Week 3 Physical Data Modeling

Assignment Solutions

Question 1 - multiple choice, shuffle

The figure shows an instance of a given relation R. Which of the given schemas is valid for this instance of R?

Game						
name	**publisher**	**releaseDate**	**developer**	**platform**	**rating**	**genre**
MoonCraft	Hail Entertainment	1/1/1998	Hail Entertainment	Doors OS	4	Strategy
Hello	Macro Studios	10/5/2001	Bun Gee Jump	Doors OS	5	FPS
Hello	Macro Studios	10/5/2001	Bun Gee Jump	Y Crate	5	FPS
Archfiend	Hail Entertainment	10/5/2001	Hail Entertainment	Doors OS	3	Strategy
OverWork	Hail Entertainment	5/28/2016	Hail Entertainment	Y Crate	4	FPS
OverWork	Hail Entertainment	5/28/2016	Hail Entertainment	Y Crate	4	Strategy
OverWork	Hail Entertainment	5/28/2016	Hail Entertainment	Doors OS	4	FPS
OverWork	Hail Entertainment	5/28/2016	Hail Entertainment	Doors OS	4	Strategy
Loigee Cart	Tentendo	11/15/2007	Tentendo	Tentendo Mee	2	Racing
Loigee Cart	Tentendo	11/15/2007	Tentendo	Tentendo Snitch	2	Racing
4-Eleven	Change	6/9/2017	No Enough Time	PrairieLearn	5	Fantasy
4-Eleven II	Change	11/9/2017	No Enough Time	PrairieLearn	5	Fantasy
MineGraft	JangMoo	7/3/2011	JangMoo	Y Crate 180	2	Sandbox
MineGraft	JangMoo	7/3/2011	JangMoo	Tentendo Mee U	2	Sandbox
OverWork	Homework Studios	5/28/2016	Homework Studios	Doors OS	1	Strategy
OverWork	Homework Studios	5/28/2016	Homework Studios	Y Pad	2	Strategy

*A: Game (name, publisher, releaseDate, developer, platform, rating, genre)

B: Game (name, publisher, releaseDate, developer, platform, rating, genre)

C: Game (name, publisher, releaseDate, developer, platform, rating, genre)

D: Game (name, publisher, releaseDate, developer, platform, rating, genre)

E: Game (name, publisher, releaseDate, developer, platform, rating, genre)

Solution: Option A

Explanation: Only option A's keys can be satisfied by the given data. All other options will violate one or more key constraints because there will be two different rows with the same values for the values associated with the attributes of the key for those options.

Question 2 - multiple choice, shuffle

The following figure shows an instance of a relation R. If {name, platform} is a key, what tuples are "incorrect"?

Game						
name	**publisher**	**releaseDate**	**developer**	**platform**	**rating**	**genre**
MoonCraft	Hail Entertainment	1/1/1998	Hail Entertainment	Doors OS	4	Strategy
Hello	Macro Studios	10/5/2001	Bun Gee Jump	Doors OS	5	FPS
Hello	Macro Studios	10/5/2001	Bun Gee Jump	Y Crate	5	FPS
Archfiend	Hail Entertainment	10/5/2001	Hail Entertainment	Doors OS	3	Strategy
OverWork	Hail Entertainment	5/28/2016	Hail Entertainment	Y Crate	4	FPS
OverWork	Hail Entertainment	5/28/2016	Hail Entertainment	Y Crate	4	Strategy
OverWork	Hail Entertainment	5/28/2016	Hail Entertainment	Doors OS	4	FPS
OverWork	Hail Entertainment	5/28/2016	Hail Entertainment	Doors OS	4	Strategy
Loigee Cart	Tentendo	11/15/2007	Tentendo	Tentendo Mee	2	Racing
Loigee Cart	Tentendo	11/15/2007	Tentendo	Tentendo Snitch	2	Racing
4-Eleven	Change	6/9/2017	No Enough Time	PrairieLearn	5	Fantasy
4-Eleven II	Change	11/9/2017	No Enough Time	PrairieLearn	5	Fantasy
MineGraft	JangMoo	7/3/2011	JangMoo	Y Crate 180	2	Sandbox
MineGraft	JangMoo	7/3/2011	JangMoo	Tentendo Mee U	2	Sandbox
OverWork	Homework Studios	5/28/2016	Homework Studios	Doors OS	1	Strategy
OverWork	Homework Studios	5/28/2016	Homework Studios	Y Pad	2	Strategy

*A: Tuples for "Overwork".

