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

A large increase in the use of Generative AI has been observed in the last few years. Data science is also a rapidly growing field with a high demand for skilled professionals. The goal of this thesis is to explore the potential of Generative AI, specifically ChatGPT, in facilitating data science education. The focus is on how ChatGPT can be used as a tutor to help solve practical exercises.

The capabilities of Generative AI were explored along with the comparison of a few different models in terms of data science. Exploratory analysis was conducted to compare Generative AI models and choose the better one for the thesis. The best practices for solving practical exercises using Generative AI were also explored. The prompt engineering practices for solving practical exercises have also been explored and described in this thesis. The effectiveness of using ChatGPT as a tutor of data science has also been evaluated in three different ways. First, 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 analyzed. Second, a study was conducted to see how helpful ChatGPT is in helping participants solve data science questions. The third approach was using another Generative AI model, Claude, to test how ChatGPT acted as a tutor for undergraduate students.

It was found that 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 to 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, itemset mining, and clustering can be learned and understood with ChatGPT. Using it for educational purposes will facilitate students faster and better learning as compared to if it is prohibited.

ChatGPT can be used for solving question and explaining 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 results of our evaluations are mostly positive with a few limitations. The results show that ChatGPT 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. Our main takeaway is that Generative AI can not substitute teachers but can act as a personalized tutor for each student. It can explain the solutions given in the textbooks in more detail and can also help with 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, SQL Fiddle, GitHub Copilot, and Claude for the research purposes.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

A large increase has been observed in the use of Generative AI in various fields. In 2020, less than 116 million people used AI daily, a figure that nearly tripled to 314 million in 2024 [2]. A great number of businesses are using Generative AI to improve their productivity and efficiency. According to a survey conducted by McKinsey, 78% of respondents say their organizations use AI in at least one business function [5]. With the rapid growth of Generative AI, it is also becoming common in the education sector. More than 50% of college students use Generative AI for their studies [7].

A study shows that the use of Generative AI by science and engineering students tend to increase with each passing year [7]. 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 models to provide better and more useful information, Generative AI is getting adopted more in the everyday lives of people. Generative AI is often looked at with a controversial image in the education sector [12]. 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 [8].

Data Science is a field of study that uses scientific methods, processes, and systems to extract knowledge and insights from data, typically extracting and storing data in databases [4]. Understanding the databases and how to get, update, modify, transform, remove, and utilize data comes within this domain. Everything related to data is included in data science. The life cycle consists of data collection, data processing, data analysis and data visualization, data communication, and data management. The industry is moving towards data-driven decision making and the demand for data scientists is increasing rapidly. According to a user study conducted in Europe with participants from 10 different countries, 56% of participants were confident that their organization collected data, and 73% of respondents indicated that decisions were made based on the data collected [3]. This is a rapidly growing field with an industry valued at 103.93 billion USD in 2023 and expected to grow to 776.86 billion USD by 2030 [10]. This shows that due to the rapid growth of data science, the demand for skilled professionals is also increasing. Data science education is also becoming more important as the demand for data scientists is increasing rapidly.

There is research 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 [11]. There has been some research on the benefits and uses of Generative AI in education. Others have studied the use of Generative AI for data science [13]. To the best of our knowledge, data science education with the help of Generative AI is still unexplored area. Generative AI can act as a tutor to help students solve practical exercise question in their data science coursework. Generative AI can also help in explaining the concepts and explain the solutions in a detailed manner. The two popular books used in the course of data science are "Database System Concepts" [14] and "Data Mining Concepts and Techniques" [9]. Generative AI has been explored in this thesis 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 [9,14]. 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, retrieving, 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 itemset mining that is used to find the frequent itemsets in the data. These frequent itemsets are 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 another 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 Generative AI can be used as a tutor for solving practical data science exercises and explaining data science concepts. The study aims to find the optimal ways and accuracy of using GenerativeAI to solve and explain the practical exercise questions related to data science for undergraduate students. The focus is on the use of Generative AI for both basic and advanced topics in data science like SQL, normalization, clustering, and frequent itemset mining. Here the tutor is a facilitator or a tool that the students can use in addition to their standard learning methods. Practical exercises are the exercises that are given to students to solve in their coursework. Students are expected to transform piece of data using algorithms and techniques related to the topics. The goal is to help students use Generative AI as a tutor to solve the practical exercises.

1.3 Research Methodology

The study initially focuses on a few different Generative AI models to find the best one for data science education. The comparison was done on the basis of the factual correctness of the answers and the ability to solve and explain data science concepts. A comparison was made between ChatGPT, Gemini, SQL Fiddle and GitHub Copilot. Then the best model was selected for further experimentation to solve practical exercises and explain the concepts. The focus has been on finding the capabilities and limitations of Generative AI for solving exercises and explaining the inner workings of the studied topics. The evaluation was done in three different ways, empirical evaluation, user study and evaluation with another Generative AI model, Claude.

1.3.1 Exploratory Analysis

Prompt engineering techniques [6] are explored to find the optimal way of providing 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 analyzing the data. The other was to explain the concepts or the solutions provided by a textbook 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 description or practical question and then it was provided the 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 a textbook. 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 explanation and the result as well, if the data is small.

For normalization, a higher level algorithm broken down into smaller steps was used as an input to ChatGPT to provide a solution. This was useful as ChatGPT acted as a tutor that students 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 itemset 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.

Empirical Evaluation

One method of evaluation was to assess the correctness of the answers given by ChatGPT. The topics are broken down into smaller solvable 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.

User Study

A user study was conducted among students with a basic to advanced background in SQL and data science. The user study was conducted in a lab setting and was divided into three parts for solving practical exercise questions. In the first part, they had to solve practical exercise questions of SQL and Normalization on their own. In the second part, they were given the same questions to solve with the help of ChatGPT. In the third part, they were given more advanced questions to solve with the help of ChatGPT. The demographics and feedback of the participants were collected. The results show that ChatGPT was the preferred Generative AI model for majority of the participants. It was found in the study that when the questions were simple, all of the participants were able to get correct results 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. Larger number of students also said that they are likely to use ChatGPT as a data science tutor in future.

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, 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 is structured in the following way:

- 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.
- Chapter 3 presents initial comparison of ChatGPT, Gemini, SQL Fiddle, and GitHub Copilot, ChatGPT outperformed other models for data science education.

- Chapter 4 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 5 presents SQL queries; writing, executing and explaining SQL queries with ChatGPT along with error resolution.
- Chapter 6 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 7 presents frequent itemset mining with ChatGPT. The chapter focuses on solving itemset mining and creating association rules with ChatGPT.
- Chapter 8 presents clustering with ChatGPT. The chapter focuses on solving k-mean clustering problems with ChatGPT.
- Chapter 9 presents the evaluation of the study. The chapter focuses on using three 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 10 presents the conclusion of the study.

Chapter 2

Background

A typical introductory data science course starts with the introduction of databases that contains relational databases. SQL is a language used to manipulate relational databases. SQL stands for Structured Query Language and is used to create, update, insert, delete and fetch the data from database. The topics for database maintenance is normalization that is the process of removing redundant data from the database. The more advanced topics related to analyses of data are clustering and itemset mining. Clustering is the process of grouping similar data points together. 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. Some other methods of data cleaning, data analysis, and data visualization are also taught in a data science course. The focus is to explore how Generative AI can be used to solve practical exercises for these topics of data science.

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 [7]. The AI can act as a tutor to help students understand their coursework in a better way [11]. 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 tables instead of long texts [13]. To the best of our knowledge, there is no previous research available on the use of Generative AI as a tutor for data science. The books used for a typical course of data science are "Database System Concepts " [14] and "Data Mining Concepts and Techniques" [9]. These books are used as a source of topics for this research. In the rest of the thesis, the term "textbook" refers to "Database System Concepts" [14] as this is a typical textbook for an introductory course in data science. Below, we introduce the topics of a typical introductory data science course along with a brief explanation of how to solve practical exercises in these topics.

Employee (person_name, street, city)
Works (person_name, company_name, salary)
Company (company_name, city)

Figure 2.1: Employee schema

section (course_id, sec_id, semester, year, building, room_number, time_slot_id)
classroom (building, room_number, capacity)
takes (ID, course_id, sec_id, semester, year, grade)
teaches (ID, course_id, sec_id, semester, year)
course (course_id, title, dept_name, credits)
prereq (course_id, prereq_id)
student (ID, name, dept_name, tot_cred)
department (dept_name, building, budget)
instructor (ID, name, dept_name, salary)
advisor (s_id, i_id)
time_slot (time_slot_id, day, start_hr, start_min, end_hr, end_min)

Figure 2.2: University schema

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 in Figure 2.1.

The answer of the question is the primary keys for each table, that is, person_name for Employee, person_name and company_name for Works, and company_name for Company [14].

The textbook uses university schema for many queries, and this thesis also used Generative AI to test generation of university schema and work on it. The primary keys are underlined. The schema has the structure given in Figure 2.2.

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

Figure 2.3: SQL practical exercise example

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

Figure 2.4: SQL query for example

2.2 SQL

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 Figure 2.2 is given in Figure 2.3.

The answer of the question in Figure 2.3 is given in Figure 2.4.

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 optimizing database performance, enhancing database scalability, improving data integrity, and reducing 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 Figure 2.2, an example of set of functional dependencies for relation course is given in Figure 2.5. 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 determines the department name and credits of the course.

