Comparing Information Extraction Between Instance-Based Data Models and Relational Data Models

by

© **Kiumars Dorani**

A thesis submitted to the School of Graduate Studies in partial fulfillment of the requirements for the degree of Master of Science.

Faculty of Business Administration

Memorial University of Newfoundland

August 2023

St. John's, Newfoundland and Labrador, Canada

Abstract

Instance-based representation has been developed to overcome the limitations of class-based models for storing data. A class-based data model organizes data into pre-defined classes that represent specific entities within a domain. However, instance-based model introduces two separated layers for representing instance and classes, freeing instances from pre-defined, fixed schemas and enabling more dynamic and flexible data representations. Despite the well-established theoretical foundations of instance-based representation, there is little empirical research that investigates its practical usefulness. In this study, we conduct an experiment to compare the effectiveness of information extraction between instance-based data models and class-based data models. Participants randomly received data represented using data structured according to one of the models and answered information extraction/retrieval questions. The results show that, depending on the type of information extraction task, one representation supported more effective retrieval than the other, suggesting that the models can be complementary. In complex use cases including extracting information about relationships of instance/entities and retrieving information involving instances from different classes, the instance-based model outperformed the class-based model. On the other hand, for simpler use cases involving extracting information about cardinalities of relationships and retrieving information involving only one entity (i.e., instances from a same class), the class-based model proved to be more effective. The findings both provide empirical evidence for the effectiveness and usefulness of the instance-based model and demonstrate how it can complement the class-based model in representing the domain.

Keywords: Conceptual modeling, data modeling, instance-based data models, class-based data models.

Acknowledgements

I would like to thank my supervisor, Dr. Jeffrey Parsons, who introduced me to the topic of instance-based data modeling and helped me greatly through the research process.

I also would like to express my gratitude to the thesis examiners, Dr. Jennifer Jewer and Dr. Sherrie Komiak, who dedicated their time and expertise to carefully review and evaluate this thesis.

Additionally, I extend my appreciation to the students of Memorial University of Newfoundland who participated in the experiment that forms the outcomes of this study. Their willingness to contribute their time and efforts has been instrumental in advancing this research.

Table of contents

List of tables

List of figures

Chapter 1

Introduction

Most of the common methods for data modeling rely implicitly or explicitly on the assumption of *inherent classification*, meaning that instances must belong to a predefined class or conform to an a priori generalized form (Parsons and Wand, 2000). This assumption imposes some limitations to data modeling, such as incompatibility with change of requirements over time, not properly reflecting the uniqueness of instances, and creating bias in user's understanding of the domain (Parsons and Wand, 2000; Lukyanenko et al., 2019).

To overcome these limitations, Parsons and Wand (2000) proposed the instance-based data model, which suggests a two-layered approach (instance layer and class layer) to separate instances from any particular classification. As a result, representations of individual instances do not need to conform to a priori abstraction (Lukyanenko et al., 2019). The instance layer represents the data about individual instances, their properties and related operations, while the class layer consists of information about classes based on a set of properties and related operations. This architecture results in independence of instances from classes, which eliminates several problems associated with class-based data modeling (Parsons and Wand, 2000; Lukyanenko et al., 2019).

There are various empirical studies on how people understand and use class-based data models. These empirical works have explored different grammars and modeling aspects, employing diverse empirical tasks to evaluate subjects' performance across multiple measures such as number of correct answers or accuracy of answers, task completion time, confidence in accuracy of answers, and perceived ease of use and understanding (Saghafi and Wand, 2014). To evaluate the conceptual models, these works typically used ontological theories or cognitive theories to explain the impact of different grammars on subject's performance (Saghafi and Wand, 2014). They have also employed different qualitative tests (including case studies, expert panels, and interviews), quantitative tests (including surveys, laboratory experiments, and field experiments), and hybrid tests (Recker et al., 2019).

In terms of empirical tasks, these employed a diverse range of question types to investigate effectiveness of different class-based data model grammars. These question types included multiple choice comprehension questions (Bodart et al., 2001; Gemino and Wand, 2005; Shanks et al., 2008, 2010), problem solving questions that require examinees to provide a verbal description of the solution (Bodart et al., 2001; Gemino and Wand, 2005; Burton-Jones and Meso, 2006; Shanks et al., 2008, 2010; Parsons, 2011; Bera et al., 2014), Cloze test or fill-in-blank questions (Gemino and Wand, 2005; Burton-Jones and Meso, 2006), Likert-type questions (Bodart et al., 2001; Gemino and Wand, 2005; Allen and March, 2006; Shanks et al., 2008; Parsons, 2011), and writing SQL codes questions (Allen and March, 2006; Bowen et al., 2009).

The instance-based data model (IBDM), as an alternative to classical relational or class-based models, has potential in representing domains so that information about the domain of interest can be extracted effectively (Parsons and Wand, 2000; Lukyanenko et al., 2019). However, relatively limited attention has been paid to the capabilities of IBDMs in representing domains and effectiveness of extracting information from such models (Lukyanenko et al., 2019). Thus, there is a limited understanding about the potential values of explicit representation of instances in conceptual modeling literature (Lukyanenko et al., 2019). Some recent studies highlighted the need for examining/developing appropriate conceptual models for the instance-based approach (Saghafi et al., 2022; Lukyanenko et al., 2014, 2019). Since the ways in which data is collected by organizations is changing and its use cases are evolving (Lukyanenko et al., 2019), there is a need to explore whether the focus on classes in data modeling is adequate for addressing the new requirements like collecting user generated content (Lukyanenko et al., 2014; Lukyanenko and Parsons, 2018) and enabling self-service analytics for non-technical content consumers (Saghafi et al., 2022).

Saghafi et al. (2022) examined the usability of instance-based data through two experiments using a control-treatment design and showed that presenting data to users in the form of instances with no predefined classification schema results in a better performance in both exploring data for pattern discovery and retrieval of information (querying). That study claims to be the first empirical attempt to examine usability and usefulness of instance-based representation compared to class-based relational models. The findings suggested that instance-based representation can be effective in meeting the growing needs to interact effectively with data for self-service analytics requirements (e.g., enabling business users to explore the data without dependence on IT experts) and using data for unanticipated purposes (i.e., using data for purposes not anticipated when it was collected). The results demonstrated that an instance-based approach provides better support for content consumers (business users with domain knowledge who lack technical skills in databases and data modeling) in exploring and analyzing data (Saghafi et al., 2022). However, the two experimental tasks in that study do not distinguish between different aspects of information retrieval and does not discuss the different types of information extraction requirements. In the first experiment, after giving the instance-based and class-based data models to the associated groups, the subjects of each group were asked to use Tableau interface to report patterns with potential value for further investigation by stakeholders. This type of data usage (i.e., listing the potential patterns) is only the requirement of specific organizational end-users responsible for generating some reports. Moreover, this experiment only assessed if participants could list some valuable patterns and did not compare the effectiveness of instance-based and class-based data model in providing data to investigate those patterns. In the second experiment, the subjects were asked to provide a verbal description of the procedure to query some information from data model. However, all the questions in this experiment focused on information that required join data of two or more entities/instances.

The purpose of this thesis is to assess and compare the effectiveness of different types of information extraction between instance-based data models and class-based relational data models. This study focuses on different aspects of information extraction requirements and compares the performance of the two models in each of these aspects. As there are very limited empirical studies about usability of instance-based representation, the main contribution of this study is to provide empirical evidence regarding the effectiveness and usefulness of the instance-based representation as a data modeling approach. It explores various dimensions of information extraction from data models and pinpoints the use cases where the instance-based representation exhibits its greatest efficacy. Furthermore, this study aims to assess the capabilities of graph data models in implementing instance-based representation and instance-based data.

The structure of the remainder of this thesis is as follows. Section 2 provides a comprehensive review of class-based and instance-based data models and their advantages and limitations. This is followed in section 3 by developing the hypotheses of the study. Section 4 describes the research methodology, including the research design, experiment materials, and data analysis techniques. Section 5 presents the results of the study and discusses their implications. Section 6 discusses the limitations of the study and suggests directions for future research. Finally, a conclusion is provided in section 7.

Chapter 2

Class-Based Data Models vs. Instance-Based Data Models

In this section we discuss the characteristics of class-based and instance-based models and introduce an implementation of instance-based representation using graph data modeling.

2.1 Class-Based Data Model (Relational Data Model)

The class-based approach is the most widely used data modeling method and is exemplified by the Entity-Relationship model at the conceptual level and the relational data model at the implementation level (Saghafi et al., 2022). It begins by identifying relevant classes in a domain (Parsons and Wand, 2000). At the conceptual level, the Entity-Relationship model uses concepts such as entity, entity-type, relationship, relationship-type, attributes of entities and relationships and cardinalities of relationships to represent a domain (Parsons and Wand, 2000). In this representation, instances only belong to an a priori form in which the schema is fixed and known in advance. At the implementation level, entity types are represented by tables and relationships between instances are represented using foreign keys that link classes/tables. There are some advantages and shortcomings associated with this data modeling paradigm.

2.1.1 Advantages of Relational Data Model

Parsons and Wand (2000) summarized some key advantages of class-based data modeling. It simplifies the complexity of the real world by classifying things as the human being's survival depends on their ability to understand similarities and differences among objects and events. It provides context in communication since using abstractions is the natural way of reasoning and communicating about domains. It can completely represent domains. It promotes inferences of unobserved attributes even though these inferences are not explicit and usually reduce storage requirements and decrease cognitive load for people working with models. Additionally, it helps to create social realities because classes are fundamental to representing social domains (Parsons and Wand, 2000).

2.1.2 Limitations/Problems of Relational Data Model

Despite the important advantages of relational data models, their assumptions impose some limitations and problems to the data models. Class-based models do not support the case in which an instance belongs to two or more classes that are not related through generalization/specialization (the problem of multiple classification). In schema integration, which entails reconciling the views of different users (model developers), the final preferred classes might not correspond to the classes identified by any users (the view integration problem). In this approach the structure of classes is fixed, which is not compatible with the change of requirements over time (the schema evolution problem). They make the exchange of information between systems difficult, especially when the related schemas of the systems are different (the interoperability problem) (Parsons and Wand, 2000).

Furthermore, class-based models cannot properly represent the uniqueness of instances as they ignore the differences between individual instances and instead focus on their similarities (Lukyanenko et al., 2019). The intended uses of data modeled through a class-based approach should be known in advance and be somehow constant over time since they do not support various ways of partitioning the instances making up reality (Lukyanenko et al., 2019). The structure of classes could bias the users' perceptions of the domain as it directs attention to some filtered features of reality, which might result in losing the opportunity of seeing the domain with another point of view to discover something new or realizing emergent patterns (Lukyanenko et al., 2019). Since these models set domain boundaries, they are unable to represent unique instances that might not fit with the boundaries. These unique instances or anomalies might be a rich source of information and insights in some contexts. Consequently, class-based models might limit a comprehensive understanding of the domain due to offering distorted impressions of instances (Lukyanenko et al., 2019).

2.2 Instance-Based Data Model (IBDM)

The core idea of the IBDM is to distinguish and separate two layers of modeling responsible for representing different aspects of a domain: instance layer and class layer. In contrast to the relational model, in which instances only exist as members of classes, in the IBDM the notion of an instance precedes the notion of class and things/instances are independent of any classification. So, based on separation of instances and classes, IBDM represents instances and their properties independent of any classification. The instance layer contains instances and their properties, while the class layer shows how things could be classified for certain purposes (Parsons and Wand, 2000).

2.2.1 Benefits of IBDM

There are different benefits associated with the IBDM. The model represents instance individuality, embraces the uniqueness of instances, and supports the standalone, independent (of classes) nature of instances. In many real-world scenarios, there might be a need to represent individual and unique characteristics of instances, which is not possible using class-based models (Lukyanenko et al., 2019).

IDBM promotes unanticipated uses. A class schema limits the flexibility to use data for purposes other than what it is developed for. However, in IBDM existence of instances is independent of how an observer might classify them, which results in fewer constraints imposed by predefined purposes and more potential for unanticipated uses of data (Lukyanenko et al., 2019).

IBDM represents and promotes open domain boundaries, as opposed to class-based modeling which creates domain boundaries. By depicting individual objects, IBDM representations are incapable of showing boundaries of the domain and therefore convey domain openness. As a result, they promote discoveries about the domain (Lukyanenko et al., 2019).

IBDM can serve as a conceptual model for NoSQL databases, which seems to be more consistent with the instance-based approach for representing instances. Despite the advancements of these modern technologies, there is a pressing need for a conceptual layer to support the understanding of data stored in these databases. IBDM can play this role and make the interpretation of interrelationships among data items in the NoSQL databases possible (Lukyanenko et al., 2019). IBDM can also guide the selection of database technology to determine which technology is most suitable for representing the semantics of a domain (Lukyanenko et al., 2019).

IBDM can facilitate reaching a common understanding between parties involved in IS development as they clarify the process of generalization and abstraction. Thus, they are useful in improving domain understanding (Lukyanenko et al., 2019).

2.2.2 Characteristics of Instance-Based Representation

The main characteristics of the instance-based representation are as follows:

- Uniqueness of instances without the need for conforming to a predefined generalized category/class;
- The standalone, independent (of classes) nature of instances (instances can exist in the database independent of any classification);
- Instances of the same class can have different attributes or properties;
- Each instance can belong to zero to more classes;
- There can be mutual relationships between instances (mutual property); and
- More than one type of relationship can exist between instances (even between instances of the same class);

2.2.3 Instance-Based Data Model and NoSQL Movement

The instance-based approach to data modeling is aligned with NoSQL databases (Saghafi et al., 2022), which do not conform to a fixed schema but, instead, adopt more flexible data models that could be considered schema-less (Davoudian et al., 2018). Although there are different types of NoSQL databases with respect to how they model data and to what types of data set they deal with, they all share some fundamental characteristics, such as being non-relational, non-ACID (Atomicity, Consistency, Isolation and Durability), and schema-less (Kaur and Rani, 2013).

