JPSDE-Journal of Pharmacy Simulation and Digital Education
Volume 1, Issue 1 | June-2025

Enhancing e-Health Competencies by Including Database Systems Education in Pharmacy Curricula

Dr. Helen Kim[1], Dr. Min-Jae Park[2]
[1]College of Pharmacy, Seoul National University, Seoul, South Korea
[2]Department of Clinical Pharmacy, Yonsei University, Seoul, South Korea

Received: 07-04-2025; Revised: 05-05-2025; Accepted: 20-05-2025; Published: 05-06-2025

Abstract

This paper focuses on how to design and implement relational databases with MS Access. Most information and application systems rely on relational databases. Many systems use them such as hybrid enterprise systems databases or as OLTP systems, representing most of today's popular database applications. From a modern perspective, they are extremely important. Instead of using a complex example, we explain a pharmacy case that is easy to understand and fully functional. Rather, it uses a commonly found approach in today's information systems where emphasis is placed on how easily something is used rather than repeated training and specialization. We wish to teach pharmacy students important knowledge of relational databases in three 90-minute sessions, ensuring they gain some practical use of databases.

Keywords: ***Databases, pharmacy, education, MS Access.***

1. Introduction

Pharmaceutical systems depend on databases like virtually all other information systems do today. Similarly, these systems are not limited to a single area, ranging from simply managing a warehouse to using advanced systems for production and having all business information tied together. Since the role of database systems is so vital in advancing modern technology, their developments require us to enhance our knowledge and technical expertise.

As the ICT sector expands at a remarkable pace, people must keep learning new skills. In this era, new algorithms, unique solutions and large-scale technology frameworks appear almost every single day. Given the pace of change in computer science, we can say that successful specialists in the future will certainly need to master basic computer knowledge, along with their specialized skills. Such competencies are already highly prized and sought after in various job areas today(1).

Healthcare highlights how technology has been adopted at a rapid pace. Technologies such as under-skin implants and continuous monitors attached to humans are used more frequently to monitor heartbeat and breathing. Today's technology allows for the development and use of tools that keep track of the health of the chronically ill or elderly. When there is an emergency, they can ask for medical help and send vital information to those who respond first. Still, these abilities require regular data collection and storage which is exactly what database systems excel at doing. If you understand database operations, it will be much easier to understand the methods of processing and organizing data.

A database main function is to manage information in an orderly and effective manner. Data is stored following a specific format in computers and you can retrieve and change it using specialized queries. Database architectures consist of the database, holding all your data and the DBMS which gives you the needed tools to work on your data. In recent technology, Structured Query Language (SQL) is used by the majority of programs to work with databases.

Over time, various patterns of organization have appeared in database modeling(2).

- Hierarchical model: All relationships are 1:1 or 1:N and their structure is similar to how computer folders are organized.
- Under a network model, entities are shown as nodes and relationships as connecting lines between nodes.
- The relational model is set up using tables or networks of tables and this is what we will be examining in our current work.
- In 1970, E.F. Codd introduced the relational data model and coined the term 'normalization' for 'processes to reduce both data redundancy and anomalies in tables of databases.' When data in a table column is repeated, the principle requires organizing them into their own tables and connecting them together. It is

because processing and storing the same information more than once wastes computing resources.

- A relational database architecture identifies three types of relationships: 1:1, 1:N and M:N. This last setup can only be made by including an intermediate join table.
- It helps to derive and understand these concepts to better appreciate how pharmacy students need to be taught how to use databases in healthcare. We ensure that these students can use databases and stay involved in the evolution of e-Health that plays a growing role in today's pharmaceuticals.