Finding Closure, Candidate Keys, and Canonical Cover are presented below.

course (course id, title, dept name, credits)
FDs = {
{course_id} $\rightarrow$ $\{title\}$
$\{course\_id\} \rightarrow \{dept\_name\}$
$\{course\_id\} \rightarrow \{credits\}$
$\{title\} \rightarrow \{dept\_name\}$
$\{title\} \rightarrow \{credits\}$
}

Figure 2.5: Functional dependencies for the course table

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 as given in Figure 2.5, 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 as given in Figure 2.5, 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 as given in Figure 2.5, 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 relationships 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 lexicographically sorted items to generate candidate sets. Initially, the 1-itemsets are generated by counting the occurrence 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 way only the items that occur a more than a threshold are kept and those bought together are kept together in higher orders. The process is repeated until no more itemsets can be generated. The algorithm is given as Algorithm 1 [1]. An example of frequent itemsets is given in Figure 2.6.

Algorithm 1: 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**
            $c.\text{count} \leftarrow c.\text{count} + 1$;
    $L_k \leftarrow \{c \in C_k \mid c.\text{count} \geq \text{minsup}\}$;
return $\bigcup_k L_k$;

Association Rules: These itemsets are then used to generate association rules. For example, the rule {Mathematics} $\rightarrow$ {Physics} can be generated from the frequent itemsets. This rule means that if a student takes Mathematics then she is more likely to take Physics as well. The quality of these rules is measured using support, confidence and lift. The formulas for these are given in Figure 2.7. Support is the percentage of transactions that contain the itemset. Confidence is the percentage of transactions that contain Y given that they contain X. 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 L_1, L_2, ..., L_k generated by the Apriori algorithm.

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
$L_1 = \{\{\text{Mathematics: } 2\}, \{\text{Physics: } 2\}, \{\text{Chemistry: } 2\}\}$
$L_2 = \{\{\text{Mathematics, Physics: } 2\}\}$

Figure 2.6: Apriori example

$$\text{Support}(X \to Y) = \frac{\text{count}(X \cup Y)}{|D|}$$

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

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

Figure 2.7: Association rules

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

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

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

Figure 2.8: Association rules example

Example: For the given example, support, confidence and lift for the rule {Mathematics} → {Physics} can be calculated as given in Figure 2.8.

2.5 Clustering

Clustering is a technique used in data analytics to find similar groups of data. K-means 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 randomly selected or selected based on some criteria. These are called centroids. Then the distance between each point and the centroid is calculated based on a 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: 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 achievers and students that require more attention. Given marks of 5 students in 2 subjects as shown in Figure 2.9. These will be used as the points for clustering. To divide the students into 2 clusters, the centroids can be randomly selected as shown in Figure 8.2. 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 shown in Figure 2.11. 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

1. [10, 20]
2. [15, 25]
3. [30, 40]
4. [35, 45]
5. [50, 60]

Figure 2.9: Clustering example

1. [20, 30]
2. [40, 50]

Figure 2.10: Randomly selecting centroids

clusters and the final clusters are shown in Figure 8.3. 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

Generative AI is the type of artificial intelligence that is used to generate new content based on the input given. It can be used to generate text, images, audio, and video. Traditional artificial intelligence focuses on analyzing or predicting data, whereas Generative AI is capable of creating new content. Mostly deep learning models like transformers are used to train Generative AI. The most common use of Generative AI is in the field of Natural Language Processing (NLP) where it is used to generate text based on the input given. Generative AI is used in many fields like education, data science, and computer vision.

Generative AI can be used in education for faster learning and better understanding of the concepts. It can be used as a personal tutor to help students clear their concepts and solve their exercises. Generative AI tutoring includes using it to solve questions, explain the answers, and provide feedback on the correctness of the answers. Just like a human tutor that helps students in addition to traditional learning, the focus is to use Generative AI tutor in the same way. The type of tutoring expected from Generative AI here is

1. [18.33, 28.33]
2. [42.5, 52.5]

Figure 2.11: Recalculating centroids

1. [10, 20], [15, 25], [30, 40]
2. [35, 45], [50, 60]

Figure 2.12: Final clusters

to help in solving practical exercise questions. For each of the topics of data science a different approach is required to solve the exercises. 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 tutor 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. Here tutoring means solving practical exercise questions and explaining the answers.

Some of the most common Generative AI models in education and data science are ChatGPT, Gemini, SQL Fiddle, Claude and GitHub Copilot. ChatGPT, Gemini and Claude are based on the transformer architecture and are used for text generation. SQL Fiddle is a web based tool that is used to test SQL queries and generate SQL queries. GitHub Copilot is a code generation tool that is used to generate code based on the input given. Gemini, ChatGPT, SQL Fiddle, and GitHub Copilot were preliminarily tested for solving practical exercise questions of data science.

2.7 Prompt Engineering

Prompt engineering is the process of optimizing the input given to generative AI models to get the desired output. It includes the finding the right prompts or questions to ask Generative AI in order to get the expected response. In some cases the prompts can be simple questions while in other cases the prompts need to be more structured and detailed. Structuring the prompts in different ways can result in different outputs from Generative AI even if the question is same. Therefore, to get the best results from Generative AI, it is important to find the right prompts and structure them in a way that the model can understand.

Chapter 3

Exploratory Analysis

An exploratory analysis was conducted for using Generative AI as a tutor for data science. The goal of this analysis was to explore the capabilities and shortcomings of Generative AI in solving data science problems and to find the best model for this purpose. The exploratory analysis was conducted to compare different Generative AI models for solving practical exercises of data science. Initially, a brief analysis was conducted to compare the capabilities of different Generative AI models. Upon the completion of this analysis, one Generative AI model was selected for the detailed analysis and research.

3.1 Generative AI Models

A few renowned models were selected for this purpose and the results were compared. The models selected for this comparison were ChatGPT, Gemini, SQL Fiddle, and GitHub Copilot. ChatGPT and Gemini were selected because these are versatile models that can be used for a variety of tasks. ChatGPT is also one of the most popular and widely used Generative AI models. SQL Fiddle was selected because it is a SQL specific model that can be used for solving SQL problems. It provides a platform to test SQL queries and get the results. GitHub Copilot was selected because it is a code generation model that is good with algorithms and coding problems.

3.2 Comparison with Respect to Topics

The models were compared based on their ability to solve SQL and relational database practical exercise questions, explain the solutions, and provide examples. These were also tested for their ability to explain concepts in a way that is easy to understand for students. The models were given the same SQL problem to solve and the results were compared.

ChatGPT Response

Example:
Let's assume:
Einstein teaches PHY101, Section 1 in Fall 2024.
Alice and Bob are enrolled in this course and section.

The query will:
Match the entries in the takes table for Alice and Bob.
Join these entries with the teaches table to confirm that the instructor for this course section is Einstein.
Filter out any students not taught by Einstein.
Return Alice's and Bob's IDs, with no duplicates.

Figure 3.1: Example provided by ChatGPT for SQL explanation

3.2.1 SQL Queries Writing

For one of the test ChatGPT and Gemini were used to solve the same SQL problem from the textbook and the results were compared. Both models gave the correct answer but it was different than the answer given in the textbook. ChatGPT explained the answer in a better way and provided more details about the solution as compared to Gemini. For SQL query writing, GitHub Copilot also provided the correct answer with a brief explanation and the SQL Fiddle provided only the SQL query without any explanation.

3.2.2 SQL Queries Explanation

The models were also tested for their ability to explain the SQL queries. All Generative AI models were also tested by giving SQL question and answer from the textbook practical exercise and asked to explain the answer. This approach was used so students understand how a specific answer in the textbook was obtained. ChatGPT also provided an example to explain the solution whereas Gemini and Github Copilot did not provide any examples. The SQL Fiddle only provided a brief explanation of the answer without any examples. The complete responses of all 4 along with the prompt given for explanation is provided in the Appendix E. The example provided by ChatGPT is given in Figure 3.1.

3.2.3 SQL Queries Execution

The models were also tested for their ability to execute SQL queries and get the results. The models were given the schema, sample data, and SQL query to execute. Then the models were asked to execute the query and provide the results. It was done so students can understand how SQL engine works and how the results are obtained. For query execution, SQL Fiddle was the best model as it provided the correct results for all queries. The reason behind this is that SQL Fiddle actually uses a SQL engine to execute the queries and get the results. ChatGPT was able to give somewhat correct results but it was not able to always provide the correct results. ChatGPT was able to provide the correct answers for most of the queries. ChatGPT executed the queries manually step by step explaining the process. Gemini was able to provide the correct result for some of the queries but it was using an SQL engine to execute the queries. Both Gemini and SQL Fiddle were not working on the queries step by step to explain the process. GitHub Copilot was not able to execute most of the queries given the data. It did provide a better explanation as compared to Gemini and SQL Fiddle.

3.3 Ease of Use

The models were also compared based on their ease of use and user-friendliness. It was tested on the basis of how easily the models can be used by the undergraduate students. For this purpose the effort taken in initial setup and the user interface were compared. ChatGPT and Gemini were the easiest to use as they are web-based models and do not require any setup. For both of these Generative AI models, the user interface is simple and easy to use. ChatGPT can also create visualizations and graphs using the data provided, making it easier for students to understand the data. GitHub Copilot requires some setup and is not as user-friendly as ChatGPT and Gemini. Although it is a code generation model, the chat option is not user-friendly. It also requires a GitHub account in order to use it. Most undergraduate students who are new to coding and programming do not have a GitHub account. SQL Fiddle is also web-based making it easily accessible but it is not as user-friendly as ChatGPT and Gemini. It is good as a SQL engine but the user interface for the Generative AI is not user-friendly. SQL Fiddle also do not save the history of the conversation and the user has to start a new chat each time.

3.4 Key Findings

The exploratory analysis was conducted to select one Generative AI model that is the best for solving data science problems. Although some Generative AI models were better than others in some aspects, the goal was to find the best model overall in terms of data science tutoring. The brief findings about each model are given below:

3.4.1 ChatGPT

ChatGPT is highly effective for solving data science problems. It is easy to use and provides detailed explanations of the solutions. It can also generate data for the schema that can be used for testing the SQL queries. It is also able to create visualizations and tables of the data, making it easier to understand the data. It also works step by step to solve the problems and explain the solutions. It can reference to previous prompts in the same chat to provide better answer as compared to Gemini. ChatGPT is also able to bridge the gap in the information provided and use its internal knowledge to provide better answers. After a detailed discussion and complete understanding of the topic, ChatGPT can provide better answers compared to the other three Generative AI models.