The practice of data modeling is significantly different between relational databases and NoSQL databases. Traditional relational databases mostly rely on Entity Relationship (ER) diagrams to conceptually represent the domain of interest. However, the question of what is an appropriate data modeling grammar, procedure, and level of representation for NoSQL databases still exists (Vera-Olivera et al., 2021). Although NoSQL databases do not adhere to a fixed, pre-defined schema, modeling such databases is still needed as it impacts data size/storage, code readability, and query performance (Vera-Olivera et al., 2021). Based on how NoSQL databases model the data, they can be categorized in four classes including Key-Value stores, Column-oriented stores, Document stores, and Graph stores. Each of these types of databases takes a distinct approach to model the data, deals with different types of data and is best suited for a specific use case and application scenario (Kaur and Rani, 2013; Davoudian et al., 2018).

Key-Value Databases:

In a key-value store, data is organized in a simple data model as key-value pairs (Kaur and Rani, 2013). Data is stored in key-based lookup structures (Davoudian et al., 2018), where keys are matched to values like a dictionary or hash (Deepak, 2016). Each key is unique and could be either simple (e.g., a URI, hash, or filename) or structured (e.g., composite keys) and is used to retrieve the associated value (Kaur and Rani, 2013; Deepak, 2016). A value represents data with any type, structure, and size/length (e.g., a string, document, image, object, or hash) which is identified by a key uniquely. As a result, they allow storing arbitrary data under a key (Corbellini et al., 2017). This simple data model creates great efficiency in querying data (Davoudian et al., 2018). Moreover, since there is no relation and structure in a key-value database, it employs a flexible, schema-less model, which makes it highly scalable as it needs less or no redesign (Deepak, 2016). These systems are suitable for applications that use a single key to access data, such as an online shopping cart, user profile/configuration, and web session information (Davoudian et al., 2018) or applications where schema is prone to evolution (Kaur and Rani, 2013).

Column-oriented Databases:

In this type of stores, data could be represented as a tabular format of rows and (a fixed number of) column-families, which are made up of columns that are related to each other and usually queried together (Davoudian et al., 2018). Columns can be nested inside other columns (Deepak, 2016), so wide-column stores could be considered as extended key-value stores because value is represented as a sequence of nested (key, value) pairs (Davoudian et al., 2018). Not having an entirely pre-structured table to work with data gives it flexibility in data definition (i.e., flexible schema), which allows the application of data compression algorithms per column (Corbellini et al., 2017), making retrieval of large amounts of a particular attribute faster (Deepak, 2016). This change in storage design results in better performance in aggregation operations and ad-hoc and dynamic querying (Kaur and Rani, 2013). As wide-column stores support multiple modeling structures such as rows, column-families, and nested column-families, they can be partitioned horizontally (by rows) and vertically (by column-families), making them suitable for storing huge datasets. Because of the high scalability and flexibility, these databases are suitable for analytical purposes, such as web analytics applications which need to keep track of their visitors' actions.

Document Databases:

These databases are extended key-value stores in which the value is represented as a document encoded in semi-structured formats such as XML, JSON, or BSON (Binary JSON) and key is always a document's ID (Davoudian et al., 2018). Documents are grouped together in the form of collections, which can be compared to relational databases: collections correspond to tables and documents to records (Kaur and Rani, 2013). They are highly flexible in nature as documents in a collection can have different fields, and any number of fields can be added to the documents without the need to add the same empty fields to the other documents in a collection (Kaur and Rani, 2013). Document-oriented databases are suitable for web applications which demand storage of semi-structured data, evolution of data schema, support of agile development methods, and execution of dynamic queries (Kaur and Rani, 2013). For example, in blogging platforms, a blog post which includes various (nested) attributes (such as tags, comments, images, and videos) can be easily represented in a document format (Davoudian et al., 2018).

Graph Databases:

These databases rely on graph-like structure containing vertices (nodes) for representing entities and edges (arcs) relating nodes for representing relationships between them (Kaur and Rani, 2013). Nodes may contain properties to describe the data included within each object. Additionally, edges may also have properties. A relationship, which is identified by a name connects two nodes, can be traversed in both directions, and may be directed to add further meaning to the relationship (Kaur and Rani, 2013). Graph databases support storing semi-structured information and they do not need a predefined schema, resulting in easier adaptation to schema evolution and ability to capture ad-hoc relationships (Kaur and Rani, 2013). They are suitable for finding relationships within huge amounts of data at a faster rate (Kaur and Rani, 2013). Graph databases are different from the three previous NOSQL databases as they focus on storing entity relationship traversals instead of storing information about entities which is the focus of the other three NOSQL types (Davoudian et al., 2018).

2.3 Graph Data Model: A Representation for Instance-Based Data

To have a proper representation of instance-base data, we need a way to depict instances and their properties (instance layer) and the classes to which they might belong or not belong (class layer). A property can inherently belong to an individual instance (intrinsic property) or be meaningful when it is shared between two or more instances (mutual property) (Wand and Weber, 1995). Saghafi et al. (2022) used a representation like Resource Description Framework (RDF) to represent instance-based data, which is the same as a simple graph-based data model. Although practitioners have begun to use graph models to explore instance-based representations, most of these attempts are happening in isolation from academic research and do not have significant empirical support (Lukyanenko et al., 2019). Thus, discussing the appropriateness of graph data models for representing instance-based data and evaluating it as an experiment is another contribution of this thesis to the overall efforts to develop conceptual models for instance-based representation.

Despite slight variations in the notion of graph data models, fundamentally a graph data model is conceptualized as a directed, possibly labeled, graph in which nodes/vertices represent data (entities and/or instances) and edges represent connections among data (Angles and Gutierrez, 2008). A property graph is a directed labeled graph in which nodes and edges can have a set (possibly empty) of property-value pairs (Angles, 2018). In such a graph, nodes represent entities/instances, edges show relationships between entities/instances, and properties describe the attributes of entities/instances and relationships. Nodes and relationships can have labels to classify them, or they may not belong to any classification (Angles, 2018). A unique characteristic of graph data models is that schema and instances can be clearly distinguished (Angles and Gutierrez, 2008).

These specific qualities make graph data models a good candidate for implementing instance-based representation. Additionally, any relational data model (i.e., entity-relationship diagram) can be transformed to a directed graph, meaning that we would have equivalent elements in the two data models (Frisendal, 2016). Thus, comparing the two models is doable.

In this representation of the instance-based approach, the instances are represented as nodes with circle shape. Intrinsic properties are shown within the circle and mutual properties between two instances are depicted as labeled edges connecting the two nodes. Figure 2.1 shows an example of instance-based data model using graph representation in the context of university. In this model there are four instances with different attributes like name, age, title, etc. There are also three types of relationships between instances including ‘take’, ‘teach’, and ‘supervise’.

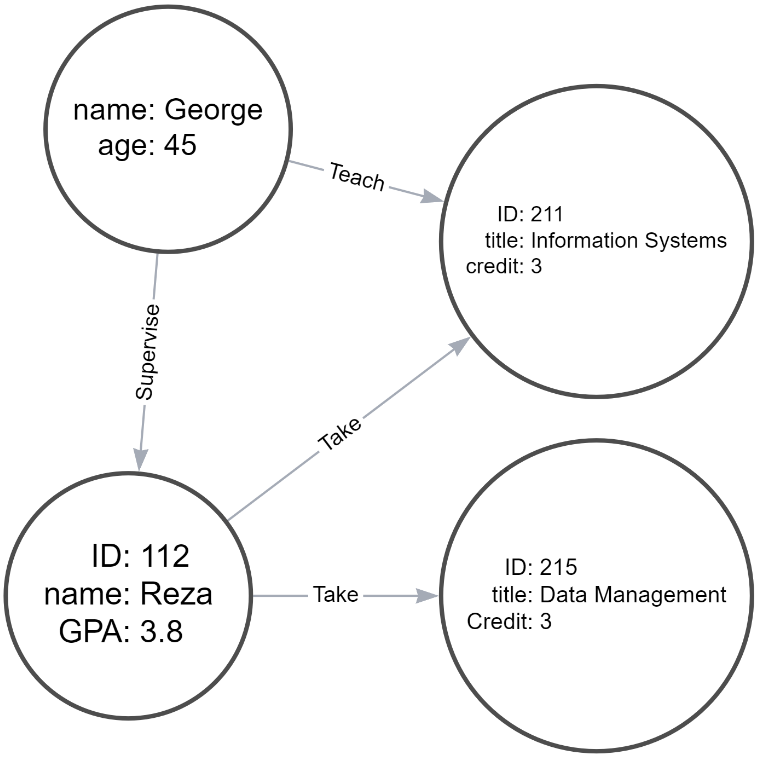


Figure 2.1: An example of instance-based data model using graph representation

Chapter 3

Hypothesis Development

The type of requirement for which a data model is being developed affects the effectiveness of extracting information from the data model. For example, in the context of user-generated contents, where discovery of unique and unknown observations is highly important, relying on class-based models may not be the most effective approach (Lukyanenko et al., 2017). Requirement determination is a critical step of IS development in which an understanding of the problem and the user's needs and expectations of the IS are acquired (Pitts and Browne, 2004). Requirements are represented as conceptual models that drive the design and development of IS components such as database schema, user interface and code (Lukyanenko et al., 2017). Moreover, conceptual models play an important role in understanding the domain, communicating with the development team, and maintaining the system (Lukyanenko et al., 2017). Conceptual data models are one of the conceptual models that are created during IS development and try to capture and represent the data requirements of the system and determine the database technology and schema appropriate for the system.

Data models represents knowledge about the domain of interest (Lukyanenko et al., 2014). Extracting information from data models usually targets different aspects of the model depending on the aims of the user and their requirements. The required information might be about the features of a thing, or mutual relationships between things and the cardinalities of such relationships. Sometimes the information that we need from a data model demand joining the data about two or more things. Thus, we can expect that based on the type of information we want to extract from the data model, the appropriate model might differ. Some data models could be more effective than others for extracting specific kinds of information.

To identify the types of information that can be extracted from a conceptual data model, it is necessary to understand how the models represents real world constructs. Information extraction can target different aspects of how the data model represents ontological constructs. Thus, in general the requirements of information extraction from data models can be grouped into the following categories, which reflect the type of knowledge we want to obtain for the models. These categories are associated with key elements of class-based data models including entities and their properties, relationship between entities, and cardinality of relationships, which can be mapped to different ontological constructs (Wand and Weber, 1995).

- **Information about the properties of instances/entities**: In this kind of information extraction the goal is to identify the properties of a thing (i.e., an instance or entity) and whether a thing possesses a specific property or not.
- **Inferences about relationships among instances/entities**: In this kind of information extraction, the goal is to make inferences about the relationship among things (instances/entities) based on the model.
- **Information about cardinalities of relationships**: In this category the

focus is gaining information about the cardinalities of relationships (one-to-one, one-to-many, many-to-many) and the required information can be obtained by exploiting the relationship's cardinalities.

- **Retrieval operations about instances that share common properties**: Here, performing retrieval operations (querying) about a set of instances that share common properties is concerned (i.e., instances that belong to the same class). The operation could be finding the record with the highest or lowest value of a specific property, finding the count of records having a certain property, or finding the average values of a property. For example, find the student with the highest or lowest age, find the average grades of students, count the number students in graduate programs (age, grade and program are the properties of the class 'Student').

- **Retrieval operation about instances that possess dissimilar properties but have relationships with each other**: In this type of information retrieval (querying), we need to combine data about two or more instances to obtain the required information. For example, suppose we have two classes: Student and Course. Finding the title of courses that a student with a specific name has taken, entails combining data about instances that belong to two potential classes but are related to each other. As another example, finding the title of courses that a student had taken in the previous Fall semester, requires joining the data of instances that belong to three potential classes: student, course, and semester. We are using the term "potential class" because in instance-based representation we do not necessarily specify the class to which instances belong. However, these classes might be attributed to the instances by users of the model.

Since a class does not capture all potential properties of an instance, class-based models result in property loss (Lukyanenko et al., 2014) and are unable to represent unique instances that are unlike other members of the class (Lukyanenko et al., 2019). Each 'thing' is unique as it has unique properties, but classification is based on finding common properties of instances, which usually ignores the properties irrelevant to the purpose of classification. When humans classify, they focus on some similarities between instances, while remaining aware of their individual differences, but this is not the case in conceptual modeling (Lukyanenko et al., 2014). In class-based modeling, users tend to assume that the model exhausts the domain, which might not be the case due to incomplete requirements elicitation or unanticipated domain changes (Lukyanenko et al., 2019). However, the instance-based model may allow representation of a class with the label dog, but at the same time allow that a particular dog has additional attributes (Lukyanenko et al., 2019). Although the IBDM is unable to depict the entire domain, its attempt to show typical members of the class does not inhibit user creativity to expand the scope of the domain (Lukyanenko et al., 2019). Thus, if in the instance-based model we present at least two instances of a class but with different properties, this could make the model reader pay attention to the individual difference of instances of the same class.

Hypothesis 1: *Instance-based data models will show a significantly better performance than class-based data models in extracting information about properties of instance/entities.*

Users with instance-based representations create their own mental models of the domain and make sense of data through their own viewpoints, which results in a better understanding of the domain and supports more effective reasoning about the data due to the flexibility and freedom that this representation gives to them (Saghafi et al., 2022). In contrast, when the users read the class-based models, they need to understand a classification created by another designer, which can lead to anchoring to a specific view (Saghafi et al., 2022). Additionally, the cognitive load of interpreting the class-based data, which is created by someone else's mindset, is higher than the cognitive load of understanding the instance-based data with no pre-classification (Saghafi et al., 2022). Likewise, understanding an example of a data model which is presented in an instance-based approach is easier than figuring out an abstraction in a relational model.