B: Tuples for “Loigee Cart”

C: Tuples for “MineGraft”

D: Tuples for “Hello”

Solution: Option A

Explanation: The game "Overwork" is the only one that has more than one tuple with the same value for the attributes in the key.

Question 3 - multiple choice, shuffle

Since a relation is a **set** of tuples, how these tuples are ordered does not matter. Furthermore, the attributes can be reordered without causing any loss of information. An instance of a relation is equivalent to another instance if it contains the permuted rows and/or columns of the other.

How many equivalent representations are there for the relation in the given figure?

Game						
name	**publisher**	**releaseDate**	**developer**	**platform**	**rating**	**genre**
MoonCraft	Hail Entertainment	1/1/1998	Hail Entertainment	Doors OS	4	Strategy
Hello	Macro Studios	10/5/2001	Bun Gee Jump	Doors OS	5	FPS
Hello	Macro Studios	10/5/2001	Bun Gee Jump	Y Crate	5	FPS
Archfiend	Hail Entertainment	10/5/2001	Hail Entertainment	Doors OS	3	Strategy
OverWork	Hail Entertainment	5/28/2016	Hail Entertainment	Y Crate	4	FPS
OverWork	Hail Entertainment	5/28/2016	Hail Entertainment	Y Crate	4	Strategy
OverWork	Hail Entertainment	5/28/2016	Hail Entertainment	Doors OS	4	FPS
OverWork	Hail Entertainment	5/28/2016	Hail Entertainment	Doors OS	4	Strategy
Loigee Cart	Tentendo	11/15/2007	Tentendo	Tentendo Mee	2	Racing
Loigee Cart	Tentendo	11/15/2007	Tentendo	Tentendo Snitch	2	Racing
4-Eleven	Change	6/9/2017	No Enough Time	PrairieLearn	5	Fantasy
4-Eleven II	Change	11/9/2017	No Enough Time	PrairieLearn	5	Fantasy
MineGraft	JangMoo	7/3/2011	JangMoo	Y Crate 180	2	Sandbox
MineGraft	JangMoo	7/3/2011	JangMoo	Tentendo Mee U	2	Sandbox
OverWork	Homework Studios	5/28/2016	Homework Studios	Doors OS	1	Strategy
OverWork	Homework Studios	5/28/2016	Homework Studios	Y Pad	2	Strategy

*A: 16! 7!

B: 8!

C: 17!

D: 17!8!

