint is applied as follows: CHECK (Start_date > '2018-02-28')

(iv) Provide the SQL to avoid a referential integrity violation that would result from a Trial being deleted from the Trials table.

[8 Marks]

Answer:

CREATE TABLE Participants … FOREIGN KEY (Trial) REFERENCES Trials(Trial_ID) ON DELETE SET NULL

Q. 3

Stocks

SKU	style	colour	size	product_name	customer_type	quant_stock
006	Skinny	Blue	Medium	Jeans	Women	110
007	Bootcut	Black	Small	Jeans	Men	150
008	bootcut	black	Medium	Jeans	Men	143

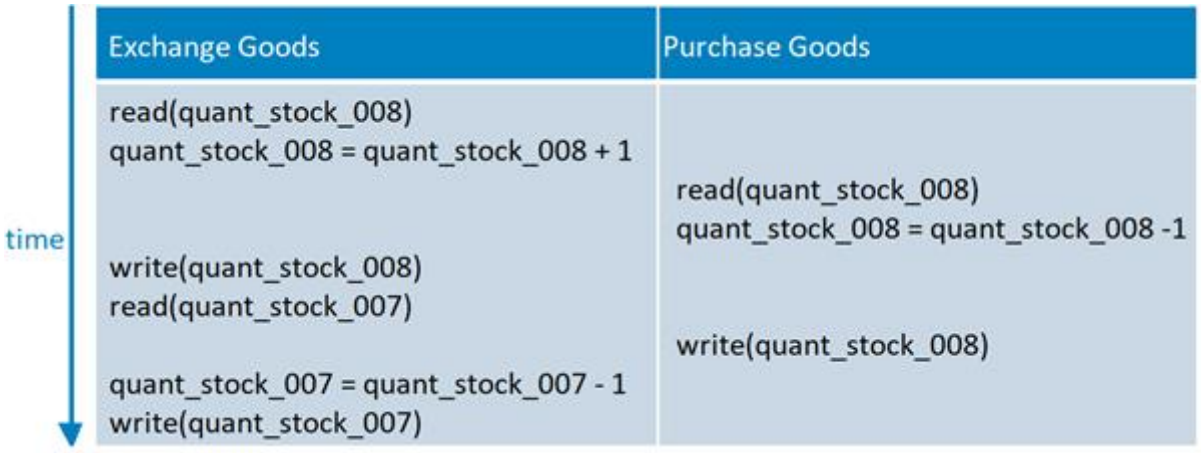
The above database is used to record the stock levels of products in an online clothes shop. Each product for sale can be uniquely identified by its stock keeping unit (SKU), a code

that is also present in the bar code used when paying for items. Data within the table should accurately reflect the actual number of items stored in boxes in the warehouse to avoid items being sold to a customer when they are in fact already sold out.

When a product is returned, via the local postal service, an employee is required to log this item as returned and purchase and send out a replacement item. This often happens when someone buys an item that is the incorrect size, for example.

In addition, the ordinary purchase of items also regularly takes place and both exchange of goods transactions and purchase transactions can potentially occur at the same time. A good database manager will ensure that if two such transactions occur at the same time, the stock levels recorded in the database after both transactions are complete, will still be correct.

Below is one possible way the transactions could be executed concurrently within such a system:



(i) Provide the schedule of operations depicted in the diagram, where Exchange Goods is $Transaction_1$ and Purchase Goods is $Transaction_2$, and a list of any potential conflicts.

[5 Marks]

ANSWER:

S1: R1(quant_stock_008); R2(quant_stock_008);

W1(quant_stock_008);R1(quant_stock_007);

W2(quant_stock_008);W1(quant_stock_007)

R1(quant_stock_008) and W2(quant_stock_008) conflict

W1(quant_stock_008) and R2(quant_stock_008) conflict

W1(quant_stock_008) and W2(quant_stock_008) conflict

(ii) Provide an alternative schedule that fixes any issues with the above schedule and describe any advantages or disadvantages of the approach you chose.

[8 Marks]

Answer:

The two transactions could be arranged into a serial schedule as follows:

S1: R1(quant_stock_008); W1(quant_stock_008);R1(quant_stock_007); W1(quant_stock_007)

R2(quant_stock_008);W2(quant_stock_008);

This would mean that operations in each transaction are no longer interleaved. This means that the schedule is correct because each transaction is correct if completed on its own and therefore the ordering of transactions in a serial schedule doesn't matter. However, a disadvantage is that this approach limits concurrency, a transaction that takes a long time to complete, leaves others waiting too long.

(A serial schedule that positions T2 before T1 is also correct with the same disadvantages.)