Hypothesis 2: *Instance-based data models will show a significantly better performance than class-based models in extracting information about relationships between instances/entities.*

The concept of cardinality is meaningful at the class level in class-based models, and it becomes irrelevant at the instance level. Therefore, the cardinalities and optionality are artifacts of class-based models. As a result, in the instance-based model there is no direct way to represent a given minimal or maximal cardinality value of a relationship (Parsons and Wand, 2000). Additionally, there is no notation of optional relationship (minimum cardinality of 0) in this type of model. On the other hand, the relational data models are very expressive in terms of showing the cardinalities and they can explicitly represent minimum and maximum cardinalities.

Hypothesis 3: *Class-based data models will show a significantly better performance than instance-based data models in extracting information about cardinalities of relationships.*

Querying a dataset for information involves finding the related things and their properties in the data, connecting them with mutual properties (if necessary) and doing an operation on them. By storing data in a flexible format instead of structured tables, querying information becomes less challenging as it eliminates the need to join data from multiple tables. Thus, we can expect that the flexibility of the data model in organizing the data would have a positive effect on the effectiveness of information retrieval (Saghafi et al., 2022). To retrieve information about instances that share common properties (i.e., instances belong to the same class), the user of relational data is required to only look at a single entity and explore its attributes, while the user of instance-based data needs to look at the entire data as in instance-based approach we instances do not have labels to identify their class and there might be more than one instance of a the same potential class in the representation.

On the other hand, when retrieving information about instance that possess dissimilar properties, but have relationships with each other (i.e., instances that belong to different potential classes), the structured way of class-based models in defining entities and connecting them with foreign keys, adds high complexity to relational data models. In relational data, working with instances across multiple classes requires using join operations to link the data from multiple tables. For example, suppose we are trying to find the title of courses that a student with a specific name has taken. In a relational data model, the user needs to identify where these attributes can be found in the schema and how they are connected by keys (Saghafi et al., 2022). A typical procedure to extract this information would be: First, identify where the entities, students and courses can be found in the model. Second, find the mutual property connecting two entities by keys (student-course entity). Third, match the values in each entity with the required value for students' name and course's title to obtain the keys. Fourth, match the keys to find the required student and the required courses and obtain the result. In contrast, unlike the class-based models, users of the instance-based data are not required to pinpoint the classes related to these attributes and their reference keys (Saghafi et al., 2022). This information could be simply extracted from an instance-based model with this typical procedure: First, find the student instance with the given name. Second, find all the links with label ‘take course’ that connect the instance to other instances and obtain the title property of the identified instances.

Thus, we can see that extracting the same information that can easily be done in instance-based representation needs complicated join operations in a relational data representation. Moreover, the more the things involved in the query, the more complex joins are needed to extract the required information.

Hypothesis 4: *Class-based data models will show a significantly better performance than instance-based data models in retrieving information about instances from the same class.*

Hypothesis 5: *Instance-based data models will show a significantly better performance than class-based data models in retrieving information about instances from different classes.*

Chapter 4

Method

To examine the hypotheses, an experiment was designed that involved reading a business scenario and the related data model and performing two tasks. The first task focused on measuring participants' performance in understanding the business scenario using the data model. The second task assessed participants' performance on extracting some information from an Excel spreadsheet containing some sample data of the data model. A control-treatment design was used in this experiment. The participants were randomly assigned to one of two groups. One group received data represented using the instance-based data model, and the other group received data represented using the relational data model. Participants in both groups performed the same tasks.

4.1 Design

For each group, the experiment involved the following steps: First, participants watched a pre-recorded training video on how to read the related data model (relational or instance-based model). Second, participants received a description of a scenario/business domain and the related data model and completed 15 multiple-choice comprehension questions targeting their understanding of the scenario through the data model. The multiple-choice format is widely used in literature to assess subjects' comprehension of data models (Bodart et al., 2001; Gemino and Wand, 2005; Shanks et al., 2008, 2010). The questions are developed according to the logic and rationale of each hypothesis to make sure different aspects of each hypothesis are covered in the questions. In the process of formulating the questions, careful thought was given to how each group could possibly answer the questions. Thus, the questions are designed in a way that covers various aspects of each hypothesis and tests the effects of the distinct characteristics of each data model, enabling a meaningful performance comparison between the two groups. The details of the reasoning for developing each specific question are provided in section 4.3. In this task, the data model was available during answering the questions and participants also had access to training materials. Third, participants watched another pre-recorded training video on how to answer questions for the second task. Fourth, participants were asked to answer eight questions about how to extract some information from the spreadsheet which was populated with some sample data of the data model. This type of problem-solving question is also frequently employed in literature to evaluate subjects' proficiency in utilizing data models (Bodart et al., 2001; Gemino and Wand, 2005; Burton-Jones and Meso, 2006; Shanks et al., 2008, 2010; Parsons, 2011; Bera et al., 2014). In this task, the data models were taken away and participants only had access to the related spreadsheet and training materials for this task. Fifth, at the end of the study participants completed a short survey about their background in data modeling and working with Excel spreadsheets and their perception of ease of use and understanding of the data model. The overall design of experiment and the allocated time for each part is provided in Figure 4.1.

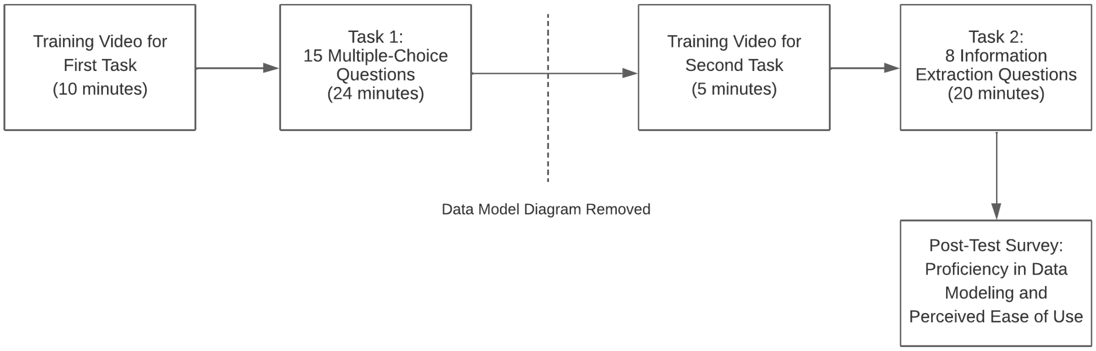


Figure 4.1: Experiment procedure

4.2 Experiment Materials

The experiment materials consisted of data models, comprehension questions (task 1), excel spreadsheets, information extraction questions (task 2), and post-test survey.

4.2.1 Task 1: Data Models

Task 1 comprised reading a data model and answering multiple-choice questions. The data model described the interactions of three entities, person, car, and company. In the data models, different kinds of interactions existed among people. There were some interactions between people and cars and between companies and cars. Companies had specific interactions between themselves. We also had mutual interactions between people and companies.

Relational Data Model Representation:

Figure 4.2 shows the relational representation of the scenario. One group of participants received this data model. In this model the boxes represent the entities/tables for which we want to collect data and the attributes (such as 'age' in the Person table) describe the properties of each entity. Some of the tables represent the relationship between the entities. For example, the 'WorkFor' table shows the people who work for each company by connecting the IDs.

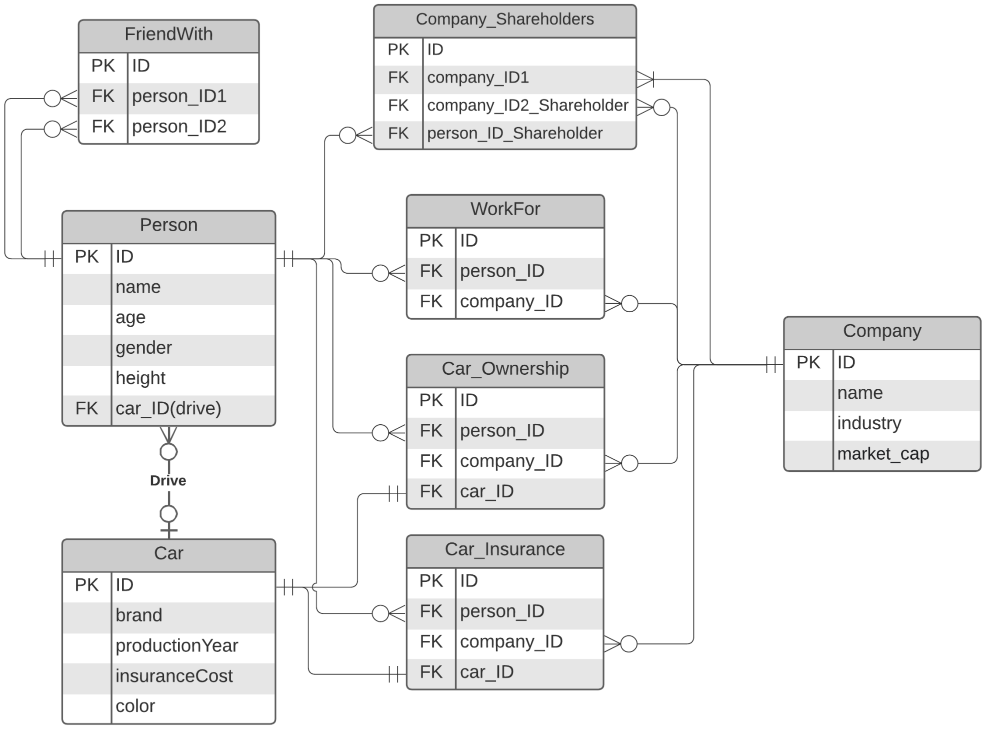


Figure 4.2: Relational data model used in the experiment

Instance-based Representation (Using a graph data model):

The instance-based representation of the data model is shown in Figure 4.3. In the model each circle shows an instance of the data. In the model circles (nodes) represent some instances of the model. The properties of each instance are shown within the circle and the edges connecting circles represent the relationships between the instances of the model. Although the model only represents some sample instances of the domain, the instances and their relationships have been carefully selected to capture various aspects of the domain. This model does not assign any labels to the instances to specify which class they belong to, indicating that this instance-based model does not have a class layer (unlike what is proposed in Parsons and Wand 2000). Labeling the instances does not create a separate class layer but creates a model that resembles a class-based model rather than an instance-based model. Thus, this model does not reflect a complete representation of the IBDM. However, this model captures the essence of IBDM by representing instances explicitly without the need to adhere to a predetermined classification.

4.2.2 Task 1: Multiple-Choice Comprehension Questions

Table 4.1 shows multiple-choice questions and their related hypothesis. As the table shows to operationalize each of hypotheses H1, H2, and H3, five questions were designed. For each question, participants were asked to specify their answer (true/false or yes/no) and the degree to which they are certain about their answer (i.e., level of confidence). It was emphasized that the scenario represented in the data model might be different from their previous understanding of automobile ownership and insurance and need to try to answer the questions based on the model, not what their prior understanding might be.

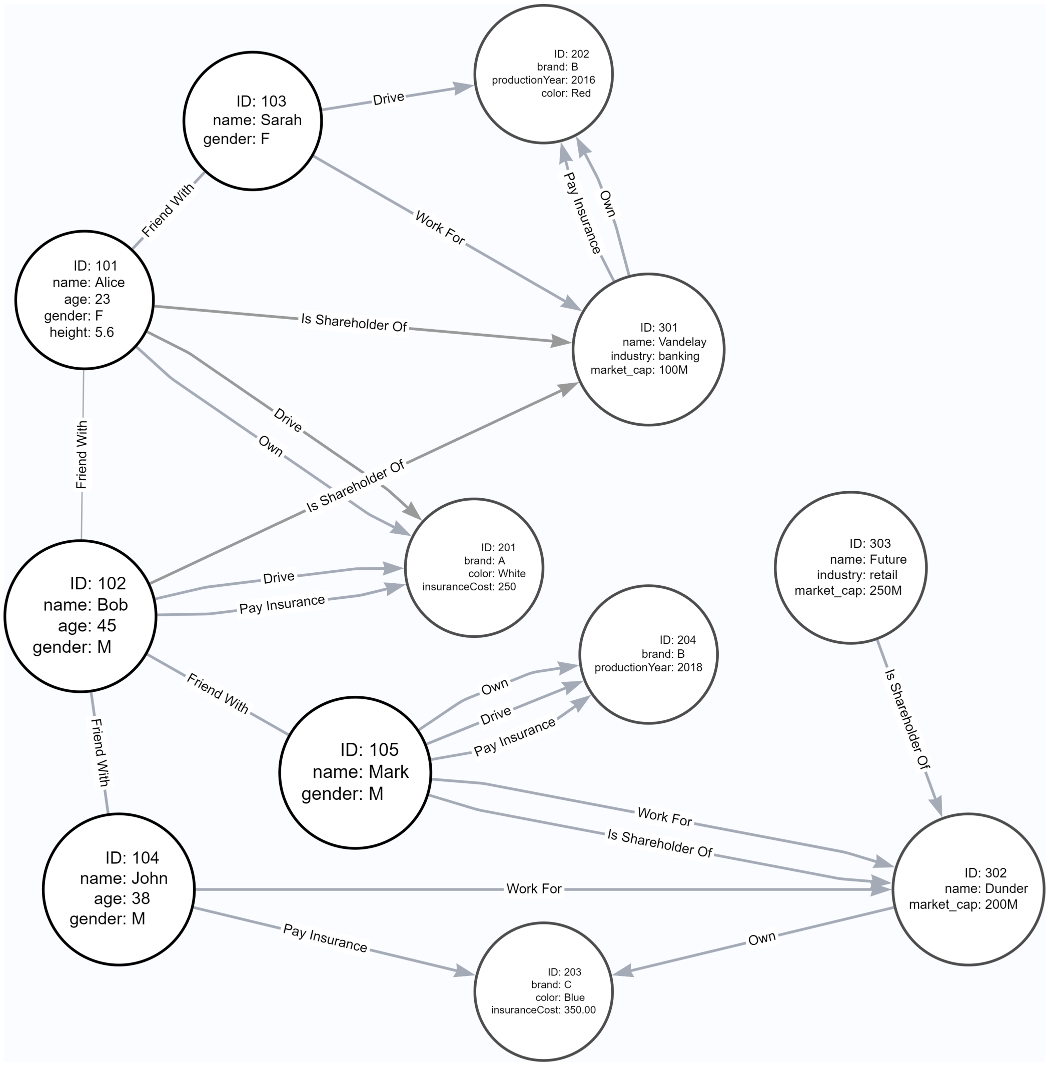


Figure 4.3: Instance-based data model used in the experiment

Table 4.1: Multiple-choice questions of task 1

Hypothesis	Question	Answer	Level of Confidence*				
			VL	L	N	H	VH
H1	1. Is it possible to represent a person in the model with the following attributes? name: 'Judi' height: 5.4	Yes/ No	□	□	□	□	□
	2. There can be a person whose name is unknown.	True/ False	□	□	□	□	□
	3. Is it possible to represent a company in the model with the following attributes? ID : 500, $market\_cap : 230M$	Yes/ No	□	□	□	□	□