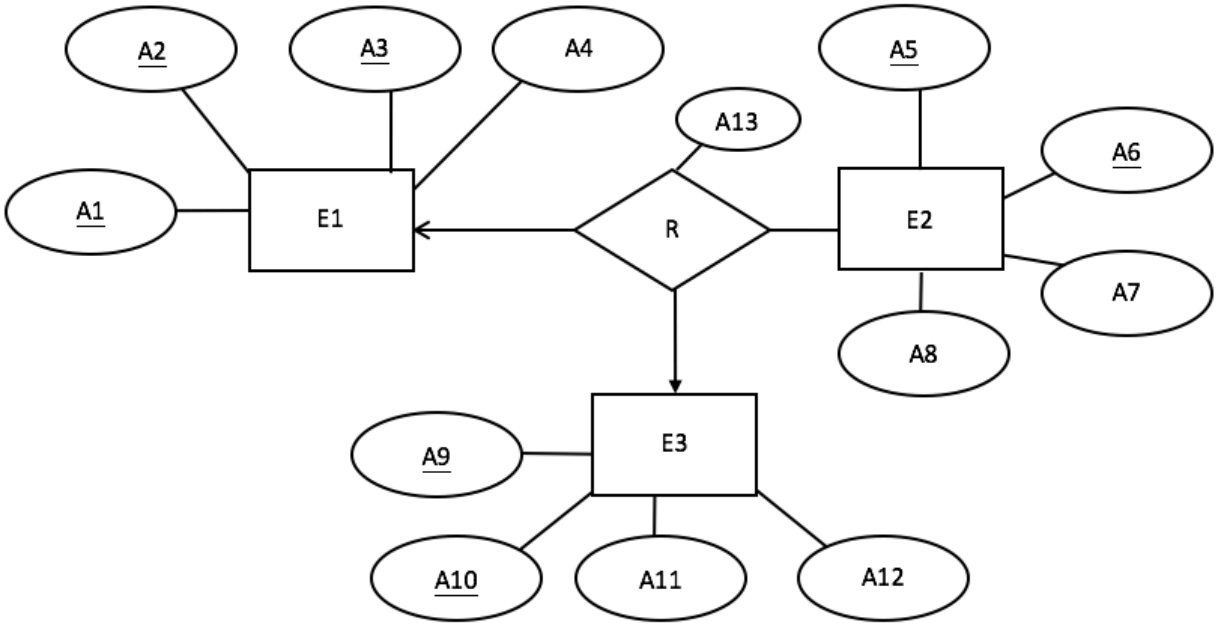
E: 1

Solution: Option A

Explanation: There are 16 rows and 7 columns. Each permutation of row and/or column in a table is equivalent to another. Total 16!7! ways to permute rows and columns.

Question 4 - multiple choice, shuffle

Select the relational database schema corresponding to the ER design in the figure. If possible, combine the relation corresponding to the relationship with the relation of one of the entities.



*A: E1 (A1, A2, A3, A4) , E2 (A1, A2, A3, A5, A6, A7, A8, A9, A10, A13) , E3 (A9, A10, A11, A12)

B: E1 (A1, A2, A3, A4) , E2 (A5, A6, A7, A8) , E3 (A9, A10, A11, A12) , R (A1, A2, A3, A5, A6, A9, A10)

C: E1 (A1, A2, A3, A4) , E2 (A5, A6, A7, A8) , E3 (A9, A10, A11, A12)

D: E1 (A1, A2, A3, A4) , E2 (A5, A6, A7, A8) , E3 (A9, A10, A11, A12) , R (A1, A2, A3, A5, A6, A9, A10, A13)

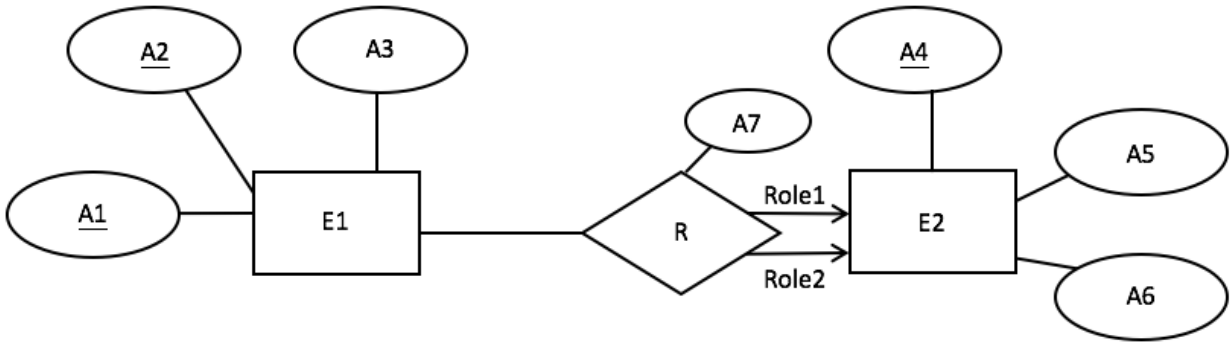
E: E1 (A1, A2, A3, A4) , E2 (A5, A6, A7, A8) , E3 (A1, A2, A3, A5, A6, A9, A10, A11, A12, A13)

Solution: Option A

Explanation: Each entity has its own relation. The relation for the relationship R contains the keys of all three entity sets and its own attribute A13. This relation can be merged into the relation for E2 as E2 is on the "many" side of the relationship. Merging relation for R with the other relations will introduce redundancy. Relevant topic is included in lecture video 2.4

Question 5 - multiple choice, shuffle

Select the relational database schema corresponding to the ER design in the figure. If possible, combine the relation for the relationship with the relation of one of the entities.



