

Proposal for the Implementation of Database Systems Teaching in Pharmacy Courses in Terms of e-Health Support

Ondrej Takáč¹ and Iveta Štempeľová^{2,3}

¹ J. Selye University
Bratislavská cesta 3322, 945 01 Komárno, Slovakia
takac.ondrej@gmail.com

² University of Veterinary Medicine and Pharmacy in Košice
Komenského 73, 041 81 Košice, Slovakia
iveta.stempelova.2013@gmail.com

³ Masaryk University, Faculty of Pharmacy
Žerotínovo nám. 617/9, 601 77 Brno, Czech Republic
iveta.stempelova.2013@gmail.com

ABSTRACT— *In our paper, we will discuss the design and implementation of relational databases using MS Access. Relational database is one of the most widely used databases in terms of practice and information systems. We often encounter their implementation in many solutions, for example as hybrid OLAP databases of enterprise systems or also as OLTP systems - currently most of the classical database applications as we understand them. Their importance is therefore great from a modern point of view. In our paper, through the concrete example of a local pharmacy, we will try to create a practical application that will be fully functional, but will not require deep expertise; on the contrary, it will follow the modern trend that is noticeable in many other information systems, where the emphasis is on usage and not on lengthy training and close long-term professionalization. Our aim is to provide pharmacy students with knowledge, the acquisition of which will not go beyond a maximum of three 90-minute classes and yet provide practical knowledge of relational databases at more than a minimal level.*

Keywords— Databases, pharmacy, education, MS Access

1. INTRODUCTION

Databases are an essential and integral part of virtually all information systems, not excluding pharmaceutical systems. On the contrary. They range from simple warehouse systems through sophisticated production management systems to integrated solutions for robust enterprise information systems. Their contribution to the modern age is not negligible which places ever greater demands on us, both in terms of our skills and knowledge. In this sphere, we need to educate ourselves continuously, and progress in the field of ICT is considerable these days. We can say that new algorithms, new solutions and entire technological solutions are being created every day. We can say that the future will require educated and skilled people, not only in the field they are working in, but also, and most importantly, in the field of the fastest-growing scientific discipline at the moment - computer science. We can already feel that competences in the field of information systems are highly valued and demanded on the labour market. [8] [9]

As an example of rapid development and currently implemented solutions, we can mention, for example, the field of healthcare. Digital elements are getting closer and closer to us, literally and literally, and are becoming part of us - subcutaneous implants, continuously working sensors placed on the human body (e.g. sensing heart activity, breathing). The technological level of modern times makes it possible not only to design, but also to implement such systems that are able to monitor the state of health of sick or elderly people and, if necessary, automatically summon medical assistance and provide the correct data to first-aid personnel. To do this, however, we need to continuously collect and store this data, and databases are used for this purpose. Therefore, a proper understanding of the working principle of database systems can greatly contribute to the understanding of data processing, sorting and proper structuring. [9] [10]

A database is a tool for collecting and organizing information. [3] It contains structured data stored in a computer system, where by using a query language we can retrieve this data. [2] A database system consists of two parts. It is the database itself (database) and the database management system. Database Management system is the software that allows

to provide all the required features of the database system and manipulate the data. One such widespread solution in modern systems is Structured Query Language called SQL language. [7] According to data models we know [6]:

- Hierarchical model - relationships can be defined as 1 : 1 or 1 : N. The internal structure is similar to a tree structure such as a set of folders in a computer.
- Network model - consists of an oriented graph in which entities are represented by nodes and associations by edges.
- Relational model - representation in the form of tables or linked tables. na Also in our paper we are dealing with relational databases.

