Using Generative AI for Tutoring Data Science

by

Yusra Khalid

Submitted to the
B. Thomas Golisano College of Computing and Information Sciences Department of Computer Science
in partial fulfillment of the requirements for the
Master of Science Degree
at the Rochester Institute of Technology

Abstract

Generative AI can be used for facilitating education in Data Science. The capabilities of Generative AI were explored along with the comparison of a few different models in terms of Data Science. ChatGPT provides more factually correct answers as compared to Gemini and is better at solving problems and explaining concepts as compared to GitHub Copilot. There are limitations of ChatGPT when it comes to computations and solution building for Data Science, but its use can facilitate the learning process of students. The topics like schema building, data creating, query writing, normalization, item set mining, and clustering can be learned and understood with ChatGPT. Using it for educational purposes will facilitate students in faster and better learning as compared to if it is prohibited. The prompt engineering practices for solving practical exercises have also been explored and described in this paper. The topics of data science can be solved with the help of ChatGPT for learning purposes. This can be used for both solving the questions and for explaining the answers. Students can work on questions step by step with the help of ChatGPT and learn the process of solving the questions. Some questions can be solved easily with simple prompts while some require more structured prompts. The effectiveness of using ChatGPT as a tutor of Data Science has also been evaluated in three different ways. First, a study was conducted to see how helpful ChatGPT is in helping participants solve data science questions. Second, a series of sessions were created with ChatGPT to help solve and explain data science concepts in a structured way and the accuracy of these answers was analysed. The third approach was using another Generative AI model to test how ChatGPT acted as a tutor for undergraduate students for learning different topics. The results are mostly positive with a few limitations. The results show that it can be used as a tutor for learning data science but can not be the only source of learning. Some guidance

or knowledge is needed for better use of the Generative AI. It can not substitute teachers but can act as a facilitator that can be a personalized tutor for each student. It can explain the solutions given in the textbooks in more detail and can also help in error resolution. A large number of participants also said that they are likely to use ChatGPT as a data science tutor in future.

In conclusion, ChatGPT has the potential to revolutionize data science education by acting as a personalized tutor, enhancing the learning experience, and bridging gaps in understanding complex concepts.

Acknowledgments

I extend my sincere thanks to Dr. Carlos Rivero for his support, guidance, encouragement, and expertise in this whole research. His thorough insights and dedication have made this research possible. I would also like to thank Prof. Zachary Butler for his valuable feedback and guidance in carrying out this research.

I would like to express my gratitude to all the faculty of Rochester Institute of Technology for their support and guidance throughout my academic journey. I want to thank all the participants who took part in the study and provided valuable feedback for the research. I also want to extend my thanks to all my teachers, friends, and family for their unwavering support and encouragement throughout this journey. I acknowledge the use of generative AI from ChatGPT, Gemini, Fidelity, GitHub Copilot, and Claude for the research purposes.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years we have observed a large increase in the use of Generative AI in various fields. A study shows that the use of generative AI in science and engineering students tend to increase with each passing year [10]. The students in the study showed interest to increase their use of Generative AI in the next year of college and also said that their use has increased from previous year. So the students' use of Generative AI is increasing day by day. With the ongoing research on improving the models to provide better and more useful information, Generative AI is getting adopted more in the everyday life of people. Generative AI is often looked at with a controversial image in the education sector [4]. Some are interested in incorporating it into the educational system, while others resist it as it could hamper the learning abilities of students. There is a debate going on about the uses of generative AI in academia. AI is already all around us; students can learn how to use it for better results [6]. More than 50% of college students use generative AI for their studies [10].

The research is available on how Generative AI can be used for the better learning of the students. The AI can act as a tutor to help students understand their coursework in a better way [3]. There has been some research on the benefits and uses of Generative AI in education. Other researches are available on using Generative AI in data science [5] but, to the best of our knowledge, education of data science with the help of Generative AI is still unexplored area. The Generative AI can act as a tutor can help students solve questions of practical exercises in their coursework. Generative AI can also help in explaining the concepts and explain the solutions in a detailed manner.

Data Science is a field of study that uses scientific methods, processes, and systems to extract knowledge and insights from data [9]. Data science uses scientific methods, statistics, and algorithms to extract the information and store it in databases. Understanding the databases and how to get, update, modify, transform, remove, and utilize data comes within this domain. The two popular books used in the course of data science are "Database System Concepts Seventh Edition" [7] and "Data Mining Concepts and Techniques" [2]. Generative AI has been explored in this paper to solve some of the practical exercises in these books.

1.1 Introductory Data Science Course

The topics being taught to undergraduate students are focused on the databases and the algorithms involved in optimizing them [2, 7]. An introductory data science course starts with the relational model that stores data in a structured way. The structure is called a schema that is made up of tables, and each table consists of attributes. The tables are linked with each other on the basis of attributes that are common in them, making them foreign keys. Each row in these tables can be uniquely identified with a unique primary key, and foreign keys point to these primary keys of other tables to make a link between two tables. There are constraints that are applied on these attributes in the tables that specify the type of information that can be stored inside each attribute. Data is added to the tables after creating the schema and tables with attributes and their constraints. The next topic is retrieving this structured data in a proper manner. Structured Query Language (SQL) is a domain-specific language that is used for storing, processing, and managing data in relational databases. The practical exercises for these topics include writing SQL queries for creating schemas and inserting, retriving, updating or deleting data in the databases. Then a foundational topic in relational databases is normalization that is defined as the process of re-organizing attributes and relations of a relational database to minimize redundant data. It makes sure that the data is stored consistently and efficiently to minimize data dependency. The practical exercises for normalization include finding the closure of attributes, finding candidate keys, and finding the canonical cover of the functional dependencies.

Another important topic in data science is item set mining that is used to find the frequent item sets in the data. This is used to find the patterns in the data that are repeating and can be used to make decisions based on them. Rule are also discovered according to data patterns called association rules. The practical exercises include finding the frequent itemsets given the transaction data and then finding the association rules from these frequent itemsets. Clustering is a major topic in data science that is used to group similar data points together. It is used to discover hidden patterns in the data and simplify the data for better understanding. The practical exercises for clustering include finding the clusters each data point belongs to.

1.2 Problem Statement

The problem under consideration is how GenerativeAI can help to solve and explain the data science concepts to undergraduate students. The study presented in this thesis aims to find how Generative AI can be used as a tutor for solving practical exercises of data science and explaining concepts of data science. The focus was on the use of Generative AI for both basic and advanced topics in data science like SQL, normalization, clustering and frequent item set mining. A research study was also be conducted to test the usability of Generative AI for solving data science questions. The feedback of participants was be collected to determine the effectiveness of ChatGPT. The prompt engineering practices for solving practical exercises were also be explored and described in this paper.

1.3 Research Methodology

The study initially focuses on a few different Generative AI model to find the best one for data science. The comparison was done on the basis of the factual correctness of the answers and the ability to solve and explain the data science concepts. ChatGPT was found to be the best model for data science among Gemini, Fidelity and Github Co-pilot. The research study with students also showed that ChatGPT is the preferred Generative AI model for majority of the participants. The study then focused on the use of ChatGPT for solving and explaining the data science concepts. The focus has been on finding the capabilities and limitation of ChatGPT for solving exercises and explaining the working.

1.3.1 Exploratory Analysis

Prompt engineering techniques are explored [1] to find the optimal way of prividing the input to ChatGPT for better results for database creation, data retrieval, and database normalization. ChatGPT was explored in mainly two forms, one was using it to solve practical exercises like writing SQL queries, creating schemas, normalizing and analysing the data. The other was to explain the concepts or the solutions in the book step by step that is easier to understand or clear any confusion that students might have. For solving practical exercises, the input provided to ChatGPT was the problem statment or practical question and then it was provided the steps of algorithm to solve the problem step by step. The output of this process was the solution to the problem that was then checked for correctness. For explanation, the input was the solution to the problem and the output is the explanation of the concept or the solution in the form of steps that are easier to understand. The data generation and schema formation was also tested with ChatGPT to see how it can be used to generate data and form schemas. The results were satisfactory and the data generated was used to solve the exercises in the book. ChatGPT has limitations in creating large scale complex data involving multiple dependencies and constraints. ChatGPT was also able to write queries that are slightly complex, and upon giving it a query it was also able to provide an explaination and the result as well, if the data is small.

For normalization, a higher level algorithm broken down into smaller step was used as an input to ChatGPT to provide a solution. This was useful as ChatGPT acted as a tutor that student can work with to solve normalization problems step by step and ask questions if there is something they don't understand. In a similar way the frequent item set mining and clustering can be explained and solved with ChatGPT. Providing a higher level algorithm step by step is beneficial as it will have the students focus on the details of the algorithm and understand it better. The use of effective prompts is also important as it helps in getting better results from ChatGPT. The prompts have been used to structure the input in a way that ChatGPT can understand it better and provide the correct output.

1.3.2 Evaluation

The effectiveness of using ChatGPT as a tutor of Data Science has also been explored with the help of students and Claude.

Research Study

A research study was conducted where students were given SQL and normalization questions to solve on their own and with the help of ChatGPT. It was found in the study that when the questions were simple, majority of the participants were able to get correct result with ChatGPT even if they were unable to solve correctly on their own. As the questions got complex the number of correct answers (with ChatGPT) decreased showing some level of guidance is required for using solving complex problems. The feedback of the students was collected and it was found that majority of the students found ChatGPT useful in solving the problems and understanding the concepts.

Empirical Evaluation

Another method of evaluation was to calculate the correctness of the answers given by ChatGPT. The topics are broken down into smaller solveable steps that could be treated as a single unit. The count of solving such units with ChatGPT was calculated and divided into three categories: worked, worked with correction, and did not work. The answers that worked were the ones that were correct and the ones that did not work were the ones that were incorrect. The worked with correction were the ones that made some mistake in the process but after correction or retry the answers were corrected.

Evaluation with Another Generative AI Model

ChatGPT was also evaluated as a tutor for undergraduate students with the help of another Generative AI model, Claude. Claude found it mostly a good tutor for undergraduate students with rating above 3 out of 5, in most cases. The rating of Claude for ChatGPT as a tutor is also provided in the evaluation section.

Overall, the ChatGPT can be used a tutor by students for learning data science concepts and solving practical exercises. The study showed that although simpler questions or SQL queries can be solved by students on their own, the complex questions require some level of guidance. This shows that although Generative AI is very powerful and used as a tutor but cannot be expected to do everything on its own.

1.4 Thesis Structure

The thesis initially starts with the introduction of the topic and defining the focus of the study in Chapter 1.

Chapter 2 presents the background of the study that explains the topics of data science that are being researched with Generative AI. The basic understanding of the topics and how to solve practical questions related to them is also provided in the background. After initial comparison of ChatGPT, Gemini, SQL Fidelity, and GitHub Copilot, ChatGPT outperformed other models for data science. The study then focused on the use of ChatGPT for solving and explaining the data science concepts.

Chapter 3 then presents the first topic of data science ”Relational Database Schemas” that is researched with Generative AI. The chapter focuses on how to generate structural schema and data with ChatGPT.

Chapter 4 presents SQL queries; writing, executing and explaining SQL queries with ChatGPT along with error resolution.

Chapter 5 presents normalization using ChatGPT. The chapter focuses on using appropriate prompts for finding closure, candidate keys and canonical cover of given set of functional dependencies.

Chapter 6 presents itemset mining with ChatGPT. The chapter focuses on solving itemset mining and creating accosiation rules with ChatGPT.

Chapter 7 presents clustering with ChatGPT. The chapter focuses on solving k-mean clustering problems with ChatGPT.

Chapter 8 presents the evaluation of the study. The chapter focuses on using 3 different approaches to evaluate the effectiveness of ChatGPT in solving data science problems. The chapter also presents the feedback of students and professors on the use of ChatGPT as a tutor for data science.

Chapter 9 presents the conclusion of the study. The chapter summarizes the study and provides the conclusion of the study.

Chapter 2

Background

Generative AI has changed the way of working for many disciplines in the past few years. It is getting more and more adopted in the field of education by students of science and engineering [10]. The AI can act as a tutor to help students understand their coursework in a better way [3]. There are a few studies that exists in the intersection of both AI and Data Science. One such study is on how to extract information from large language models (LLM) using structured query languages. This study focused on the extraction of important information from the LLM in the structured form like table instead of long texts [5]. The focus here is to use Generative AI, to help students learn data science concepts by solving practical exercise questions. The topics of data science researched in this thesis with Generative AI are based on the book "Database System Concepts Seventh Edition" [7] and "Data Mining Concepts and Techniques" [2]. An overview of the data science topics explored in this thesis are given below along with a brief explaination of how to solve practical exercises in these topics.

2.1 Relational Databases

The topic starts with the relational model that stores data in a structured way. The structure is called a schema that is made up of tables, and each table consists of attributes. The tables are linked with each other on the basis of attributes that are common in them, making them foreign keys. Each row in these tables can be uniquely identified with a unique primary key, and foreign keys point to these primary keys of other tables to make a link between two tables. There are constraints that are applied on these attributes in the tables that specify the type of information that can be stored inside each attribute. Data is added to the tables after creating the schema and tables with attributes and their constraints. The example exercise question is to find appropriate primary keys for a given schema. The schema is given as:

```
Employee (person_name, street, city)
```

```
Works (person_name, company_name, salary)
Company (company_name, city)
```

The answer of the question is the primary keys for each table i.e. person_name for Employee, person_name for Works, and company_name for Company [7].

2.1.1 University Schema

The book uses University Schema for many queries, and this paper also used Generative AI to test generation of university schema and work on it. The schema has the structure:

Table: section (course_id, sec_id, semester, year, building, room_number, time_slot_id)
Primary Keys (course_id, sec_id, semester, year)
Foreign Keys (classroom[building, room_number], time_slot[time_slot_id])

Table: classroom (building, room_number, capacity)
Primary Keys (building, room_number)

Table: takes (ID, course_id, sec_id, semester, year, grade)
Primary Keys (ID, course_id, sec_id, semester, year)
Foreign Keys (section[course_id, sec_id, semester, year], student[ID])

Table: teaches (ID, course_id, sec_id, semester, year)
Primary Keys (ID, course_id, sec_id, semester, year)
Foreign Keys (section[course_id, sec_id, semester, year], instructor[ID])

Table: course (course_id, title, dept_name, credits)
Primary Keys (course_id)
Foreign Keys (department[dept_name])

Table: prereq (course_id, prereq_id)
Primary Keys (prereq_id, course_id)
Foreign Keys (course[course_id], course[prereq_id])

Table: student (ID, name, dept_name, tot_cred)
Primary Keys (ID)
Foreign Keys (department[dept_name])

Table: department (dept_name, building, budget)
Primary Keys (dept_name)

Table: instructor (ID, name, dept_name, salary)
Primary Keys (ID)

Foreign Keys (department[dept_name])

Table: advisor (s_id, i_id)
Primary Keys (s_id)
Foreign Keys (instructor[i_id], student[s_id])

Table: time_slot (time_slot_id, day, start_hr, start_min, end_hr, end_min)
Primary Keys (time_slot_id, day, start_hr, start_min)

2.2 SQL

The SQL topic works by teaching this and then after adding the data, it moves towards retrieving this structured data in a proper manner. Structured Query Language (SQL) is a domain-specific language that is used for storing, processing, and managing data in relational databases. The use of SQL is taught as both Data Definition Language (DDL) and Data Manipulation Language (DML) along with the syntax and semantics. The SQL statements start from basic select statements that fetch data from one table, then go into more complex statements that involve data retrieval from multiple tables involving both join operations and sub-queries. The book uses many examples to make students better understand the way to write SQL queries. An example of practical exercise question for SQL using university schema described in section 2.1.1 is [7]:

Find the title of courses in the Comp. Sci. department that have 3 credits.