*A: E1 (A1, A2, A3, role1_A4, role2_A4, A7) , E2 (A4, A5, A6)

B: E1 (A1, A2, A3), E2 (A4, A5, A6)

C: E1 (A1, A2, A3), E2 (A4, A5, A6), R (A1, A2, A4, A7)

D: E1 (A1, A2, A3, A4, A7), E2 (A4, A5, A6)

E: E1 (A1, A2, A3), E2 (A4, A5, A6), R (A1, A2, role1_A4, role2_A4, A7)

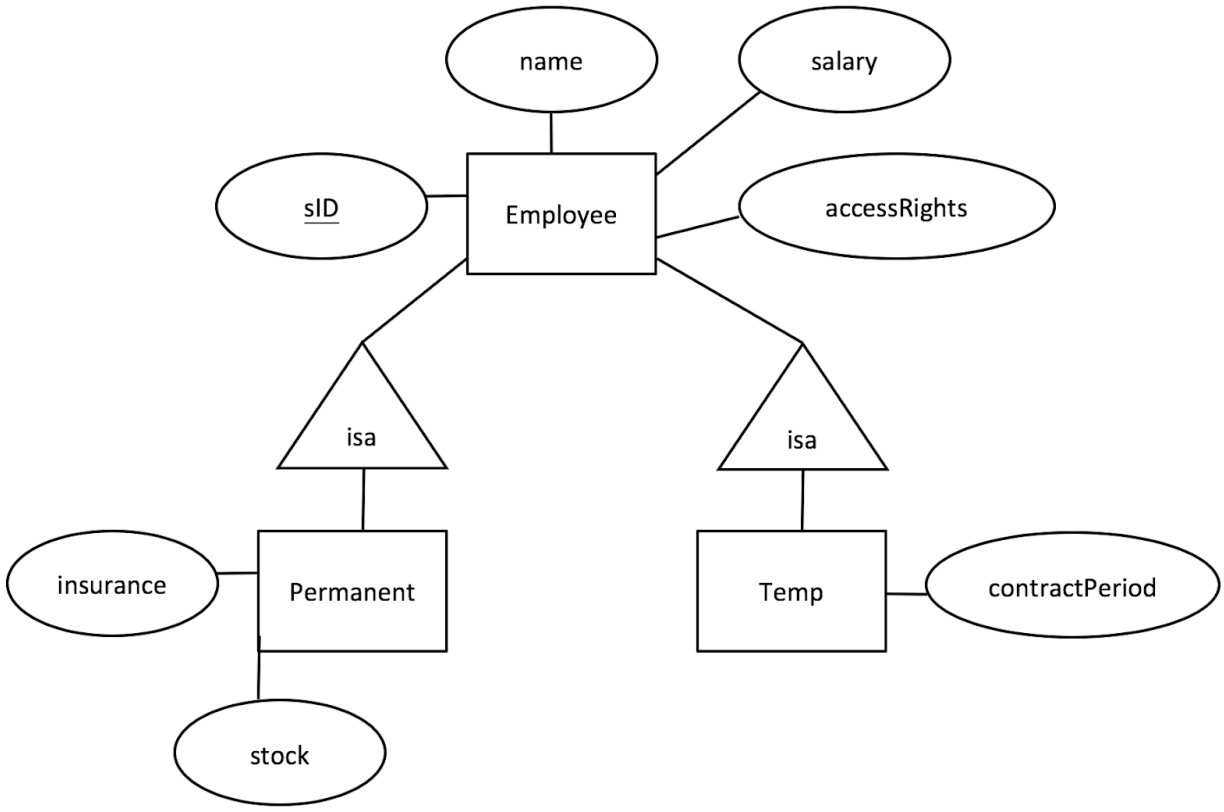
Solution: Option A

Explanation: In lecture video 2.4, we discussed the Rule 3: Combining Many-One Relationships, which is to combine the relation of a many-one relationship with the relation of the "many"-side entity set. In this diagram, E1 is on the 'many' side and E2 is on the 'one' side, hence Relation R can be merged into E1

Each role introduces an attribute corresponding to the key of E2 suitably renamed to the relation R, which also has attribute A7.