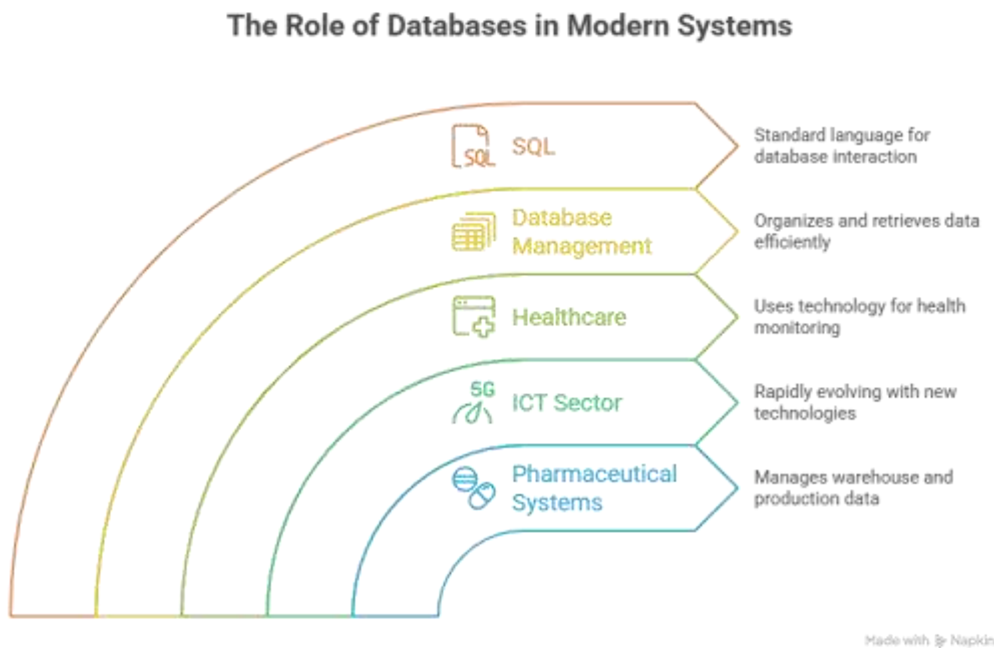

FIGURE 1 The Role of Databases in Modern Systems

2.Methodology

Since informatics is featured only sporadically in their studies, any teaching of these courses should focus on maintaining awareness of what they already know and the skillset needed to use this information. So, we should seek instructional methods that you do not need to be skilled in programming or database query language to use and are simple for both undergraduates and high school students in pharmacy. Therefore, our main educational goal is to equip students with abilities that support future learning or work(3).

The secondary goal of education is to develop practical applications. Students should be able to use their knowledge in practical situations at work and in their daily lives. Because of this method, our instructional platform is Microsoft Access. A number of aspects influenced me to choose MS Access: it fits into the usual Microsoft Office environment with Word and Excel, is easy to learn and use, can create custom forms and reports, requires little installation, doesn't use separate or extra files and still handles many normal types of use cases. Even ordinary users find it simple to get installed and start using the device.

If the database application needs to handle many complex calculations, Microsoft SQL Server would be a sensible option. It is important to note that most database systems use characteristic programming languages such as Microsoft SQL which MS Access also relies on. Sometimes, you may rely on the power of SQL in MS Access, especially for making queries. You can also use Visual Basic for Applications to customize your app which is available through its own interface. Still, since these features require knowledge most people do not get in school, they are used less frequently. We still focus on the basic operations needed in a database: building tables, arranging them through relationships, learning how to work with queries and partly automating how operations follow in sequence(4).

Users gain knowledge by getting involved in a real example within MS Access. We pretend the student pharmacists are in charge of a warehouse in the region and look after a group of customers who require help with database management. Because of this framework, we are able to specify the key skills and knowledge students must learn:

- Developing and putting together database tables
- Knowing and implementing the principles on how tables are linked
- Building queries that are suited for how the application is built
- Making and using summary and action queries
- Creating user interfaces by using forms
- How to create reports that are formatted correctly
- Getting familiar with the core principles of macros for MS Access

For these goals to be accomplished, we break the big assignment into several smaller parts. In this model, emphasis is given to presenting problems well and developing clear steps to solve problems using proper interpretation of constants, operators and functions. It fits within the course time schedule and meets students at the level where they are likely unfamiliar with MS Access.

Each phase in the solution is studied with the help of the instructor. The teaching process follows the given sequence as it checks if students have mastered the objectives from before.

1. Instructions on table design to include making fields, setting parameters and picking the data type
2. Rules for making relationships: teaching how to build relationships in which tables share one key, multiple keys or share no keys
3. When constructing a selection query, use the wizard to help with the use of the operators () and functions ().
4. Learning to use aggregates and noticing the importance of the red exclamation point icon that runs an action query
5. Ensuring users can complete forms and connect to reports, as well as manage wizards, controls and form layout.
6. Creating macros that help automate the database and perform actions.