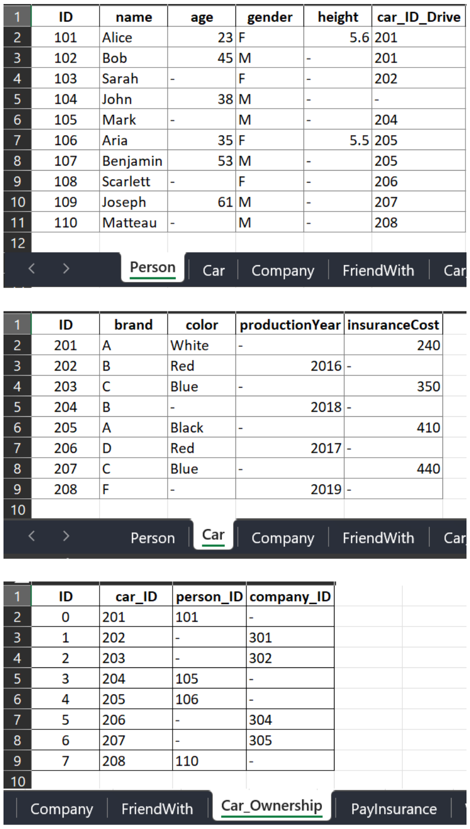
Hypothesis	Question	Answer	Level of Confidence				
			VL	L	N	H	VH
	4. For all instances of companies in the model, the industry in which they belong is known.	True/False	□	□	□	□	□
	V5. Insurance cost is fixed for a car no matter who insures it.	True/False	□	□	□	□	□
H3	6. A company can be a shareholder of many companies.	True/False	□	□	□	□	□
	7. A person can work for multiple companies.	True/False	□	□	□	□	□
	8. There could be a car that is driven by no one.	True/False	□	□	□	□	□
	9. A person can drive a maximum of one car.	True/False	□	□	□	□	□
	10. A person can drive more than one car.	True/False	□	□	□	□	□
H2	11. A person can drive a car that is not insured by herself/himself.	True/False	□	□	□	□	□
	12. A person who is not the owner can drive a car.	True/False	□	□	□	□	□
	13. A person can own a car and drive another car.	True/False	□	□	□	□	□
	14. If two people are not friends with each other, it is possible that they have a mutual friend	True/False	□	□	□	□	□
	15. A person can work for a company and be its shareholder at the same time.	True/False	□	□	□	□	□
* Level of Confidence: VL (Very Low), L (Low), N (Neutral), H (High), VH (Very High)							

4.2.3 Task 2: Spreadsheets

Task 2 involved reading an Excel spreadsheet containing some sample data based on the data model and answering some information extraction questions.

Relational Data:

In the relational spreadsheet, we had a spreadsheet with different sheets containing some sample data of our entities. Thus, data of each entity in our data model (Figure 4.2) was stored in a different sheet. Figure 4.4 Shows screenshots of sheets for 'Person', 'Car', and 'Car-Ownership' sheets.



ID	name	age	gender	height	car_ID_Drive
101	Alice	23	F	5.6	201
102	Bob	45	M	-	201
103	Sarah	-	F	-	202
104	John	38	M	-	-
105	Mark	-	M	-	204
106	Aria	35	F	5.5	205
107	Benjamin	53	M	-	205
108	Scarlett	-	F	-	206
109	Joseph	61	M	-	207
110	Matteau	-	M	-	208

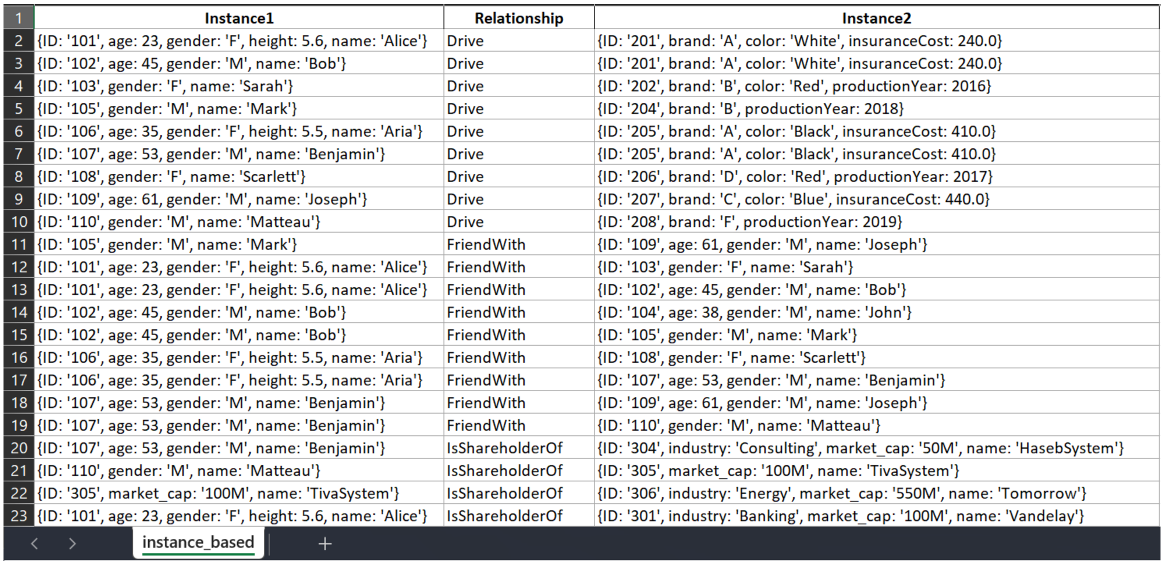
ID	brand	color	productionYear	insuranceCost
201	A	White	-	240
202	B	Red	2016	-
203	C	Blue	-	350
204	B	-	2018	-
205	A	Black	-	410
206	D	Red	2017	-
207	C	Blue	-	440
208	F	-	2019	-

ID	car_ID	person_ID	company_ID
0	201	101	-
1	202	-	301
2	203	-	302
3	204	105	-
4	205	106	-
5	206	-	304
6	207	-	305
7	208	110	-

Figure 4.4: Screenshot of some sheets of relational data

Instance-Based Data:

In the instance-based representation there are no classes and the instances do not have any labels to identify to which class they belong. Thus, the instance-based data consisted of a single sheet with three columns (Figure 4.5). Aligned with the instance-based representation (Figure 4.3), the first and third columns of the instance-based spreadsheet contains data of instances (i.e., circles or nodes in the representation) and their attributes. The second column shows the relationship between instances. Adopting this format for storing instance-based data can increase the complexity of the data; therefore, in the case of undirected relationship (i.e., bidirectional relationship) such as 'friend with' relationship, only one row/record was used to represent the related data to avoid representing unnecessary, duplicate data. In this case, representing 'Mark is friends with Joseph' is the same as representing 'Joseph is friends with Mark', because the properties of Mark and Joseph (i.e., ID, gender, age, name) are the same in this undirected relationship. Therefore, showing one record (i.e., one direction of this relationship) is sufficient to retain all relevant information of this relationship, and showing the reverse direction as a separate row/record in the data only creates data duplication. Thus, the reverse relationship was not shown in the data.



1	Instance1	Relationship	Instance2
2	{ID: '101', age: 23, gender: 'F', height: 5.6, name: 'Alice'}	Drive	{ID: '201', brand: 'A', color: 'White', insuranceCost: 240.0}
3	{ID: '102', age: 45, gender: 'M', name: 'Bob'}	Drive	{ID: '201', brand: 'A', color: 'White', insuranceCost: 240.0}
4	{ID: '103', gender: 'F', name: 'Sarah'}	Drive	{ID: '202', brand: 'B', color: 'Red', productionYear: 2016}
5	{ID: '105', gender: 'M', name: 'Mark'}	Drive	{ID: '204', brand: 'B', productionYear: 2018}
6	{ID: '106', age: 35, gender: 'F', height: 5.5, name: 'Aria'}	Drive	{ID: '205', brand: 'A', color: 'Black', insuranceCost: 410.0}
7	{ID: '107', age: 53, gender: 'M', name: 'Benjamin'}	Drive	{ID: '205', brand: 'A', color: 'Black', insuranceCost: 410.0}
8	{ID: '108', gender: 'F', name: 'Scarlett'}	Drive	{ID: '206', brand: 'D', color: 'Red', productionYear: 2017}
9	{ID: '109', age: 61, gender: 'M', name: 'Joseph'}	Drive	{ID: '207', brand: 'C', color: 'Blue', insuranceCost: 440.0}
10	{ID: '110', gender: 'M', name: 'Matteau'}	Drive	{ID: '208', brand: 'F', productionYear: 2019}
11	{ID: '105', gender: 'M', name: 'Mark'}	FriendWith	{ID: '109', age: 61, gender: 'M', name: 'Joseph'}
12	{ID: '101', age: 23, gender: 'F', height: 5.6, name: 'Alice'}	FriendWith	{ID: '103', gender: 'F', name: 'Sarah'}
13	{ID: '101', age: 23, gender: 'F', height: 5.6, name: 'Alice'}	FriendWith	{ID: '102', age: 45, gender: 'M', name: 'Bob'}
14	{ID: '102', age: 45, gender: 'M', name: 'Bob'}	FriendWith	{ID: '104', age: 38, gender: 'M', name: 'John'}
15	{ID: '102', age: 45, gender: 'M', name: 'Bob'}	FriendWith	{ID: '105', gender: 'M', name: 'Mark'}
16	{ID: '106', age: 35, gender: 'F', height: 5.5, name: 'Aria'}	FriendWith	{ID: '108', gender: 'F', name: 'Scarlett'}
17	{ID: '106', age: 35, gender: 'F', height: 5.5, name: 'Aria'}	FriendWith	{ID: '107', age: 53, gender: 'M', name: 'Benjamin'}
18	{ID: '107', age: 53, gender: 'M', name: 'Benjamin'}	FriendWith	{ID: '109', age: 61, gender: 'M', name: 'Joseph'}
19	{ID: '107', age: 53, gender: 'M', name: 'Benjamin'}	FriendWith	{ID: '110', gender: 'M', name: 'Matteau'}
20	{ID: '107', age: 53, gender: 'M', name: 'Benjamin'}	IsShareholderOf	{ID: '304', industry: 'Consulting', market_cap: '50M', name: 'HasebSystem'}
21	{ID: '110', gender: 'M', name: 'Matteau'}	IsShareholderOf	{ID: '305', market_cap: '100M', name: 'TivaSystem'}
22	{ID: '305', market_cap: '100M', name: 'TivaSystem'}	IsShareholderOf	{ID: '306', industry: 'Energy', market_cap: '550M', name: 'Tomorrow'}
23	{ID: '101', age: 23, gender: 'F', height: 5.6, name: 'Alice'}	IsShareholderOf	{ID: '301', industry: 'Banking', market_cap: '100M', name: 'Vandelay'}

Figure 4.5: Screenshot of instance-based data

4.2.4 Task 2: Information Extraction Questions

Table 4.2 shows the information extraction questions for the second task and the related hypotheses. Three questions were defined to operationalize H4, and five questions were formulated to address H5. For each question, the participants were asked to provide a verbal description of the procedure needed to extract the information from the spreadsheet. They were supposed to specify the related excel sheets, columns, operations, and steps required to pull out the information.

Table 4.2: Information extraction questions of task 2

Hypothesis	Question
H4	1. Find the average insurance cost.
	2. Find the highest market capitalization $(market_cap)$.
	3. Find the number of females.
H5	4. Find the color of the car that Sarah drives.
	5. Find the name of the company that John works for.
	6. Find the name and age of all people who are friend with Bob.
	7. Find the name of all shareholders of Vandelay company.
	8. John and Mark are not friend with each other. Find a mutual of friend of them that could introduce them to each other.

4.2.5 Post-Test Survey (Part 1): Participants Background in Data Modeling

In the survey some general information about the participants' backgrounds in data modeling and working with Excel spreadsheet was collected (Table 4.3).

Table 4.3: Post-test survey questions: participants background

Questions
1. Have you received any formal/academic or informal training in data modeling with entity relationship diagrams or class diagrams? ☐ Yes ☐ No

Questions
2. Please indicate your level of knowledge/proficiency in data modeling with entity relationship diagrams or class diagrams. ☐ No Knowledge ☐ Beginner ☐ Intermediate ☐ Advanced
3. Have you received any formal/academic or informal training in instance-based data modeling, or graph data modeling with directed graphs? ☐ Yes ☐ No
4. Please indicate your level of knowledge/proficiency in instance-based data modeling, or graph data modeling with directed graphs. ☐ No Knowledge ☐ Beginner ☐ Intermediate ☐ Advanced
5. Please indicate your level of knowledge/proficiency in working with Excel spreadsheets for creating reports, extracting information, or building charts. ☐ No Knowledge ☐ Beginner ☐ Intermediate ☐ Advanced

4.2.6 Post-Test Survey (Part 2): Perceived Ease of Understanding and Interpreting

This survey targeted the participants' perceived ease of understanding and interpreting when answering the questions and doing the tasks. Table 4.4 shows the questions utilized to assess perceived ease of use and understanding (Gemino and Wand 2005).