The answer of this question is:

```
SELECT title
FROM course
WHERE dept_name = 'Comp. Sci.' AND credits = 3;
```

2.3 Normalization

Normalization is defined as the process of re-organizing attributes and relations of a relational database to minimize redundant data. It makes sure that the data is stored consistently and efficiently to minimize data dependency. The main improvements normalization makes are optimize database performance, enhance database scalability, improve data integrity, and reduce data redundancy. Finding functional dependencies and optimizing them are an important part of normalization. A functional dependency is similar to a business rule or constraint that appears in our data. For example, for each SSN there is always the same first name so this is a functional dependency. Using university schema from section section 2.1.1, an example fo set of functional dependencies for relation course would be:

```
course (course id, title, dept name, credits)
FDs = {
    course_id → title
    course_id → dept_name
    course_id → credits
    title → dept_name
    title → credits
}
```

These functional dependencies (FDs) define that title of a course can be obtained by using course_id. In a similar way dept_name and credits can also be obtained by using course_id. The title of a course can also determine the department name and credits of the course.

Finding Closure, Candidate Keys, and Canonical Cover are a few important steps in normalization.

Closure: Given the set of functional dependencies, closure of α is defined as all the attributes that can be retrieved from α using the functional dependencies. For example, for the set of functional dependencies of course table from university schema; FD = $\{\{course\_id\} \rightarrow \{title\}, \{course\_id\} \rightarrow \{dept\_name\}, \{course\_id\} \rightarrow \{credits\}, \{title\} \rightarrow \{dept\_name\}, \{title\} \rightarrow \{credits\}\}$, the closure of {course_id} denoted as $course\_id^+$ is {course_id, title, dept_name, credits}. It can be obtained by starting with {course_id} and then adding the attributes that can be obtained from course_id using the functional dependencies.

Candidate Keys: A Candidate Key is the one that is able to find all attributes in the relation through the given set of functional dependencies. Upon taking closure of the candidate key, it should be equal to the set of all attributes in the relation. For example, for the set of functional dependencies of course table from university schema; FD = $\{\{course\_id\} \rightarrow \{title\}, \{course\_id\} \rightarrow \{dept\_name\}, \{course\_id\} \rightarrow \{credits\}, \{title\} \rightarrow \{dept\_name\}, \{title\} \rightarrow \{credits\}\}$, the candidate keys are {course_id} and {course_id, title}.

Extraneous Attributes: The attributes in the functional dependencies whose removal does not impact functional dependencies. These are the attributes that we can disregard in functional dependencies.

Canonical Cover: Canonical cover is the simplified set of functional dependencies that behaves similarly to the original set of functional dependencies. This is obtained by removing all the extraneous attributes in the set of functional dependencies. For example, for the set of functional dependencies of course table from university schema; FD =

$\{\{course\_id\} \rightarrow \{title\},\ \{course\_id\} \rightarrow \{dept\_name\},\ \{course\_id\} \rightarrow \{credits\},\ \{title\} \rightarrow \{dept\_name\},\ \{title\} \rightarrow \{credits\}\}$, the canonical cover is $\{\{course\_id\} \rightarrow \{title\},\ \{title\} \rightarrow \{dept\_name,\ credits\}\}$.

2.4 Itemset Mining

Itemset mining is the process of finding items that occur together frequently in a dataset. This technique is used in data analysis to generate association rules. These association rules are used to find the relationship between different items in a dataset. It can be used in various fields like market basket analysis, intrusion detection, and bioinformatics. For university schema, the itemset mining can be used to find the courses that are frequently taken together by students or the courses that are frequently offered together by the university.

Apriori Algorithm: Apriori algorithm is used to find the frequent itemsets in a dataset. The algorithm uses lexiographically sorted items to generate candidate sets. Initially, the 1-itemsets are generated by counting the occurance of each item in all transactions and only keeping the ones above the minimum support. Then higher order itemsets are created using the previous order itemsets by joining them with each other and keeping only the ones with count greater than minimum support. In this wasy only the items that occur a more than a threshold are kept and those bought togher are kept together in higher orders. The process is repeated until no more itemsets can be generated. The algorithm is given below [8]:

Algorithm 2: Apriori Algorithm

Input: Dataset D, Minimum Support Threshold minsup
Output: Frequent Itemsets

$L_1 \leftarrow \{\text{large 1-itemsets}\}$;
for $k \leftarrow 2$ ***to*** $L_{k-1} \neq \emptyset$ **do**
    $C_k \leftarrow \text{apriori-gen}(L_{k-1})$; `// Generate new candidates`
    foreach *transaction* $t \in D$ **do**
        $C_t \leftarrow \text{subset}(C_k, t)$; `// Candidates contained in` t
        foreach *candidate* $c \in C_t$ **do**
            $c.\text{count} \leftarrow c.\text{count} + 1$;
        end
    end
    $L_k \leftarrow \{c \in C_k \mid c.\text{count} \geq \text{minsup}\}$;
end
return $\bigcup_k L_k$;

Example: For students taking following courses together:
1. Mathematics, Physics, Chemistry
2. Mathematics, Physics
3. Biology, Chemistry

Using minimum support as 2, the frequent itemsets are
L1 = {{Mathematics: 2}, {Physics: 2}, {Chemistry: 2}}
L2 = {{Mathematics, Physics: 2}}

Association Rules: These itemsets are then used to generate association rules. The association rules can be generated from these frequent itemsets. For example, the rule {Mathematics} ⇒ {Physics} can be generated from the frequent itemsets. This rule would mean that if a student takes Mathematics then he is more likely to take Physics as well. The quality of these rules is measured using support, confidence and lift. The formulae for these are:

$$\text{Support}(X \Rightarrow Y) = \frac{\text{count}(X \cup Y)}{\text{total transactions}}$$

$$\text{Confidence}(X \Rightarrow Y) = \frac{\text{count}(X \cup Y)}{\text{count}(X)}$$

$$\text{Lift}(X \Rightarrow Y) = \frac{\text{Support}(X \Rightarrow Y)}{\text{Support}(X) \times \text{Support}(Y)}$$

or

$$\text{Lift}(X \Rightarrow Y) = \frac{\text{count}(X \cup Y)}{\text{count}(X) \times \text{count}(Y)} \text{total transactions}$$

The support is the percentage of transactions that contain the itemset. The confidence is the percentage of transactions that contain Y given that they contain X. The lift is the ratio of the observed support to that expected if X and Y were independent. The counts here comes from the frequent itemsets L1, L2, ..., Lk generated by the Apriori algorithm.

Example: For the given example, the support, confidence and lift for the rule {Mathematics} ⇒ {Physics} can be calculated as:

$$\text{Support}(\{Mathematics\} \Rightarrow \{Physics\}) = \frac{2}{3} = 0.67$$

$$\text{Confidence}(\{Mathematics\} \Rightarrow \{Physics\}) = \frac{2}{2} = 1$$

$$\text{Lift}(\{Mathematics\} \Rightarrow \{Physics\}) = \frac{0.67}{\frac{2}{3} \times \frac{2}{3}} = 1.5$$

2.5 Clustering

Clustering is a technique used in data analytics to find similar groups of data. K mean clustering is a popular clustering algorithm that is used to group data points into clusters. The number of clusters are pre defined in which the data will be divided. These points are randomnly slected or selected based on some criteria. These are called centroids. Then the distance between each point and the centroid is calculated based on distance formula defined. Each point is assigned to the closest cluster based on the minimum distance from the centroid. The centroid is recalculated based on the points that were assigned to that cluster using mean of the points. The process is repeated until the centroids do not change or the number of iterations are completed. The points in one cluster are grouped together and shows the similarity between the points in that cluster.

Example: For example the students can be grouped together based on their marks in different subjects. The marks can be used as the points and the students can be divided into clusters based on the marks. This can group similar students based on their academic performance. This type of data can be used for finding high achivers and students that require more attention. Given marks of 5 students in 2 subjects as:
1. {10, 20}
2. {15, 25}
3. {30, 40}
4. {35, 45}
5. {50, 60}

These will be called points. To divide the students into 2 clusters, the centroids can be randomly selected as:
1. {20, 30}
2. {40, 50}

The distance between each point and the centroid is calculated using Manhattan distance and the points are assigned to the closest cluster. Points 1, 2, and 3 are assigned to cluster 1 and points 4 and 5 are assigned to cluster 2. The new centroids are recalculated as:
1. {18.33, 28.33}
2. {42.5, 52.5}

The distance is calculated again and the points are assigned to the closest cluster. The process is repeated until the centroids do not change. After this iteration, the points remained in the same clusters and the final clusters are:
1. {10, 20}, {15, 25}, {30, 40}
2. {35, 45}, {50, 60}

This shows that first three students are similar in their marks and the last two students are similar in their marks.

2.6 Generative AI Tutoring

The type of tutoring expected from Generative AI here is to help in solving these practical exercise questions. For each of the topics of data science that are defined above a different approach is required to solve the exercises. Gemini, ChatGPT, SQL Fidelity, and Github Copilot were preliminarily tested for solving these questions. ChatGPT outperfomed by giving more logical and factually correct answers with detialed explanation as compared to other models. Generative AI takes input in the form of text and output is also given in a similar way. The questions need to be converted into the format that the model understands and process like required. The tutor explains the answers given in the solution and can also show the working. It can also guide the student in the right direction in case of error. The model can also be used to cross check the answers given by the students and provide feedback on the correctness of the answers. It can also be used to practice the exercises and learn the concepts in a better way by solving the exercises step by step with the model.

Chapter 3

Using ChatGPT for Relational Database Schema

Generative AI, especially ChatGPT, can help in generating relational databases. ChatGPT can take the tables along with fields in text format to generate SQL queries for data creation in SQL. It will also take primary and foreign keys and set up the relations among tables.

3.1 Database Schema

The database schema is given to ChatGPT to create SQL queries for schema creation. The example format to give this information to the AI model for University Schema is given below. It is of format table_name(attribute1, attribute2, attribute3, ...) - Primary Keys (primary_key1, primary_key2, ...) - Foreign Keys (foreign_key1[attribute1, attribute2], foreign_key2[attribute3, attribute4], ...) The primary keys of each table are written after the table with '-' as separator. The foreign keys are written after the primary keys with '-' as separator. The foreign keys define the table it is referring and the attributes in that table that will be mapped to the attributes in the current table. The prompt given to ChatGPT is as follows.

Create in SQL the University Schema with these details.
section (course_id, sec_id, semester, year, building, room_number, time_slot_id) - Primary Keys (course_id, sec_id, semester, year) - Foreign Keys (classroom[building, room_number], time_slot[time_slot_id])
classroom (building, room_number, capacity) - Primary Keys (building, room_number)
takes (ID, course_id, sec_id, semester, year, grade) - Primary Keys (ID, course_id, sec_id, semester, year) - Foreign Keys (section[course_id, sec_id, semester, year] , student[ID])
teaches (ID, course_id, sec_id, semester, year) - Primary Keys (ID, course_id, sec_id, semester, year) - Foreign Keys (section[course_id, sec_id, semester, year] , instructor[ID])

course (course_id, title, dept_name, credits) - Primary Keys (course_id) - Foreign Keys (department[dept_name])
prereq (course_id, prereq_id) - Primary Keys (prereq_id, course_id) - Foreign Keys (course[course_id], course[prereq_id])
student (ID, name, dept_name, tot_cred) - Primary Keys (ID) - Foreign Keys (department[dept_name])
department(dept_name, building, budget) - Primary Keys (dept_name)
instructor (ID, name, dept_name, salary) - Primary Keys (ID) - Foreign Keys (department[dept_name])
advisor (s_id, i_id) - Primary Keys (s_id) - Foreign Keys (instructor[i_id], student[s_id])
time_slot (time_slot_id, day, start_hr, start_min, end_hr, end_min) - Primary Keys (time_slot_id, day, start_hr, start_min)

As a result of this ChatGPT will return a list of SQL queries that can be executed to create a database with the given schema. As in this instance the ChatGPT responded with this

Here's the SQL schema for the university database based on the details provided:

```sql
-- Table: department
CREATE TABLE department (
    dept_name VARCHAR(50),
    building VARCHAR(50),
    budget DECIMAL(10, 2),
    PRIMARY KEY (dept_name)
);

-- Table: course
CREATE TABLE course (
    course_id VARCHAR(10),
    title VARCHAR(100),
    dept_name VARCHAR(50),
    credits INT,
    PRIMARY KEY (course_id),
    FOREIGN KEY (dept_name) REFERENCES department(dept_name)
);

-- Table: classroom
CREATE TABLE classroom (
    building VARCHAR(50),
    room_number VARCHAR(10),
    capacity INT,
    PRIMARY KEY (building, room_number)
```

```
);

-- Table: student
CREATE TABLE student (
    ID INT,
    name VARCHAR(100),
    dept_name VARCHAR(50),
    tot_cred INT,
    PRIMARY KEY (ID),
    FOREIGN KEY (dept_name) REFERENCES department(dept_name)
);

-- Table: instructor
CREATE TABLE instructor (
    ID INT,
    name VARCHAR(100),
    dept_name VARCHAR(50),
    salary DECIMAL(10, 2),
    PRIMARY KEY (ID),
    FOREIGN KEY (dept_name) REFERENCES department(dept_name)
);

-- Table: section
CREATE TABLE section (
    course_id VARCHAR(10),
    sec_id VARCHAR(10),
    semester VARCHAR(6),
    year INT,
    building VARCHAR(50),
    room_number VARCHAR(10),
    time_slot_id VARCHAR(10),
    PRIMARY KEY (course_id, sec_id, semester, year),
    FOREIGN KEY (course_id) REFERENCES course(course_id),
    FOREIGN KEY (building, room_number) REFERENCES classroom(
       building, room_number),
    FOREIGN KEY (time_slot_id) REFERENCES time_slot(time_slot_id)
);

-- Table: time_slot
CREATE TABLE time_slot (
    time_slot_id VARCHAR(10),
    day VARCHAR(10),
    start_hr INT,
    start_min INT,
    end_hr INT,
    end_min INT,
```

```
    PRIMARY KEY (time_slot_id, day, start_hr, start_min)
);

-- Table: prereq
CREATE TABLE prereq (
    course_id VARCHAR(10),
    prereq_id VARCHAR(10),
    PRIMARY KEY (course_id, prereq_id),
    FOREIGN KEY (course_id) REFERENCES course(course_id),
    FOREIGN KEY (prereq_id) REFERENCES course(course_id)
);

-- Table: takes
CREATE TABLE takes (
    ID INT,
    course_id VARCHAR(10),
    sec_id VARCHAR(10),
    semester VARCHAR(6),
    year INT,
    grade CHAR(2),
    PRIMARY KEY (ID, course_id, sec_id, semester, year),
    FOREIGN KEY (ID) REFERENCES student(ID),
    FOREIGN KEY (course_id, sec_id, semester, year) REFERENCES
       section(course_id, sec_id, semester, year)
);

-- Table: teaches
CREATE TABLE teaches (
    ID INT,
    course_id VARCHAR(10),
    sec_id VARCHAR(10),
    semester VARCHAR(6),
    year INT,
    PRIMARY KEY (ID, course_id, sec_id, semester, year),
    FOREIGN KEY (ID) REFERENCES instructor(ID),
    FOREIGN KEY (course_id, sec_id, semester, year) REFERENCES
       section(course_id, sec_id, semester, year)
);

-- Table: advisor
CREATE TABLE advisor (
    s_id INT,
    i_id INT,
    PRIMARY KEY (s_id),
    FOREIGN KEY (s_id) REFERENCES student(ID),
    FOREIGN KEY (i_id) REFERENCES instructor(ID)
```

```
);
```

Using this approach ChatGPT can be used to create SQL queries for schema that can be directly run in an SQL environment. It also takes into account the primary keys and foreign keys while creating the tables. If the primary keys are not explicitly defined then ChatGPT takes a decision on its own to either use an attribute as primary key or create a new attribute for primary key, depending upon the attributes in the table. The schema creation can also be invoked by giving an ERD to the ChatGPT. In this scenario the table creation is accurate but the primary keys if not defined explicitly will not be accurate.