3.4.2 Gemini

Gemini is also a good model for solving data science problems but it is not as effective as ChatGPT. It is able to create data and generate SQL queries but it is not able to create visualizations and tables of the data. It does not work step by step to solve problems and explain solutions. As compared to ChatGPT, it is not able to reference to previous prompts in the same chat effectively.

3.4.3 SQL Fiddle

SQL Fiddle is effective for executing SQL queries and getting the results. It is a good platform if the user is already familiar with SQL and does not need much help in understanding the concepts. It is not as user-friendly as ChatGPT and Gemini and does not provide detailed explanations of the solutions.

3.4.4 GitHub Copilot

GitHub Copilot is a code generation model that is good for solving coding problems and algorithms. It is not as good as ChatGPT for solving and explaining practical exercise questions of data science. It is also not able to write code for algorithms of data science (like normalization) completely on its own. It can write the code if a high level algorithm is provided. It can provide explanation of the practical exercise questions and also somewhat correct answers. It is not as user-friendly as ChatGPT and Gemini and requires some setup to use it. This makes it difficult for undergraduate students to use it as a tutor for data science.

3.5 Best Model

The best model based on the correctness of responses, quality of responses, ease of use, and user-friendliness is ChatGPT. ChatGPT is also the most preferred Generative AI model among students, that was also shown in the survey conducted in the user study in Section 9.2.2. ChatGPT will be used for detailed analysis of each topic of data science. This analysis will be done to find the best practices for using ChatGPT as a tutor for solving practical exercises of data science. For each topic, the best practices will be explored along with the limitations of using ChatGPT as a tutor.

3.6 Further Analysis of ChatGPT

After the selection of ChatGPT, the further analysis was done on ChatGPT for solving data science problems. The practical exercises required some transformation of the data and some computations to be done. It was found that ChatGPT can provide the answers in two different ways. One was descriptive way, like solving on paper, and the other was through code, returning the answer after running the code.

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 4

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. For creation of relational database schema and data generation, a few different approaches were analyzed. The details specific to the topic are given below.

4.1 Database Schema

The goal here is to create a relational database structure in the SQL engine given the schema of the database. The schema can be provided to ChatGPT and it will return a list of SQL queries that can be executed to create the database.

4.1.1 Prompt Engineering

A detailed analysis of the prompt engineering was done to understand how to give the schema to ChatGPT. Different types of prompts were until a format was found that can be used to create the schema. The format of the schema is important for ChatGPT to understand the tables and their relations. Upon giving ChatGPT an ERD (Entity Relationship Diagram) it can create the tables but the primary keys and foreign keys are not always accurate. It is able to parse the image of ERD and create the basic table but the constraints are not always accurate. Upon not explicitly defining the primary keys, ChatGPT will create a new attribute for primary key or use an attribute as primary key depending upon the attributes in the table. This showed that the more information is given to ChatGPT, the better it can understand the schema and create the tables. Instead of just highlighting the primary keys and foreign keys, explicitly defining them in the prompt

User Prompt

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)

Figure 4.1: Prompt for university schema

will help ChatGPT to understand the schema better and create the tables with correct constraints.

The optimal way to give database schema to ChatGPT to create SQL queries for schema creation is analyzed. The example format to give this information to ChatGPT for university schema is 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 given in Figure 4.1. The prompt is given in this way so ChatGPT can easily understand the schema and create the tables with the given attributes.

ChatGPT Response

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

Figure 4.2: ChatGPT response for university schema

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 ChatGPT responded with SQL queries for creating all the tables in the schema. A part of the response is given in Figure 4.2. The complete ChatGPT response is formatted in a way that it can be directly run in SQL environment.

4.1.2 Tutoring

The creation of schema in this way can facilitate learning. 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.

Using this approach ChatGPT can be used to create SQL queries for schema that can be directly run in an SQL environment. 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.

4.1.3 Key Takeaways

- Providing schema in the form of schema diagram is not the best way.
- Providing schema in the form of text with primary keys and foreign keys is the best way.
- Constraints should also be explicitly defined in the prompt.
- For average sized schema, typically used in education, the complete schema should be provided in a single prompt.

4.2 Data Generation

The goal here is to populate the relational database with sample data. In order to understand the working of a database, it is important to have some data in the tables.

4.2.1 Prompt Engineering

A detailed analysis of the prompt engineering was done to understand how to give the schema to ChatGPT. Different types of prompts were used until a format was found that can be used to create the data. 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. This data is not always logical, like students names as "Student1", "Student2" etc. Although this can be used for testing purposes, it is not the ideal data creation. To make a more logical data, specific instructions can be given to ChatGPT for that. This is useful if specific data values are needed in the database. One way of doing that is given in Figure 4.3.

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 given in Figure 4.4.

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 was also tested for exporting the data in a PDF format or directly writing the queries in a PDF file. It was able to create a PDF file with all the queries that can be run in SQL environment. This can be used to create a database and run the queries on it.

User Prompt

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 may be located in several cities, specially "Small Bank Corporation". Create SQL queries for relations creation and data insertion.

Figure 4.3: Prompt for data generation

4.2.2 Tutoring

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.

4.2.3 Key Takeaways

- The ChatGPT can be used for small scale data generation.
- It can also create a PDF file with all the queries that can be run in SQL environment.
- Specific instructions can be given to ChatGPT to create data with specific values. But these should not be too complex.
- If simple data is required that does not have any interdependencies, then ChatGPT is able to create data as big as having 500 rows in each table.
- If sample data is required to test a specific query, then providing information needed for that query in the prompt will help ChatGPT to create the data expected data.

ChatGPT Response

– 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

Figure 4.4: Data insertion queries

- This will be sample data that might not exist in real world but can be used for testing purposes.

Chapter 5

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.

5.1 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.

5.1.1 Prompt Engineering

Different ways of giving prompts to ChatGPT for writing queries were tested. It was found that if the schema was given in the beginning of the chat or many prompts earlier then the possibility of getting correct query were lower. Giving the complete schema in each prompt is more effective. The schema can be given in the form of text stating the tables and their columns or in the form of SQL queries that create the schema. 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 textbook, 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 textbook for different schemas. For example, attach a PDF containing

User Prompt

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.

Figure 5.1: Prompt for SQL query

ChatGPT Response

```
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;
```

Figure 5.2: SQL query generated by ChatGPT

SQL schema creation and data insertion queries and then ask the information needed from that database. Ask it to write a query like given in the Figure 5.1.

For the prompt in Figure 5.1, the schema was given in the form of a PDF file containing the SQL queries to create the schema and insert data. The query generate by ChatGPT as a response is given in the Figure 9.1. A detailed explanation of the query was also provided by ChatGPT. The explanation contained the reason for selecting each field, tables, join type and the aggregate function used.

5.1.2 Tutoring

ChatGPT can be used as a tutor for helping students learn how to write SQL queries. The students can get help in writing queries by asking ChatGPT to write the query step by step. ChatGPT also provides a detailed explanation of each step along with the reasoning behind it. This helps students understand the thought process behind writing the query and how to approach similar problems in the future. The students can also ask ChatGPT to write a query to cross check the results of their query. This will help them to validate their results and if the results are not as expected, they can ask ChatGPT to explain the query and how it works. They can also ask ChatGPT the difference between the two queries to understand better how the queries are structured. The students can find the answers quickly and learn faster by using ChatGPT as a tutor.

5.1.3 Key Takeaways

- The schema can be provided in the form of SQL queries or simple text containing the table names and their columns.
- Providing schema in each prompt is more effective than giving it once at the beginning of the chat.
- The expected query definition or question should be well defined in plain English.
- The schema can be given in the prompt as text or by attaching a PDF file containing the SQL queries to create the schema and insert data.

5.2 Query Execution

SQL Query is executed on a set of data to get the required information. The working of the SQL engine and how it executes the query can be done by manually processing the query over small set of data.

5.2.1 Prompt Engineering

ChatGPT was tested for its ability to execute queries. It was found that it can be used to execute queries over a small set of data. ChatGPT can also be used for its ability to act as an SQL engine that can go over the execution step by step. It can execute a query manually so the process of getting the output is explained. 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 query in Figure 9.1 ChatGPT explained the working of query and showed the result on sample data. The output is given in Table 5.1 and the explanation is given in the Appendix B.

5.2.2 Tutoring

ChatGPT can help students understand how an SQL engine works and how it executes queries. As part of the practical exercises ask students to manually run query over a small set of data and write the results. For these types of exercises, ChatGPT can be used to run the query and show the results as well. In this way help students understand the as well as cross check their work. It is highly useful for students that struggle with the engine.

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 5.1: SQL query output

By using this method they can understand how each part of the query contributed to the result and how the final result is computed.

5.2.3 Key Takeaways

- The schema can be provided as SQL queries or simple text with table names and their columns.
- The data can be provided in the form of SQL insert queries or as a table.
- The data and schema can be provided through a PDF file.
- Adding the data and schema in the prompt along with SQL query is more effective than giving it once at the beginning of the chat.
- The schema or data should not be too complex or large.
- ChatGPT can execute queries more effectively over a small set of data.

5.3 Query Explanation

Explanation of a SQL query is important when the query is complex and the student does not understand how it works.

ChatGPT Response

Explanation:

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.