Question 6 - multiple choice, shuffle

Select the relational database schema corresponding to the ER design in the figure **using the ER approach**. Note that an employee must either be a permanent staff or a temp and cannot be both. That is, the **subclasses Permanent and Temp are disjoint**.



A: PermanentEmployee (sID, name, salary, accessRights, insurance, stock), TempEmployee (sID, name, salary, accessRights, contractPeriod)

*B: Employee (sID, name, salary, accessRights), Permanent (sID, insurance, stock), Temp (sID, contractPeriod)

C: Employee (sID, name, salary, accessRights), Permanent (sID, insurance, stock), Temp (sID, contractPeriod), IsPermanentEmployee(sID, insurance, stock), isTempEmployee (sID, contractPeriod)

D: Employee (sID, name, salary, accessRights), Permanent (sID, insurance, stock), Temp (sID, contractPeriod)

Solution: Option B

Explanation: In lecture 2.5, we discussed three options of translating subclasses. Here we will take the ER approach. First, we create schemas for the three entity sets. Each subclass entity set has its own 'additional' attributes. And notice that in the ER approach, the subclass table is different from the superclass, but the key is the primary key of the superclass.

Question 7 - multiple choice, shuffle

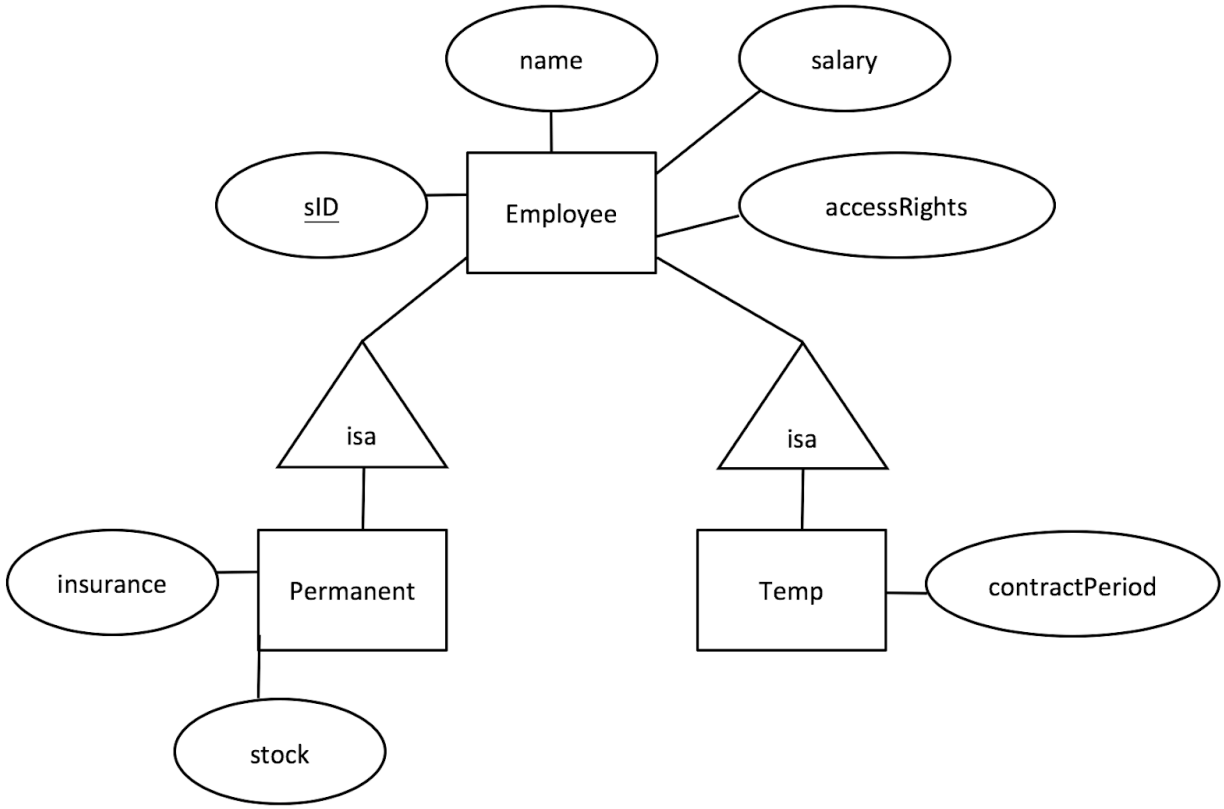
Select the relational database schema corresponding to the ER design in the figure **using the object-oriented approach**.