(iii) Add locking to the schedule and explain what will occur when the wait-die protocol is applied, making use of a wait-for graph in your explanation.

[12 Marks]

Answer:

writelock(quant_stock_008)

read(quant_stock_008)

quant_stock_008 = quant_stock_008 + 1

writelock(quant_stock_008)

read(quant_stock_008);

quant_stock_008 = quant_stock_008 -1

write(quant_stock_008)

unlock(quant_stock_008)

writelock(quant_stock(007)

read(quant_stock_007);

write(quant_stock_008);

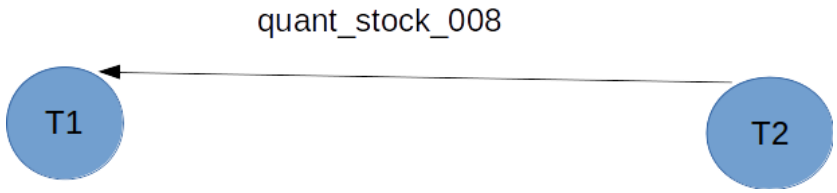
unlock(quant_stock_008)

quant_stock_007 = quant_stock_007 - 1

write(quant_stock_007)

unlock(quant_stock_007)

Deadlock does not occur. Since T2 is younger than T1 when it seeks a lock on quant_stock_008 it is allowed to wait. The following wait-for-graph is generated when T2 must wait for a lock to be released from T1

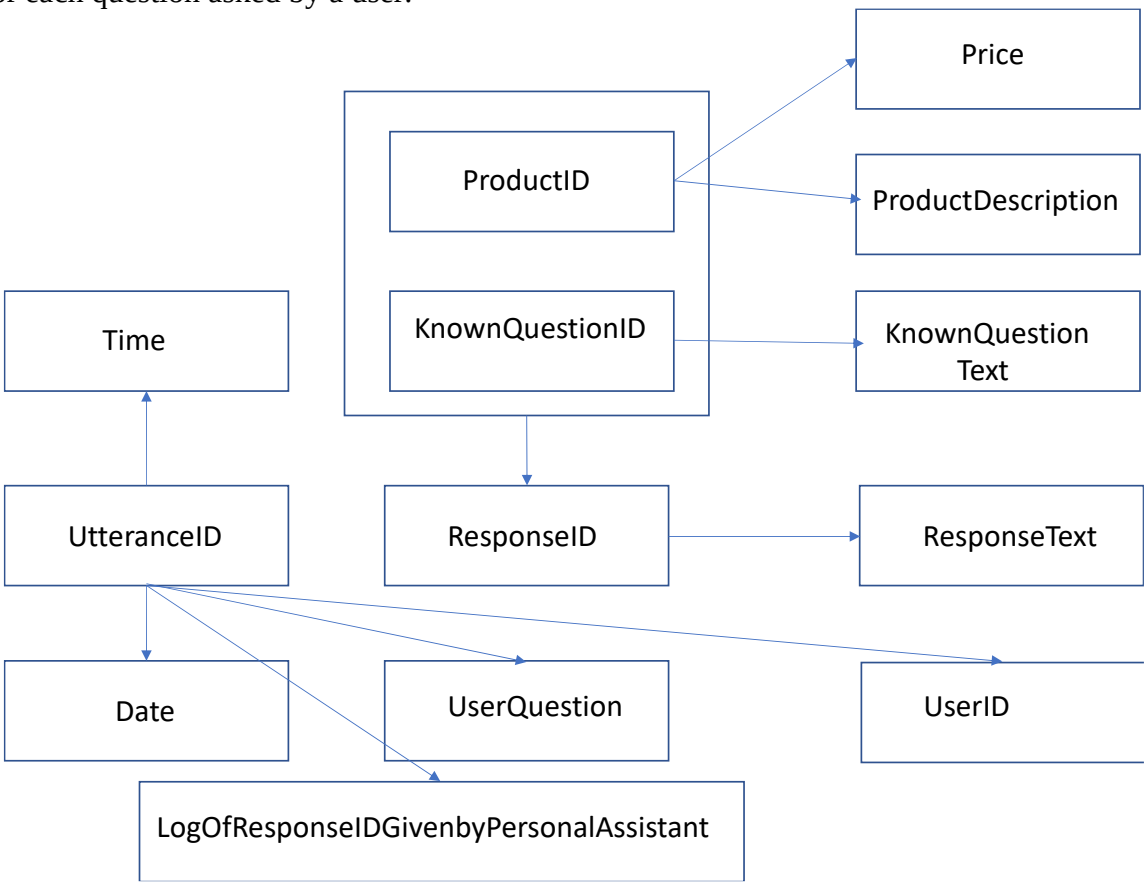


T1 continues as it is not waiting on a lock to be released from T2. When T1 eventually releases the lock from quant_stock_008, the directed edge is removed from the wait-for-graph:

This means that T2 can continue and no more waiting is encountered in either transaction as they both involved different data items from that point on.