Table 4.4: Post-test survey questions: perceived ease of use and understanding.

Question	Disagree Strongly	Disagree	Neither Disagree nor Agree	Agree	Agree Strongly
1. I believe that it was easy for me to understand the data model that we were trying to model	☐	☐	☐	☐	☐
2. Overall, I believe the data model was easy to use	☐	☐	☐	☐	☐
3. Learning how to read the data model was easy for me	☐	☐	☐	☐	☐
4. Using the data model was often frustrating	☐	☐	☐	☐	☐

4.3 Reasoning Behind Developing Questions

This section provides an insight into the rationale behind the questions related to each hypothesis. To illustrate this, we present a question from each hypothesis, along with guidance on how participants in each group can deduce the correct answer. Appendix A presents a detailed explanation for all the questions.

H1: Extracting information about properties of instances/entities:

The questions associated with this hypothesis aim to assess participants comprehension of whether instances/entities can possess a set of attributes or not. The goal is to evaluate their performance when they are confronted with instances that exhibit properties that do not completely match the data model. Thus, in the questions the properties do not exactly match the properties in the data model. This can be a measure of how participants can deal with new observations in the domain that do not fully comply with their prior understanding of the domain.

For example, in question 1 of task 1 (i.e., question 1 in Table 4.1), the instance appears to be a new instance that does not fit precisely into either data model. In the class-based model, the entity 'person' has six properties. In the instance-based model the instances that could belong to 'person' class possess three to five properties. The participant is expected to comprehend if this new instance can be an instance of the model. This can be measure of whether the model embraces the uniqueness of instances and how it can deal with new observations in the domain that do not fully comply with the prior understanding of the domain.

In the instance-based model, we have instances with three properties (ID, name, gender), four properties (ID, name, age, gender), and five properties (ID, name, age, gender, height), that could belong to 'person' class. This different combination of properties means that it is highly likely the instance in the question be an instance of the model. On the other hand, in the class-based model the 'person' entity has six properties (ID, name, age, gender, height, car Id). This model does not contain any notation to specify if an instance of this model may not possess any of these attributes (i.e., the model does not specify if the attributes are optional or not). Additionally, unlike the instance-based model, the class-based model only shows the name of properties and does not assign any sample value to them. As a result, it is less likely the users of the class-based model could answer this question correctly.

H2: Extracting information about relationships among instances/entities:

The questions related to this hypothesis aim to assess participants' ability to extract information concerning two or more relationships in the model. Answering these questions requires combining information from multiple relationships and drawing inferences based on it. Here, simply knowing the cardinalities of the relationships will not necessarily lead to the correct answer. Instead, the focus is on the chain of relationships between instances.

For example, in question 15 of task 1 (i.e., question 15 in Table 4.1), focuses on two relationships in the model 'work for' and 'shareholder' relationships. It targets the chain of relationships and answering it requires following this chain of relationships. Essentially, this question requires an understanding of how 'work for' and 'shareholder' relationships are connected and how they impact each other. In the instance-based model, the participants can visually see how these two relationships are connected. Thus, following this chain of relationship is straightforward in this model. On the other hand, the class-based model uses two separate intermediate entities to show ‘work for’ and ‘shareholder’ relationships. Thus, creating the chain of relationships and making inferences based on that is challenging in this model.

H3: Extracting information about cardinalities of relationships:

Finding the correct answer to the questions of this hypothesis demands understanding the cardinalities of the relationships. This requires identifying the minimum and maximum cardinality for each relationship. Understanding these cardinalities is enough for answering the questions.

For example, in question 8 of task 1 (i.e., question 8 in Table 4.1), the goal is to evaluate whether the participants can understand that the ‘drive’ relationship is a zero-to-many relationship. The class-based model shows that on the ’car’ side, the cardinality is constrained to a minimum of 0 and a maximum of 1. Additionally, on the ‘person’ side the minimum and maximum cardinality are 0 and many respectively. Thus, with this model, participants can conclude that there could be a car that is driven by no one, a person can only drive a minimum of one car, and a car might be driven by zero or many people. However, a minimum cardinality of zero could not be represented in the instance-based model. Furthermore, the existence of a relationship cannot imply the maximum cardinality. Even if the model contains an instance that has more than one ‘drive’ relationship, it only implies ’many’ relationship and does not necessarily specify the maximum cardinality of such relationship. As a result, answering this question with the instance-based model is very challenging.

H4: Retrieving information about instances that share common properties:

The questions of this hypothesis aim to assess participants performance in retrieving/querying information about instances with common properties (information about instance belonging to the same class). The key requirement here is that the required information is only about a single entity/class. For example, question 1 of task 2 (i.e., question 1 in Table 4.2), focuses on instances with the common property of 'insurance cost'. To retrieve this information, the participants using the class-based model simply need to go the related sheet in the data (in this example 'car' sheet) and look at the 'insurance cost' column to find the related data. However, the participants using the instance-based model need to explore the whole dataset and come up with a mechanism to filter the data and find all distinct instances with 'insurance cost' property. Thus, it is expected that answering this question be more challenging for the instance-based group compared to the class-based group.

H5: Retrieving information about instances that possess dissimilar properties but have relationships with each other:

The questions of this hypothesis attempt to measure participants performance when retrieving/querying information involving instances with dissimilar properties that are related to each (information about instance from different classes/entities). The goal here is to assess their performance when retrieving the required information demands joining/combining data of two instances/entities. In other words, retrieving the required information entails exploring the relationships between instances. For example, in question 5 of task 2 (i.e., question 5 in Table 4.2), the required information is related to John (first instance) and the company (second instance) that he works for (the relationship between two instances). In the instance-based data all instances are represented in a single view and their relationships are explicitly specified. Thus, the participants using this model need to filter the 'work for' relationship and then look for the required information. However, retrieving the same information using the class-based data requires joining three tables: person, company, work for. From the 'person' table they need to extract John's ID, then from 'work for' table extract the ID of the company related to John's ID, and then in the 'company' table extract the name of the required company using its ID. Thus, the procedure in the class-based model is more complicated. As a result, it is expected that answering this question be more challenging for the class-based group compared to the instance-based group.

4.4 Dependent Variables

Table 4.6 shows the dependent variables of this study and the related measures for each variable. Performance of participants in tasks 1 and 2 is the main dependent variable, which reflects the hypothesis of the study. To compare the performance between instance-based group and class-based group, a one-way independent T-test was used. In the first task, along with answering each question, the participants specified their level of confidence in the answers as well. Thus, level of confidence in answers is an additional dependent variable of the study.

Time taken to complete each task is another dependent variable of the study, which can be a measure of difficulty or complexity of the tasks (Gemino and Wand, 2005). Time assigned to first and second task was 24 minutes and 20 minutes respectively. Participants were aware of the allocated time and were informed that they can complete the tasks sooner and go to the next section. Time taken to do each task was collected automatically by the online testing platform. Perceived ease of use and understanding is also a dependent variable of the study, which measured using four post-test survey questions. The questions addressed perceived ease of use, ease of understanding and ease of learning data models on a 5-point scale.

Table 4.6: Dependent variables and their related measures

Category	Dependent Variable	Measure	Hypothesis	Test
Main Dependent Variable: Performance	Performance in Task 1, Questions 1-5	Number of correct answers in questions 1-5 of task 1 (scale: 0-5)[a]	H1	One-way Independent T-test
	Performance in Task 1, Questions 11-15	Number of correct answers in questions 11-15 of task 1 (scale: 0-5)[a]	H2	
	Performance in Task 1, Questions 6-10	Number of correct answers in questions 6-10 of task 1 (scale: 0-5)[a]	H3	
	Performance in Task 2, Questions 1-3	Average score in questions 1-3 of task 2 (scale: 0-5)[b]	H4	
	Performance in Task 2, Questions 4-8	Average score in questions 4-8 of task 2 (scale: 0-5)[b]	H5	
Confidence Level	Level of Confidence in Questions 1-5 of Task 1	Average level of confidence in questions 1-5 of task 1 (scale: 1-5)[c]	-	Two-way Independent T-test
	Level of Confidence in Questions 6-10 of Task 1	Average level of confidence in questions 6-10 of task 1 (scale: 1-5)[c]	-	
	Level of Confidence in Questions 11-15 of Task 1	Average level of confidence in questions 11-15 of task 1 (scale: 1-5)[c]	-	
	Overall Level of Confidence	Average level of confidence in questions 1-15 of task 1 (scale: 1-5)[c]	-	
Time	Time to Complete Tasks	Total time taken to complete tasks 1 and 2	-	Two-way Independent T-test
Perceived Ease of Use	Perceived Ease of Use and Understanding	Answers to post-survey questions (scale: 1-5) [d]	-	

[a] scale (0: zero correct answer, 1-4: based on the number of correct answers, 5: five correct answers)

[b] scale (0: no answer or unacceptable answer, 1:4: based on number mistakes, 5: a complete answer)

[c] scale (1: very low, 2: low, 3: neutral, 4: high, 5: very high)

[d] scale (1: disagree strongly, 2: disagree, 3: neither disagree nor agree, 4: agree, 5: agree strongly)

4.5 Scoring and Encoding Answers

In the first task, the scoring was based on the number of correct answers. A 5-point scale was used to encode the level of confidence: 1 for 'very low', 2 for 'low', 3 for 'neutral', 4 for 'high', and 5 for 'very high'. To encode the Perceived Ease of Use and Understanding measure, we utilized a 5-point scale with 'disagree strongly' coded as 1, 'disagree' as 2, 'neither disagree nor agree' as 3, 'agree' as 4, and 'agree strongly' as 5. For scoring the answers to the second task the following metrics were applied:

- A complete answer received a score of 5. An answer was considered complete if it described the procedure using correct excel sheets, columns, operations, and steps to extract the required information.
- No answer or illogical answers with major mistakes received 0.
- For any other satisfactory, logical, but incomplete answers which contained some minor mistakes, one point was deducted for each mistake.

Appendix B provides details of how we scored the answers to the second task. For example, the following answers is for the first question for instance-based group. This answer has described the logical procedure to extract the required information. However, this answer has two mistakes: 1) it may not include all cars, 2) it may not include distinct cars. Thus, this answer got a score of 3.

> “Find all of the cars from the from relationship payinsurance. select the ones that have the insurance variable. use that to calculate the avg.”

4.6 Participants

The participants were bachelor and graduate students at Memorial University of Newfoundland, mainly from Faculty of Business Administration and Computer Science Department. Most of the participations were voluntary, however the bachelor students of Faculty of Business Administration received a course credit of one percent as compensation for their participation. Altogether 38 students participated in this study and were randomly assigned to one of the two groups. Participants had different levels of data modeling experience and could mirror both business users of data modeling and technical practitioners of data modeling. Table 4.7 summarizes the number of participants with prior experience in relational data modeling and instance-based data modeling in each group. A two-way independent T-test demonstrated that there was no significant different between class-based group and instance-based group in the number of participants with prior training in relational data modeling ($t = 0.66$ and $p - value = 0.515$) and graph data modeling ($t = 0.86$, $p - value = 0.397$).

Table 4.7: Participants prior experience in data modeling and data extraction

	Previous training in relational data modeling: Yes (%)	Previous training in graph data modeling: Yes (%)
Class-based Group	5 (29%)	3 (18%)
Instance-based Group	8 (40%)	6 (30%)

Figure 4.6 compares participant's proficiency in relational data modeling, graph data modeling, and working with spreadsheets between class-based group and instance-based group. Using two-way independent T-test showed that level of proficiency in relational data modeling ($t = -0.8$, $p{-}value = 0.428$), graph data modeling ($t = 1.12$, $p{-}value = 0.268$), and working with Excel spreadsheets ($t = 0.96$, $p{-}value = 0.343$) was not significant between the two groups (level of training was encoded to values

from 0 for ‘No Knowledge’ to 3 for ‘Advanced’).

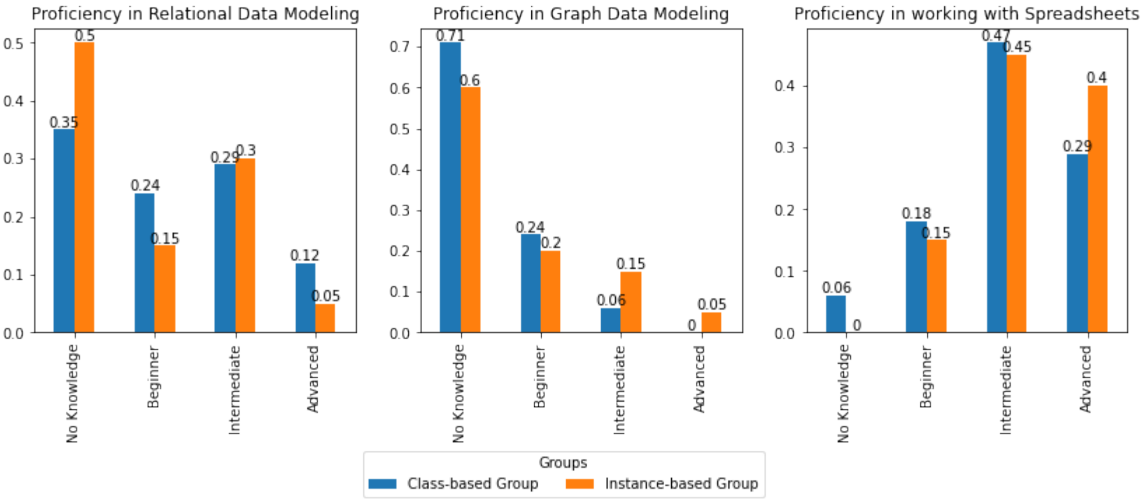


Figure 4.6: Proficiency of participants in data modeling and information extraction

4.7 Addressing Missing Data

Participants left some questions unanswered in both task 1 and task 2, resulting in some missing data. To handle missing values, we initially divided the questions of task 1 and task 2 into separate datasets. For each dataset, in the first step the records where all the columns/questions were missing were removed from datasets. In the remaining data there were only few missing responses. After this process, we had 20 acceptable responses in the instance-based group and 18 acceptable responses in the class-based group for the first task. For the second task, both groups had 14 acceptable responses. Analyzing the participants’ responses revealed that the number of missing data in the second task was higher than that in the first task. Additionally, some participants did not attempt the second task and left all questions unanswered, which were removed from the data according to the first step. After deleting unacceptable responses, to address the remaining missing data in the two datasets, we used the average scores of the relevant questions to impute the values.