Phase	Learning Objectives	Pedagogical Strategies
1. Table Design	Field creation, data types, parameter setup	Use simple data grouping, intuitive data types; defer complex ideas like primary keys to later phases
2. Relationships	Understanding 1:1, 1:N, M:N relationships	Visualize using ER diagrams (e.g., Dia), introduce primary/foreign keys, hands-on linking of tables
3. Query Construction	Basic selection, criteria setup, use of operators/functions	Utilize query wizards; emphasize visual tools and expression building
4. Aggregate/Action Queries	Introduce summaries (e.g., totals), actions (e.g., delete, update)	Highlight query execution indicators (e.g., red exclamation icon), reinforce SQL logic through wizard examples
5. User Interface Design	Form and report creation, interface control	Use form/report wizards, emphasize layout and functional grouping, design for user-friendliness
6. Macro Introduction	Automate database tasks	Introduce simple macros with guided automation examples; contextualize within warehouse scenario

TABLE 1 Instructional Phases in Pharmacy-Focused MS Access Curriculum

Dividing the curriculum in the described way enables students to become more skilled step by step. We first check what information is being collected and the suitable format for the table to students when we start to create them. For now, we do not use terms like primary key or primary/secondary fields and introduce them while making relationships. Our priority is to determine which information should be recorded and how it can be grouped in tables.

If we want to better show attributes, entities and relationships, we can use tools such as Dia to make entity-relationship models(5). With Tastydat, you choose date fields for dates, expect numbers for quantities, pick currency for prices and so on. We suggest that any field without math should store text types of data (for example, telephone numbers and building codes). If the discipline does not depend on arithmetic or mathematical multiplication or division, this heuristic assigns it a text name.

Using this system, we ensure that the classwork respects the limits of the curriculum and focuses on the needs of pharmacy students, all while providing important database knowledge they can use in the workplace.

3.Results

During the process of designing the tables for students, we took time to analyze which pieces of information needed to be saved and how many tables were needed. During this time, we did not go through confusing topics like primary keys and primary or secondary fields; instead, we discussed those issues in the following session on relationships. Instead, we worked only on deciding which information we need and organizing it logically into the database tables. Whenever we saw an advantage in extra visualization, we drew entity-relationship models with Dia to highlight what each attribute and entity represents and how they are linked(6).

Instructional Component	Key Activities & Tools Used	Student Learning Outcomes
Table Design & Data Types	Identified data to be stored, grouped into tables; avoided complex keys; used intuitive data types	Students structure data logically and select appropriate data types for pharmacy datasets
Relationships & Keys	Used graphical ER modeling (Dia); implemented referential integrity; explained primary/foreign keys practically	Students learn linking strategies and enforce consistency through table relationships
Query Design	Used wizards; built selection, parameter, and summary queries; practiced operators and functions	Students construct dynamic queries, aggregate data, and apply logical conditions
Action Queries	Built append/delete queries (e.g., processed orders to archive table)	Students manage record flows and understand database state transitions
Expression Builder	Created calculated fields (e.g., price with VAT); concatenated names using &	Students enhance queries using mathematical and textual expressions
User Interface Development	Designed navigation forms, headers/footers, user-focused layouts	Students create accessible interfaces for interacting with data
Reports	Used reporting tools to present analysis (e.g., yearly profits)	Students generate and format print-ready summaries for pharmaceutical decisions
Automation via Macros	Built macro "ClickIt!" to streamline query execution	Students automate repetitive tasks, improving workflow efficiency
Business Intelligence Concepts	Created statistics queries using GROUP BY, SUM, AVG, COUNT, filtered by date	Students apply BI logic for analyzing profitability and operational data

TABLE 2 MS Access Implementation Activities & Learning Outcomes for Pharmacy

Data type was chosen through principles that reflect what the data is meant to be used for. Dates were provided in date/time formats, the total number of pieces was shown with numbers and prices used currency. When it comes to telephone numbers, building identifiers and other non-computational data, we always picked text for its data type. As a result, any information that did not involve math calculations went into text format. As shown in Figure 1, the table layout we suggested follows a systematic database model.