Creating schema in this way, ChatGPT can tutor students on how to write SQL queries for creating tables and setting up relations among them. This also shows how the tables are created before their attributes can be referenced as foreign keys in other tables. Students can double check their work by comparing the SQL queries generated by ChatGPT with their own queries. They can also ask questions if they find something different in the queries generated by ChatGPT. In this way they can learn how to write Create queries with constraints and how to define primary and foreign keys in SQL. This is also beneficial for students as they can quickly create a schema for a database and start working on it. If students are required to work on writing queries then they can use this method to create a schema and then write queries on that schema. This will reduce the time taken to create a schema and they can focus on writing queries and understanding the working of the database.

3.2 Data Generation

The abilities of ChatGPT can also be used for data generation. Upon giving the db schema, ChatGPT can generate a sample data for that schema. This can be used for fast learning and easy setup. Upon only asking it to create the data, it creates a simplified data. To make a more logical data, specific instructions can be given to ChatGPT for that. One way of doing that is given below.

```
For this Employee Schema
Employee (id, person_name, street, city) - Primary Keys ( id)
Works (id, company_name, salary) - Primary Keys (id)
Company (company_name, city) - Primary Keys ( company_name)
Manages (id, manager_id) - Primary Keys (id)

Create data that has companies "First Bank Corporation", "Small Bank
     Corporation" and 4 more companies. Employees should earn more
    than $10,000 as well as less than that. "Small Bank Corporation"
    employees have less salary but not lesser then everyone.
    Companies maybe located in several cities, specially "Small Bank
    Corporation".
Create SQL queries for relations creation and data insertion.
```

Giving this prompt, ChatGPT first writes queries for creating the tables and then write queries for inserting data into the table according to the given conditions. It also gives the queries to verify the data that was inserted into the tables. The insertion part of the response after the create statements is as follows.

```
-- Inserting data into the Company table
INSERT INTO Company (company_name, city) VALUES
('First Bank Corporation', 'New York'),
('First Bank Corporation', 'Chicago'),
('Small Bank Corporation', 'Los Angeles'),
('Small Bank Corporation', 'New York'),
('Big Bank Corporation', 'Boston'),
('Medium Bank Corporation', 'San Francisco'),
('Tiny Bank Corporation', 'Dallas'),
('Large Bank Corporation', 'Miami');

-- Inserting data into the Employee table
INSERT INTO Employee (id, person_name, street, city) VALUES
(1, 'Alice Johnson', '123 Main St', 'New York'),
(2, 'Bob Smith', '456 Oak Ave', 'Chicago'),
(3, 'Carol White', '789 Pine Rd', 'Los Angeles'),
(4, 'David Green', '321 Elm St', 'San Francisco'),
(5, 'Eve Black', '654 Cedar St', 'Boston'),
(6, 'Frank Brown', '987 Birch Ln', 'Miami');

-- Inserting data into the Works table with salary conditions
-- Employees at 'Small Bank Corporation' have relatively lower
   salaries
-- Ensure that some employees earn more than $10,000 and some
   less
INSERT INTO Works (id, company_name, salary) VALUES
(1, 'First Bank Corporation', 12000),
(2, 'First Bank Corporation', 9500),
(3, 'Small Bank Corporation', 10500),
(4, 'Small Bank Corporation', 9800),
(5, 'Big Bank Corporation', 14000),
(6, 'Medium Bank Corporation', 9000);

-- Inserting data into the Manages table
-- Let's assume Alice manages Bob, and Carol manages David
INSERT INTO Manages (id, manager_id) VALUES
(2, 1),  -- Bob is managed by Alice
(4, 3);  -- David is managed by Carol
```

Although it is easier to generate data this way for learning purposes, it cannot be relied on for larger data sets. With the increase in the size of data and complexity, the data tends to become erroneous. The main things that it was not able to identify correctly were complex relationships among tables and attribute constraints. Even though it is not useful for large data sets, it can be used for learning purposes. Creating data instantly can be used for running queries and understanding the working of a database.

ChatGPT can act as a tutor for students to help them understand how to generate data for a database schema. This can be used to learn writing create and insert queries and understanding the working of a database. As a tutor ChatGPT can provide the students with this sample data that they can quickly create a working database with. The students can run their queries on this data for practice and learning purposes. This reduces the time needed to setup a database before students can get hands on experience with SQL queries. This can also be used to generate data for testing purposes. The data generated can be used to test the queries and the results obtained from the queries can be compared with the expected results. This can be used to validate the queries and the results obtained from them.

Chapter 4

Using ChatGPT for SQL Queries

The use of ChatGPT for writing, executing, and explaining queries has been explored. It can be used to facilitate writing queries for understanding or explanation of queries that are complex.

4.1 Query Writing

Query writing is an important skill in Data Science as it is used for the retrieval of data. Students are taught the structure of writing queries, and the focus is that they are able to extract diverse types of information from the database with ease. For finding information from a database, ChatGPT can write a query in SQL.
In order to get better results from ChatGPT and help it write better queries, the schema or structure of the table should be provided along with the description of what is required in the query. ChatGPT has the tendency to format the output column names differently from those found in the data science book, but the overall answer is correct for relatively simple queries.
It can work well with subqueries and multiple joins. The user can also ask it to convert a query from subqueries to joins and from joins to subqueries. It can be used to find the optimal query for faster results as well. If the information required is very complex, then ChatGPT can make mistakes. In order to avoid that, give a clear explanation of what is required along with the schema or table structure. It was tested on most of the queries from the data science textbook for different schemas. For example, attach a PDF containing SQL schema creation and data insertion queries and then ask the information needed from that database. Ask it to write a query like:

```
Write a query to Display a list of all instructors, showing each
   instructor's ID and the number of sections taught. Make sure to
   show the number of sections as 0 for instructors who have not
```

```
    taught any section. Your query should use an outer join, and
    should not use subqueries.
Write the query on University Schema and Data
```

The query given by ChatGPT along with the explanation is

```
SELECT I.ID, I.name, COUNT(T.course_id) AS number_of_sections
FROM instructor I
LEFT JOIN teaches T ON I.ID = T.ID
GROUP BY I.ID, I.name;
```

4.2 Query Execution

ChatGPT can also be used for its ability to act as a manual engine for SQL. It can execute a query manually step by step so the process of getting the output is explained. It is highly useful for students that struggle with the engine. By using this method they can understand how each part of the query contributed to the result and how the final result is computed.
For overall learning if the expected output of a given query over a limited set of data is required, it can be done by ChatGPT as well. Give the data to ChatGPT in the form of a table or SQL queries and as the output of the query and it will calculate it. If the query is complex then the output can be wrong. To avoid that ask for the steps of finding the answer and then if something is wrong, point it out and ChatGPT will correct it. So, this feature cannot be used for large scale but it can be used by students to learn in a faster and easier environment. For the above query ChatGPT explained the wokring of query and showed the result on sample data. The output is given below and the explanation is given in the Appendix B.

4.3 Query Explanation

ChatGPT can be used for explaining queries as well. This behavior is particularly useful for students that are new to the field and have difficulty understanding how a query is structured and why it is written in a specific way. Upon asking the explanation of query writing exercises in the textbook, ChatGPT acts as a tutor to explain each step in detail. A good prompt to ask for this is to give the statement in which what is required from the query is given along with the schema or giving the query along with the schema and asking ChatGPT to explain what this query is doing and how. The explaination of the query above provided by ChatGPT is as follows:

```
Explanation:
```

ID	Name	Number of Sections
10101	Srinivasan	3
12121	Wu	1
15151	Mozart	1
22222	Einstein	1
32343	El Said	1
33456	Gold	0
45565	Katz	2
58583	Califieri	0
76543	Singh	0
76766	Crick	2
83821	Brandt	3
98345	Kim	1

Table 4.1: Instructor Sections Data

```
LEFT JOIN: This ensures that all instructors from the instructor
   table are included in the result, even if they do not have
   matching records in the teaches table.

COUNT(T.course_id): Counts the number of sections each instructor
   has taught. If the instructor has not taught any sections, the
   count will be 0.

GROUP BY I.ID, I.name: Groups the results by the instructor's ID and
    name to aggregate the section count for each instructor.

This query will return a list of instructors with their ID, name,
   and the number of sections they have taught, with a count of 0
   for those who haven't taught any sections.
```

4.4 Error Resolution

One of the most common problems encountered during learning data science is getting errors. A student who is new to the field is not familiar with most of the errors that can occur while running queries. Currently, those issues are resolved by extensively searching the internet, websites like Stack Overflow and GeeksforGeeks. With generative AI, the time consumed to resolving such issues can be reduced. ChatGPT has proven to be useful in identifying the issues and giving an appropriate solution. In one example, it starts by explaining the reason for the error and then gives the corrected query. The error explaination is given below:

```
The error you're seeing, ERROR 1248 (42000): Every derived table
    must have its own alias, is occurring because you're using a
    subquery, and in SQL, every subquery (or derived table) must
    have an alias. In your case, you're using a subquery in the
   FROM clause but not providing an alias for it.
```

Chapter 5

Using ChatGPT for Normalization

Normalization is an important process in data science. It is used to make sure that data is formatted in a way that reduces redundancy and improves data integrity. It focuses on the dependency of data to divide attributes into relations that help achieve an organized database. The processes of normalization can be better understood and achieved using generative AI as well. ChatGPT was tested upon some of the major normalization steps to find the optimal way of acquiring its help. It has proven useful in solving the algorithms of normalization along with explanations.

5.1 Closure

Closure of a set means all the attributes that can be retrieved from the given set of attributes depending on the functional dependencies. The retrieved attributes are the ones that are dependent on the attributes. Upon exploring, ChatGPT proved useful in finding closure of attributes given the functional dependencies. The way for ChatGPT to give the closure of the given set over the set of functional dependencies is to give the high-level algorithm to the model to run the data on. For optimal results, the set notation is easier for ChatGPT to interpret as input. It also gives a step-by-step explanation of finding the closure. The algorithm given to ChatGPT for finding closure is in this form.

Algorithm 3: Closure of β under F

Input: Attribute set β, Functional Dependencies F
Output: Closure of β under F

$result \leftarrow \beta$;
repeat
  $result\_previous \leftarrow result$;
  foreach $\gamma \rightarrow \delta \in F$ **do**
    if $\gamma \subseteq result$ **then**
      $result \leftarrow result \cup \delta$;
    end
  end
until $result\_previous = result$;
return $result$;

The ChatGPT model is able to interpret this form of algorithm and find the solution based on this. The method of giving the data for solving is as:

```
Given a relation R(apple, mango, orange, grapes) and the set of
   functional dependencies
FD: {{grapes} → {orange}, {apple} → {grapes}, {mango} → {orange,
   grapes}, {apple} → {mango}}
Find the closure of {apple}
```

The closure of the given attribute was founded by ChatGPT step by step that can be used as a learning process. It guides on how the closure is found and what are the steps taken to find the closure. The result is also provided by ChatGPT along with the explanation of each step taken to find the closure. The response of ChatGPT is:

```
To find the closure of the attribute `{apple}` under the given
   set of functional dependencies (FDs), we'll iteratively apply
   the FDs starting with the attribute `{apple}` and keep
   adding attributes that can be inferred until no more can be
   added.

### Given Functional Dependencies:
1. {apple} → {mango}
2. {apple, orange} → {grapes}
3. {mango} → {orange, grapes}
4. {grapes} → {orange}

### Closure of `{apple}` (denoted as `{apple}+`):

1. **Start with `{apple}` in the closure**:
   `{apple}+ = {apple}`
```

```
2. **Apply FD 1: {apple} → {mango}**
   `{apple}+ = {apple, mango}`

3. **Apply FD 3: {mango} → {orange, grapes}**
   `{apple}+ = {apple, mango, orange, grapes}`

4. **Apply FD 4: {grapes} → {orange}**
   (Already included, no change.)

5. **Apply FD 2: {apple, orange} → {grapes}**
   (Already included, no change.)

### Final Closure of `{apple}`:
The closure of `{apple}` is:
    ```
    {apple, mango, orange, grapes}
    ```
```

5.2 Candidate Keys

Candidate keys are the attributes of combinations of attributes that can be used to uniquely identify each row. For each value of a candidate key, there will be only one dataset that exists. These are beneficial in normalization because they show the dependency of attributes. ChatGPT is able to detect new candidate keys upon giving functional dependencies, but it has the capacity to miss some candidate keys as well. ChatGPT is able to find if closure is not fulfilled, then it's not a candidate key, but it is unable to determine it with respect to minimal key. This can be overcome by giving it the step-by-step algorithm to find candidate keys.
Giving the whole algorithm of finding the candidate keys at once will make the model run it over without using the iterative. The best way to find candidate keys with the help of ChatGPT and learn during the process is to give it a small portion of the algorithm in one prompt and ask for the solution of that portion. Like asking for it to convert the attributes into 4 categories of those that are not in functional dependencies: Case1; that are neither on the left side nor on the right side of any functional dependencies, Case2; those that are only on the right side of functional dependencies, Case3; those only on the left side of functional dependencies, Case4, those on both sides of the functional dependencies.
Upon giving ChatGPT the functional dependencies and the algorithm for these 4 cases, it returns the attributes in each case. The algorithm for finding cases that can be given to ChatGPT is as given in Algorithm 4.