Chapter 5

Results and Discussion

5.1 Results

Table 5.1 contains the means, standard deviations, number of acceptable participations and test results for each dependent variable. The results showed that four of our hypotheses were supported and one hypothesis was not supported. Additionally, the findings indicated there was a significant difference in completion time and perceived ease of use and understanding between instance-based and class-based groups. However, with regards to the overall level of confidence, there was no statistically significant difference observed between the two groups.

Extracting information about properties of instances/entities (H1):

For this dependent variable, which was used to operationalize H1, the results of a one-way independent T-test revealed no significant difference in performance between the instance-based group ($M = 2.45$, $SD = 1.23$) and the class-based group ($M = 2.39$, $SD = 0.92$) with $t = 0.17$ and $p > 0.05$. Thus, Hypothesis 1 was not supported, meaning that with respect to identifying the properties of instance/entities there were no significant differences between the two groups.

Table 5.1: Test results for each dependent variable

Dependent Variable	**Instance-based Group**	**Class-based Group**	**T-test**	**P-value**	**Conclusion**
	Mean (SD) n	Mean (SD) n			
Performance in Task 1, Questions 1-5 (H1)	2.45 (1.23) n = 20	2.39 (0.92) n = 18	0.17	0.432	Not Significant (Not Supported)
Performance in Task 1, Questions 11-15 (H2)	3.65 (1.23) n = 20	2.73 (1.28) n = 18	2.27	0.015	Significant (Supported)
Performance in Task 1, Questions 6-10 (H3)	2.6 (0.88) n = 20	3.6 (1.33) n = 18	-2.75	0.005	Significant (Supported)
Performance in Task 2, Questions 1-3 (H4)	4.33 (0.23) n = 14	4.93 (0.19) n = 14	-7.47	0.000	Significant (Supported)
Performance in Task 2, Questions 4-8 (H5)	4.32 (0.54) n = 14	3.83 (0.50) n = 14	2.49	0.010	Significant (Supported)
Level of Confidence in Questions 1-5 of Task 1	3.67 (0.79) n = 20	3.29 (0.52) n = 18	1.73	0.092	Not Significant
Level of Confidence in Questions 11-15 of Task 1	3.75 (0.67) n = 20	2.98 (0.62) n = 18	3.66	0.001	Significant
Level of Confidence in Questions 6-10 of Task 1	3.2 (0.84) n = 20	3.47 (0.53) n = 18	-1.16	0.254	Not Significant
Overall Level of Confidence	3.54 (0.64) n = 20	3.25 (0.35) n = 18	1.73	0.093	Not Significant
Time to Complete Tasks	1557.34s (478.02s) n = 15	1962.69s (445.74s) n = 16	-2.44	0.021	Significant
Perceived Ease of Use and Understanding	3.49 (0.56) n = 20	2.81 (0.63) n = 17	3.48	0.001	Significant

Extracting information about relationships among instances/entities (H2):

A one-way independent T-test revealed that participants in the instance-based group showed significantly higher performance ($M = 3.65$, $SD = 1.23$) than those in the class-based group ($M = 2.73$, $SD = 1.28$), $t = 2.27$, $p < 0.05$. Hypothesis 2 was therefore supported. The finding provided evidence that instance-based models outperform class-based models in extracting information and making inferences about relationships among instances/entities.

Extracting information about cardinalities of relationships (H3):

A one-way independent T-test showed that participants in the class-based group demonstrated significantly better performance ($M = 3.60$, $SD = 1.33$) than those in the instance-based group ($M = 2.60$, $SD = 0.88$), $t = -2.75$, $p < 0.05$. Therefore, Hypothesis 3 was supported, indicating that relational data models offer greater expressiveness in representing cardinalities of relationships.

Retrieving information about instances that share common properties (H4):

Using a one-way independent T-test indicated that participants in the class-based group performed significantly better ($M = 4.93$, $SD = 0.19$) than those in the instance-based group ($M = 4.33$, $SD = 0.23$), $t = -7.47$, $p < 0.05$. Thus, Hypothesis 4 was supported. The results provide evidence that class-based data are more effective than instance-based data for retrieving (querying) information about instances that belong to one class (i.e., information about a single entity).

Retrieving information about instances that possess dissimilar properties but have relationships with each other (H5):

A one-way independent T-test shows that there is a significant difference in performance scores between the instance-based group ($M = 4.32$, $SD = 0.54$) and the class-based group ($M = 3.83$, $SD = 0.50$), $t = 2.49$, $p < 0.05$. This result supports Hypothesis 5 and suggests that, for retrieving information that involved instances from more than one class (i.e., information that involved two or more entities/tables), the instance-based data is more effective than class-based data.

Level of Confidence:

Using a two-way independent T-test demonstrated that level of confidence in questions 1 to 5 of task 1, was not significantly different between the instance-based group ($M = 3.67$, $SD = 0.79$) and the class-based group ($M = 3.29$, $SD = 0.52$), $t = 1.73$, $p > 0.05$. Similarly, the difference was not significant in questions 6 to 10 of task 1 between the instance-based group ($M = 3.2$, $SD = 0.84$) and the class-based group ($M = 3.47$, $SD = 0.53$) using two-way independent T-test ($t = -1.16$, $p > 0.05$). However, the difference was significant in questions 11 to 15 of task 1 between the instance-based group ($M = 3.75$, $SD = 0.67$) and the class-based group ($M = 2.98$, $SD = 0.62$), $t = 3.66$, $p < 0.05$. With respect to the overall level of confidence (in all questions of first task), the outcome did not show a significant difference between the instance-based group ($M = 3.54$, $SD = 0.64$) and the class-based group ($M = 3.25$, $SD = 0.35$) using two-way independent T-test ($t = 1.73$, $p > 0.05$).

Time to Complete Tasks:

To examine how the two groups differed in terms of time taken to complete tasks, a two-way independent T-test was performed, and the result indicated that the differences in the means were significant ($t = -2.44$, $p < 0.05$) between the instance-based group ($M = 1557.34s$, $SD = 478.02s$) and the class-based group ($M = 1962.69s$, $SD = 445.74s$). Thus, participants in the instance-based group spent less time completing the two tasks.

Perceived Ease of Use and Understanding:

A two-way independent T-test was conducted to compare the perceived ease of use and understanding between the two groups. The test revealed that perceived ease of use and understanding were significantly higher in the instance-based group ($M = 3.49$, $SD = 0.56$) than in the class-based group ($M = 2.81$, $SD = 0.63$), $t = 3.48$, $p < 0.05$. Therefore, the instance-based model was easier to learn, understand and use.

After conducting the above T-tests to compare the performance of the two groups, a subsequent Mann-Whitney U test was performed to verify the robustness of the findings. The Mann-Whitney U test was chosen due to concerns about the normality of the data, as the T-test assumes a normal distribution. It is noteworthy that the Mann-Whitney U test is a non-parametric alternative, making fewer distributional assumptions and focusing on rank comparisons. The results of the Mann-Whitney U test aligned with those of the T-test, indicating consistency in the observed differences between the groups. This convergence reinforces the confidence in the initial findings, suggesting that the conclusions drawn from the T-test were not overly influenced by the assumption of normality, and the observed differences in performance between the groups are likely to hold true regardless of the data distribution.

5.2 ANCOVA Analysis: Exploring the Effects of Covariates

Although the results showed that there is no significant difference between the two groups regarding their training and proficiency levels in the three areas of relational data modeling, graph data modelling and working with spreadsheets, it would be insightful to examine if this covariate (i.e., level of proficiency) moderated the primary effect. To do so, the average level of proficiency in these three topics was computed, and then an ANCOVA analysis was performed to evaluate the effects of using different data models on the dependent variables while controlling the influences this covariate. Table 5.2 shows the results of ANCOVA analysis for each dependent variable. The result of ANCOVA analysis is same as the result of T-test in Table 5.1. The findings demonstrate that prior training and proficiency did not influence the impacts of utilizing different data models. In other words, the significant difference between two groups in H2, H3, H4, H5, Task completion time, and perceived ease of use and understanding can not be explained by the differences in their level of training.

Table 5.2: ANCOVA Analysis

Dependent Variable	**F**	**P-Value**	**Conclusion**	**Covariate Moderated the Primary Effect?**
Performance in Task 1, Questions 1-5 (H1)	0.255	0.617	Not Significant (H1 Not Supported)	No (Result is same as one-way independent T-test)
Performance in Task 1, Questions 11-15 (H2)	5.758	0.022	Significant (H2 Supported)	No (Result is same as one-way independent T-test)
Performance in Task 1, Questions 6-10 (H3)	13.586	0.001	Significant (H3 Supported)	No (Result is same as one-way independent T-test)
Performance in Task 2, Questions 1-3 (H4)	50.494	0.000	Significant (H4 Supported)	No (Result is same as one-way independent T-test)

Continued on next page

Table 5.2 – continued from previous page

Dependent Variable	F	P-Value	Conclusion	Covariate Moderated the Primary Effect?
Performance in Task 2, Questions 4-8 (H5)	7.8757	0.0096	Significant (H5 Supported)	No (Result is same as one-way independent T-test)
Level of Confidence in Questions 1-5 of Task 1	2.528	0.121	Not Significant	No (Result is same as two-way independent T-test)
Level of Confidence in Questions 11-15 of Task 1	14.313	0.001	Significant	No (Result is same as two-way independent T-test)
Level of Confidence in Questions 6-10 of Task 1	3.559	0.068	Not Significant	No (Result is same as two-way independent T-test)
Overall Level of Confidence	2.833	0.102	Not Significant	No (Result is same as two-way independent T-test)
Time to Complete Tasks	13.703	0.001	Significant	No (Result is same as two-way independent T-test)
Perceived Ease of Use and Understanding	11.751	0.002	Significant	No (Result is same as two-way independent T-test)

5.3 Discussion

Although instance-based representation has a strong theoretical foundation (Parsons and Wand, 2000; Lukyanenko et al., 2019), empirical research examining the usefulness of instance-based data representation has been scarce (Saghafi et al., 2022). The purpose of this study was to evaluate and compare the effectiveness of instance-based data models versus class-based data models in different dimensions of information extraction, including extracting information regarding the properties of things, extracting information concerning the cardinalities of relationships between instances/entities, extracting information related to relationships between instance/entities, retrieving information about instances that share common properties (i.e., instances belong to the same class), and retrieving information about instances that possess dissimilar properties (i.e., instances that belong to different classes). The study was conducted using a treatment-control experimental design and involved 38 participants who were randomly assigned to either the instance-based group or the class-based group.

The first hypothesis focused on differences in extracting information about the properties of things, which was not supported by the data. For this hypothesis, both one-way and two-way independent T-tests showed that there is no significant difference between the two representations, suggesting that they are equally useful. The rationale behind this hypothesis was that, since in instance-based representation there might be instances with varying properties (even from the same class), the model consumers would gain a better understanding of how individual instances differ from one another. As a result, users of instance-based representation would perform better in identifying properties of things. However, upon further consideration, we recognized that some of the questions (such as question 2 and question 4) which we had formulated to operationalize this hypothesis were somewhat ambiguous and could not fully capture our intention to assess participants' understanding of the properties of instances. Another possible explanation for this outcome could be attributed to the use of small models in the experiment.

The second hypothesis, which the data supported, investigated drawing inferences about the relationships between entities/instances. Instance-based models do not impose any pre-defined classifications on users, allowing for greater flexibility and customization in understanding the domain of concern. They represent relationships between instances by connecting them with proper labels and may utilize multiple edges to represent different types of relationships that may exist between instances, making it easier to derive meaningful insights from the data. On the other hand, relational data models rely on foreign keys to show the relationships and do not use any labels for relationships. Additionally, in cases where many-to-many relationships exist, additional entities may be required to represent the connections (i.e., changing a many-to-many relationship to two one-to-many relationships), which can make it more challenging to understand the relationships. The findings confirmed this rationale and demonstrated that relational data models are superior to instance-based models in extracting information about relationship cardinalities.

The third hypothesis, which was supported by the data, concentrated on cardinalities of relationships. Class-based models (i.e., relational data models) offer high expressivity in showing minimum and maximum cardinalities (even when minimum cardinality is zero) and can effectively represent one-to-one, zero-to-one, zero-to-many, one-to-many, and many-to-many relationships. Furthermore, cardinalities including optional cardinality are meaningful at class level. At the instance level, however, there are only relationships connecting instances. Besides, in instance-based representation, it's only possible to display a few sample instances from the domain, making it impractical to represent all possible relationships. Thus, as expected, the class-based group outperformed the instance-based group in this aspect, as demonstrated by the higher performance results.