The relational data model was proposed by E. F. Codd in 1970, and he also proposed normalization, where by normalization we mean: "the process of modifying and correcting the structure of tables in a database so as to minimize data redundancy and data anomalies." [6] This means that if there are fields (columns of a table) with data that are repeated, these fields need to be separated into a separate table, and the tables need to be linked by a relation. This is because redundancy has a great impact on the optimal use of computing power (the processor unnecessarily consumes time when searching the same data multiple times) and storage space (for multiple storage of the same information, we also need multiple storage space allocations). [4] [5]

We know three types of relationships between tables. Namely, 1:1 when one record (table row) from one table is matched by one and only one record from the other table (a specific drug and a drug manufacturer), 1:N when one record from one table is matched by multiple records from the other table (one drug manufacturer and multiple drugs) and M:N when multiple records from one table are matched by multiple records from the other table (active ingredients and drug names). The last type of relationship cannot be directly created, a so-called join table is needed. [1]

1. METODOLOGY

Pharmacy students will encounter informatics subjects only to a limited extent and therefore it is necessary to present the activity system effectively with respect to the knowledge and expertise level of students as well as the nature of the activity. Thus, we need to find a way of presentation that does not require a deep knowledge of programming language or database query language and yet is acceptable and understandable to students, whether undergraduate or high school students. This is our main and primary goal. That is, to attain such skills and competencies to serve them as a good foundation for further study or use in practice. And the aforementioned use in practice is our second objective. If they learn something, it should not remain only in theoretical or quasi-knowledgeable terms, but they should be able to use the knowledge they have gained in this way both in their professional and personal life. From this point of view, our choice fell on MS Access. Firstly, it is part of the Microsoft Office package - like MS Word or MS Excel, but it also has a user-friendly environment. It also allows you to create your own forms and reports, with full automation support via macros. It does not require complex installation, requires no additional files, modules or settings, yet is powerful enough for even moderately demanding applications. Both installation and operation can be handled by the average user. For complex and computationally intensive solutions we recommend Microsoft SQL Server. We have to mention the obvious fact that database systems use so-called query languages, such as Microsoft SQL, under which MS Access also runs. If necessary, the SQL language is also fully available in MS Access and can be used - especially in queries. If the application nature of the projects requires it, the VisualBasic for Applications programming language is also available via a separate interface. However, these already require specialist knowledge and will not be covered in our paper. We will focus our attention on creating tables, links between them, understanding queries and partially automating sequential operations.

The next step is to learn how to work with MS Access itself. We will demonstrate this with a practical example. Suppose we are a regional pharmaceutical warehouse and we have our customers. We need an information system with database support. So let's define our objectives, knowledge and skills.

The knowledge and skills we want students to achieve:

- To learn how to create spreadsheets,
- learn and understand the principles of creating relationships between tables,
- learn to create queries given the application nature of the projects,
- learn how to create and use summary and action queries,
- learn how to design and create user interfaces using forms,
- learn to design and create print reports,

- understand the principles of macros in MS Access.

Field Name	Data Type	Description (Optional)
ATC_1_level_cat	Short Text	A, B, C ...V
ATC_1_level_description	Short Text	

Field Name	Data Type	Description (Optional)
ATC_code_full	Short Text	
ATC_title	Short Text	
_Value	Short Text	
_Unit	Short Text	

Field Name	Data Type	Description (Optional)
Medicament_title	Short Text	
Supplement	Short Text	
Unit_price	Currency	
Type_of_medicine	Short Text	
Medical_prescription	Yes/No	

Field Name	Data Type	Description (Optional)
D_issued	Date/Time	
D_due	Date/Time	
Issued	Short Text	
Medicament	Number	
Customer	Number	
Number_of_pieces	Number	
paid	Yes/No	
Shipped	Yes/No	
D_shipped	Date/Time	
Conclude	Short Text	

Field Name	Data Type	Description (Optional)
Customer	Short Text	
Headquarters_city	Short Text	
Headquarters_street	Short Text	
Headquarters_number	Short Text	
Headquarters_ZIP	Short Text	
Contact_e-mail	Short Text	
Contact_tel	Short Text	

Field Name	Data Type	Description (Optional)
D_issued	Date/Time	
D_due	Date/Time	
Issued	Short Text	
Medicament	Number	
Customer	Number	
Paid	Yes/No	
Shipped	Yes/No	
D_dispatch	Date/Time	
Finished	Short Text	
Unit_prices	Currency	
Number_of_pieces	Number	
Price_without_VAT	Short Text	
Price_with_VAT	Short Text	