Algorithm 4: Candidate Keys Part 1

Input: Functional Dependencies (FD) of type $\alpha \rightarrow \beta$
Output: Attributes classified into cases 1, 2, 3, and 4

left $\leftarrow$ Union of all α in FD;
right $\leftarrow$ Union of all β in FD;
Initialize case_1, case_2, case_3, case_4 as empty sets;
foreach *x in attributes* **do**
  if $x \notin left$ ***and*** $x \notin right$ **then**
    case_1.add(x);
  if $x \notin left$ ***and*** $x \in right$ **then**
    case_2.add(x);
  if $x \in left$ ***and*** $x \notin right$ **then**
    case_3.add(x);
  if $x \in left$ ***and*** $x \in right$ **then**
    case_4.add(x);
return *attributes in case_1, case_2, case_3, case_4*;

The output sometimes contain the explanation of the steps taken to get the answer and sometimes it only contains the final result. In case only the final answer is given without explanation, the model can be asked to explain the steps taken to get the answer. The explanation of the steps taken to get answer for previous example over the algorithm 4 is as follows:

```
    Let's break down the output by analyzing the classification
       process step by step.

### **Step 1: Compute Left and Right Sets**
From the given functional dependencies:

1. **{apple} → {mango}**
2. **{apple, orange} → {grapes}**
3. **{mango} → {orange, grapes}**
4. **{grapes} → {orange}**

We extract:
- **Left set** (LHS of FDs): `{apple, mango, grapes, orange}`
- **Right set** (RHS of FDs): `{mango, grapes, orange}`

### **Step 2: Categorize Attributes**

We classify each attribute into one of the four cases based on
   whether it appears in the left set (LHS) and/or right set (RHS).
```

```
1. **Case 1 (Neither in left nor in right)**
   - Attributes that do not appear in either set.
   - Since all attributes `{apple, mango, orange, grapes}` appear in
      at least one of the sets, **this case is empty**.

2. **Case 2 (Only in right)**
   - Attributes that appear in RHS but not in LHS.
   - Every attribute in RHS (`{mango, grapes, orange}`) is also in
     LHS, **so this case is empty**.

3. **Case 3 (Only in left)**
   - Attributes that appear in LHS but not in RHS.
   - `{apple}` appears only in LHS and not in RHS, **so case 3
     contains `{apple}`**.

4. **Case 4 (In both left and right)**
   - Attributes that appear in both LHS and RHS.
   - `{mango, grapes, orange}` are present in both, **so case 4
     contains `{mango, grapes, orange}`**.

### **Final Output Interpretation**
    - **Case 1 (   ):** No attributes are completely independent.
    - **Case 2 (   ):** No attributes exist only as a dependent
      variable (i.e., only in RHS).
    - **Case 3 ({apple}):** "Apple" is only determining other
      attributes but is never determined by anything.
    - **Case 4 ({mango, grapes, orange}):** These attributes are
      both determinants (in LHS of some FDs) and dependents (in RHS
       of some FDs).

This classification helps analyze the role of each attribute in the
   dependency structure of the relation.
```

In this way, the AI tutor breaks down the working of first half of the algorithm and explain how the attributes are classified into the 4 cases. As a tutor it can help students work on the examples side by side to learn the process of finding candidate keys.

The remaining algorithm of finidng candidate keys is given in Algorithm 5.

Algorithm 5: Candidate Keys Part 2

Input: List of all attributes and attributes in Case 1, Case 3 and Case 4
Output: Candidate Keys

$core \leftarrow$ Attributes in Case 1 $\cup$ Case 3;
$\text{case}_4 \leftarrow$ Attributes in Case 4;
if $Closure(core) = FD$ **then**
  return $core$;
$candidate\_keys \leftarrow \emptyset$;
foreach $x \subseteq \mathcal{P}(case_4)$; `// Iterate over all subsets of Case 4`
do
  if $Closure(core \cup x) = FD$ **then**
    $candidate\_keys \leftarrow candidate\_keys \cup x$;
return $candidate\_keys$;

For the second part of the algorithm, the student can work together with AI tutor to find all candiate keys one by one. After initial division into 4 cases, ask ChatGPT to find the closure of core attributes (Case1 + Case3) and check if it is equal to the set of functional dependencies. If so, this is the candidate key; else, ask ChatGPT for the closure of $core \cup \{x | x \subseteq Case4\}$. Asking for closure of a limited number of subsets at a time will ensure the correctness of the result. The result can be improved by giving the algorithm of closure while asking for closure. The closure can be found in the similar way as defined in above subsection. At the end, the set of all attributes whose closure is equal to the set of functional dependencies can be combined to find the required set of candidate keys.
This approach is beneficial in education because it does not directly give the final result but provides students with the chance to calculate the solution step by step with the help of AI. This can also be used to cross-reference any work that the students are doing for finding candidate keys. ChatGPT also provides a detailed explanation of each step to help understand how the result is extracted.

5.3 Canonical Cover

The removal of extraneous attributes from the set of functional dependencies retrieves the canonical cover. The removal of extraneous attributes does not change the semantics of the functional dependencies. The algorithm of canonical cover combines together the functional dependencies that have the same left-hand side and then checks each attribute in the functional dependencies to remove the extraneous attributes. The canonical cover can be found using ChatGPT by breaking down the algorithm into two parts and iterating it until there is no change. The following algorithm is used to find the canonical cover.

Algorithm 6: Canonical Cover

Input: Functional Dependencies (FDs)
Output: Canonical Cover of FDs

result $\leftarrow$ FDs;
repeat
    prev_result $\leftarrow$ *result*;
    Replace all functional dependencies of the form $\alpha \rightarrow \beta_1$ and $\alpha \rightarrow \beta_2$ with $\alpha \rightarrow \beta_1 \cup \beta_2$;
    foreach *functional dependency f in result* **do**
        foreach *attribute in f* **do**
            if *attribute is extraneous* **then**
                Remove it from f;
            end
        end
    end
until *prev_result* $=$ *result*;
return *result*;

Inorder to find the canonical cover with the ChatGPT, first ask to merge all functional dependencies of type $\alpha \implies \beta_1$ and $\alpha \implies \beta_2$. After it's done, from the merged set of functional dependencies, ask by giving each attribute of the functional dependencies at each position if that is extraneous or not. Also give the formula for finding the extraneous attributes along with the attribute and the functional dependency in question. If there are any extraneous attributes then repeat the same steps until there are no more extraneous attributes. Then the final set of functional dependencies are the canonical cover.

The formula for finding extraneous attributes is given below
Given a functional dependency $\alpha \rightarrow \beta$:

- An attribute $a \in \alpha$ is **extraneous** if $\beta \subseteq (\alpha - \{a\})^+$ under F.
- An attribute $a \in \beta$ is **extraneous** if $a \in \alpha^+$ under $F' = (F - \{\alpha \rightarrow \beta\}) \cup \{\alpha \rightarrow (\beta - \{a\})\}$.

The students can use this approach to learn the process of finding the canonical cover in the same way as finding the candidate keys. ChatGPT can act as a tutor helping students find canonical cover step by step and explaining any doubts they have. Similar to other topics, the students can also cross refernce their work and practice such questions with the help of ChatGPT. Trying the same example to check if apple is extraneous in the functional dependencies given above. ChatGPT shows detailed processing and explanation of the steps taken to find the extraneous attributes. The response of ChatGPT is:

```
    Yes, **{orange} is an extraneous attribute** in the left-hand
       side of the functional dependency **{apple, orange} → {grapes
       }**.

### **Reasoning:**
1. We check whether **{grapes}** is still functionally determined if
     we remove **{orange}** from the LHS.
2. Compute **Closure({apple})** under the given functional
   dependencies:

   - Start with **{apple}**
   - **{apple} → {mango}**, so add **{mango}**
   - **{mango} → {orange, grapes}**, so add **{orange, grapes}**
   - **{grapes} → {orange}** is redundant since **orange** is
      already in the set.

   **Closure({apple}) = {apple, mango, orange, grapes}**

3. Since **{grapes}** is already in **Closure({apple})**, it means
   that **{orange}** was not necessary in **{apple, orange} → {
   grapes}**.

Thus, **{orange} is extraneous in the LHS** of the functional
   dependency **{apple, orange} → {grapes}**, and the dependency can
     be simplified to **{apple} → {grapes}**.
```

In this way the AI tutor can help students find the extraneous attributes in the functional dependencies. The students can then remove these extraneous attibutes and ask again for another attribute. The removing process and combining functional dependencies with same left hand side can also be done with the help of AI tutor. This process can be repeated until there are no more extraneous attributes left. The final set of functional dependencies will be the canonical cover. Going through each step with guidance can improve the learning process of the students.

5.4 Structuring Functional Dependencies for Clarity

The best way to structure functional dependencies in a way that is most easily interpretable by ChatGPT is using the set notation. The interpretation is found to be more accurate when words are used instead of alphabets for attributes of the functional dependencies. The arrow can be either $->$ or $=>$ for defining the functional dependencies. One example of the format in which the functional dependencies are given to ChatGPT that is comprehendible by it is given below.

```
For Relation R(apple, mango, orange, grapes, peach) and Functional
    dependencies FD: { {grapes} → {orange, peach} , {apple} → {grapes
    }, {mango} → {orange, grapes}, , {apple} → {mango}, {peach} → {
    apple}} give all the candidate keys
```

5.4.1 Renaming

In order to give ChatGPT the functional dependencies containing words as attributes instead of alphabets, it needs to convert the functional dependencies. This can be done with the help of ChatGPT as well. Upon giving ChatGPT the functional dependency in the normal alphabet form, it can be asked to convert it into meaningful words. The model will give the mapping of alphabets to words. This converted set of functional dependencies can then be given to ChatGPT for further processing. A sample prompt to ask ChatGPT for conversion is

```
rename attributes in this relation and functional dependencies to
    meaningful words

     r5 ( A ,B ,C ,D ,E ,F);
     F  = {
     {C, E} → {A},
     {C} → {D},
     {A} → {B},
     {D} → {B, E},
     {B} → {F},
     {A, D} → {C, F}
     }
```

ChatGPT converts the attributes given in the functional dependencies to meaningful words. These can not only help the AI tutor understand the question better but also helps the students comprehend the functional dependencies in a better way. The mapping of ChatGPT along with the updated functional dependencies is given below:

- A = OrderID
- B = CustomerID
- C = ProductID
- D = ShippingID
- E = PaymentID
- F = DeliveryDate

```
Renamed Functional Dependencies (F)
    {ProductID, PaymentID} → {OrderID}
    Meaning: A combination of product and payment information
       uniquely identifies an order.

    {ProductID} → {ShippingID}
    Meaning: A product determines the shipping ID for its shipment.

    {OrderID} → {CustomerID}
    Meaning: An order uniquely identifies the customer who placed it
       .

    {ShippingID} → {CustomerID, PaymentID}
    Meaning: The shipping information identifies the customer and
       their payment details.

    {CustomerID} → {DeliveryDate}
    Meaning: The customer determines the delivery date for their
       orders.

    {OrderID, ShippingID} → {ProductID, DeliveryDate}
    Meaning: A combination of order and shipping details determines
       the product being shipped and its delivery date.
```

5.5 Code Vs Descriptive Solution

ChatGPT can provide both descriptive manually computed answers as well as answers after running the values through code. For learning, the descriptive answers are more interactive and explain each step taken. It is easier to understand how the process works through a manual process. If only the answer is required, like if the students want to match the answer they found themselves at the end, then the solution through code can be used.
If the type of solution required is not specified, then ChatGPT decides on its own which mode is to be used. If the computation is long, then the code form is used; else, the descriptive form is used. If a specific type of solution is required, then it can be added to the prompt, and the answer will be in the required format. For getting the final result, a high-level algorithm like the one described above will be given to ChatGPT and asked to give the final result after running it through code.

Chapter 6

Using ChatGPT for Itemset Mining

Itemset is used to find frequent itemsets in given data. The process of finding frequent itemsets given the transaction through Apriori algorithm was explored with the help of ChatGPT. Each transactions contains the set of items that occur together, for example, the grocery items purchased together in a single transaction. Trying to find all frequent itemsets in a single prompt is not possible as the model is not able to keep track of all the items and their counts. If the model is required to solve the complete process then there are chances of incorrect results. In this case ChatGPT uses a method similar to Apriori but without sorting and the results can vary. Inorder to achieve better results the ChatGPT model can be used like a tutor that helps solve the exercise together. The whole process can be broken down into smaller steps and the model can be asked to solve each step one by one. The working of the ChatGPT as a tutor is shown with the help of a similar example of finding frequent itemsets using Apriori algorithm. The list of transactions re given below:

- {Orange}, {Apple}, {Mango}
- {Mango}, {Grapes}, {Orange}, {Apple}
- {Orange}, {Grapes}, {Apple}
- {Apple}, {Grapes}
- {Orange}, {Mango}, {Apple}, {Grapes}
- {Mango}, {Orange}

6.1 Lexicographical Sorting

Apriori algorithm works on the principle of lexicographically sorted items in each transaction. Without any guidance the model does not sort the items and the result can be incorrect. The first step is to get the items in each transaction sorted lexicographically. In this way the AI is able to use the items lexiographically throughout the process. For this simple prompt with the list of items in each transaction is given to the model and it is asked to sort the items in each transaction lexicographically. The result of this step is then used in the next step to find the frequent itemsets. The example of ChatGPT response for above example is given below:

```
Here are the transactions with items sorted lexicographically:

    1 - {Apple}, {Mango}, {Orange}
    2 - {Apple}, {Grapes}, {Mango}, {Orange}
    3 - {Apple}, {Grapes}, {Orange}
    4 - {Apple}, {Grapes}
    5 - {Apple}, {Grapes}, {Mango}, {Orange}
    6 - {Mango}, {Orange}

Let me know if you need further modifications!
```

6.2 Create Ck

The next step is to ask about the unique items that exists in all the transactions. This will be C1 (Candidate 1-itemsets). For the first step C1 is simple and is the list of all unique items in all transactions. After this L1 is found by removing some of the items from C1 that are below the minimum support. In the next step C2 is generated from L1 and the process is repeated to create Ck until no more Ck can be created from L(k-1). After C1, generating Ck can be tricky as the model needs to understand the process of generating Ck from L(k-1). This is a complex statement and is difficult for model to understand. There are two ways to get this step done:

- Give the model the complete algorithm of generating CK and ask it to generate CK for the given Lk.

```
    Create Ck from L(k-1) such that Ck = {x| a      b & a1=b1,
       a2=b2, a3=b3, ..., a(k-2)=b(k-1), a(k-1)<b(k-1) where a
           L(k-1) and b      L(k-1)}
```

- Give the model the complete algorithm of generating CK and ask it to generate CK for the given Lk step by step. For example, (A, B) is in L(k-1), then create all candidate itemsets Ck that contains (A,B) by:

```
give all combinations of X = (A, B) with all other 2-
    itemsets Y such that X[:-1] = Y[:-1] & X[-1] < Y[-1] and
     return the union of each valid combination
```

In this way for one Ck the prompts will be given one by one for each itemset in L(k-1).

The prune step can then be used to remove any itemsets, the subset of which are not present in L(k-1). Using the above example the C1 is create by ChatGPT is C1 = Apple, Grapes, Mango, Orange. After creating L1, the C2 is created by ChatGPT as follows:

```
The Candidate 2-itemset (C2) is:

    C2 = { (Apple, Grapes), (Apple, Mango), (Apple, Orange), (Grapes
        , Mango), (Grapes, Orange), (Mango, Orange) }
```

The students can check their manual work to see if their working is correct. The model can be used as a tutor to help solve the exercises and learn the process of finding frequent itemsets. Sometimes an itemset can be missed so working on problems with model can help catch those mistakes. Even if the model makes a mistake, the student can point it out and the model can correct it. This way the student will look into the possible mistakes and learn from them.

6.3 Calculate Lk

Then model can be asked to count the occurance of each item in all transactions, if the list of items is too long it is better to divide it into smaller parts. Ask the model to find the frequent 1-itemsets (L1) by keeping only the items that are above the minimum support. Then ask the model to generate CK as shown in the previous step. Using that Ck, Lk is created by counting the occurance of each itemset in Ck and keeping only those that are above the minimum support. This process is repeated until there are mo more Ck that can be generated. The ChatGPT can then combine all the frequent itemsets to find the final result. The example of ChatGPT response for finding L2 for above example is given below:

```
The L2 (Frequent 2-itemset) is:

    L2 = { (Apple, Grapes, 4), (Apple, Mango, 3), (Apple, Orange, 4)
        , (Grapes, Orange, 3), (Mango, Orange, 4) }
```

ChatGPT can also help determine the stopping criteria for the Apriori algorithm. The stopping criteria is when no more Ck can be generated from L(k-1). The model can be asked if the current step is the stopping criteria or not.