The fourth and fifth hypotheses, both supported, evaluated the effectiveness of retrieving (querying) information from class-based and instance-based data. In class-based data, we utilized separate Excel sheets for each entity (e.g., person, car, company, etc.). In contrast, instance-based data was organized using a single Excel sheet that contains three columns: instance1, relationship, and instance2 (Figure 4.4 and Figure 4.5). To answer the questions related to hypothesis 4, the class-based group simply had to examine one Excel sheet. For example, to find the average insurance cost they only needed to look at the Car sheet and InsuranceCost column to find all distinct values for calculating the average. However, the instance-based group had to utilize the entire dataset and develop a method to extract the necessary information. Thus, as anticipated, the class-based group outperformed the instance-based group in answering questions that required retrieving information about instances belonging to a single class (i.e., information about a single entity). On the other hand, to address the questions of hypothesis 5, the class-based group had to join data from two or more sheets (tables) to obtain the necessary information. In contrast, the instance-based group simply needed to search for the relationship name in the relevant column and focus on the selected instances to obtain the same information. Therefore, aligned with our expectations, the instance-based data was more effective in retrieving information involving instances from more than one class (i.e., information that involved two or more entities/tables). This holds particular importance since, in most real-world use cases, the users usually seek to extract information that need combining data of two or more entities, in which the instance-based representation proved to be more effective.

In the remaining three dependent variables: level of confidence, time taken to complete tasks and perceived ease of use and understanding, the results were also noteworthy. The overall level of confidence was not significantly different between the two groups. This might be explained by the fact that most participants had not received prior training in either relational data modeling or instance-based data modeling. Time taken to complete the tasks was significantly shorter for the instance-based group than for the class-based group, suggesting that the tasks were less difficult or complex for the instance-based group. Furthermore, the perceived ease of use and understanding was greater for the instance-based group, demonstrating that the participants of this group found it easier to learn, understand and use the instance-based data model.

In summary, the findings provided strong evidence that the most effective data model for information extraction depends on the type of information being extracted. As expected, the participants in the instance-based group demonstrated significantly better performance in extracting information about the relationship between instances/ entities and retrieving information that involved instances from more than one class. The class-based group exhibited significantly higher performance in extracting information related to cardinalities of relationships and retrieving information pertaining to only one entity, which was according to our proposition. However, there was no statistically significant difference between the two groups in their performance at extracting information about properties of instances/ entities. Additionally, with respect to completion time and perceived ease of use and understanding, the instance-based group showed significant superiority (lower completion time and higher perceived ease of use and understanding) over the class-based group. However, in terms of overall level of confidence, there was no difference between the two groups.

5.4 Contributions and Implications

While the class-based data model proved to be more effective for two types of information extraction (i.e., extracting information about cardinalities of relationships and retrieving information involving only one entity), it is worth noting that these two dimensions represent relatively simple and straightforward use cases which may not be the primarily focus of most data analysis and information extraction attempts. On the other hand, for more complex use cases including extracting information about relationships of instance/entities and retrieving information involving instances from different classes, the instance-based model demonstrated significantly better performance. Furthermore, it is important to acknowledge that the instance-based model remains effective even for the two simple use cases (i.e., Hypothesis 3 and Hypothesis 4). Specifically, it achieved average scores of 2.6 and 4.33 out of 5 in Hypothesis 3 and Hypothesis 4, respectively. However, its effectiveness is not on par with the class-based model, mainly due to the highly expressive representation of the class-based model in the corresponding dimensions. Thus, through an exploration of various facets of information extraction use cases, this study establishes the superiority and effectiveness of instance-based data models. These findings align with previous research that highlighted the usefulness and functionality of instance-based representations (Lukyanenko et al., 2019; Saghafi et al., 2022). However, this study explored different aspects of information extraction, and identified the use cases in which the instance-based representation demonstrated its highest capabilities.

This study has some important implications for both researchers and practitioners. For researchers, the study highlights the importance of empirically examining usefulness of instance-based representations, as it has shown a remarkable ability to extract information. This study also emphasizes the capabilities of graph data modeling in implementing instance-based representation, which merits greater academic attention. The results indicate that the graph data model proves to be a highly effective approach for implementing both instance-based representation (Figure 4.3) and instance-based data (Figure 4.5). This finding underscores the need for further research into the practical applications of instance-based representation, implemented with graph data models, in the fields of data analysis and information extraction.

For practitioners, our findings suggest that combining relational data models and instance-based models could complement each other, resulting in a more accurate representation of the domain and a greater understanding of it. However, in most real-world use cases, it is needed to deal with joining data of multiple entities/instances and exploring the chain of relationships between them, in which case employing an instance-based representation is more effective than class-based models. The instance-based data model proves to be highly effective in representing domains, while also eliminating the constraints associated with the class-based model. Hence, in the context of user-generated contents, adopting the instance-based representation could be advantageous. Moreover, considering that data is often utilized for unforeseen purposes, opting for an instance-based representation, which offers a more flexible form of data modeling, is a preferable choice.

Chapter 6

Limitations and Future Research

Although this study provided valuable insights about the different aspects of information extraction from relational and instance-based data models, there are some limitations associated with the research methodology and the outcome which should be considered when interpreting the results. First, the sample size was relatively small with only 18 participants in the class-based group and 20 participants in the instance-based group, which may limit the generalizability of the findings. Second, most of the participants did not have previous training in data modeling, which may have impacted their performance and could affect the generalizability of the results. Third, regarding Hypothesis 1, which was not supported by the findings, further deliberation showed that the result might be because of ambiguity in two of the questions (question 2 and question 4). In these questions by 'unknown' name/industry, we meant that the instance does not possess name/industry attribute. Our intention was to assess participants' performance when faced with an instance that does not entirely correspond to the data model. However, the intention may have not be conveyed by the wording of the question, which limits the interpretation of results. Fourth, subjects' answers in the second task of the experiment were only coded and scored by one researcher. The use of multiple researchers to review the scored answers could have potentially mitigated any potential coding bias. Finally, the test presented to the participants was challenging and complex, resulting in some participants not completing the second task of the experiment, which could have potentially affected the accuracy of the results. It is worth noting that although all the participants were students, given the general nature of the tasks, there is no reason to expect students to behave differently than others potential participants (Parsons and Cole, 2005). Thus, using exclusively students as participants has not imposed any limitations on the study's outcomes. Furthermore, the lack a pre-test in the experiment is unlikely to have influenced the results as the questions demonstrated adequate precision in revealing distinctions between the two groups.

Despite these limitations, the current study produced some promising results about the effectiveness of instance-based data models. However, further empirical research is warranted to investigate the effectiveness of this type of representation and explore these findings in more depth. Future studies could investigate various dimensions of information extraction using a larger and more diverse and representative sample, including participants from various backgrounds, with advanced technical training and expertise in data modeling to explore whether the prior experience affected the results. Furthermore, providing participants with some detailed training or exposure to data modeling concepts prior to the study could mitigate the impact of limited prior knowledge and improve the overall quality of responses. Additionally, in future research, the second task of this study could be redesigned to require participants to generate actual queries for extracting information from relational and instance-based data, rather than providing written verbal descriptions.

Chapter 7

Conclusion

In this thesis, various aspects of information extraction were compared between class-based data models and instance-based data models. The findings demonstrated that to extract information about the relationships of instances/entities and retrieve information involved instances from more than one class (i.e., instances that have relationship with each other), the instance-based model was more effective. The results suggest that the instance-based model has potential practical implications for tasks where accurate identification of relationships is critical. In contrast, the class-based model showed a higher performance in extracting information regarding cardinalities of relationships and retrieving information involving only one entity, which are relatively simpler and less practically demanding use cases. Therefore, it can be concluded that both models have unique strengths that can complement each other, depending on the specific requirements of the information extraction task. Nonetheless, in complex real-world domains that involves exploring various relationships and using data for unanticipated purposes, instance-based representation becomes an advantageous approach to data modeling.

Bibliography

Allen, G. N. and March, S. T. (2006). The effects of state-based and event-based data representation on user performance in query formulation tasks. *Mis Quarterly*, pages 269–290.

Angles, R. (2018). The property graph database model. *AMW*.

Angles, R. and Gutierrez, C. (2008). Survey of graph database models. *ACM Computing Surveys (CSUR)*, 40(1):1–39.

Bera, P., Burton-Jones, A., and Wand, Y. (2014). How semantics and pragmatics interact in understanding conceptual models. *Information Systems Research*, 25(2):401–419.

Bodart, F., Patel, A., Sim, M., and Weber, R. (2001). Should optional properties be used in conceptual modelling? a theory and three empirical tests. *Information Systems Research*, 12(4):384–405.

Bowen, P. L., O'Farrell, R. A., and Rohde, F. H. (2009). An empirical investigation of end-user query development: The effects of improved model expressiveness vs. complexity. *Information Systems Research*, 20(4):565–584.

Burton-Jones, A. and Meso, P. N. (2006). Conceptualizing systems for understanding:

An empirical test of decomposition principles in object-oriented analysis. *Information Systems Research*, 17(1):38–60.

Corbellini, A., Mateos, C., Zunino, A., Godoy, D., and Schiaffino, S. (2017). Persisting big-data: The nosql landscape. *Information Systems*, 63:1–23.

Davoudian, A., Chen, L., and Liu, M. (2018). A survey on nosql stores. *ACM Computing Surveys (CSUR)*, 51(2):1–43.

Deepak, G. (2016). A critical comparison of nosql databases in the context of acid and base. *Culminating Projects in Information Assurance.*

Frisendal, T. (2016). Graph data modeling for nosql and sql. *Visualize structure and meaning. Technics Publications.*

Gemino, A. and Wand, Y. (2005). Complexity and clarity in conceptual modeling: Comparison of mandatory and optional properties. *Data and Knowledge Engineering*, 55(3):301–326.

Kaur, K. and Rani, R. (2013). Modeling and querying data in nosql databases. *In 2013 IEEE international conference on big data*, pages 1–7.

Lukyanenko, R. and Parsons, J. (2018). Beyond micro-tasks: Research opportunities in observational crowdsourcing. *Journal of Database Management*, 29(1):1–22.

Lukyanenko, R., Parsons, J., and Samuel, B. M. (2019). Representing instances: the case for reengineering conceptual modeling grammars. *European Journal of Information Systems*, 28(1):68–90.

Lukyanenko, R., Parsons, J., Wiersma, Y., Wachinger, G., Huber, B., and Meldt, R. (2017). Representing crowd knowledge: Guidelines for conceptual modeling of user-generated content. *Journal of the Association for Information Systems*, 18(4):2.

Lukyanenko, R., Parsons, J., and Wiersma, Y. F. (2014). The iq of the crowd: Understanding and improving information quality in structured user-generated content. *Information Systems Research*, 25(4):669–689.

Parsons, J. (2011). An experimental study of the effects of representing property precedence on the comprehension of conceptual schemas. *Journal of the Association for Information Systems*, 12(6):1.

Parsons, J. and Cole, L. (2005). What do the pictures mean? guidelines for experimental evaluation of representation fidelity in diagrammatical conceptual modeling techniques. *Data and Knowledge Engineering*, 55(3):327–342.

Parsons, J. and Wand, Y. (2000). Emancipating instances from the tyranny of classes in information modeling. *ACM Transactions on Database Systems (TODS)*, 25(2):228–268.

Pitts, M. G. and Browne, G. J. (2004). Stopping behavior of systems analysts during information requirements elicitation. *Journal of management information systems*, 21(1):203–226.

Recker, J., Indulska, M., Green, P., Burton-Jones, A., and Weber, R. (2019). Information systems as representations: A review of the theory and evidence. *Journal of the Association for Information Systems*, 20(6):5.

Saghafi, A. and Wand, Y. (2014). Do ontological guidelines improve understandability of conceptual models? a meta-analysis of empirical work. *In 2014 47th Hawaii International Conference on System Sciences*, pages 4609–4618.

Saghafi, A., Wand, Y., and Parsons, J. (2022). Skipping class: improving human-driven data exploration and querying through instances. *European Journal of Information Systems*, 31(4):463–491.

Shanks, G., Moody, D., Nuredini, J., Tobin, D., and Weber, R. (2010). Representing classes of things and properties in general in conceptual modelling: An empirical evaluation. *Journal of Database Management (JDM)*, 21(2):1–25.

Shanks, G., Tansley, E., Nuredini, J., Tobin, D., and Weber, R. (2008). Representing part-whole relations in conceptual modeling: An empirical evaluation. *MIS Quarterly*, pages 553–573.

Vera-Olivera, H., Guo, R., Huacarpuma, R. C., Da Silva, A. P. B., Mariano, A. M., and Holanda, M. (2021). Data modeling and nosql databases-a systematic mapping review. *ACM Computing Surveys (CSUR)*, 54(6):1–26.

Wand, Y. and Weber, R. (1995). On the deep structure of information systems. *Information Systems Journal*, 5(3):203–223.

Appendix A

Reasoning Behind Developing Question

This appendix provides an explanation of the reasoning behind each question, as well as guidance on how participants in each group can arrive at the correct answer.