Field Name	Data Type	Description (Optional)
Id_zamestnanca	AutoNumber	
Name_first	Short Text	
Name_second	Short Text	
Title_before_name	Short Text	
Title_after_name	Short Text	
Sex	Yes/No	
Birth_date	Date/Time	
Birth_place	Short Text	
Birth_number	Short Text	
Address_street	Short Text	
Address_number	Short Text	
Address_ZIP	Short Text	IM: ::ZIP: xxx xx. > 0 - characters are saved with data, 1 - characters are not saved, but only displayed. IM format: "000\ 00:
Address_city	Short Text	
Contact_mobil	Short Text	IM: ::+421 xxx/xxx xxx. > 0 - characters are saved with data, 1 - characters are not saved, but only displayed. TvarIM: "+421 "000
Contact_e_mail	Hyperlink	
Photo	OLE Object	
Curriculum_vitae	Attachment	

Figure 1: Design of tables

In order to achieve our goal, we divide the above assignment into several smaller stages, emphasizing the quality of the presentation of the problem to be solved but also the solution with respect to methods and procedures (interpretation of constants, operators, functions, etc.), since we are limited by the time allotment and there is a high assumption that the students have not yet encountered MS Access at this level.

The students solve each stage of the solution with the help of the teacher, with the teacher using the following procedures and monitoring the mastery of the objectives identified above:

- Table design - learning how to work with fields, their parameters and data types,
- implementation of relationships between tables - learning how to create 1:1, 1:N and M:N relationships,
- creating selection queries with emphasis on using the wizard in the criteria section - working with operators [!]; ! and functions,
- creating aggregate and action queries in order to learn how to work with aggregate functions and aggregates, learning the meaning of the red exclamation point icon - running an action query,
- creating user interfaces - using forms and print reports, students will learn how to work with controls, wizards and the layout of the forms themselves,
- creating macros - learning how to work with macro actions and automating the running of a database system.

2. RESULTS

When creating tables with students, we will do a short analysis of what we want to record in the tables and in how many tables. In particular, we emphasize that students do not yet know the concept of a primary key or a primary or secondary field. We will deal with these when we create the relationships. At this stage, we will only lay out what data will need to be recorded and in what group structures - that is, tables. If necessary, we can use an entity-relational model for visualization, specifying attributes, entities and relationships, created e.g. in Dia software. We choose data types intuitively, e.g. invoice date - date and time, number of pieces - number, unit price - currency, etc. For telephone numbers, house numbers, etc., we always choose the data type text. We keep to the simple point of view that anything we don't want or can't mathematically process later is always data type text. E.g. we will never want to multiply or divide or factor or square root, etc. a phone number with anything. We show our proposed tables in Figure 1.

Table 1: Design of fields for relationships

Relationship:	1 : N		1 : 1	
+ field with name:	ID_name	fk_name	ID_name	fk_name
Data Type:	Autonumber	Number	Autonumber	Number
Primary Key:	Yes	No	Yes	Yes

In creating the relations, we assume that the students have no close experience and therefore, we focus not on technical interpretation but rather on practical implementation procedures. To create the relations, we create additional fields, for which we adhere to the rules listed in table no. 1.