Figure 5.3: Explanation of SQL query

5.3.1 Prompt Engineering

ChatGPT was tested for its tutoring ability to explain how a query was written and how it works. ChatGPT can be used for explaining queries in detail. 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 explanation of the query in Figure 9.1 is provided by ChatGPT in Figure 5.3. The explanation contains the reason for selecting each field, tables, join type and the aggregate function used.

5.3.2 Tutoring

In the practical exercises of the textbook students are asked to write queries. In the answer section of the textbook, the queries are given but without any explanation. In such cases, ChatGPT can be used to explain the queries. The students can ask ChatGPT to explain how the textbook came up with that answer and what each part of the query is doing. This will help them understand the thought process behind writing the query and how to approach similar problems in the future. 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. In addition to that students can also ask ChatGPT to explain the queries students see in class or other resources. ChatGPT can provide a detailed explanation that can facilitate the learning process.

5.3.3 Key Takeaways

- The schema can be provided as SQL queries or simple text with table names and their columns.
- The schema or data can be provided through a PDF file.
- For explanation of a specific part of the query, the prompt should ask for the explanation of that part.
- For general explanation of the query, the prompt should ask to explain the query as a whole.
- If the confusion is not cleared by the explanation, the user can ask for more details or clarification.

Chapter 6

Using ChatGPT for Normalization

Normalization is an important process in data science that was analyzed with the help of ChatGPT. The processes of normalization can be better understood and achieved using Generative AI as well.

6.1 Closure

Closure of a set means all the attributes that can be retrieved from the given set of attributes depending on the functional dependencies. This is an important part of normalization that can be done with the help of ChatGPT. The retrieved attributes are the ones that are dependent on the starting attributes.

6.1.1 Prompt Engineering

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 algorithm of closure along with explanations. Upon exploring, ChatGPT proved useful in finding closure of attributes given the functional dependencies. ChatGPT can use its internal knowledge to find the closure of the given attributes but it has a tendency to make a mistake. Upon giving the functional dependencies in without set notation or proper formatting, ChatGPT sometimes get confused in interpreting the functional dependencies. ChatGPT does not work well with the functional dependencies that are given in the form of single letters. It is able to interpret the functional dependencies better when they are given in the form of words. If the closure of more than 5 attributes is asked at the same time, there is a higher chance of error. For closure of less than 5 attributes, ChatGPT is able to find the closure with a good accuracy with the algorithm. A good 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 ChatGPT to run the data on. For optimal results, the set notation is easier for ChatGPT to interpret as

User Prompt

Using this algorithm for closure: Provide Algorithm 2

And for this set of functional dependencies:
course (course_id, title, dept_name, credits) FDs = {
{course_id} $\rightarrow$ {$title$}
$\{course\_id\} \rightarrow \{dept\_name\}$
$\{course\_id\} \rightarrow \{credits\}$
$\{title\} \rightarrow \{dept\_name\}$
$\{title\} \rightarrow \{credits\}\}$

Give closure of $\{course\_id\}$

Figure 6.1: Prompt for finding closure

input. It also gives a step-by-step explanation of finding the closure. The algorithm given to ChatGPT for finding closure is in the form of Algorithm 2.

Algorithm 2: Closure of β under F

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

```
result ← β;
repeat
    result_previous ← result;
    foreach γ → δ ∈ F do
        if γ ⊆ result then
            result ← result ∪ δ;
        end
    end
until result_previous = result;
return result;
```

The ChatGPT is able to interpret this form of algorithm and find the solution based on this. The method of giving the prompt for finding closure is given in Figure 6.1.

The ChatGPT is able to follow the Algorithm provided and will work on it step by step. The response of ChatGPT is given in Figure 6.2.

ChatGPT Response

We will apply the closure algorithm to compute the closure of {course_id} under the given set of functional dependencies.

—

Given:
Attributes: course(course_id, title, dept_name, credits)
FDs:
1. course_id $\rightarrow title$
2.$course\_id \rightarrow dept\_name$
3.$course\_id \rightarrow credits$
4.$title \rightarrow dept\_name$
5.$title \rightarrow credits$

—

Closure of course_id:
Step 0:
Initialize result := course_id

—

Step 1: Apply FD 1:
Since course_id $\subset result, addtitle \rightarrow result := course\_id, title$
Step 2: Apply FD 2:
course_id $\subset result, adddept\_name \rightarrow result := course\_id, title, dept\_name$
Step 3: Apply FD 3:
course_id $\subset result, addcredits \rightarrow result := course\_id, title, dept\_name, credits$
Step 4: Apply FD 4:
title $\subset result, dept\_namealreadyinresult \rightarrow nochange$
Step 5: Apply FD 5:
title $\subset result, creditsalreadyinresult \rightarrow nochange$

—

No new attributes are being added. Loop terminates.

—

Closure of course_id = course_id, title, dept_name, credits
This means course_id is a **superkey** for this relation.
Would you like to check if it's also a **candidate key**?

Figure 6.2: Response of ChatGPT for closure

6.1.2 Tutoring

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 students can use this technique to practice finding the closure of different attributes. They can also do solve their own question of finding closure and then ask ChatGPT to find the closure of the same attributes. This way they can compare their answers with the answers given by ChatGPT and learn if they made any mistakes.

6.1.3 Key Takeaways

- Providing the algorithm of closure to ChatGPT will help it follow that algorithm.
- Finding closure of less than 5 attributes at the same time is easier for ChatGPT.
- The functional dependencies should be provided in set notation for better understanding.
- The functional dependencies should be provided as words instead of single letters for better results.
- The attributes of the functional dependencies can be renamed by ChatGPT to make it easier for it to understand.

6.2 Candidate Keys

Candidate keys are the attributes or combinations of attributes that can be used to uniquely identify each row. ChatGPT can be used to find candidate keys as well. 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.

6.2.1 Prompt Engineering

ChatGPT was tested for its ability to help find candidate keys. 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 is 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 ChatGPT run it over without the iterations. 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 3.