Since we realized that students had only limited experience in mapping tables together, we chose to focus more on how to do something in practice rather than explain the theory. To help create relationships, we added more fields using the rules presented in Table 1. Following this process allowed the design of the database to be the same, including how it handles primary and foreign keys for various relationships(7).

We worked with the database tools interface (accessed by selecting Relationships) to drag fields from the tables labeled with ID_ to the tables with key fields labeled fk_. While working on this, we prioritized enforcing referential integrity and ensuring that data updates and deletions cascaded so that our data remained consistent.

Seven queries were designed and implemented as part of the application. To gather ATC_1_level_cat, ATC_code_full, Medicament_title and Medical_prescription, the query assigned Q_medicament_prescription brought together information from T_ATC_cat_global, T_ATC_cat_complet and T_medicament. When we used the "True" standard on Medical_prescription (which was recorded as yes/no), only results for prescription drugs were displayed. Similarly, with Q_medicament_according_code, square brackets are used around the criterion [Enter ATC code] to develop a parameter query that allows users to enter an ATC code when the command is run.

As part of financial analysis, we made Q_statement_1_paid that makes use of multiplication to work out pricing. As a result of this question, Price without VAT and Price with VAT were added to the query.

- Price without VAT: [T_medicament]![Unit_price]*[T_Orders]![Number_of_pieces]
- Price with VAT: [T_clothes]![Unit_price]*[T_Orders]![Number_of_pieces]*1.2

As a result, students were able to use the expression builder in practice and continue practicing math in their queries. We built on Q_statement_1_No_payment_shipped and Q_statement_2_SUM_paid_but_not_shipped by creating the next query, to look for instances where medications were reimbursed without being shipped.

Q_issued was made to improve user experience with personal data by combining all the names into a single field using "&" as the text operator.

Expression1: T_Employees![name_first] and T_Employees![name_second]; T_Employees![Title_before_name] & T_Employees![Title_after_name

The wizard was used to make this feature work properly by linking the T_orders table's Issued and Closed fields to the expression and making them display full person names instead of IDs.

We applied action queries to manage the records processed by the database. It worked such that Q_statement_3_data_connection removed all completed orders from the T_order table, placing them on the T_equipped table through an append query and Q_statement_4_data_deleting removed all active records from the T_order table, thereby ensuring smooth functioning of the system. We decided to keep the fields Price without VAT and Price with VAT computed in T_equipped to improve reporting and support archiving. shows how the record configurations did not affect any unprocessed records(8).

Another major part of our implementation was designing the user interface. This resulted in four forms named F_employees, F_orders, F_customers and F_equipped_orders that are similar to T_employees, T_orders, T_customer and T_equipped. The picture in illustrates that I used navigation and special header and footer sections to present the web pages clearly and efficiently, as recommended by good practice.

We developed some statistics queries that used functions that aggregate groups of data. A summary of every medication's yearly profits and every customer's profits was created with Q_statement_5_statistics_a and Q_statement_5_statistics_b. To create business intelligence using these queries, the Group By function was joined with Sum, Avg and Count operations. we limited the query to last year's data by using the expression Year(Date())-The information from the reports was presented in an easy-to-read and print-ready form, as reveals. It is apparent from this report that two Salofalk 500 tablets were included in two invoices for €108, with each invoice giving us an average profit of €54. On the other hand, PANZYNORM FORTE-N was purchased just once with a total value of €5.76 that includes VAT.

We automated the execution of queries by using macros. We prepared a macro named "ClickIt!" that allowed us to execute all the required steps in one go. Because of this automation, performing tasks became quicker and more user-friendly.

When creating new orders, the system places the current date into the order creation date fields and increases those dates by seven days when calculating due dates (this is done using Date() and Date()+7 properties in the default values). We designed this table by separating user-visible fields from those focused on relationships, so that users would not see them.

4.Discussion

Pharmacy education has responded to changes in healthcare information management by including database systems in their curriculum. Even knowing only the basics required for this exercise, pharmacy students can easily learn to use databases useful in their future. It accounts for the fact that pharmacy students do not become database administrators, yet they use data systems in nearly all aspects of their work in healthcare(9).