It is possible for an entity to belong to multiple subclasses. In this case, **someone who is both a permanent and temp employee is a "probationary employee" .**

Probationary employees are hired as a permanent staff but has a probation period (contractPeriod from the temp subclass) during which their contracts can be terminated for poor performance or if they are ill-suited for the job.

Note that an employee must belong to exactly one of the three categories: permanent employee, temp employee, or a "probationary employee".

Simplify the database schema by merging appropriate relations if it will not result in any loss of information.



*A: PermanentEmployee (sID, name, salary, accessRights, insurance, stock), TempEmployee (sID, name, salary, accessRights, contractPeriod), ProbationaryEmployee (sID, name, salary, accessRights, insurance, stock, contractPeriod)

B: Employee (sID, name, salary, accessRights), Permanent (sID, insurance, stock), Temp (sID, contractPeriod)

C: PermanentEmployee (sID, name, salary, accessRights, insurance, stock), TempEmployee (sID, name, salary, accessRights, contractPeriod)

D: Employee (sID, name, salary, accessRights), Permanent (sID, insurance, stock), Temp (sID, contractPeriod), IsPermanentEmployee(sID, insurance, stock), isTempEmployee (sID, contractPeriod)

Solution: Option A

Explanation: Under the OO approach, we should first end up with four relations – Employee, PermanentEmployee, TempEmployee, and ProbationaryEmployee. Employee is an empty table as all employees must be exactly one of the other three and so can be discarded.

Question 8 - multiple choice, shuffle

Which of the following is **NOT** a valid JSON data?

*A: { “Title” : “Fifa 2017”, “Platform” : { “PlayStation”, “Windows”, “Mac OS X 10.12” }, “Genre” : “Sports” }

B: ["John Doe"]

C: [false]

D: {"Accounts" : 12356 }

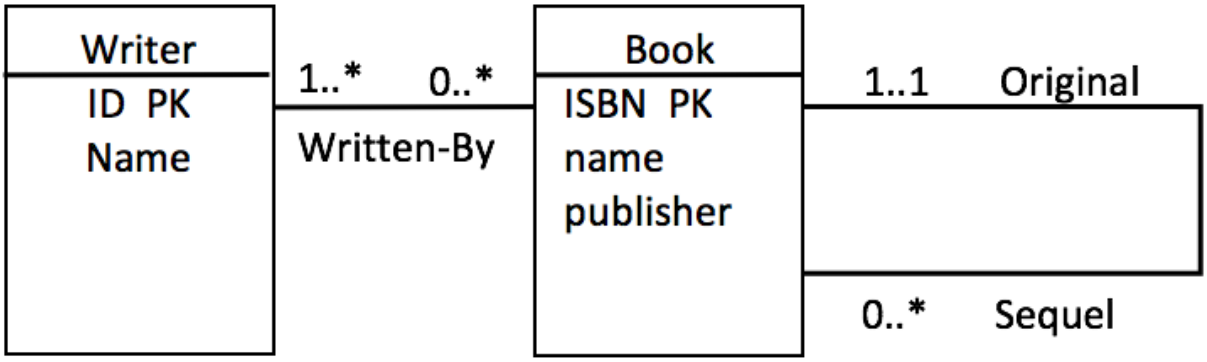
E: [{}]

Solution: Option A

Explanation: Objects in JSON must be field-value pairs so the value of “Platform” is invalid. It should have been a list [“PlayStation,” “Windows,” “Mac OS X 10.12”].

Question 9 - multiple choice, shuffle

Convert the model specified by the UML diagram to a relational schema. Combine appropriate relations for the relationships if it does not result in any loss of information or introduce any redundancy.