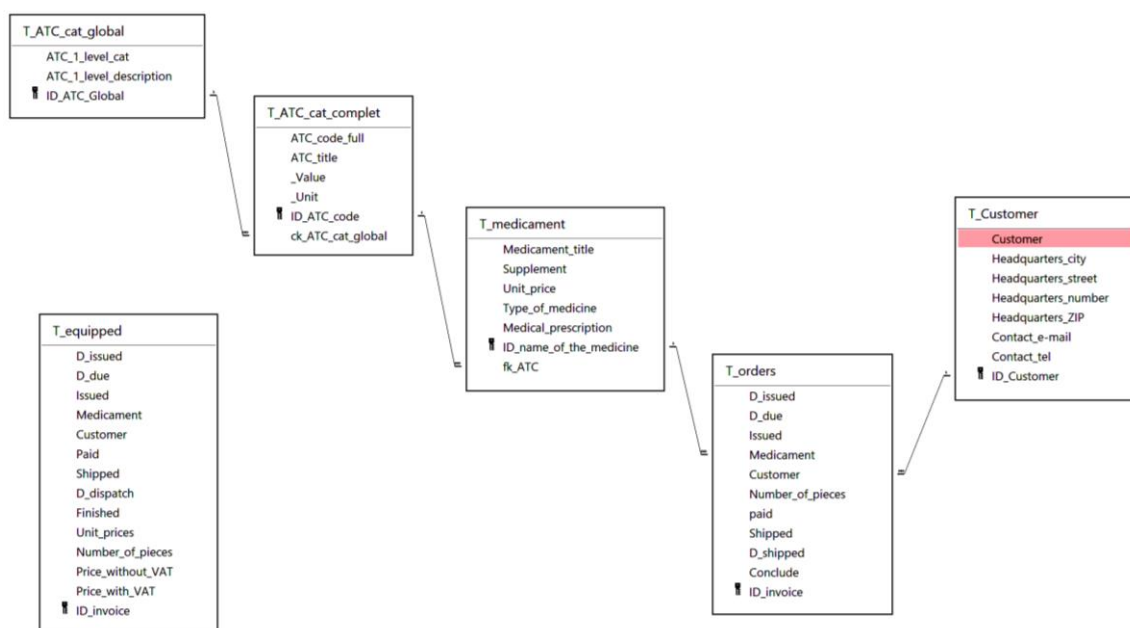


Figure 2: Creating relationships between tables

We show the established relationships between the tables in Fig. 2. When creating, we proceed by dragging the mouse (Menu - database tools - relationships) always from the parent table - the primary field (ID_) with the primary key to the secondary table to the created field with the foreign key (fk_). Of course, we treat the creation of orphaned records by referential integrity along with cascading update and deletion.

We have created seven queries in our application. Among them we mention here the following: In the query Q_medication_prescription we used three tables, namely T_ATC_cat_global, T_ATC_cat_complet and T_medication. From the three tables we have chosen the output fields: ATC_1_level_cat, ATC_code_full, Medicament_title and Medical_prescription field. Since the Medical_prescription field is of the yes/no data type, we used the constant True as a criterion, which selected only the prescription drug output fields. Similarly, we queried Q_medication_according_code, where we put the following in square brackets as the criterion: [Enter ATC code]. Since the [] operator refers to an object that is not present in the query, however, the MS Access asks for this information from the user - Figure 3.

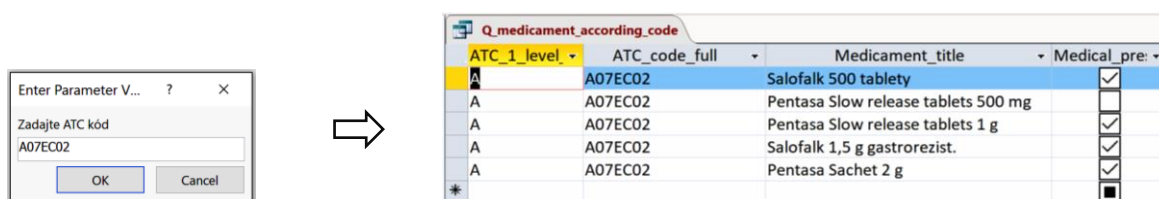


Figure 3: Query result Q_medication_according_code

Query Q_statement_1_paid is similar to the first query, except that here we wanted to know the total price for each medicine. To do this, we created two new fields in the query where we also used multiplication, namely one without and one with VAT. The students here will master working with the build helper.