We found Microsoft Access to be ideal for our educational needs due to some important features. Its graphical user interface helps students learn it quickly and using Excel Standard with the Microsoft Office suite makes it simpler for students. Moreover, since all aspects of making databases happen inside Access, there is no need for different design, development and management tools like those in enterprise systems. Because everyone in the class is doing the same thing and the researchers explain the results right away, students can instantly notice the impact of different structures.

We have chosen to teach students mainly by using practical examples, not abstract concepts. When we help students develop a pharmacy database and focus on its function, rather than simply listing technical concepts, they understand and retain more of what they are taught. It is consistent with how constructivist learning sees knowledge as acquired best when placed in real-life contexts that reflect applications in practice. The case of pharmaceutical warehousing demonstrates to students that database skills can be important in their future work life.

As we implemented the plan, we noticed a few educational problems that affected how we designed lessons. The main difficulty was in combining the new concepts with their simplified use within the given amount of time. As a result, some database concepts were simplified, mainly those related to normalization theory and creating relationships. We found that although explanations were brief, it did not greatly affect what students were able to perform with their applications.

Our way of understanding relationships is very practical. While one could examine the meaning of referential integrity or the structure of relational algebra, we clarified how to define keys in practice. They learned to follow proper relationships without needing a detailed understanding of theories. We also picked data types intuitively: we said that if you are not going to do math with it, store it as text.

Teaching was also made easier thanks to simpler instructions in the form pages. We taught students to design user interfaces step by step to prevent them from being confused by the rules of controls and event handling. By starting out with the bare minimum and gradually growing the program, students avoided boredom since they gained results early and continued to learn more by using improved software.

How we create queries may demonstrate our approach to teaching the best. Essentially, we started with basic selections, introduced the idea of accepting input from users, then handled math operations and finally covered actions that changed data. With each new type of query, the sequence offers a simple extension to existing concepts.

The process of implementing macro software allowed the course students to see how they could use what they built to automate their work. Using automation to carry out the order archiving process, students understood how databases can help simplify and streamline tedious manual duties in companies.

We saw that the majority of students were able to carry out the technical requirements of the major concepts used in the functional examples. However, not all participants shared the same level of understanding about designing a database. It was easy for all students to follow the provided application patterns, but some faced issues when using them in new areas. From this, it seems that highlighting design principles and how they can be used in different situations could be useful to future mockups.

Compared to other teaching modules, the curriculum allowed students to acquire knowledge in a very efficient manner. After approximately three hours and 30 minutes of learning (180 minutes), students who had little experience with databases could build programs that use special elements, including parameterized queries, calculated fields, automated storage and statistical data. The efficient learning in Access is partly due to its visual interface, meaning students spend less time learning how to type code and can instead concentrate on how Access works.

When planning the next stages, several new features should be considered. The relevance to students may be increased by explaining pharmacy topics such as monitoring medications or databases of different drugs, with practical examples. Also, teaching the basic ideas of validating and securing data is necessary for today's healthcare professionals. Offering a session on how databases work with different health information systems may add value to students' understanding of how these systems are interconnected today.

It introduces you to concepts that are useful in any database, rather than only Microsoft Access. Access serves as an easy entry, but knowing how to create data structure, relate entities, ask data questions and automate tasks is important any time you work with data. As a result, the knowledge gained stays useful as healthcare information systems advance.

All in all, our case points out that teaching database skills in pharmacy education is both useful and useful for students when put to practice. Highlighting the use of data systems and limiting theory allows pharmacy students to immediately use these skills in their work. Since healthcare is moving digital, the ability to use and manage data will grow in importance for pharmacy staff. Remember to check over all your work.

5.Conclusion and Future work

The design presented in this paper is simple but has all the necessary functions for use in education. We wanted the tables and queries to be understandable by students with little technical knowledge, so we designed the framework with this in mind rather than making it suitable for production use. For example, details about employees such as account and social security numbers, could be added to the T_employees table in an enterprise environment. To ensure the subject was easy to learn and motivated students, we reduced the number of fields to a minimum.