*A: Book (ISBN, name, publisher, ISBNofOriginal), Writer (ID, name), WrittenBy (ISBN, ID)

B: Book (ISBN, name, publisher), Writer (ID, ISBNofBook, name), Sequels (ISBNofOriginal, ISBNofSequel)

C: Book (ISBN, ISBNofSequel, name, publisher), Writer (ID, name), WrittenBy (ISBN, ID,)

D: Book (ISBN, ISBNofSequel, name, publisher), Writer (ID, ISBNofBook, name)

Solution: Option A

Explanation:

Each class converts to one relation containing the attributes of the class. Primary key for each class is specified in the diagram and they become primary key of relation. Associations are analogous to relationships so they convert to a relation as well containing the keys of the classes connected to it along with any attributes of its own (none here). The association of Book to itself denoting the sequel relationship becomes a table as well containing the key for the original book and the sequel (similar situation as the case for ER modelling). However, this is a many-one relationship and so can be merged (required by this question) into the Book relation so the key for the original (ISBNofOriginal) is added to the Book relation. Adding ISBNofSequel to Book is wrong as an “original” book can have many sequels so the relationship from original to sequel is one-many. Trying to add ISBNofSequel will introduce one row for every original and its sequel and the information of the original will be repeated (a form of redundancy).

- B is wrong - Missing relation for the Written-By association. Sequels should be merged into Book.
- C is wrong - A sequel is not unique. Merging the relation for the association between Book and itself into Book this way will result in rows containing duplicate information about the book. ISBNofOriginal should be stored in Book instead.
- D is wrong - Missing relation for the Written-By association. Also, the way the sequel association is merged into Book is incorrect. See correct answer for explanation.

Question 10 - multiple choice, shuffle

Scenario 1:

You need to store a large collection of images. Each image has tags indicating the objects in that image. The most common query is to retrieve all images associated with a given set of labels.

Scenario 2:

You run an online discussion forum BufferOverflow for programmers. Your system needs to keep track of registered users and allow them to post questions, answers, and comments. The posts can be tagged with topics. Some common queries are: retrieve all posts containing a set of keywords, retrieve posts related to a given set of topics within a time period, and retrieve all posts written by a particular user.

Scenario 3:

You are tasked with building the backend of a social network. This requires you to construct a knowledge base, which is essentially a collection of interconnected data. The system must be able to discover products that are similar to those "liked" by a user or people close to her (i.e., her family, friends, or friends of friends, etc.) so that the system can display relevant advertisements to the user.

Which type of database, relational, key-value, or graph, is the most appropriate for scenarios 1, 2, and 3?

*A: 1: Key-value, 2: Relational, 3: Graph

B: 1: Relational, 2: Key-value, 3: Graph

C: 1: Key-value, 2: Key-value, 3: Graph

D: 1: Relational, 2: Graph, 3: Key-value

Solution: Option A

Explanation:

- Scenario 1: Query is simple – retrieve values (images, or more specifically, image filenames / URLs) given keys, which are the labels tagged to each image. For a key-value DB such as Redis, images can be represented as URLs/filenames in a sorted set associated with a tag. Intersect sets of images to get images containing a set of tags. A relational DB is not well-suited for this, especially if the user wants to look up images that has a large set of specified tags as that might require many table joins. Graph DB is not suited for this as there are practically no significant relationships to be modelled here.

- Scenario 2: This is a typical discussion forum application. Each entity has a fixed set of attributes, which means the schema is fixed and can be determined right from the start. The queries are also fairly complicated and require use of various attributes from various entities. Relational DB is

good for such use cases but the key-value DB is not because of the complicated queries.

- Scenario 3: The social network in this case can be well modelled using a graph. Entities are represented as nodes and relationships as edges. Graph DB is suitable for this as it makes it easy to discover relationships between entities by examining relationships between them. Relational DB is not well suited for this. The information for various entities will be spread across many tables. Same for the relationships. This means any query to find relationships between entities and entities not directly related to them will require several complicated operations involving many joins. Queries will also be complex and unwieldy. Key-value DB is not good for this case as it does not model connections between entities well.