The accuracy of the results is provided in the evaluation section. In this way the model can act as a tutor to help solve exercises so the student can work along the model to learn the process of finding frequent itemsets. The students can use this way of learning to facilitate their learning process and doing better practice for exams as well.

6.4 Association Rules

The association rules are generated from the frequent itemsets. Generative AI can also be used in finding these rules from the data by calculating confidence and support. The support, confidencfe and lift can be calculated with the help of Generative AI. Incase of ChatGPT OpenAI model 4o the model already knows the formulas for these and can calculate them easily. Calculating these values for all association rules altogether can cause the model to skip a few rule. One way to get the results is to ask the model to calculate the values for each rule one by one. Another ways is to ask the model to create association rules for one frequent itemsets at a time, it does so by calculating confidence and if confidence is above a threshold then include it in the association rules. In this way the model can be used by students to learn the process of generating association rules from frequent itemsets and validating them using support, confidence and lift. For finding the confidence, support and lift for the association rule Orange -¿ Apple the ChatGPT response is as follows:

```
We need to compute **Confidence, Support, and Lift** for the
    association rule:

{Orange} → {Apple}

using the given itemsets and their counts.

---

### **Step 1: Calculate Support**
Support of a rule ( X → Y ) is:


Support(X → Y) = Support(X∪Y) / TotalTransactions


From the given data:
- ( Support({Apple, Orange}) = 4 )
- Total Transactions = 6


Support({Orange} → {Apple}) = 4/6 = 0.6667 ( or 66.67
```

—

```
### **Step 2: Calculate Confidence**
Confidence of a rule ( X → Y ) is:
```

Confidence(X → Y) = $\frac{Support(X \cup Y)}{Support(X)}$

```
From the given data:
- ( Support({Orange}) = 5 )
- ( Support({Apple, Orange}) = 4 )
```

Confidence({Orange} → {Apple}) = $\frac{4}{5}$ = 0.8 (or 80%)

—

```
### **Step 3: Calculate Lift**
Lift measures the strength of association relative to independence:
```

Lift(X → Y) = $\frac{Confidence(X->Y)}{Support(Y)}$

```
From the given data:
- ( Support({Apple}) = 5 )
- ( Confidence({Orange} → {Apple}) = 0.8 )
- ( Support({Apple}) = 5/6 )
```

Lift({Orange} → {Apple}) = $\frac{0.8}{5/6} = \frac{0.8 \times 6}{5} = \frac{4.8}{5} = 0.96$

```
---

### **Final Results**
- **Support:** **66.67% (0.6667)**
- **Confidence:** **80% (0.8)**
- **Lift:** **0.96**

Since Lift < 1, the presence of **Orange** slightly decreases the
   likelihood of **Apple**, meaning they are negatively correlated.
```

In this way the ChatGPT can teach the students how to generate association rules from frequent itemsets. The students can also learn how to validate the rules using support, confidence and lift. The overall learning process can get some benefit from using the tutor this way for solving these practical exercise questions.

Chapter 7

Clustering

Clustering is the process of grouping similar data points together. Using generative AI for clustering small number of data points can be beneficial for educational purposes. The questions related to clustering can be solved using generative AI. K-mean clustering was tested and optimal ways for finding the answers were explored. Like other topics, working on the problem step by step is the best way to get the correct results. The model can be used as a tutor to help solve the exercises. The process of clustering is iterative and asking for each iteration separately improves the accuracy of results. Breaking down the steps of each iteration helps the model to understand the problem better. Using the example of customers who bought items (from above example) together to group them together. The count of items bought by each customer is given below:

Customer	**Apple**	**Orange**
c1	1	1
c2	4	0
c3	5	6
c4	3	3
c5	9	8
c6	7	2

Table 7.1: Items bought by each customer

Centroids:

- Centroid 1: (1, 3)
- Centroid 2: (4, 0)

For this example each item will be used as a dimension and the distance will be calculated considering the count as the value of the dimension. The model can be asked to find the distance of each point from the centroids and assign to the cluster with shortest distance. The result of the first iteration is given below:

```
The points assigned to Cluster 1 are (1,1) and (3,3).
The new centroid for Cluster 1, calculated as the mean of these
   points, is (2.0, 2.0).

The points assigned to Cluster 2 are (4,0), (5,6), (9,8), (7,2).
The new centroid for Cluster 2, calculated as the mean of these
   points, is (6.25, 4.0).
```

The model can also be asked to find the new centroids by taking the mean of all points in the cluster. If no point is assigned to a cluster, the model randomly selects a new centroid to ensure that the clustering process continues. These steps can be repeated until the centroids do not change. The model can also be asked if this is the right time to stop the iterations. The model does not always shows the working and calculations but can be asked to show the working for better understanding.

After 2 more steps the final clustering of the customers is given below:

```
Cluster 1: c1, c2, c4
Cluster 2: c3, c5, c6
```

This shows that the customers c1, c2 and c4 bought similar items and are grouped together. Similarly customers c3, c5 and c6 are grouped together. Using this approach the students can use the ChatGPT as a tutor to improving their understanding of clustering and the K-mean clustering algorithm. The step by step approach can be used for faster learning as the steps that are easier but time consuming can be done by the model and the students can focus on the more complex steps. This AI tutor can also be used to solve the exercises in the text book that are difficult to understand. Unlike the traditional tutor, the AI tutor can be used at any time and can be asked to solve the same problem multiple times. If a student is stuck at a specific step, the model can be asked to solve that step and then the student can continue from there. This way the student can learn at their own pace and can get help whenever needed. For clustering also, the students can learn the process of data analyses by working with the model.

Chapter 8

Evaluation

In this study, 3 different methods were used for evaluation of the responses of generative AI. These methods determined the quality of answers given by ChatGPT with respect to learning perspective.

One of the methods used for evaluation was carrying out an experimental study with participants. In this study the participants were asked to solve a few practical exercise questions manually and then using ChatGPT. Their responses and feedback were recorded and analysed.

The second method used empirical evaluation to test the responses of ChatGPT against questions of data science. An exploratory analysis was done to see how ChatGPT responds to different types of questions. The correctness and clarity of the responses were evaluated along with the complexity of the structure of prompts needed to get the correct answers.

The third method used Claude to test the ability of ChatGPT to act as a tutor for students learning data science. Claude was given the task of evaluating the responses ChatGPT gave, as a tutor, to the questions of data science. The ratings given by Claude were used as a measure of the effectiveness of ChatGPT as a tutor for data science.

Details of these are given in the following sections.

8.1 Experimental Study

A study was conducted to evaluate the effectiveness of using ChatGPT as a tutor for learning data science topics. Participants were asked to answer a series of question using their previous knowledge and with the help of ChatGPT. They were not given any guidance on how to use ChatGPT and were given a reference sheet with basic concepts and methods to answer the questions manually. The questionnaire was divided into 5 sections. The first section was about the background of the participants, the second section was about using the participant's previous knowledge of the topic to answer the basic questions, the third section was about using the ChatGPT to answer the same basic questions, the fourth section was about using the ChatGPT to answer the advanced questions, and the fifth section was about the experience of the participants using the ChatGPT and their feedback. It also evaluated how participants from different degree programs and education levels, benefited from ChatGPT in learning the topic. The complete questionnaire is provided in Appendix A. The questions given to the participants to solve are as follows:

Basic - Using Previous Knowledge

The following questions are to be answered without the help of Generative AI but the participants were given a reference sheet with basic concepts and methods to answer the questions manually. The reference sheet is also provided in the Appendix A.

1. Write a single SQL Query to find the id of each employee who works for "First Bank Corporation" and earns more than $10000. Use this Schema:
    - Employee (id, person_name, street, city) - Primary Keys (id)
    - Works (id, company_name, salary) - Primary Keys (id)
    - Company (company_name, city) - Primary Keys (company_name)
    - Manages (id, manager_id) - Primary Keys (id)

    Show your work.

    *The expected answer is

```sql
SELECT W.id
FROM Works W
WHERE W.company_name = 'First Bank Corporation'
AND W.salary > 10000;
```

2. Find the Closure of {BookID} from the set of functional dependencies:

$$Books(BookID, AuthorName, Genre, Publisher)$$

{ {BookID } → {AuthorName, Genre},
{AuthorName } → {Genre},
{Publisher } → {BookID},
{BookID, AuthorName } → {Publisher}}.

Show your work.

*The expected answer is {BookID, AuthorName, Genre, Publisher}

3. Find the Candidate Keys of the set of functional dependencies:

$$Books(BookID, AuthorName, Genre, Publisher)$$

{ {BookID } → {AuthorName, Genre},
{AuthorName } → {Genre},
{Publisher } → {BookID},
{BookID, AuthorName } → {Publisher}}.

Show your work.

*The expected answer is {BookID}, {Publisher}

Basic - Using ChatGPT

Participants were asked to answer the same questions with the help of ChatGPT. A basic guidance sheet was provided to the participants on how to use ChatGPT for tutoring, incase they get stuck. The guidance sheet is also provided in the Appendix A.

Write a single SQL query to find the id of each employee who works for "First Bank Corporation" and earns more than $10000. Use this Schema:

- Employee (id, person_name, street, city) - Primary Keys (id)
- Works (id, company_name, salary) - Primary Keys (id)
- Company (company_name, city) - Primary Keys (company_name)
- Manages (id, manager_id) - Primary Keys (id)

Show your work. You can copy and paste from the AI tutor if that is useful.
*The expected answer is

```
SELECT W.id
FROM Works W
WHERE W.company_name = 'First Bank Corporation'
AND W.salary > 10000;
```

Provide the session URL for the above question. (Session URL)

Find the Closure of {BookID} from the set of functional dependencies:

$$Books(BookID, AuthorName, Genre, Publisher)$$

{ {BookID } → {AuthorName, Genre},
{AuthorName } → {Genre},
{Publisher } → {BookID},

{BookID, AuthorName } → {Publisher}}.

Show your work. You can copy and paste from the AI tutor if that is useful.
*The expected answer is {BookID, AuthorName, Genre, Publisher}

Provide the session URL for the above question. (Session URL)

Find the Candidate Keys of the set of functional dependencies:

$$Books(BookID, AuthorName, Genre, Publisher)$$

{ {BookID } → {AuthorName, Genre},
{AuthorName } → {Genre},
{Publisher } → {BookID},
{BookID, AuthorName } → {Publisher}}.

Show your work. You can copy and paste from the AI tutor if that is useful.
*The expected answer is {BookID}, {Publisher}

Provide the session URL for the above question. (Session URL)

Advanced - Using ChatGPT

Participants were asked to answer more advanced questions with the help of ChatGPT.

1. Write a single SQL Query to find the id of each employee who earned more than every employee of "Small Bank Corporation". Use this Schema:
   - Employee (id, person_name, street, city) - Primary Keys (id)
   - Works (id, company_name, salary) - Primary Keys (id)
   - Company (company_name, city) - Primary Keys (company_name)
   - Manages (id, manager_id) - Primary Keys (id)

   Show your work. You can copy and paste from the AI tutor if that is useful.

   *The expected answer is

```
SELECT W1.id
FROM Works W1
WHERE W1.salary > ALL(
    SELECT W2.salary
    FROM Works W2
    WHERE W2.company_name = 'Small Bank Corporation'
);
```

2. Provide the session URL for the above question. (Session URL)

3. Find the Candidate Keys of the set of functional dependencies:

$$Class(StudentID, CourseID, Semester, Instructor)$$

{ {CourseID } → {StudentID, Semester},
{Semester } → {Instructor},
{StudentID, Instructor } → {CourseID},
{StudentID, Semester } → {CourseID, Instructor}}.

Show your work. You can copy and paste from the AI tutor if that is useful.

*The expected answer is {CourseID}, {StudentID, Semester}, {StudentID, Instructor}

4. Provide the session URL for the above question. (Session URL)

5. Find the Candidate Keys of the set of functional dependencies:

$$Inventory(ProductCode, ProductName, Category, Supplier, WarehouseLocation, StockQuantity)$$

{{ProductCode } → {ProductName}
{ProductCode } → {WarehouseLocation}
{Category } → {ProductName, StockQuantity}
{ProductName, Category } → {StockQuantity}
{ProductName } → {ProductCode, Supplier}
{Supplier, StockQuantity } → {Category, WarehouseLocation}}

Show your work. You can copy and paste from the AI tutor if that is useful.

*The expected answer is {Category}, {ProductCode, StockQuantity}, {ProductName, StockQuantity}, {Supplier, StockQuantity}

6. Provide the session URL for the above question. (Session URL)

8.1.1 Participants' Feedback

Participants were asked to provide feedback on their experience using ChatGPT in this study. They were asked their opinions on ChatGPT's helpfulness as a data science tutor. They were also given the opportunity to provide any additional feedback or comments.

8.1.2 Result of Experimental Study

The participants were recruited from the Rochester Institute of Technology. The call for participation was sent out through email to the students with background in computer science, data science or related fields. Most of the participants were graduate students who had taken courses in databases and data science. However, some participants were not familiar with SQL and normalization. Specifically, 9.1% of participants were not familiar with SQL, 22.7% were somewhat familiar, and the rest were familiar with SQL. The total number of participants was 22 and an overview of the subject that they have studied is provided in the figure 8.1. This shows that most of the participants have studied databases and are familiar with a few concepts. The confidence of participants in SQL was normally distributed across likert scale between not confident at all to extremely confident. The confidence of participants in normalization was skewed towards lower confidence. This shows that most participants had basic knowledge of SQL but were not so confident in normalization.

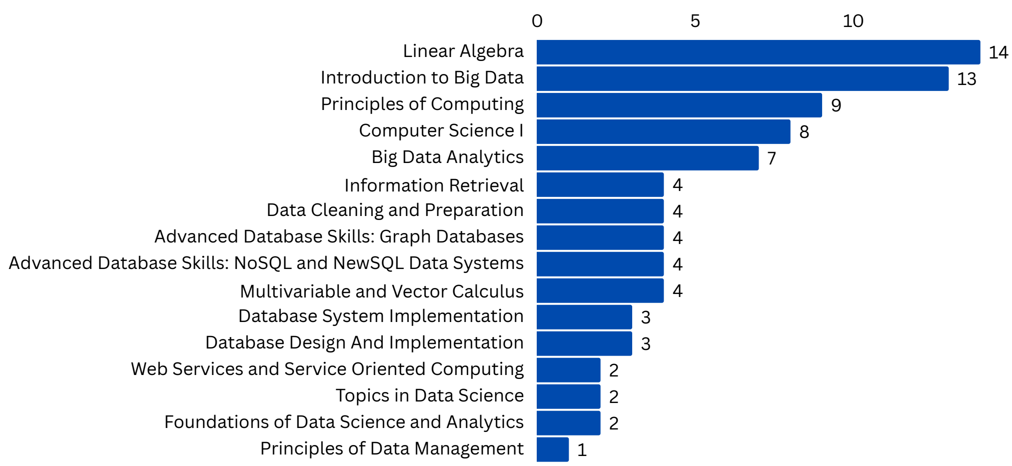
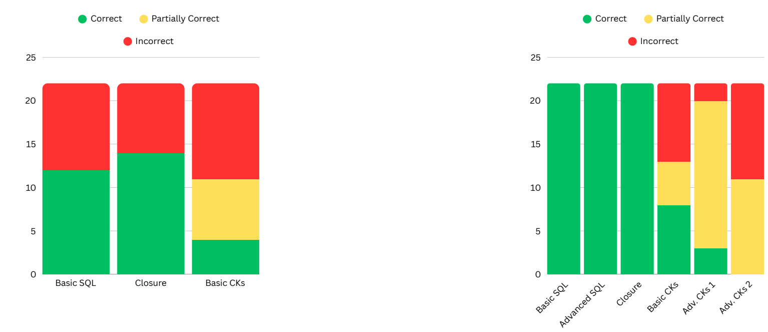


Figure 8.1: Background subjects of participants

The results of the study showed that ChatGPT was able to answer the questions correctly for SQL queries and simpler normalization questions. The answers provided by

ChatGPT were accurate for these topics. The exact statements of the answers differed, but the answers were correct for simple topics. Some SQL queries contained an additional join, which ChatGPT explained as a validation factor. While it was not the most optimal, it was still a correct answer. For advanced topics, all of the participants were able to solve SQL queries as well with ChatGPT. When students were asked to write SQL queries on their own with the help of a reference sheet and their knowledge, only 54.5% of the participants were able to solve it for basic SQL. With ChatGPT, all those participants, even those that were unfamiliar with SQL were able to write correct queries for both basic SQL and advanced SQL.