This is because, even if essential data are limited, we still demonstrate that they can produce a strong and reliable database with logical structure. When our goals are accomplished, that shows our way of delivering database knowledge to pharmacy students is effective. It simulates standard duties in a pharmacy in an accessible way that is easy for most users to handle.

We recommend arranging lab sessions in three 90-minute meetings, each with particular objectives to be learned. During the first session, the main topic would be learning how to create a database by making tables. During the next session, information would be collected from the client and converted into queries. In the final session, learners would use report generation and create macros to conclude the course. The step-by-step approach aids in gaining new skills and recalling the basics as they are applied in practice.

It should be noted that the work in the lab during training covers only the initial part of understanding the topic. They gain the most when they apply what they have learned to tackle special issues in their preferred area of interest. It is well accepted that students in pharmacy care less about programming courses, so instruction should not focus too much on such topics. Three 90-minute sessions include the basics of databases and do not prevent students from spending adequate time on other pharmacy subjects.

This data is available for everyone to review and use at https://cloud.ujs.sk/index.php/s/flUxiq7uc9vOfLm, making it easier for teachers to integrate the lessons into their classes. This guide allows for keeping things consistent yet adapting according to what the institution requires.

This paper describes a framework that meets a key requirement in pharmacy education. As hospitals and clinics incorporate more technology and data-based methods, pharmacists must learn how to manage data to work successfully today. The system ensures students learn the key information without becoming professionals in computer science if that is not needed. Concentrating on what we do in drugstores helps us remember and fully appreciate these technical concepts.

Also, because Microsoft Access has a visual interface and built-in features, it provides an excellent place to learn. In Access, the steps for designing, coding and managing a database are blended, so new users pick up the system more quickly than they might with other databases. This way, students can rapidly notice how the design of tables impacts queries, forms and reports.

Guiding students to learn by exploring challenges is given priority over lecturing in Soar Kripke's courses. As they develop a useful system that applies to pharmacies, students improve their programming skills and also gain general understanding of how to organize data in different systems. Because of this, you will develop useful knowledge that can be used no matter what new technology appears.

Students improve their knowledge of effective database design by practicing creating calculated fields, using conditional queries and automating operations with workflows. The frameworks explained here will still be useful whenever and wherever students apply their database knowledge in their careers.

To put it simply, the results show that giving pharmacy students effective training in databases is possible even when time is limited. The knowledge they gain allows pharmacy students to take part more effectively in the healthcare field, since managing data is vital nowadays. Because healthcare information systems are expanding, pharmacy practitioners must know the basic concepts of databases.

Acknowledgement: Nil

Conflicts of interest

The authors have no conflicts of interest to declare

References

1. Lin M, Hsu C. Self-efficacy and readiness for practice among Taiwanese Pharm.D. versus B.S. graduates. Am J Pharm Educ. 2020;84(6):1120–1128.
2. Chang C, Lee W. Confidence and competence in pharmaceutical care: Comparing Pharm.D. and B.S. graduates in Taiwan. J Formos Med Assoc. 2019;118(4):762–769.
3. Wu J, Huang H. Educational outcomes of Pharm.D. and B.S. programs: A comparative study. Taiwan J Public Health. 2021;40(3):345–351.
4. Tsai Y, Kuo F. Readiness for advanced practice among Taiwanese pharmacy graduates. Int J Pharm Pract. 2020;28(3):210–216.
5. Chen YL, Liao C. Differences in clinical preparedness between Pharm.D. and B.S. graduates. Curr Pharm Teach Learn. 2018;10(2):239–246.
6. Wang C, Yu J. Evaluating pharmacist competence using self-assessment tools in Taiwan. J Eval Clin Pract. 2021;27(5):1074–1081.
7. Huang L, Wu S. Self-perception of skills in clinical settings among pharmacy graduates. Asian J Pharm Educ Res. 2019;8(4):33–39.
8. Lai P, Lin H. Bridging the gap: How Pharm.D. and B.S. graduates view professional readiness. Res Social Adm Pharm. 2022;18(6):3115–3121.
9. Chiang Y, Chang H. The evolution of pharmacy education in Taiwan and its impact on graduate confidence. Pharm Educ. 2020;20(1):155–162.