Q. 4 Personal Assistants are programmes which respond to user utterances (spoken or typed commands). For example, they can be used to provide helpful answers to a user's question about products on a website. A simple Personal Assistant works by (approximately) matching a user's query about a product to a previously known question about that product and then choose the appropriate answer or response for that known question. If the simple Personal Assistant cannot match the user's query to a known question & product, it just returns a default error message "Sorry I don't understand, can you rephrase your question?").
Suppose a Simple-PersonalAssistant contains a database to store 'known questions' about products available on a retail website and a set of appropriate responses for such known questions. The Simple-PersonalAssistant works by taking each user's utterance (question) and matching it to the most similar known question stored in the database. You may assume the Simple-PersonalAssistant uses an internal algorithm to match the user question about a particular product to the most similar known question. It then looks up which response is appropriate for that known product related question and retrieves that response. Thus, the Simple-PersonalAssistant database stores a list of known questions about the website products, the product(s) to which each known question relates and the IDs of the response appropriate for that known question. For each product the database stores the product ID, its description and its current retail price. The database also stores each predefined Simple-PersonalAssistant's response (as text) and a unique ResponseID to identify each response. For logging purposes, the database also stores the original text of a user's question (utterance), the user ID of the person who asked the question, the time &

date of which the question was asked and a log of the ResponseID which the Simple-PersonalAssistant gave for the question (utterance). As it cannot be assumed that every user question is unique, the Simple-PersonalAssistant also generates a unique utterance ID for each question asked by a user.



(i) Identify the primary keys, foreign keys and other attributes of EACH of the tables in the above Functional Dependency diagram.

[9 Marks]

UtteranceTable(UtteranceID, Date, Time,
LogofResonseIDGivenByPersonalAssistant, User Question, UserID)
ProductTable(ProductID, ProductDescription, Price)
QuestionTable(KnownQuestionID, QuestionText)
ProductQuestionTable(*ProductID*, *KnownQuestionID*, *ResponseID*)
ResponseTable (ResponseID, ResponseText)
(Primary keys are underlined, foreign keys are in italic. If attributes are both primary key(s) and foreign keys that are underlined & in italics).

(ii) Using examples from the tables in (i) above, explain 1st, 2nd and 3rd Normal Forms.

[3 Marks]

Answer should give example of 1st Normal Form: where the domain of each attribute in a relation contains only atomic values and the value of each attribute contains only a single value from that domain

Answer should give example of 2nd Normal Form where a relation in addition to satifying the criteria for 1st normal form, is such that every non-key column in that relation is *fully functionally dependent* on the entire primary key.

Answer should give example of 3rd Normal Form where a relation, in addition to satisfying the criteria for 2nd Normal Form, and is such that no non-key attributes in that relation are *transitively dependent* upon the primary key

(iii) Write SQL commands to perform the following:

(a). List all of the KnownQuestions and their ResponseTexts for the ProductID 10090.

[5 Marks]

Answer should give the appropriate SELECT command across the ProductQuestionTable and ResponseTable using the condition ProductID=10090

(b). Give the SQL commands to store a new Response where the ResponseID is 555 and the ResponseText is "This product can only be sold to a person over 18 years of age". Suppose this response is the appropriate response for KnownQuestionIDs 77 & 78, about the Product with ProductId 10010. Give the SQL commands needed to relate the response to these KnownQuestionIDs and ProductID. You may assume that the Product information for ProductId 10010, and related KnownQuestionTexts for KnowQuestionIDs 77 & 78 already exists in the database but had not yet been assigned a ResponseID

[5 Marks]

Answer should give the appropriate INSERT COMMAND to insert a new ResponseID and ResponseText into the Response Table. Answer should also give the two additional INSERT commands needed to store the ProductID, KnownQuestionID and ResponseId into the ProductQuestionTable as required.

(c). Give the SQL command to define the table which contains User Question, Utterance ID, the User Id, the Time and Date of the question and the log of ResponseID given by personalAssistant for the question. In your answer indicate how you would ensure that the UtteranceID would be automatically created and always unique.

[3 Marks]

Answer should give the SQL CREATE TABLE command for the Utterance table with AutoIncrement on the primary key (utteranceId).