Algorithm 3: Candidate Keys Part 1

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

```
left ← Union of all α in FD;
right ← 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 ∉ left and x ∉ right then
        case_1.add(x);
    if x ∉ left and x ∈ right then
        case_2.add(x);
    if x ∈ left and x ∉ right then
        case_3.add(x);
    if x ∈ left and x ∈ 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, ChatGPT can be asked to explain the steps taken to get the answer. The part of the output provided by ChatGPT for the algorithm in Figure 3 is given in Figure 6.3.

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 finding candidate keys is given in Algorithm 4.

For the second part of the algorithm, the student can work together with AI tutor to find all candidate 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

ChatGPT Response

. . .

Step 1: Compute the left and right sets

Left (all LHS attributes) = {course_id, title}

Right (all RHS attributes) = {title, dept_name, credits}

Step 2: Full attribute set = {course_id, title, dept_name, credits}

Now classify each attribute into cases:

Attribute	In Left	In Right	Case
course_id	Yes	No	Case 3
title	Yes	Yes	Case 4
dept_name	No	Yes	Case 2
credits	No	Yes	Case 2

Final Classification:

Case 1 (Not in left or right): $\emptyset$

$Case2(Onlyinright) : dept\_name, credits$

$Case3(Onlyinleft) : course\_id$

$Case4(Inbothleftandright) : title$

.

.

.

Figure 6.3: Response of ChatGPT for candidate keys

Algorithm 4: 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;

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*;

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.

6.2.2 Tutoring

The AI tutor can help students find the candidate keys step by step. 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. The students can use this approach to learn the process of finding candidate keys in the same way as finding closure.

6.2.3 Key Takeaways

- The algorithm of candidate keys should be given to ChatGPT in parts to get the best results.
- The prompts should be divided into parts and work on it step by step.
- The functional dependencies should be provided in set notation for better understanding.
- The functional dependencies should be provided as words instead of single letters for better results.
- The attributes of the functional dependencies can be renamed (from single letters) by ChatGPT to make it easier for it to understand.

6.3 Canonical Cover

The removal of extraneous attributes from the set of functional dependencies retrieves the canonical cover. This is an important step in normalization as this removed extra attributes from relations. The removal of extraneous attributes does not change the semantics of the functional dependencies.

6.3.1 Prompt Engineering

ChatGPT was tested for its ability to help find canonical cover. 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 algorithm 5 is used to find the canonical cover.

Algorithm 5: 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*;

In order to find the canonical cover with 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 prompt. 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 in Figure 6.4 for a functional dependency $\alpha \rightarrow \beta$.

Trying the same example to check if course_id 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 given in Figure 6.5.

- 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 $F' = (F - \{\alpha \to \beta\}) \cup \{\alpha \to (\beta - \{a\})\}$.

Figure 6.4: Extraneous Attributes

ChatGPT Response

Step-by-Step:
$\alpha = \{course\_id\},\ \beta = \{title\},\ a = course\_id$
We want to check if: title $\subseteq$ closure of $\emptyset$ under F
—
Initialize result = $\emptyset$
- No FD has LHS $\subseteq \emptyset$, so no RHS can be added.
$\emptyset^+ = \emptyset$
—
Step 2: Check if $\beta \subseteq (\alpha - a)^+$
$\beta = title, (\alpha - a) = \emptyset\ \to\ \emptyset^+ = \emptyset$
So, title $\not\subseteq \emptyset^+ \to$ **condition fails**
—
Conclusion:
course_id **is not** an extraneous attribute in {course_id} $\to$ *title*.

Figure 6.5: Response of ChatGPT for extraneous attributes

6.3.2 Tutoring

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 reference their work and practice such questions with the help of ChatGPT.

6.3.3 Key Takeaways

- The algorithm of canonical cover should be given to ChatGPT without the iterative part to get the best results.
- The iterative portion of the algorithm should be done manually. For each attribute in each functional dependency, ask ChatGPT if it is extraneous or not.
- The rules for finding extraneous attributes should be given to ChatGPT in the prompt.
- The functional dependencies should be provided in set notation for better understanding.
- The functional dependencies should be provided as words instead of single letters for better results.

Chapter 7

Using ChatGPT for Itemset Mining

Itemset is used to find frequent itemsets in given data. This is a common problem in data mining and is used to find the association rules. The process of finding frequent itemsets given the transaction through Apriori algorithm was explored with the help of ChatGPT.

7.1 Lexicographical Sorting

Apriori algorithm works on the principle of lexicographically sorted items in each transaction.

7.1.1 Prompt Engineering

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 ChatGPT is not able to keep track of all the items and their counts. If ChatGPT 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. In order to achieve better results ChatGPT can be used like a tutor that helps solve the exercise together. The whole process can be broken down into smaller steps and ChatGPT can be asked to solve each step one by one. The working of ChatGPT as a tutor is shown with the help of a similar example of finding frequent itemsets using Apriori algorithm. The list of transactions are given in Figure 7.1.

Without any guidance ChatGPT 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 lexicographically throughout the process. For this simple prompt with the list of items in each transaction is given to ChatGPT and it is asked to

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

Figure 7.1: Transactions

ChatGPT Response

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

Figure 7.2: Lexicographically sorted transactions

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 the example in Figure 7.1 is given in Figure 7.2.

7.1.2 Tutoring

The ChatGPT can be used as a tutor to help students learn the process of finding frequent itemsets. The students can work on the problem step by step with the help of ChatGPT and learn the process of solving the practical exercise questions. The first step of finding itemsets through Apriori algorithm is to sort the items in each transaction lexicographically. This step can help students in quickly sorting the items in each transaction and then they can work on the next step of finding the frequent itemsets.

User Prompt

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

Figure 7.3: Prompt 1 for generating C_k

7.1.3 Key Takeaways

- The transactions can be given in any order and ChatGPT can sort them lexicographically.
- The transactions should be clearly divided into separate sets of items.
- The complete itemsets can not be found in a single prompt.
- A step by step approach is needed to get the correct results.

7.2 Create C_k

The next step is to ask about the unique items that exists in all the transactions. This will be C_1 (Candidate 1-itemsets).

7.2.1 Prompt Engineering

The prompt engineering techniques were tested to see which works better for ChatGPT. For the first step C_1 is simple and is the list of all unique items in all transactions. After this L_1 is found by removing some of the items from C_1 that are below the minimum support. In the next step C_2 is generated from L_1 and the process is repeated to create C_k until no more C_k can be created from $L_(k-1)$. After C_1, generating C_k can be tricky as ChatGPT needs to understand the process of generating C_k from $L_(k-1)$. Giving the instructions in the form of SQL query or simple english can be difficult for ChatGPT to understand. This is a complex statement and is difficult for ChatGPT to understand. There are two ways to get this step done:

- Give ChatGPT the complete algorithm of generating C_k and ask it to generate C_k for the given L_k. The user prompt for this is given in Figure 7.3.
- Give ChatGPT the complete algorithm of generating C_k and ask it to generate C_k for the given L_k step by step. For example, (A, B) is in $L_(k-1)$, then create all candidate itemsets C_k that contains (A,B) as shown in Figure 7.4. In this way for one C_k the prompts will be given one by one for each itemset in $L_(k-1)$.

User Prompt

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

Figure 7.4: Prompt 2 for generating C_k

ChatGPT Response

The Candidate 2-itemset (C_2) is:
C_2 = { (Apple, Grapes), (Apple, Mango), (Apple, Orange), (Grapes, Mango), (Grapes, Orange), (Mango, Orange) }

Figure 7.5: ChatGPT response for C_2

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 C_1 is create by ChatGPT is C_1 = Apple, Grapes, Mango, Orange. After creating L_1, the C_2 is created by ChatGPT as given in Figure 7.5.

7.2.2 Tutoring

The ChatGPT can be used as a tutor to help students learn the process of finding frequent itemsets. The students can check their manual work to see if their working is correct. The ChatGPT 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 ChatGPT can help catch those mistakes. Even if ChatGPT makes a mistake, the student can point it out and ChatGPT can correct it. This way the student will look into the possible mistakes and learn from them.

7.2.3 Key Takeaways

- Simple prompts for generating C_k do not work well.
- A well defined prompt is needed to get the correct results.
- The mathematical notation to create C_k works well as well as the step by step approach.
- Each C_k should be created separately for each itemset in L_{k-1} to get the final result.

ChatGPT Response

The L_2 (Frequent 2-itemset) is:
$L_2 =$ (Apple, Grapes, 4), (Apple, Mango, 3), (Apple, Orange, 4), (Grapes, Orange, 3), (Mango, Orange, 4)

Figure 7.6: ChatGPT response for L2

7.3 Calculate L_k

After generating C_k, the next step is to calculate the support for each itemset in C_k. Then keep only those itemsets that are above the minimum support. This will create L_k that is the frequent itemsets.

7.3.1 Prompt Engineering

Then ChatGPT can be asked to count the occurrence of each item in all transactions, if the list of items is too long it is better to divide it into smaller parts. Ask ChatGPT to find the frequent 1-itemsets (L_1) by keeping only the items that are above the minimum support. Then ask ChatGPT to generate C_k as shown in the previous step. Using that C_k, L_k is created by counting the occurrence of each itemset in C_k and keeping only those that are above the minimum support. This process is repeated until there are no more C_k 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 L_2 for above example is given in Figure 7.6.

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

7.3.2 Tutoring

The ChatGPT can be used to help students understand the process of finding frequent itemsets. In this way ChatGPT can act as a tutor to help solve exercises so the student can work along ChatGPT to learn the process of finding frequent itemsets. The students can also work on the problems step by step along with ChatGPT to practice the exercises. If the students are stuck at a specific step, ChatGPT can be asked to solve that step and then the student can continue from there. If the students do not understand the process, they can ask ChatGPT to explain it in detail. The students can use this way of learning to facilitate their learning process and doing better practice for exams as well.

7.3.3 Key Takeaways

- L_k can be calculated by counting the occurrence of each itemset in C_k.
- One C_k should be created at a time and in the following prompt the same L_k should be created.
- The stopping criteria can be determined if no more C_k can be generated from L_{k-1}.
- The ChatGPT can be asked to combine all the frequent itemsets found in previous steps at the end to get the final result.

7.4 Association Rules

The association rules are generated from the frequent itemsets. These rules are used to find the relationship between the items in the transactions.

7.4.1 Prompt Engineering

ChatGPT can also be used in finding these rules from the data by calculating confidence and support. The support, confidence and lift can be calculated with the help of ChatGPT. Incase of ChatGPT OpenAI model 4o already knows the formulas for these and can calculate them easily. This was found by for the formulas and to calculate the values for support, confidence and lift. The response was correct and ChatGPT was able to calculate the values for support, confidence and lift. Calculating these values for all association rules altogether can cause ChatGPT to skip a few rule. One way to get the results is to ask ChatGPT to calculate the values for each rule one by one. Another ways is to ask ChatGPT 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 ChatGPT 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 $\rightarrow$ Apple ChatGPT response is as follows:

7.4.2 Tutoring

The ChatGPT can be used as a tutor to help students learn the process of generating association rules from frequent itemsets. In this way 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.

ChatGPT Response

Step 1: Calculate Support Support of a rule (X $\rightarrow$ Y) is:

Support(X $\rightarrow$ Y) = $(\frac{Support(X \cup Y)}{TotalTransactions})$

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

Support(Orange $\rightarrow$ Apple) = $(\frac{4}{6} = 0.6667$ (or 66.67%))

Step 2: Calculate Confidence Confidence of a rule (X $\rightarrow$ Y) is:

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

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

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

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

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

From the given data: - (Support(Apple) = 5) - (Confidence(Orange $\rightarrow$ Apple) = 0.8) - (Support(Apple) = 5/6)

Lift(Orange $\rightarrow$ 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.

Figure 7.7: ChatGPT response for association rules

7.4.3 Key Takeaways

- Given the frequent itemsets, ChatGPT can be used to generate association rules.
- Support, confidence and lift can be calculated without giving the formulas. ChatGPT can calculate these using the formulas it already knows.
- The ChatGPT can be used to validate the association rules using support, confidence and lift.
- If complete set of transactions are given then ChatGPT uses that to calculate support, confidence and lift. If frequent itemsets are given then ChatGPT uses that to calculate the support, confidence and lift.

Chapter 8

Clustering

Clustering is the process of grouping similar data points together. This is an important technique data science for data analysis.

8.1 Points Assignment

Initially a set of data points is randomly selected as centroids. The distance of each point from the centroid is calculated and the point is assigned to the cluster with the closest centroid.

Prompt Engineering Using ChatGPT for clustering small number of data points can be beneficial for educational purposes. The questions related to clustering can be solved using ChatGPT. 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 Chatgpt 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 ChatGPT to understand the problem better. Using the example of data point in x-y plain given in Table 8.1, the clustering process can be explained.

The each point will be assigned to one of the clusters that is closest to. The distance between the points and the centroids can be calculated using the Euclidean distance formula or Manhattan distance formula. ChatGPT can be given the task to calculate the distance of each point from the centroids using the distance formula required. The ChatGPT can then assign each point to the cluster with the shortest distance. The ChatGPT can be asked to show the working and calculations for better understanding. The result of the first iteration is given in Figure 8.2. Here ChatGPT found the clusters that each point was assigned to and the new centroids for each cluster.

Point	X	Y
c1	1	1
c2	4	0
c3	5	6
c4	3	3
c5	9	8
c6	7	2

Centroids:

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

Figure 8.1: Data points and centroids

ChatGPT Response

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).