Table A.1: Logic for developing questions of H1

Question	Reasoning behind question and expected way of answering it
1. Is it possible to represent a person in the model with the following attributes? name: 'Judi' height: 5.4	Reasoning behind the question: For both data models, the instance in this question is somehow a new instance. In the class-based model, the entity 'person' has six properties. In the instance-based model the instances that could belong to 'person' class possess three to five properties. On the other hand, unlike the instance-based model, the class-based model only shows the name of properties and does not assign any sample value to them. The participant is expected to comprehend if this new instance can be an instance of the model. This can be measure of whether the model embrace uniqueness of instances and how it can deal with new observations in the domain that do not fully comply with the prior understanding of the domain. Correct Answer: Yes. How to answer the question with instance-based model: In this model, we have instances with three properties (ID, name, gender), four properties (ID, name, age, gender), and five properties (ID, name, age, gender, height), that could belong to 'person' class. This different combination of properties means that it is highly likely the instance in the question be an instance of the model. How to answer the question with class-based model: In this model the 'person' entity has six properties $(ID, name, age, gender, height, car_id)$. This model does not contain any notation to specify if an instance of this model may not possess any of these attributes (i.e., the model does not specify if the attributes are optional or not). As a result, it is highly unlikely the users of this model could answer this question correctly.
2. There can be a person whose name is unknown.	Reasoning behind the question: The goal is to assess whether a new instance that lacks the attribute 'name' can still be considered a valid instance of the model. Here by unknown name, we mean not having the property 'name'. This could be measure of if participants can make inferences about the properties of instances/entities based on the model. Here again we are trying to assess participants understanding in dealing with instances that do not completely match the data model. Correct Answer: True How to answer the question with instance-based model: In this model, we have instances of 'person' class that possess different combination of attributes. For example, we have an instance that does not have the attribute 'age'. Thus, considering all instances of this class, it is expected the participant makes inference that an instance without attribute 'name' is acceptable. How to answer the question with class-based model: In the class-based model we do not have any notation to specify if each property is optional or mandatory. So, the participant cannot be sure if the new instance without attribute 'name' is an acceptable instance or not.

Question	Reasoning behind question and expected way of answering it
3. Is it possible to represent a company in the model with the following attributes? ID : 500 $market_cap$: 230M	See question 1.
4. For all instances of companies in the model, the industry in which they belong is known.	See question 2.
5. Insurance cost is fixed for a car no matter who insures it.	Reasoning behind the question: The goal of this question is to assess if 'insurance-cost' attribute is fixed for a specific car, or it changes depending on who insures it. In other words, are participants able to identify whether the cost is an attribute of the car or attribute of its relationship with a person. Correct Answer: True. How to answer the question with instance-based model: In this model, this attribute is shown within the circle, meaning that it is an attribute of car instance. If it was shown on the relationship connecting two instances, then it would be the attribute of the relationship. How to answer the question with class-based model: : In this model, this attribute is part of 'car' entity. So, it is an attribute of a car.

Table A.2: Logic for developing questions of H3

Question	Reasoning behind question and expected way of answering it
6. A company can be shareholder of many companies.	Reasoning behind the question: The goal of this question is to assess if the participants can comprehend that 'shareholder' relationship is a zero-to-many relationship. Correct Answer: True. How to answer the question with instance-based model: Answering the question using this model, depends on how many sample instances and relationships are represented in the data model and if the model contains an instance that has more than one 'is shareholder' relationships. Since this model only shows a few sample instances of the domain, it might be difficult to answer the question with the model correctly. How to answer the question with class-based model: This model has well-defined notations to specify minimum and maximum cardinality. In the data model, the minimum and maximum cardinalities on the side of *company* entity are 1 and 1, and they are 1 and many on the side of $company_shareholders$ entity respectively. This means that a company and be shareholder of many companies.
7. A person can work for multiple companies.	See question 6.

Question	Reasoning behind question and expected way of answering it
8. There could be a car that is driven by no one.	Reasoning behind the question: The goal of this question is to evaluate whether the participants can understand that 'drive' relationship is a zero-to-many relationship. Correct Answer: True. How to answer the question with instance-based model: A minimum cardinality of zero could not be represented in this model. On the other hand, the existence of a relationship cannot imply the maximum cardinality. Even if the model contains an instance that has more than one 'drive' relationship, it does not necessarily specify the maximum cardinality of such relationship. How to answer the question with class-based model: The model shows that on the 'car' side, the cardinality is constrained to a minimum of 0 and a maximum of 1. However, on the 'person' side the minimum and maximum cardinality are 0 and many. This means that there could be a car that is driven by no one, a person can only drive a minimum of one car, and a car might be driven by zero or many people.
9. A person can drive a maximum of one car.	See question 8.
10. A person can drive more than one car.	See question 8.

Table A.3: Logic for developing questions of H2.

Question	Reasoning behind question and expected way of answering it
11. A person can drive a car that is not insured by herself/himself.	Reasoning behind the question: This question asks about information that is related to two relationships in the model: 'car insurance' and 'drive'. It targets a chain of relationships and answering it requires following this chain of relationships. Essentially, this question requires an understanding of how car ownership and car insurance are connected and how they impact each other. Correct Answer: True. How to answer the question with instance-based model: In the model, the participants can visually see how these two relationships are connected. So, following this chain of relationship is straightforward in this model. How to answer the question with class-based model: This model uses a separate intermediate entity to show 'car insurance' relationship. Thus, connecting the two relationships, creating the chain of relationships, and then making inferences about it is very challenging in this model.
12. A person who is not the owner can drive a car.	Reasoning behind the question: The idea of this question is same as question 11, but it focuses on 'car ownership' and 'drive' relationships. Correct Answer: True. How to answer the question with instance-based model: See question 11. How to answer the question with class-based model: See question 11.

Question	Reasoning behind question and expected way of answering it
13. A person can own a car and drive another car.	Reasoning behind the question: See question 12. Correct Answer: True. How to answer the question with instance-based model: See question 11. How to answer the question with class-based model: See question 11.
14. If two people are not friends with each other, it is possible that they have a mutual friend.	Reasoning behind the question: The idea of this question is same as question 11, but it focuses on the chain of friendships between people. Answering this question demands understanding this chain of relationships and drawing inferences based on that. Correct Answer: True. How to answer the question with instance-based model: In the model, the participants can visually see how the friendship relationship between people. Thus, following this chain of friendships is straightforward in this model. How to answer the question with class-based model: This model uses a separate intermediate entity to show 'friend with' relationship. Thus, connecting the data of people and the data of friendship relationships, is very challenging in this model.
15. A person can work for a company and be its shareholder at the same time.	Reasoning behind the question: The idea of this question is same as question 11, but it focuses on 'work for' and 'shareholder' relationships. Correct Answer: True. How to answer the question with instance-based model: In the model, the participants can visually see how these two relationships are connected. Thus, following this chain of relationship is straightforward in this model. How to answer the question with class-based model: This model uses two separate intermediate entities to show 'work for' and 'shareholder' relationships. Thus, creating the chain of relationships and making inferences based on that is challenging in this model.

Table A.4: Logic for developing questions of H4

Question	Reasoning behind question and expected way of answering it
1. Find the average insurance cost	Reasoning behind the question: This question focuses on finding average insurance cost of instances with the common property of 'insurance cost'. Thus, it only targets instances that belong to the same class/entity. The goal is to evaluate participants' performance in this type of information retrieval. How to answer the question with instance-based model: Participants using this model need to explore the whole dataset and come up with a mechanism to filter the data and find all distinct instances with 'insurance cost' property. Thus, it is expected that answering this question be more challenging for the instance-based group compared to the class-based group. How to answer the question with class-based model: Participants using this model only need to go the related sheet in the data (in this example 'car' sheet) and look at the 'insurance cost' column to find the related data.
2. Find the highest market capitalization ($market_cap$)	Reasoning behind the question: This question uses the same logic as question 1 and concentrates on instances with the 'market capitalization' property. How to answer the question with instance-based model: See question 1. How to answer the question with class-based model: See question 1.
3. Find the number of females	Reasoning behind the question: This question uses the same logic as question 1 and targets instances with 'gender' property. How to answer the question with instance-based model: See question 1. How to answer the question with class-based model: See question 1.

Table A.5: Logic for developing questions of H5

Question	Reasoning behind question and expected way of answering it
4. Find the color of the car that Sarah drives.	Reasoning behind the question: See question 5 below. The rationale for this question is the same as question 2 below, but it asks about Sarah (first instance) and the car (second instance) that she drives (the relationship between two instances).. How to answer the question with instance-based model: See question 5. How to answer the question with class-based model: See question 5.

Question	Reasoning behind question and expected way of answering it
5. Find the name of the company that John works for.	Reasoning behind the question: Here the required information is related to John (first instance) and the company (second instance) that he works for (the relationship between two instances). How to answer the question with instance-based model: In the instance-based data all instances are represented in a single view and their relationships are explicitly specified. Thus, the participants using this model need to filter the 'work for' relationship and then look for the required information. How to answer the question with class-based model: Retrieving the same information using the class-based data requires joining three tables: person, company, work for. From the 'person' table they need to extract John's ID, then from 'work for' table extract the ID of the company related to John's ID, and then in the 'company' table extract the name of the required company using its ID. Thus, the procedure in the class-based model is more complicated. As a result, it is expected that answering this question be more challenging for the class-based group compared to the instance-based group.
6. Find the name and age of all people who are friend with Bob.	Reasoning behind the question: See question 5. The logic of this question is the same as question 5. This question focuses on Bob (first instance) and the people (second instance) who are his friend (the relationship between two instances). How to answer the question with instance-based model: See question 5. How to answer the question with class-based model: See question 5.
7. Find the name of all shareholders of Vandelay company.	Reasoning behind the question: See question 5. This question uses the same logic as the above questions and concentrates on Vandelay company (first instance) and other companies/people (second instance) which are its shareholders (the relationship between two instances). How to answer the question with instance-based model: See question 5. How to answer the question with class-based model: See question 5.
8. John and Mark are not friends with each other. Find a mutual of friend of them that could introduce them to each other.	Reasoning behind the question: The rationale for this question is same as question 5. The required information is about John (first instance), Mark (second instance), John's friends (third instance), Mark's friends (fourth instance) and their chain of friendship (the relationship between two instances). How to answer the question with instance-based model: See question 5. How to answer the question with class-based model: See question 5.

Appendix B

Scoring the Second Task

This appendix provides details of how we scored the second task with some examples. To be considered complete and receive a score of 5, an answer must provide a clear and accurate description of the process involved in extracting the required information from the Excel file, including the use of appropriate sheets, columns, operations, and steps. On the other hand, if an answer was deemed satisfactory and logical but contained some minor mistakes, one point was deducted for each mistake. An example answer for each question is presented below, along with a discussion of the rationale for its score.

Question 1 for the class-based group: This answer received a score of 5 because it correctly identified the related sheet, column, and procedure to extract the required information.

> "We need to look at the 'Car' sheet and look at the 'insuranceCost' column. We sum all the values in the 'insuranceCost' column and divide it by the number of rows that have an insuranceCost greater than 0"

Question 1 for the instance-based group: This answer received a 4 because it did not select distinct cars. We might have duplicate instances in the data when representing different relationships.

> “Search data to find all of cars. From the results, use their insurance cost and calculate average cost.”

Question 2 for the class-based group: This answer received a 5 due to describing the correct sheet, column, and operation.

> “In the Company sheet, find the Market Cap column, then find the highest value in this column.”

Question 2 for the instance-based group: This answer received a 4 because using this procedure may not select all companies. We might have companies without a shareholder relationship.

> “Search the Relationship column, get rows with value Is Shareholder of. Next, get instances that have market capitalization feature. Next, obtain their maximum values.”

Question 3 for the class-based group: This answer received a 5 because it used correct sheet, column, and steps.

> “1- I look at the sheet ”Person.” 2- I scan through column D ”gender” to count how many Fs there 3- I find 4 females.”

Question 3 for the instance-based group: This answer received a 4 because it did not select distinct instances.

> “We look at the ‘instance 2’ column, from this column we select all the instances with the attribute ‘gender’. From these attributes we all the letters ’F’ in front of the gender”

Question 4 for the class-based group: This answer scored a 5 due to its accurate use of the correct procedure, sheets, and columns without any mistakes.

> “Start by person sheet and locate Sarah and her car’s Id. After that, switch to cars data and find her car’s color.”

Question 4 for the instance-based group: : This answer received a 5 because the described procedure is accurate and complete.

> "Search the Relationship column and get rows with value Drive. Next, in the instance1 column, get the row with name Sarah. Next, search the instance2 column and obtain the color of the car."

Question 5 for the class-based group: This answer received a 4 because it should have stated that use 'work for' sheet to find his company Id.

> "Use person sheet. Find John and his company id. Look at company sheet and find his company and its name."

Question 5 for the instance-based group: This answer received a 5 because the described procedure is correct and can extract the required information.

> "You could search the Relationship column and find Work for value. In the first column find John. Then in the third column find the name of his company."

Question 6 for the class-based group: : The only mistake in this answer is that at the end it did not state to go back to the 'Person' sheet and find their name and age. As a result, it received a 4.

> "In the Person sheet, find the row with name Bob and find the related ID. Then go to Friend With sheet, find the related IDs and their name and age."

Question 6 for the instance-based group: This answer received a 5 because it identified the required columns and steps correctly.

> "select relation friend. find all instances 1 that has name bob. select instance 2. find name and ages of those people."

Question 7 for the class-based group: This answer was awarded a score of 5 because it accurately utilized the correct procedure, sheets, and columns without any errors.

> “In the *Company* sheet, we identify the ID of the required company. Then, in the $Company_Shareholders$ sheet, we identify all the $company_ID1$ be the same as the previously identified ID. After that, we get all the $person_ID_Shareholder$ list and go to the $Person$ sheet. After that, we select all the rows in the $Person$ sheet that have the ID value equaling to the $person_ID_Shareholder$ list of IDs. After that, we get the name of those rows.”

Question 7 for the instance-based group: This answer received a 4 because instead of using instance2 column to find the name of the company, it used instance1 column.

> “You need to look at the Relationship column and find Is shareholder of value. In instance 1 column find instance name Vandelay. Then in instance 2 column find names of related ones.”

Question 8 for the class-based group: This answer used the correct sheets, however the procedure to extract the required information is not clear. It should have mentioned that in the ‘friends’ data, use the Id of John to find his friend list and use Id of Mark to find his friend list and then match them to find mutual friends. This answer received a 2.

> “Use person data and find John and Mark. Next, use friends data to find a mutual friend of theirs.”

Question 8 for the instance-based group: This answer received a 5 because the required information can be extracted correctly using this procedure.

> “You need to look at the Relationship column and find Friend with value. In instance 1 column find people with John’s name and Mark. Then find friends of each one and match them to find mutual friends.”