63.6% participants were able to solve the basic closure question on their own with the help of a reference sheet and their previous knowledge. With ChatGPT, all the participants were able to solve the basic closure question. This shows that the accuracy of answers for basic questions like closure and SQL queries that are not extremely complex is almost 100% with the help of ChatGPT according to these results. The participants were not given much guidance on how to use ChatGPT and direct their prompts. It was an observational study to see how participants use ChatGPT and how it helps them in solving the questions. Since participants were given a free hand, and all the participants wrote correct answers for SQL and closure questions, it can be said that for these type of questions very structured prompts are not required.



(a) Histogram of questions solved manually

(b) Histogram of questions solved with ChatGPT

Figure 8.2: Histogram of questions solved manually and with ChatGPT

However, finding candidate keys was somewhat challenging for the participants. When the participants were asked to find basic candidate keys manually, only 18.2% of the participants were able to find all candidate keys. When the participants were given the option to use ChatGPT, only 36.4% of the participants were able to find complete correct answer. This shows that the percentage of correct answers increased when using ChatGPT but still a large percentage of participants were not able to find the complete correct answer. Some of the participants also found some of the candidate keys but not all. In the manual attempt 31.8% of the participants found the partial answer and with ChatGPT 22.7% of the participants found the partial answer.

For advanced questions of candidate keys, the first question was relatively less complex as compared to the second question. The first advanced question was solve correctly by only 13.6%, partially solved by 77.3% and incorrectly solved by 9.1% of the participants. The correct answer for this contained 3 candidate keys. This required multiple iterations to find all the candidate keys. When multiple steps are required ChatGPT can skip some steps and provide a partial answer. This shows why most of the participants only got a partial answer. The second advanced question of candidate keys required 4 candidate keys to be found. This question was not answered completely correctly by any of the participants. 50% of the participants were able to find partial answer and 50% of the participants answered incorrectly. This shows that the complexity of the question also impacts the ability of the participants to find the correct answer. As the complexity of the question increases, the rate of incorrect answers also increases. Generative AI can assist as a tutor and work alongside learners but cannot serve as the sole source of learning. This also shows that some guidance is needed as the tasks become more complex.The rate of incorrect answers is also higher for basic candidate keys question while the rate of partial answers is higher for advanced questions. The histogram of the results is provided in the figure 8.2. It is found that the rate of correct answers increased when using ChatGPT as compared to solving questions manually.

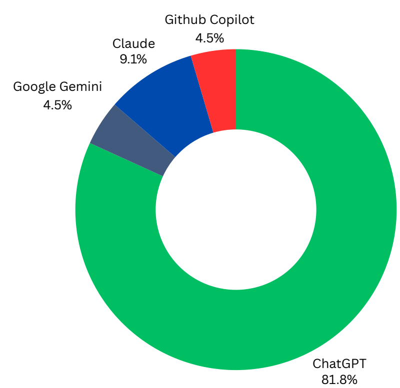


Figure 8.3: Preferred generative AI tool

It was also found that majority of the participants preferred ChatGPT over other Generative AI models, shown in figure 8.3. Those who preferred other models have also said that they have used ChatGPT. Showing that it is the most commonly used tool among college students for learning purposes.

The fluency of the participants also impacted their ability to yield correct answers. The participants who were more fluent in ChatGPT were able to achieve better results as compared to those who were not fluent in ChatGPT. For basic candidate keys question the participants who were able to solve correctly with ChatGPT were also fluent in ChatGPT. Among the participants who were able to find complete correct result for advanced candidate keys question 1 are only those who were almost fluent and very fluent in ChatGPT.

The feedback of the participants was also collected on their experience and the results are mostly positive. For the question "How likely are you to use ChatGPT as a Data Science tutor in the future?" 72% of the participants said that they are somewhat likely or extremely likely to use ChatGPT as a Data Science tutor in the future. This is shown in figure 8.4. This shows that the participants found ChatGPT helpful in solving the questions and are likely to use it in future.

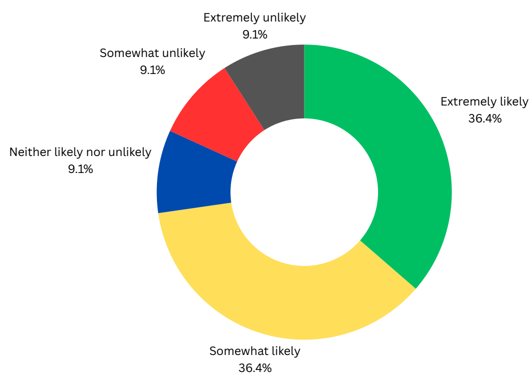


Figure 8.4: Likelihood of using ChatGPT as a Data Science tutor in future

8.2 Empirical Evaluation

A detailed evaluation of the responses generated by ChatGPT is essential to understand the quality of the answers. ChatGPT was given prompts of different types of questions related to Data Science and the responses were evaluated based on the correctness of the answers. In most cases, the answers were incorrect when the model was asked to solve complex problems, such as normalization, itemset mining, and clustering, in one go. The model performed better when these problems were broken into smaller steps. The model performed better when the problem was broken down into smaller steps and the model was asked to solve each step one by one. Some questions were solvable with simple commands and layman terminology, while others required complex commands and advanced algorithms. This was analyzed by checking the correctness of answers provided by ChatGPT and also by the complexity of the prompts needed to get these results. The details are given in the following sections.

8.2.1 Accuracy of ChatGPT Responses

The model was explored to find the ways that worked best to solve a specific type of questions. The accuracy of the responses given by ChatGPT using structured prompts is provided in the table 8.1. Here the unit for count is one complete process of calculation of the specific step. For example, if distance is mentioned once in clustering then it means that the distance of all points from all centroids is calculated once. The unit here is not the number of prompts but the number of times a specific step is calculated. In case of itemset mining the calculation of Lk is considered as one unit but it may take multiple prompts to get Lk. The total numbers given in the table correspond to the number of times the model was asked to solve a specific step. It counts the number of units of solution provided or asked by ChatGPT. The number of steps calculated without any correction are the ones that were calculated in the first attempt and were correct, these are given as "Worked" in the table. The number of steps worked with correction are the ones that were corrected after being pointed out by the user or by asking again without pointing out the mistake or misunderstanding. These are given as "Worked with Correction" in the table. The number of steps that did not work are the ones that were incorrect and could not be corrected. These are the ones that could not be solved even after multiple attempts. These are given as "Did not Work" in the table.

8.1 shows the result of total units of solution provided or asked by ChatGPT. More broken down results for each topic are provided in their respective tables.

As per 8.2, ChatGPT was able to create large datasets for populating the tables but that data did not make much sense. It had information like student 1, student 2, student 3, and so on instead of proper names. For basic working understanding, this approach can be used a little to create large sclae data but not so beneficial for real world applications. For creating data, the unit for measurement is the number of times the data for complete

Topic	**W**	**WC**	**N**	**Total**	**Percentage**
Relational SQL Database Creation	14	2	0	16	87.5
SQL Queries	100	6	2	108	92.6
Normalization	60	12	1	74	81
Association Rules	54	6	0	60	90
Clustering	58	2	0	60	96.66

Table 8.1: Combined Performance Evaluation Table

Note: W = Worked, WC = Worked with Correction, N = Did not Work

schema was created or complete data for a new table was created. For creating database, the unit is the number of times the complete database schema was created or a new table was added in the existing database.

Topic	**W**	**WC**	**N**	**Total**	**Percentage**
Creating Relational Databases From Schema	6	0	0	6	100
Creating Sample Data for Tables	6	2	0	8	75
Creating Large Datasets for Populating Tables	2	0	0	2	100
Total	14	2	0	16	87.5

Table 8.2: Relational SQL Database Creation

Note: W = Worked, WC = Worked with Correction, N = Did not Work

In the table 8.3, the did not work category includes the SQL queries or execution results that were incorrect the first time and were not corrected. The attempt to correct these were not made on the incorrect results. The query execution works for smaller amount of sample data that ChatGPT can walk through by explaining the process. For larger sample data the query execution needs to be derived and is not very accurate. As the conditions for data creation gets complex and size of data increases, ChatGPT is unable to follow along. So creating large datasets with complex conditions are not included in the table.

The unit for closure refers to the calculation of given number of closures together. The calculation of extraneous attributes can range from 1 to more attributes being tested at once. In the table, 1 refers to finding one extraneous attribute. The accuracy when ChatGPT uses code based computation is almost always correct. The table does not include most of the attemps made to solve all the steps together, specially the complex ones, as most of these ended up in failure. The simpler questions could be solved without breaking up but the complex ones needed to be broken down into smaller steps to get the correct results. The complete process can be solved when using code based computation.

Topic	W	WC	N	Total	Percentage
Simple SQL (Single-table Select/Where)	12	0	0	12	100
Simple SQL (Groupby and Aggregation)	10	0	0	10	100
SQL Queries With 1-2 Joins and Where	17	0	0	17	100
Advanced SQL (Groupby and Aggregation)	10	1	0	11	90.9
Advanced SQL (Multiple Joins)	4	1	0	5	80
Advanced SQL (Subqueries)	10	0	0	10	100
Explaining SQL Queries	19	0	0	19	100
Finding Errors in Queries	7	0	0	7	100
Queries Execution	11	4	2	17	64.7
Total	100	6	2	108	92.6

Table 8.3: SQL Queries

Note: W = Worked, WC = Worked with Correction, N = Did not Work

The table 8.4 shows that when asking ChatGPT to find closure of 1-5 attributes, the model was able to solve a higher percentage of questions correctly. But when asked to find closure of a large number of attributes together, it made mistakes that needed to be corrected. In the similar way, for canonical cover and candidate keys, when the complete questions were asked, the model made mistakes. But when the questions were broken down into smaller steps, the model was able to solve them correctly.

Topic	W	WC	N	Total	Percentage
Closure of 1-5 Attributes	27	0	0	27	100
Closure of 6-10 Attributes	2	2	0	4	50
Closure of 10+ Attributes	0	0	1	1	0
Divide into 4 Cases for CKs	8	3	0	11	73
Candidate Keys	8	2	0	10	80
Canonical Cover	5	2	0	7	71
Finding Extraneous Attribute	12	3	0	14	78
Total	60	12	1	74	81

Table 8.4: Normalization

Note: W = Worked, WC = Worked with Correction, N = Did not Work

Topic	W	WC	N	Total	Percentage
Lexicographical Sorting	5	0	0	5	100
Create Ck	17	4	0	21	80.9
Calculate Lk	18	2	0	20	90
Stopping Criteria	3	0	0	3	100
Find All Apriori Item-Sets	5	0	0	5	100
Find Support and Confidence	2	0	0	2	100
Find Association Rules	2	0	0	2	100
Find Lift	2	0	0	2	100
Total	54	6	0	60	90

Table 8.5: Association Rules

Note: W = Worked, WC = Worked with Correction, N = Did not Work

Topic	W	WC	N	Total	Percentage
Calculating Distance	14	0	0	14	100
Assigning Clusters	13	0	0	13	100
Calculating Centroid	15	0	0	15	100
Updating Centroid	12	1	0	13	92.3
Stopping Criteria	4	1	0	5	80
Total	58	2	0	60	96.6

Table 8.6: Clustering

Note: W = Worked, WC = Worked with Correction, N = Did not Work

8.2.2 Complexity of Prompts

Some of the topics do not need any guidance or well defined prompts to find the answers from ChatGPT. These are simpler topics that can be answered with a simple question. However, as the questions get more complex, the prompts start to get more complex as well. Some of these require a little better wording or defining the problem in a slightly more structured way. The complexity of prompts is analyzed by checking the complexity of the prompts needed to get the results. The prompts that are needed to get the results are analyzed and the complexity of these prompts is calculated on a scale of 1 to 5. If the prompt is a simple question that can be asked in layman's terms then it is given a complexity level of 1. These are the ones where multiple people can give prompts in their own way and still get the similar results. If the prompt is a little more structured then it is given a complexity level of 2. These are the ones that need a little attention into how the question is asked. If the prompt is a standard technical question then it is given a complexity level of 3. These are the ones where a generic explanation of how to get the answer or what is expected is given in a the prompt. If the prompt is well defined then it is given a complexity level of 4. These include proper well defined instructions on how to get the answer. These include some algorithm or mathematical instructions but not very complex ones. If the prompt includes algorithms or mathematical instructions then it is given a complexity level of 5. These are the ones that include complex algorithms or mathematical instructions that are needed to get the answer.

For normalization, easer topics like finding closure of a few attributes does not need much structured prompts. Providing the algorithm of closure improves the accuracy of the results. But for more complex topics like finding candidate keys, canonical cover, and extraneous attributes, more focus is required on how the prompts are given. Properly breaking down the steps of the algorithm are needed to get the desired results.

Topic	**1**	**2**	**3**	**4**	**5**
Relational SQL Database Creation	X				
SQL Queries:	X				
Normalization				X	
Association Rules					X
Clustering		X			

Table 8.7: Higher Level Complexity of Prompts

Note: Prompt Complexity Levels –

- **1:** Basic (Layman's terms)
- **2:** Lightly structured
- **3:** Standard technical
- **4:** Well-defined
- **5:** Includes algorithms or mathematical instructions

8.7 shows how complex the combination of prompts can be for each topic. This includes the maximum complexity of prompts needed to get the results. Some steps in a topic are straight forward and the model is able to solve these without any guidance. While some steps require more structured prompts to get the results. This was also found by a user study conducted during this research (see Section **??** for details), where participants were not given any guidance on how to use the model. As the complexity of the questions increased, the count of correct answers decreased.

8.3 Evaluation with Claude

The ability of ChatGPT to solve data science problems was tested with the help of Claude. Claude was given the task of asking question assuming it is a student learning SQL, learning clustering and itemset mining. Then those questions were given to ChatGPT to answer. The answers were provided to Claude to evaluate the correctness of the answers. A sample conversations between ChatGPT and Claude are provided in the Appendix C. The evaluations of multiple ChatGPT sessions by Claude are summarized in Table 8.8. It contains the evaluation of ChatGPT by Claude on the basis of Technical accuracy, Clarity of explanation, Educational value and Appropriate level for Undergraduate Students. The rating for multiple sessions of same topic is averaged to get the final rating for that topic.