Figure 8.2: Points assigned to clusters and new centroids

8.1.1 Tutoring

The ChatGPT can be used in this way to help students learn the clustering process. In the example ChatGPT gave a brief answer to the question that is beneficial if the student already knows the clustering process. In this way the student can cross check their own work or work quickly on these kind of problems and focus on the more complex problems. The ChatGPT can also provide more detailed answers if needed. That is useful for students who are new to this concept and need more help. They can ask for more details and ChatGPT provides a detailed explanation of the steps involved in the clustering process. The ChatGPT can be asked to show the working and calculations for better understanding as well.

8.1.2 Key Takeaways

- ChatGPT can be asked to show the working and calculations for better understanding.

ChatGPT Response

The final clustering of the points is:
Cluster 1: c1, c2, c4
Cluster 2: c3, c5, c6

Figure 8.3: Final clusters

- ChatGPT can be asked to just provide brief answers to the questions, if only answers are needed.
- The process needs to be iterative and ChatGPT needs to be provided with one step at a time.
- Calculating all steps at the same time is not recommended as it can lead to incorrect results.
- The ChatGPT can be asked if this is the right time to stop the iterations.

8.2 Recalculation of Centroids

After the first iteration, the centroids are recalculated. This is done by taking the mean of all points in the cluster. In this way the centroids move to the center of the cluster.

8.2.1 Prompt Engineering

The ChatGPT 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, ChatGPT randomly selects a new centroid to ensure that the clustering process continues. These steps can be repeated until the centroids do not change. The ChatGPT can also be asked if this is the right time to stop the iterations. The ChatGPT does not always shows the working and calculations but can be asked to show the working for better understanding. The process of finding distances between the points and centroids, assigning the point to the clusters and recalculating the centroids can be done in one prompt. This is an iterative process, so keep asking ChatGPT to repeat the process until the centroids do not change. If more detailed calculations are needed then it is recommended to not do all these steps of one iteration in one prompt. After 2 more steps for the above example the final clustering of the points is given in Figure 8.3. This shows that the points c1, c2 and c4 are similar and grouped together. Similarly points c3, c5 and c6 are grouped together.

8.2.2 Tutoring

Using this approach the students can use 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 ChatGPT and the students can focus on the more complex steps. This AI tutor can also be used to solve the exercises in the textbook 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, ChatGPT 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 ChatGPT.

8.2.3 Key Takeaways

- This is an iterative process, so asking for calculations for each iteration provides better results.
- ChatGPT can be asked to show more details and calculations for better understanding.
- ChatGPT can be asked to show less working for faster results.
- Once the centroids do not change, the process can be stopped. This can also be asked to ChatGPT.
- The prompts are not usually very complex but these need to be given step by step.

Chapter 9

Evaluation

In this study, three 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 user 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 analyzed. 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.

9.1 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 ChatGPT was asked to solve complex problems, such as normalization, itemset mining, and clustering, in one go. The ChatGPT performed better when these problems were broken into smaller steps and solved step by step. 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.

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 9.1: Combined Performance Evaluation Table

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

9.1.1 Accuracy of ChatGPT Responses

The ChatGPT 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 9.1. 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 L_k is considered as one unit but it may take multiple prompts to get L_k. The total numbers given in the table correspond to the number of times ChatGPT 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.

9.1 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 9.2, 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 scale 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 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 9.2: Relational SQL Database Creation

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

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 9.3: SQL Queries

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

In the table 9.3, 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 attempts made to solve all the steps together, especially the complex ones, as most of these ended up in failure. The simpler questions could be solved without

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 9.4: Normalization

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

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. The table 9.4 shows that when asking ChatGPT to find closure of 1-5 attributes, ChatGPT 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. Similarly, for canonical cover and candidate keys, when the complete questions were asked, ChatGPT made mistakes. But when the questions were broken down into smaller steps, ChatGPT was able to solve them correctly.

9.1.2 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

Topic	**W**	**WC**	**N**	**Total**	**Percentage**
Lexicographical Sorting	5	0	0	5	100
Create Ck	17	4	0	21	80.9
Calculate L_k	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 9.5: 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 9.6: Clustering

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

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

Table 9.7: 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

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, easier 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.

Table 9.7 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 ChatGPT 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 9.2 for details), where participants were not given any guidance on how to use ChatGPT. As the complexity of the questions increased, the count of correct answers decreased.

The complexity of the prompts given in the Table 9.7 is calculated based on the collective complexity of the prompts needed to get the results. In some cases of difficulty level 1 and 2, the prompts needed to be more complex to get the results. But these instances are very few. For example, a more complex prompt in SQL query execution is given in Figure 9.1 where ChatGPT is asked to show the output of the query. For this both the schema and the data is provided to ChatGPT along with the query.

Similarly, for the topic of clustering, the complexity of the prompts is calculated on the

ChatGPT Response

using this schema:
create table classroom
(building varchar(15),
room_number varchar(7),
capacity numeric(4,0),
primary key (building, room_number));

create table department
(dept_name varchar(20),
building varchar(15),
budget numeric(12,2) check (*budget* > 0),
primary key (dept_name));
. . .
insert into classroom values ('Packard', '101', '500');
insert into classroom values ('Painter', '514', '10');
insert into classroom values ('Taylor', '3128', '70');
insert into classroom values ('Watson', '100', '30');
insert into classroom values ('Watson', '120', '50');
insert into department values ('Biology', 'Watson', '90000');
insert into department values ('Comp. Sci.', 'Taylor', '100000');
insert into department values ('Elec. Eng.', 'Taylor', '85000');
insert into department values ('Finance', 'Painter', '120000');
insert into department values ('History', 'Painter', '50000');
insert into department values ('Music', 'Packard', '80000');
insert into department values ('Physics', 'Watson', '70000');
. . .
What will be the output of "
WITH section_enrollment AS (
SELECT course_id, sec_id, COUNT(ID) AS enrollment_count
FROM takes
WHERE semester = 'Fall' AND year = 2017
GROUP BY course_id, sec_id
) SELECT MAX(enrollment_count) AS max_enrollment
FROM section_enrollment; "

Figure 9.1: SQL Query Prompt

ChatGPT Response

For points:
(1, 2), (3, 4), (2, 2), (4, 3), (2, 5), (7, 3), (1, 6), (4, 0) and centroids (5, 2) and (1, 0) calculate based on Euclidean distance, to which centroid each point is closer to. The centroid a point is closer to is the cluster it belongs to in this run.

Figure 9.2: Clustering Prompt

fact that most of these prompts are relatively simpler. But for these to work a previous knowledge of the algorithm is needed. Some questions can be a more complex and need to be broken down into smaller steps, like calculating only the distance of points from centroids and not assigning the points to clusters. In other cases, a more detailed explanation of the algorithm is required in the prompt to get the correct results. An example of this is given in Figure 9.2 where ChatGPT is asked to show the working and calculations for better understanding.

For the topics where the complexity of the prompts is simpler, it is due to the fact that most of the individual prompts are simpler. Some of the prompts in these topics need to be more complex, specially when the problem get more complex. The complexity of the prompts is calculated based on the collective complexity of the prompts needed to get the required results.

9.2 User 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 ChatGPT to answer the same basic questions, the fourth section was about using ChatGPT to answer the advanced questions, and the fifth section was about the experience of the participants using ChatGPT and their feedback. It also evaluated how participants from different degree programs and education levels, benefited from ChatGPT in learning the topic.

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

Figure 9.3: Expected answer for basic SQL query

9.2.1 User Study Quentionnaire

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.

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.

{BookID, AuthorName, Genre, Publisher}

Figure 9.4: Expected answer for closure of BookID

{BookID}, {Publisher}

Figure 9.5: Expected answer for candidate keys of BookID

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.

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, in case they get stuck. The guidance sheet is also provided in the Appendix A.

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.

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

Figure 9.6: Expected answer for advanced SQL query

{CourseID}, {StudentID, Semester}, {StudentID, Instructor}

Figure 9.7: Expected answer for candidate keys of Class

2. 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.

3. 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.

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.

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

Figure 9.8: Expected answer for candidate keys of Inventory

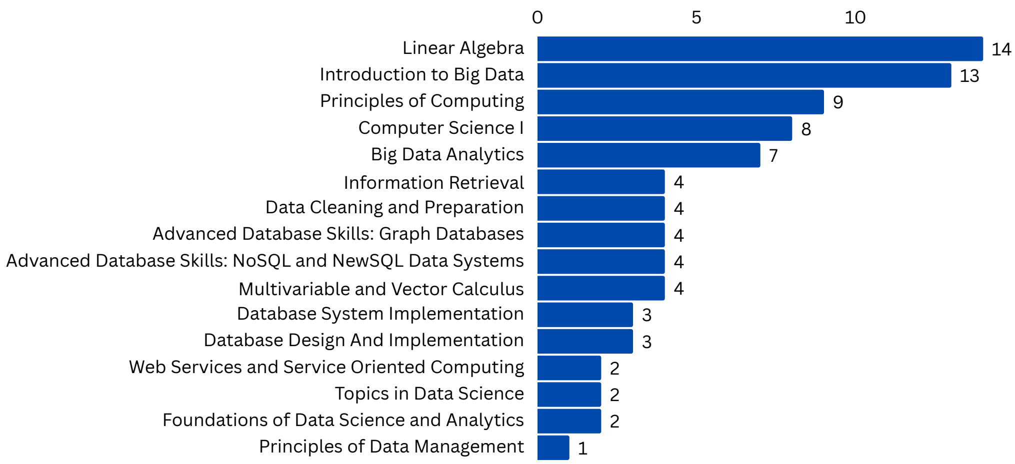
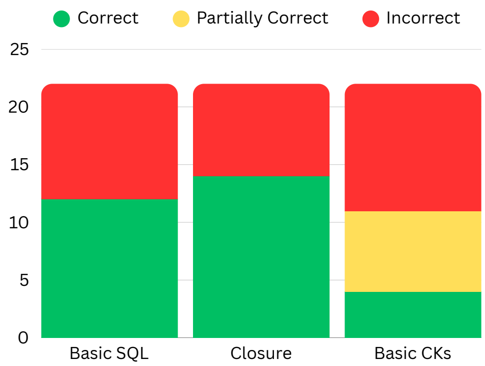


Figure 9.9: Background subjects of participants

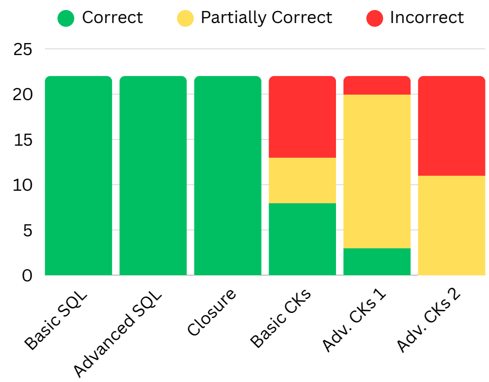
They were also asked to rate the usefulness of ChatGPT for each section of the questionnaire. They were asked how easy it was to use ChatGPT to get the answers and how effective it was as a tutor. They were also given the opportunity to provide any additional feedback or comments.

9.2.2 Result of User 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 9.9. 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



(a) Histogram of questions solved manually



(b) Histogram of questions solved with ChatGPT

Figure 9.10: Histogram of questions solved manually and with ChatGPT

normalization.

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.

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

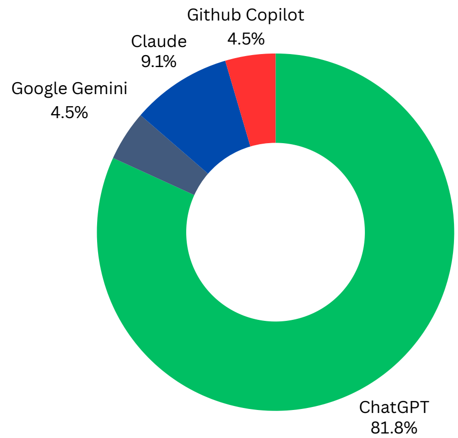


Figure 9.11: Preferred Generative AI tool

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 9.10. It is found that the rate of correct answers increased when using ChatGPT as compared to solving questions manually.

It was also found that majority of the participants preferred ChatGPT over other Generative AI models, shown in Figure 9.11. Those who preferred other models have also

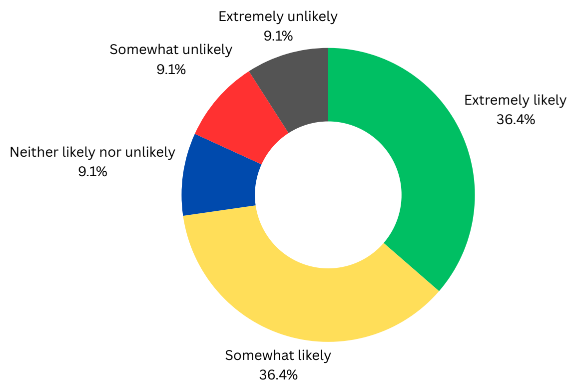


Figure 9.12: Likelihood of using ChatGPT as a data science tutor in future

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 9.12. This shows that the participants found ChatGPT helpful in solving the questions and are likely to use it in future.

9.3 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 9.8. It contains the evaluation of ChatGPT by Claude on the basis of Technical accuracy, Clarity of expla-

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 9.8: 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

nation, 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.

The other methods of evaluation involved human analysis of the responses generated by ChatGPT. In this method, the responses of ChatGPT 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 9.8 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 9.3. 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 10

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 data science education. 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 itemset mining can also be learned and understood with ChatGPT. It can help solve practical exercise questions of itemset 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 science tutor were used. The study shows that correctness of the answers improved by using ChatGPT for SQL queries and simpler normlaization problems. All the participants 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 evaluate the effectiveness of ChatGPT as a tutor of data science. 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] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB)*, pages 487–499, Santiago, Chile, 1994. Morgan Kaufmann.

[2] Edge AI and Vision Alliance. Global ai adoption to surge 20%, exceeding 378 million users in 2025, February 2025. Retrieved April 20, 2025 from https://www.edge-ai-vision.com/2025/02/global-ai-adoption-to-surge-20-exceeding-378-million-users-in-2025/.

[3] Daina Bilkštytė-Skanė and Vita Akstinaite. Strategic organizational changes: Adopting data-driven decisions. *Strategic Change*, 33(2):107–116, 2024.

[4] U.S. Census Bureau. Data science, 2022. Retrieved March 1, 2025 from https://www.census.gov/topics/research/data-science.html.

[5] McKinsey & Company. The state of ai in early 2024: Gen ai adoption spikes and starts to generate value. 2024. Retrieved April 20, 2025 from https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai-2024.

[6] Google Developers. Prompt engineering resources. Retrieved from https://developers.google.com/machine-learning/resources/prompt-eng.

[7] Yael Erez and Orit Hazzan. How science and engineering students use genai tools throughout their academic journey: A four-year analysis, 2024. Retrieved March 1, 2025 from https://cacm.acm.org/blogcacm/how-science-and-engineering-students-use-genai-tools-throughout-their-academic-journey-a-four-year-analysis/.

[8] Suélia Fleury, Gloria Washington, and Poonam Chaudhary. How 3 educators are using generative ai. *IEEE*, 2024. Retrieved from https://transmitter.ieee.org/how-3-educators-are-using-generative-ai/.

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

[10] Fortune Business Insights. Data science platform market size, share covid-19 impact analysis, by component (platform, services), by deployment (on-premises, cloud), by end-user (bfsi, healthcare, it telecommunications, transportation logistics, retail e-commerce, manufacturing, others), and regional forecast, 2024–2032, April 2025. Retrieved April 20, 2025 from https://www.fortunebusinessinsights.com/data-science-platform-market-107017.

[11] 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. IEEE, 2024.

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

[13] Mohammed Saeed, Nicola De Cao, and Paolo Papotti. Querying large language models with sql. 2023.

[14] Avi Silberschatz, Henry F. Korth, and S. Sudarshan. *Database System Concepts*. McGraw-Hill, 7th edition, 2019.

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

```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}

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 } → {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)

A.5 Participants' Feedback

Please provide your feedback on your experience using 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 ChatGPT for the basic SQL query? (Likert Scale - 1: Not at all, 5: Very Helpful)
2. How helpful was ChatGPT for the advanced SQL query? (Likert Scale - 1: Not at all, 5: Very Helpful)
3. How helpful was ChatGPT for the basic Closure question? (Likert Scale - 1: Not at all, 5: Very Helpful)
4. How helpful was ChatGPT for the advanced Closure question? (Likert Scale - 1: Not at all, 5: Very Helpful)
5. How helpful was ChatGPT for the basic Candidate Keys question? (Likert Scale - 1: Not at all, 5: Very Helpful)
6. How helpful was 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 6: Finding Candidate Keys from Functional Dependencies

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

$left \leftarrow \bigcup \alpha$ in F;
$right \leftarrow \bigcup \beta$ in F;
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**
    $case_1 \leftarrow case_1 \cup \{x\}$;
  if $x \notin left$ ***and*** $x \in right$ **then**
    $case_2 \leftarrow case_2 \cup \{x\}$;
  if $x \in left$ ***and*** $x \notin right$ **then**
    $case_3 \leftarrow case_3 \cup \{x\}$;
  if $x \in left$ ***and*** $x \in right$ **then**
    $case_4 \leftarrow case_4 \cup \{x\}$;
$core \leftarrow case_1 \cup case_3$;
if $Closure(core) = F$ **then**
  return *core as the only candidate key*;
else
  $candidate\_keys \leftarrow \emptyset$;
  foreach *combination x of* $case_4$ **do**
    if $Closure(core \cup x) = F$ **then**
      $candidate\_keys \leftarrow candidate\_keys \cup (core \cup x)$;
  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

```
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;
```

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

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:

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/ChatGPTAsDataScienceTutor.git`.

Appendix E

SQL Explanation

E.1 ChatGPT Query Explanation

The following is the prompt given to ChatGPT to explain the SQL query and the response provided by ChatGPT.

User Said:

Explain this query " select distinct takes.ID from takes, instructor, teaches where takes.course_id = teaches.course_id and takes.sec_id = teaches.sec_id and takes.semester = teaches.semester and takes.year = teaches.year and teaches.id = instructor.id and instructor.name = 'Einstein'" as the answer to this question "Find the IDs of all students who were taught by an instructor named Einstein; make sure there are no duplicates in the result."

ChatGPT said:

Sure! Let me explain this SQL query in relation to the question: **"Find the IDs of all students who were taught by an instructor named Einstein; make sure there are no duplicates in the result."**

Query:

Listing E.1: SQL Query

```
SELECT DISTINCT takes.ID
FROM takes, instructor, teaches
WHERE takes.course-id = teaches.course-id
```

```
AND takes.sec-id = teaches.sec-id
AND takes.semester = teaches.semester
AND takes.year = teaches.year
AND teaches.id = instructor.id
AND instructor.name = "Einstein";
```

Explanation:

1. **Goal**: The query is finding **student IDs** (`takes.ID`) of students who have taken a course taught by an instructor named "Einstein," ensuring that there are **no duplicate student IDs** in the result.

2. **Tables Involved**:

   – `takes`: This table records which students took which courses (`course_id`, `sec_id`) and in which semester/year.
   – `instructor`: This table holds instructor information (`id`, `name`, etc.).
   – `teaches`: This table records which instructors taught which sections of a course in a given semester and year (`course_id`, `sec_id`, `semester`, `year`, `id`).