Topic	**TA**	**CE**	**EV**	**AL**
Relational SQL Database Creation	4.67	4.16	4.67	4.83
SQL Queries	5	5	5	5
Normalization (Closure & Candidate Keys)	4.9	4.45	4.9	4.72
Normalization (Canonical Cover)	5	4.8	4.8	4.8
Association Rules	4.66	4.5	4.58	4.58
Clustering	4.25	3.75	3.88	4.25

Table 8.8: Summary of ChatGPT's Evaluation by Claude Across Various Topics

Note: TA = Technical accuracy, CE = Clarity of explanation, EV = Educational value, AL = Appropriate level for Undergraduate Students

The other methods of evaluation involved human analysis of the responses generated by ChatGPT. In this method, the responses of Generative AI model were evaluated by another Generative AI model. Some of the evaluations done by Claude had critical analyses of the responses generated by ChatGPT. It was observed that some mistakes, which Claude is also prone to, were not identified by Claude. This analysis is similar to the other evaluations done by humans. This shows that there is some accuracy in the responses generated by ChatGPT and Claude's evaluation of those responses. In Table 8.8 the SQL queries were given a rating of 5 for all the categories. Similarly in the study, participants were able to get the correct answers for the SQL queries. In the empirical analyses, the accuracy of SQL queries is above 90% for most of the subtopics, except query execution and Advanced SQL (Multiple Joins) shown in table 8.3. The overall accuracy of SQL queries in empirical analyses is 92.6%. This relationship between the accuracy of responses, participants answers, and the evaluation by Claude shows that the evaluation by Claude is consistent with the human evaluation. Similarly, the evaluation of relational SQL database creation is also consistent with the human evaluation. In this way, the Generative AI model can be used to evaluate the responses generated by another Generative AI model. This can be used to evaluate the responses of the model in real-time and provide feedback to the model to improve the responses.

Chapter 9

Conclusion

The conclusion of this is that there are many generative AI models available that can help us in learning better. ChatGPT has predominantly worked better for Data Science concepts, both in explanation and solution and is the top choice of students. It has the capacity to give wrong answers, so it cannot be trusted 100%. Even with imperfect accuracy, it can be highly useful in the education of Data Science. The good practices for using ChatGPT for helping in education are giving clear and well-defined prompts, giving small steps of algorithms to solve at a time, and giving information in the way the model understands. By using these few techniques, ChatGPT can be used for learning and solving Data Science questions.

ChatGPT can generate SQL queries for relational database schema creation, including primary and foreign keys. It can also generate sample data for the schema, with the option to provide specific instructions for more logical data.

ChatGPT can generate data for learning purposes, but it is not reliable for large data sets due to errors. It can assist in writing, executing, and explaining SQL queries, providing step-by-step explanations and error resolution. ChatGPT is also useful for understanding normalization processes, such as finding attribute closures based on functional dependencies.

ChatGPT can assist in finding candidate keys and canonical cover for functional dependencies. It can also structure functional dependencies for clarity and convert them into meaningful words. However, it may miss some candidate keys and requires step-by-step guidance to determine minimal keys.

Data analysis techniques like item set mining can also be learned and understood with ChatGPT. It can help solve practical exercise questions of item set mining step by step, to facilitate learning and understanding. Clustering problems can also be solved with ChatGPT using relatively simpler prompts and step-by-step guidance.

ChatGPT can generate solutions in code or descriptive form, and can be guided to use a specific format. It can be a valuable tool for Data Science education, but its accuracy should be verified.

Three techniques of evaluation of ChatGPT as a data sciecne tutor were used. The study shows that correctness of the answers improved by using ChatGPT for SQL queries and simpler normlaization problems. All the paticipants provided correct answers when they used ChatGPT for SQL queries and closure. For more complex questions the correctness of the answers decreased, showing some guidance is needed for better results. The students found ChatGPT to be useful for learning Data Science concepts, but it should not be relied upon solely for learning. It can be used as a tutor to help understand concepts and solve problems, but the answers should be verified for correctness.

The evaluation with Claude showed that ChatGPT can be a good tutor for undergraduate students learning these topics. It gave an average rating of more than 4 out 5 on the basis of Technical accuracy, Clarity of explanation, Educational value, and Appropriate level for Undergraduate Students.

The calculation of correctness of the answers given by ChatGPT was also used to evalaute the effectiveness of ChatGPT as a tutor of data sciecnce. Some topics needed more complex prompts than others. Using these prompts the correctness of answers by ChatGPT were more than 80% on average for all the topics.

Overall, this can be used for better understanding of data science and improve the learning process. Using the step by step approach has proved to be useful in solving the data science problems.

Bibliography

[1] Google Developers. Prompt engineering resources, n.d. Retrieved from `https://developers.google.com/machine-learning/resources/prompt-eng`.

[2] Pei Jian Tong Hanghang Han, Jiawei. *Data Mining Concepts and Techniques (4th Edition)*. Elsevier, 2023.

[3] Carlos Delgado Kloos, Carlos Alario-Hoyos, Iria Estévez-Ayres, Patricia Callejo-Pinardo, Miguel A. Hombrados-Herrera, Pedro J. Muñoz-Merino, Pedro Manuel Moreno-Marcos, Mario Muñoz-Organero, and María Blanca Ibáñez. How can generative ai support education? In *2024 IEEE Global Engineering Education Conference (EDUCON)*, pages 1–7, 2024.

[4] Katy Major and Clay Chiarelott. Slow down: Generative ai, faculty reactions, and the role of critical thinking in writing instruction. *Double Helix*, 11, 2023.

[5] Paolo Papotti Mohammed Saeed, Nicola De Cao. Querying large language models with sql. In *Proceedings of the 27th International Conference on Extending Database Technology (EDBT 2024)*. OpenProceedings, 2024.

[6] n.d. How 3 educators are using generative ai. *IEEE*, page 1, 2024.

[7] Avi Silberschatz, Henry F. Korth, and S. Sudarshan. *Database System Concepts*. McGraw-Hill, 7th edition, 2019. Face the Real World of Database Systems Fully Equipped.

[8] Hannu Toivonen. *Apriori Algorithm*, pages 39–40. Springer US, Boston, MA, 2010.

[9] U.S. Census Bureau. Data science, 2025. Accessed: March 1, 2025.

[10] Dongwon Xu, Saeyeon Lee, Nate Anderson, and Sajeev Mohan. How science and engineering students use genai tools throughout their academic journey: A four-year analysis, 2024. Accessed: March 1, 2025.

Appendix A

Questionnaire

In this study you will be asked to answer a series of questions related to SQL, Closure, and Candidate Keys in the context of relational databases and database normalization. You will asked to use ChatGPT as a tutor to help you answer the questions. You will use ChatGPT model GPT-4o as a tutor for this study. The sessions you will work on will be collected as well for further analysis.

A.1 Background

1. I hereby consent to participate in this research study. My responses will be included in the study for research purposes. This includes demographic data to observe the patterns of learning with ChatGPT, answers to Data Science questions, feedback, and ChatGPT sessions specifically created for this study. Personally identifiable information will not be collected or shared in any publication. The information will be kept confidential and will only be used for the research. There are no anticipated risks associated with this study. I understand that I have the right to withdraw from the study at any time. (Binary Choice)

2. What is the highest level of education you have completed? (Single Choice)
    - High School
    - Undergraduate
    - Graduate
    - Doctorate
    - Other (Please specify)

3. Which degree are you currently pursuing? (Single Choice)
    - Bachelor's

- Master's
- Doctorate
- Other (Please specify)
- Not pursuing a degree

4. What is your major field of study? (Single Choice)
   - Computer Science
   - Data Science
   - Artificial Intelligence
   - Cyber Security
   - Human Computer Interaction
   - Computing and Information Systems
   - Game Design and Development
   - Software Engineering
   - Information Technology and Analytics
   - Mathematics
   - Other (Please specify)

5. Which of the following courses have you taken or have experience working on a project in? (Multiple Choice)

   Computer Science
   - Introduction to Big Data
   - Big Data Analytics
   - Database System Implementation
   - Data Security and Privacy
   - Information Retrieval
   - Data Cleaning and Preparation
   - Data Analytics with Cognitive Computing
   - Advanced Database Skills: Graph Databases
   - Web Services and Service Oriented Computing
   - Advanced Database Skills: NoSQL and NewSQL Data Systems
   - Topics in Data Science

   Data Science

- Principles of Data Management
- Foundations of Data Science and Analytics
- Applied Data Science I
- Applied Data Science II
- Database Design And Implementation

Mathematics

- Principles of Computing
- Computer Science I
- Multivariable and Vector Calculus
- Linear Algebra
- Other (Please specify)

6. Are you familiar with SQL? (Yes-No Binary selection)
7. How confident are you in each of these topics? Do you know how to solve problems related to these topics? (Likert Scale - 1: Not at all, 5: Very Confident)
   - Database Modeling
   - SQL
   - Database Normalization
   - Data Cleaning
   - Association Rules
   - Decision Trees
   - Clustering
   - Outlier Detection
8. How much do you use ChatGPT regularly? (Likert Scale - 1: Not at all, 5: Daily)
9. How fluent do you think you are with ChatGPT? (Likert Scale - 1: Not at all, 5: Very Fluent)
10. Which Generative AI tools have you used before? (Multiple Choice)
    - ChatGPT
    - Claude
    - Goggle Gemini
    - Github Copilot

- OpenAI Codex
- Jasper
- Anyword
- Shortwave
- Other (Please specify)

11. Which is your preferred Generative AI tool? (Single Choice)
    - ChatGPT
    - Claude
    - Goggle Gemini
    - Github Copilot
    - OpenAI Codex
    - Jasper
    - Anyword
    - Shortwave
    - Other (Please specify)

A.2 Basic - Using Previous Knowledge

The following questions are to be answered without the help of Generative AI but you are allowed to use the reference sheet provided. You are timed for each question.

1. Write a single SQL Query to find the id of each employee who works for "First Bank Corporation" and earns more than $10000. Use this Schema:
   - Employee (id, person_name, street, city) - Primary Keys (id)
   - Works (id, company_name, salary) - Primary Keys (id)
   - Company (company_name, city) - Primary Keys (company_name)
   - Manages (id, manager_id) - Primary Keys (id)

   Show your work.

   *The expected answer is

```
SELECT W.id
FROM Works W
WHERE W.company_name = 'First Bank Corporation'
AND W.salary > 10000;
```

2. Find the Closure of {BookID} from the set of functional dependencies:

$$Books(BookID, AuthorName, Genre, Publisher)$$

{ {BookID } → {AuthorName, Genre},
{AuthorName } → {Genre},
{Publisher } → {BookID},
{BookID, AuthorName } → {Publisher}}.

Show your work.

*The expected answer is {BookID, AuthorName, Genre, Publisher}

3. Find the Candidate Keys of the set of functional dependencies:

$$Books(BookID, AuthorName, Genre, Publisher)$$

{ {BookID } → {AuthorName, Genre},
{AuthorName } → {Genre},
{Publisher } → {BookID},
{BookID, AuthorName } → {Publisher}}.

Show your work.

*The expected answer is {BookID}, {Publisher}

A.3 Basic - Using ChatGPT

You can use the provided guidance sheet of ChatGPT on how to use it for tutoring. Please answer the following simple questions with the help of ChatGPT. You are timed for each question.
(Hint: Try dividing the question into smaller parts and work on it manually along with the AI)

Write a single SQL query to find the id of each employee who works for "First Bank Corporation" and earns more than $10000. Use this Schema:

- Employee (id, person_name, street, city) - Primary Keys (id)
- Works (id, company_name, salary) - Primary Keys (id)
- Company (company_name, city) - Primary Keys (company_name)
- Manages (id, manager_id) - Primary Keys (id)

Show your work. You can copy and paste from the AI tutor if that is useful.
*The expected answer is

```
SELECT W.id
FROM Works W
WHERE W.company_name = 'First Bank Corporation'
AND W.salary > 10000;
```

Provide the session URL for the above question. (Session URL)

Find the Closure of {BookID} from the set of functional dependencies:

$$Books(BookID, AuthorName, Genre, Publisher)$$

{ {BookID } → {AuthorName, Genre},
{AuthorName } → {Genre},
{Publisher } → {BookID},
{BookID, AuthorName } → {Publisher}}.

Show your work. You can copy and paste from the AI tutor if that is useful.
*The expected answer is {BookID, AuthorName, Genre, Publisher}

Provide the session URL for the above question. (Session URL)

Find the Candidate Keys of the set of functional dependencies:

$$Books(BookID, AuthorName, Genre, Publisher)$$

{ {BookID } → {AuthorName, Genre},
{AuthorName } → {Genre},
{Publisher } → {BookID},
{BookID, AuthorName } → {Publisher}}.

Show your work. You can copy and paste from the AI tutor if that is useful.
*The expected answer is {BookID}, {Publisher}

Provide the session URL for the above question. (Session URL)

A.4 Advanced - Using ChatGPT

The following questions are more complex and require a deeper understanding of the topic. Please answer the following questions with the help of ChatGPT. You are timed for each question.

1. Write a single SQL Query to find the id of each employee who earned more than every employee of "Small Bank Corporation". Use this Schema:
   - Employee (id, person_name, street, city) - Primary Keys (id)
   - Works (id, company_name, salary) - Primary Keys (id)
   - Company (company_name, city) - Primary Keys (company_name)
   - Manages (id, manager_id) - Primary Keys (id)

   Show your work. You can copy and paste from the AI tutor if that is useful.

   *The expected answer is

```
SELECT W1.id
FROM Works W1
WHERE W1.salary > ALL(
    SELECT W2.salary
    FROM Works W2
    WHERE W2.company_name = 'Small Bank Corporation'
);
```

2. Provide the session URL for the above question. (Session URL)
3. Find the Candidate Keys of the set of functional dependencies:

$$Class(StudentID, CourseID, Semester, Instructor)$$

{ {CourseID } $\rightarrow$ {StudentID, Semester},
{Semester } $\rightarrow$ {Instructor},
{StudentID, Instructor } $\rightarrow$ {CourseID},
{StudentID, Semester } $\rightarrow$ {CourseID, Instructor}}.

   Show your work. You can copy and paste from the AI tutor if that is useful.

   *The expected answer is {CourseID}, {StudentID, Semester}, {StudentID, Instructor}

4. Provide the session URL for the above question. (Session URL)
5. Find the Candidate Keys of the set of functional dependencies:

$$Inventory(ProductCode, ProductName, Category, Supplier, WarehouseLocation, StockQuantity)$$

{{ProductCode } $\rightarrow$ {ProductName}
{ProductCode } $\rightarrow$ {WarehouseLocation}

{Category } → {ProductName, StockQuantity}
{ProductName, Category } → {StockQuantity}
{ProductName } → {ProductCode, Supplier}
{Supplier, StockQuantity } → {Category, WarehouseLocation}}

Show your work. You can copy and paste from the AI tutor if that is useful.

*The expected answer is {Category}, {ProductCode, StockQuantity}, {ProductName, StockQuantity}, {Supplier, StockQuantity}

6. Provide the session URL for the above question. (Session URL)

A.5 Participants' Feedback

Please provide your feedback on your experience using the ChatGPT in this study. The following questions are regarding this study and how effective you found ChatGPT in helping you as a data science tutor. You are also encouraged to provide any additional feedback or comments.