3. **SELECT DISTINCT**:

   – `DISTINCT` ensures that duplicate student IDs are removed from the final result. For instance, if a student was taught by Einstein in multiple courses, the `DISTINCT` clause ensures that the student's ID appears only once in the output.

4. **FROM and WHERE Clauses**:

   – **FROM takes, instructor, teaches**: This indicates that the query is referencing the `takes`, `instructor`, and `teaches` tables. It implies an implicit cross join (although modern SQL practices recommend using `JOIN` instead).
   – **WHERE takes.course_id = teaches.course_id AND takes.sec_id = teaches.sec_id AND takes.semester = teaches.semester AND takes.year = teaches.year**: These conditions link the `takes` table to the `teaches` table. This ensures that the courses the students took match the courses taught by an instructor in terms of course ID, section, semester, and year.

- **AND teaches.id = instructor.id**: This condition joins the `teaches` table with the `instructor` table to match the instructor who taught the course.
- **AND instructor.name = 'Einstein'**: This condition filters the results to include only those courses taught by the instructor named "Einstein."

Process:

- The query first gathers all student enrollments from the `takes` table.
- It checks which sections these students enrolled in and matches them with the corresponding records in the `teaches` table (i.e., matching `course_id`, `sec_id`, `semester`, and `year`).
- It then ensures that only records where the `instructor.name` is "Einstein" are selected.
- Finally, the query retrieves the `takes.ID` (i.e., the student ID) for all students who were taught by Einstein, ensuring no duplicates by using the `DISTINCT` keyword.

Example:

Let's assume:

- Einstein teaches `PHY101`, Section `1` in Fall 2024.
- Alice and Bob are enrolled in this course and section.

The query will:

1. Match the entries in the `takes` table for Alice and Bob.
2. Join these entries with the `teaches` table to confirm that the instructor for this course section is Einstein.
3. Filter out any students not taught by Einstein.
4. Return Alice's and Bob's IDs, with no duplicates.

Final Output:

– If the conditions match multiple students, all of their `IDs` will be returned with duplicates removed.

Why This Query Answers the Question:

The query explicitly finds all student IDs (`takes.ID`) for courses that were taught by an instructor named Einstein. The use of `DISTINCT` ensures that no student is listed more than once, even if they took multiple courses taught by Einstein. Thus, this query perfectly addresses the question of finding **all students taught by Einstein without duplicates**.

E.2 Gemini Query Explanation

The following is the prompt given to Gemini to explain the SQL query and the response provided by Gemini.

User Prompt:

Explain this query " select distinct takes.ID from takes, instructor, teaches where takes.course_id = teaches.course_id and takes.sec_id = teaches.sec_id and takes.semester = teaches.semester and takes.year = teaches.year and teaches.id = instructor.id and instructor.name = 'Einstein'" as the answer to this question "Find the IDs of all students who were taught by an instructor named Einstein; make sure there are no duplicates in the result."

Gemini Response:

Absolutely, this SQL query retrieves the IDs of students who were taught by an instructor named "Einstein" while eliminating duplicates. Here's a breakdown of how it works:

1. Tables involved:

– `takes`: This table holds information about student enrollments, including student ID (ID), course ID (course_id), section ID (sec_id), semester, and year.

– `teaches`: This table links instructors to courses they teach. It contains instructor ID (id), course ID (course_id), section ID (sec_id), semester, and year.

  - `instructor`: This table stores information about instructors, including their ID (id) and name.

2. **Finding Students Taught by Einstein:**

  - `SELECT DISTINCT takes.ID`: This part defines what you want to retrieve - distinct student IDs (`takes.ID`) from the `takes` table. The `DISTINCT` keyword ensures you get only unique student IDs.
  - `FROM takes, instructor, teaches`: This specifies the tables involved in the query.
  - `WHERE` **clause**: This is where the filtering happens to identify students taught by "Einstein":

    * `takes.course_id = teaches.course_id`: This condition ensures the course a student is enrolled in (`takes.course_id`) matches the course taught by the instructor (`teaches.course_id`).
    * `takes.sec_id = teaches.sec_id`: This guarantees the section a student is enrolled in (`takes.sec_id`) matches the section taught by the instructor (`teaches.sec_id`).
    * `takes.semester = teaches.semester`: This verifies that the semester the student took the course (`takes.semester`) matches the semester the instructor taught it (`teaches.semester`).
    * `takes.year = teaches.year`: This confirms that the year the student took the course (`takes.year`) aligns with the year the instructor taught it (`teaches.year`).
    * `teaches.id = instructor.id`: This connects the `teaches` table to the `instructor` table. It ensures the instructor ID (`teaches.id`) matches the instructor's record in the `instructor` table.
    * `instructor.name = 'Einstein'`: This is the core filtering condition. It selects only instructors whose name is "Einstein" from the `instructor` table.

3. **Putting it Together:**

By combining these conditions in the `WHERE` clause, the query effectively retrieves the student IDs from the `takes` table where the corresponding instructor (based on `teaches.id` and `instructor.id`) has the name "Einstein". Using `DISTINCT` ensures only unique student IDs are returned.

Therefore, this query successfully finds and eliminates duplicates of student IDs who were enrolled in courses taught by an instructor named "Einstein".

E.3 SQL Fiddle Query Explanation

The following is the prompt given to SQL Fiddle to explain the SQL query and the response provided by SQL Fiddle.

User Prompt:

Explain this query " select distinct takes.ID from takes, instructor, teaches where takes.course_id = teaches.course_id and takes.sec_id = teaches.sec_id and takes.semester = teaches.semester and takes.year = teaches.year and teaches.id = instructor.id and instructor.name = 'Einstein'" as the answer to this question "Find the IDs of all students who were taught by an instructor named Einstein; make sure there are no duplicates in the result."

SQL Fiddle Response:

This query retrieves the unique IDs of students who were taught by an instructor named "Einstein". Here's how it works:

Tables Involved:

- `takes`: Contains student enrollments.
- `teaches`: Links instructors to the courses they teach.
- `instructor`: Contains instructor details.

Joins:

- The query uses implicit joins (comma-separated tables) to connect takes, teaches, and instructor based on matching course IDs, section IDs, semester, year, and instructor IDs.

Conditions:

- It ensures that the course and section details match between takes and teaches.
- It filters the results to only include records where the instructor's name is "Einstein".

Distinct:

- The distinct keyword ensures that duplicate student IDs are removed from the final result.

SQL Query:

Listing E.2: SQL Query

```
SELECT DISTINCT takes.ID
FROM takes, instructor, teaches
WHERE takes.course_id = teaches.course_id
  AND takes.sec_id = teaches.sec_id
  AND takes.semester = teaches.semester
  AND takes.year = teaches.year
  AND teaches.id = instructor.id
  AND instructor.name = 'Einstein';
```

Summary:

This query effectively finds all unique student IDs for those who were taught by "Einstein", ensuring no duplicates in the output.

E.4 GitHub Copilot Query Explanation

The following is the prompt given to GitHub Copilot to explain the SQL query and the response provided by GitHub Copilot.

User Prompt: Explain this query " select distinct takes.ID from takes, instructor, teaches where takes.course_id = teaches.course_id and takes.sec_id = teaches.sec_id and takes.semester = teaches.semester and takes.year = teaches.year and teaches.id = instructor.id and instructor.name = 'Einstein'" as the answer to this question "Find the IDs of all students who were taught by an instructor named Einstein; make sure there are no duplicates in the result."

GitHub Copilot Response:

The query provided is a valid SQL query that answers the question: **"Find the IDs of all students who were taught by an instructor named Einstein; make sure there are no duplicates in the result."**

Listing E.3: SQL Query

```
SELECT DISTINCT takes.ID
FROM takes, instructor, teaches
WHERE takes.course_id = teaches.course_id
  AND takes.sec_id = teaches.sec_id
  AND takes.semester = teaches.semester
  AND takes.year = teaches.year
  AND teaches.ID = instructor.ID
  AND instructor.name = 'Einstein';
```

Explanation

1. **Tables Involved:**
   - `takes`: This table contains information about students and the courses they are enrolled in. It includes columns like `ID` (student ID), `course_id`, `sec_id`, `semester`, and `year`.
   - `teaches`: This table contains information about instructors and the courses they teach. It includes columns like `ID` (instructor ID), `course_id`, `sec_id`, `semester`, and `year`.
   - `instructor`: This table contains information about instructors, including their `ID` and `name`.
2. **Joins:** The query uses implicit joins (via the `WHERE` clause) to connect the three tables:
   - `takes.course_id = teaches.course_id`: Matches students with the courses taught by instructors.
   - `takes.sec_id = teaches.sec_id`: Ensures the section of the course matches.
   - `takes.semester = teaches.semester` and `takes.year = teaches.year`: Ensures the semester and year match between the student enrollment and the course being taught.
   - `teaches.ID = instructor.ID`: Links the `teaches` table to the `instructor` table to identify which instructor is teaching the course.
3. **Filter:** The condition `instructor.name = 'Einstein'` filters the results to include only courses taught by an instructor named "Einstein."
4. **Selection:** The query selects the `ID` column from the `takes` table, which represents the student IDs of students enrolled in courses taught by Einstein.
5. **DISTINCT:** The `DISTINCT` keyword ensures that duplicate student IDs are removed from the result. This is important because a student might have taken multiple courses or sections taught by Einstein, but the result should only include each student ID once.

Final Output

The query returns a list of unique student IDs (`takes.ID`) for students who were taught by an instructor named "Einstein."

How It Answers the Question

The query correctly identifies all students (`takes.ID`) who were taught by Einstein by:

- Matching students (`takes`) with the courses they took.
- Matching those courses with the courses taught by Einstein (`teaches` and `instructor`).
- Ensuring no duplicate student IDs are included in the result using `DISTINCT`.

This aligns perfectly with the question's requirement to find all student IDs taught by Einstein without duplicates.