1. How helpful was the ChatGPT for the basic SQL query? (Likert Scale - 1: Not at all, 5: Very Helpful)
2. How helpful was the ChatGPT for the advanced SQL query? (Likert Scale - 1: Not at all, 5: Very Helpful)
3. How helpful was the ChatGPT for the basic Closure question? (Likert Scale - 1: Not at all, 5: Very Helpful)
4. How helpful was the ChatGPT for the advanced Closure question? (Likert Scale - 1: Not at all, 5: Very Helpful)
5. How helpful was the ChatGPT for the basic Candidate Keys question? (Likert Scale - 1: Not at all, 5: Very Helpful)
6. How helpful was the ChatGPT for the advanced Candidate Keys question? (Likert Scale - 1: Not at all, 5: Very Helpful)
7. How useful was the guidance sheet for ChatGPT for the basic SQL query? (Likert Scale - 1: Not at all, 5: Very Useful)
8. How useful was the guidance sheet for ChatGPT for the advanced SQL query? (Likert Scale - 1: Not at all, 5: Very Useful)

9. How useful was the guidance sheet for ChatGPT for the basic Closure question? (Likert Scale - 1: Not at all, 5: Very Useful)
10. How useful was the guidance sheet for ChatGPT for the advanced Closure question? (Likert Scale - 1: Not at all, 5: Very Useful)
11. How useful was the guidance sheet for ChatGPT for the basic Candidate Keys question? (Likert Scale - 1: Not at all, 5: Very Useful)
12. How useful was the guidance sheet for ChatGPT for the advanced Candidate Keys question? (Likert Scale - 1: Not at all, 5: Very Useful)
13. How likely are you to use ChatGPT as a data science tutor in the future? (Likert Scale - 1: Not at all, 5: Very Likely)
14. Are there any advantages of using ChatGPT for data science tutoring in your opinion? (Descriptive)
15. Are there disadvantages of using ChatGPT for data science tutoring in your opinion? (Descriptive)
16. Any other comments or feedback? (Descriptive)

*Expected answers will not be provided in the questionnaire.

A.6 Reference Sheet

This sheet contains some of the basic concepts and methods that you can use to answer the questions.

A.6.1 SQL

For Schema:
- Employee (id, person_name, street, city) - Primary Keys (id)
- Works (id, company_name, salary) - Primary Keys (id)
- Company (company_name, city) - Primary Keys (company_name)
- Manages (id, manager_id) - Primary Keys (id)

- Select All Columns from Employee Table:

```
SELECT *
FROM Employee;
```

- Select person_name, street from Employee Table:

```
SELECT person_name, street
FROM Employee;
```

- Finding company names in New York:

```
SELECT company_name
FROM Company
WHERE city = 'New York';
```

- Finding name and salary of all employees:

```
SELECT person_name, salary
FROM Works
    JOIN Employee
    ON Works.id = Employee.id;
```

- Select unique person_name, from Employee Table:

```
SELECT DISTINCT person_name
FROM Employee;
```

- SOME

```
SELECT column_name(s)
FROM table_name
WHERE column_name operator ANY
    (SELECT column_name
    FROM table_name
    WHERE condition);
```

- SOME

```
SELECT column_name(s)
FROM table_name
WHERE column_name operator ALL
    (SELECT column_name
    FROM table_name
    WHERE condition);
```

A.6.2 Closure

The closure of β in a set of functional dependencies F, denoted by β^+, is the set of all functional dependencies that can be inferred from β.

Following is an example of finding the closure of a set of functional dependencies: For R(A, B, C, D, E)

1. $\{A, B\} \rightarrow \{C\}$
2. $\{C\} \rightarrow \{D\}$
3. $\{E\} \rightarrow \{C, A\}$
4. $\{A, D\} \rightarrow \{E\}$
5. $\{C, D\} \rightarrow \{B\}$

The closure of {C} starts with {C}
then using $\{C\} \rightarrow \{D\}$ we add {D} and closure becomes {C, D}
then using $\{C, D\} \rightarrow \{B\}$ we add {B} and closure becomes {C, D, B}.
Since there are no more functional dependencies that can be inferred from the closure, the final closure is {C, D, B}.

Another example of finding the closure of a set of functional dependencies is: For Books(BookID, AuthorName, Genre, Publisher)

{ {BookID } → {AuthorName, Genre},
{AuthorName } → {Genre},
{Publisher } → {BookID},
{BookID, AuthorName } → {Publisher}}.

The closure of {AuthorName} starts with {AuthorName}
then using {AuthorName} → {Genre} we add {Genre} and closure becomes {AuthorName, Genre}.
Since there are no more functional dependencies that can be inferred from the closure, the final closure is {AuthorName, Genre}.

A.6.3 Candidate Keys

A candidate key is a minimal set of attributes that can uniquely identify a tuple in a relation. The algorithm for finding the candidate keys of a set of functional dependencies F is as follows:

Algorithm 7: Finding Candidate Keys from Functional Dependencies

Input: A set of Functional Dependencies (F) of type $\alpha \to \beta$

Output: Candidate Keys

```
left ← ⋃ α in F;
right ← ⋃ β in F;
Initialize case_1, case_2, case_3, case_4 as empty sets;
foreach x in attributes do
    if x ∉ left and x ∉ right then
        case_1 ← case_1 ∪ {x};
    if x ∉ left and x ∈ right then
        case_2 ← case_2 ∪ {x};
    if x ∈ left and x ∉ right then
        case_3 ← case_3 ∪ {x};
    if x ∈ left and x ∈ right then
        case_4 ← case_4 ∪ {x};
core ← case_1 ∪ case_3;
if Closure(core) = F then
    return core as the only candidate key;
else
    candidate_keys ← ∅;
    foreach combination x of case_4 do
        if Closure(core ∪ x) = F then
            candidate_keys ← candidate_keys ∪ (core ∪ x);
    return candidate_keys;
```

A.7 Guidance Sheet for ChatGPT

The following are some of the methods you can use to get help from ChatGPT in answering the questions.

- Switch the account if you are not using GPT-4o
- Divide the questions into smaller parts can give better results
- Work on each step along with the AI tutor
- Use the AI tutor to understand the concepts
- Use mathematical notation to explain the concepts

- Give parts of algorithm to AI tutor to work on it step by step
- If there is a mistake in reponse from AI tutor, ask AI to correct it
- If you are stuck, try rephrasing the question
- include the schema in the prompt can help AI tutor understand the question better

Appendix B

SQL Query Execution

I will walk you through how the query executes step by step using the provided **University Schema and Data**.

B.1 Understanding the Schema

We have two relevant tables:

- **Instructor**
  - Stores information about instructors.
  - **Key columns**: `ID` (Primary Key), `name`.
- **Teaches**
  - Stores records of instructors teaching sections.
  - **Key columns**: `ID` (Foreign Key referencing Instructor), `course_id`, `sec_id`, `semester`, `year`.

B.2 Writing the SQL Query

The SQL query used for analysis:

Listing B.1: SQL Query for Counting Sections per Instructor

```sql
SELECT I.ID, I.name, COUNT(T.course_id) AS number_of_sections
FROM instructor I
LEFT JOIN teaches T ON I.ID = T.ID
GROUP BY I.ID, I.name;
```

B.3 Breakdown of Query Execution

1. **Selecting Instructors**: The query starts by selecting all instructors from the `instructor` table.
2. **Performing a LEFT JOIN**: It joins each instructor with records in `teaches`, linking `instructor.ID` to `teaches.ID`.
   - Since it's a **LEFT JOIN**, all instructors appear in the result, even if they do not have matching entries in `teaches`.
3. **Counting Sections**: The query counts how many sections each instructor teaches.
   - If an instructor does not teach any sections, `T.course_id` will be `NULL`, and `COUNT()` will return `0`.
4. **Grouping Data**: The data is grouped by instructor `ID` and `name` to aggregate the count correctly.

B.4 Sample Data from Schema

The `instructor` table:

ID	Name
10101	Srinivasan
12121	Wu
15151	Mozart
22222	Einstein
32343	El Said
33456	Gold
45565	Katz
58583	Califieri
76543	Singh
76766	Crick
83821	Brandt
98345	Kim

Table B.1: Instructor Table

The `teaches` table:

ID	Course ID	Sec ID	Semester	Year
10101	CS-101	1	Fall	2017
10101	CS-315	1	Spring	2018
10101	CS-347	1	Fall	2017
12121	FIN-201	1	Spring	2018
15151	MU-199	1	Spring	2018
22222	PHY-101	1	Fall	2017
32343	HIS-351	1	Spring	2018
45565	CS-101	1	Spring	2018
45565	CS-319	1	Spring	2018
76766	BIO-101	1	Summer	2017
76766	BIO-301	1	Summer	2018
83821	CS-190	1	Spring	2017
83821	CS-190	2	Spring	2017
83821	CS-319	2	Spring	2018
98345	EE-181	1	Spring	2017

Table B.2: Teaches Table

B.5 Execution Result

The query returns B.3:

B.6 Explanation of Result

- **Instructors with sections**:
  - Srinivasan teaches 3 sections.
  - Katz teaches 2 sections.
  - Brandt teaches 3 sections.
  - Many others teach 1 section.
- **Instructors with 0 sections**:
  - `LEFT JOIN` ensures that instructors without teaching records are included.
  - In this dataset, Gold, Califieri, and Singh have 0 sections.

B.7 Key Learnings

- **JOIN Types**:

ID	Name	Number of Sections
10101	Srinivasan	3
12121	Wu	1
15151	Mozart	1
22222	Einstein	1
32343	El Said	1
33456	Gold	0
45565	Katz	2
58583	Califieri	0
76543	Singh	0
76766	Crick	2
83821	Brandt	3
98345	Kim	1

Table B.3: Query Result: Number of Sections per Instructor

  * `INNER JOIN` would exclude instructors who don't teach.
  * `LEFT JOIN` includes all instructors, ensuring those with `0` sections are counted.
- **Aggregation with COUNT()**:
  * `COUNT(column_name)` ignores NULL values.
  * This ensures instructors without sections return `0`.

B.8 Conclusion

This query effectively lists **all instructors** along with the **number of sections they teach**, ensuring even those **not teaching any sections** are included. This is a common use case in database systems for reporting and analytics.

Appendix C

ChatGPT Evaluation with Claude

This appendix contains detailed chat logs used in the research evaluation of ChatGPT. The conversations are between ChatGPT and Claude.

C.1 Clustering

User:

This is a session containing a few questions and their solutions by a tutor for undergradute students who are new to this topic. Evaluate the overall responses of the tutor on the basis of: 1. Technical accuracy 2. Clarity of explanation 3. Educational value 4. Appropriate level for undergraduate students. Only rating number is needed

Shared Clustering1.pdf Clustering2.pdf
Claude:

Based on my analysis of the provided documents, I'll evaluate the tutor's responses across the requested criteria:

1. Technical Accuracy: 9/10

The tutor demonstrates strong technical accuracy in explaining and implementing k-means clustering. They correctly calculate distances (both Euclidean and Manhattan), properly assign points to clusters, and accurately compute new centroids. A minor issue appears in Clustering1.pdf where there seems to be some inconsistency in the cluster assignments in Student Said-3 vs. Student Said-4.

2. Clarity of Explanation: 8/10

The tutor's explanations are generally clear and structured. They break down the k-means algorithm into logical steps and walk through each iteration methodically. The use of tables to display distances and cluster assignments is particularly helpful. However, occasionally the formatting of mathematical expressions is unclear (especially in Clustering2.pdf).

3. Educational Value: 9/10

The tutor provides excellent educational value by:

- Connecting concepts to intuitive understanding (”natural grouping”)
- Explaining why certain steps are taken
- Discussing implications like convergence criteria
- Adding context about variations (k-means++, effects of different k values)
- Building on student understanding incrementally

4. Appropriate Level for Undergraduate Students: 9/10

The material is pitched very appropriately for undergraduate students. The tutor:

- Uses accessible language
- Follows the student's pace
- Encourages questions
- Reinforces correct understanding
- Provides visual support through tables
- Relates technical details to conceptual understanding

Overall Rating: 9/10

User:

Clustering3.pdf

Claude:

Based on my analysis of Clustering3.pdf, I'll evaluate the tutor's responses:

1. Technical Accuracy: 8/10

The tutor accurately applies the k-means clustering algorithm using Manhattan distance, correctly calculating distances between points and centroids, assigning points to clusters, and computing new centroids. However, there's a mistake in Student Said-9's response analysis - the tutor incorrectly states that ”cluster assignments have changed compared to the last iteration” when they actually remained the same.

2. Clarity of Explanation: 8/10}

The tutor provides clear, step-by-step breakdowns of the calculations with organized sections and headings. The use of tables to display distances and cluster assignments makes the information easy to follow. The explanation of the random assignment for empty clusters is straightforward. However, some mathematical notation formatting appears inconsistent.

3. Educational Value: 7/10

The tutor methodically shows the iterative process of k-means clustering with detailed calculations. However, unlike in the previous documents, there's less explanatory content about why certain steps are taken or the conceptual understanding behind the algorithm. The tutor focuses primarily on the mechanics rather than building deeper understanding.

4. Appropriate Level for Undergraduate Students: 8/10

The content is presented at an appropriate technical level for undergraduate students. The step-by-step calculations are detailed enough for students to follow along, and the tabular presentation of results makes the information accessible. The tutor responds directly to student questions without overwhelming them with unnecessary complexity.

Overall Rating: 8/10

C.2 Itemset Mining

User:

This is a session containing a few questions and their solutions by a tutor for undergradute students who are new to this topic. Evaluate the overall responses of the tutor on the basis of: 1. Technical accuracy 2. Clarity of explanation 3. Educational value 4. Appropriate level for undergraduate students. Only rating number is needed

Shared ItemsetMining4th.pdf

Claude:

I'll evaluate the tutor's responses based on the requested criteria:

1. Technical Accuracy: 5/5 The tutor demonstrates excellent technical accuracy throughout the session. All itemset mining calculations, counting of transactions, and application of the Apriori algorithm steps are performed correctly. The tutor even catches and corrects mistakes, such as when the student points out an incorrect count for Detergent.
2. Clarity of Explanation: 4/5 The tutor's explanations are generally clear with good step-by-step breakdowns, especially when generating candidate itemsets and checking support counts. The use of structured formatting with headers like "Step 1" and "Final Result" helps organize complex procedures. However, some explanations could benefit from more context about why certain steps are performed, particularly for students new to itemset mining.
3. Educational Value: 4/5 The session provides high educational value by walking through the complete Apriori algorithm process. The tutor demonstrates proper lexicographic ordering, support counting, candidate generation, and pruning. The hands-on approach with the grocery dataset helps reinforce concepts. The tutor could enhance educational value by occasionally connecting steps back to the bigger picture of association rule mining.
4. Appropriate Level for Undergraduate Students: 4/5 The material is presented at an appropriate level for undergraduate students new to the topic. The tutor breaks down complex operations into manageable steps and uses clear notation. The step-by-step approach to candidate generation and verification is particularly helpful for beginners. Some additional context about the purpose of each step in the algorithm would help students better understand why they're performing certain operations.

Overall Rating: 4.25/5

Appendix D

All Generative AI Chats

All the conversations with generative AI models are provided in this github repository: `https://github.com/YusraKhalid/GenerativeAIChats`.