Price without VAT: [T_medication]![Unit_price]*[T_Orders]![Number_of_pieces]

Price incl. VAT: [T_clothes]![Unit_price]*[T_Orders]![Number_of_pieces]*1,2

A separate task for students here may be to simply combine the criteria from the previous queries, where we will be curious about the price per drug for reimbursed but not yet shipped drugs - query: Q_statement_2_SUM_paid_but_not_shipped. Since it is an unwritten rule that we enter data in the tables as granularly as possible - i.e. not first name with last name and title, but separate first name, separate last name, and separate titles before and after the first name, in the case of determining who created the order, who closed the order, etc. it may not be appropriate for us to refer to the person by last name, first name, number, etc. only. Therefore, we created a query Q_issued where we concatenated multiple fields into one using the & text operator:

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

Then, using the Table Lookup Wizard, we linked this field to the T_orders table (specifically the Issued and Closed fields) and created a drop-down field where the user simply selects their name - Figure 5, Form F_orders (red arrow).

Field:	Issued	Medicament	Customer	Number_of_pieces	paid	Shipped	D_shipped	Conclude
Table:	T_orders	T_orders	T_orders	T_orders	T_orders	T_orders	T_orders	T_orders
Delete:	Where	Where	Where	Where	Where	Where	Where	Where
Criteria:						True		
or:								

Figure 4: Set to delete only dispensed medication records in a delete query. Similarly for a join query.

In our database, we often work with the order table. A large number of records can lead to a lack of clarity between orders and make work more difficult, which is obviously not desirable. Therefore, we use action queries to move those records that are already processed - i.e. paid or shipped - to the T_equipped table with a join query Q_statement_3_data_connection and remove them from the T_order table with a delete query

Q_statement_4_data_deleting. When creating the queries, we followed the rule that anything that is derivable or calculable should not be stored in the tables. However, we deliberately broke this rule for the T_equipped table. However, we were aiming to optimize the functionality of the database, and practical considerations also played a role - we moved all the processed orders here for archiving purposes, and from the point of view of subsequent statistics, we also needed to make the shape of the table clearer, which led to the necessity to break this rule. Specifically, we stored the fields Price without VAT and Price with VAT. In order to ensure that the records are moved and deleted correctly, we need to treat them with the correct criteria - see. Figure 4.

We will then focus our attention on user interface design. Here we have created the forms F_employees - based on the T_employees table, F_orders - based on the T_orders table, F_customers - based on the T_customer table and F_equipped_orders - based on the T_equipped table. An example of two forms with navigation buttons and modified header/footer can be seen in Figure 5.

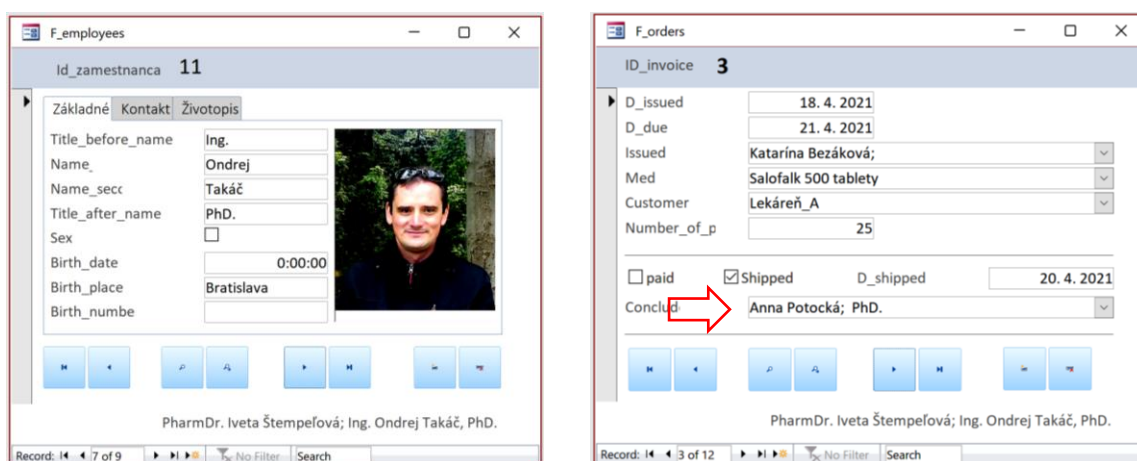


Figure 5: F_Employees and F_Orders forms

In practice, we are often curious about various statistics. We have chosen two for illustration. We wanted to know the average - Avg but also the total - Sum profit for last year by drug and by customer. As a solution to present the results we chose the reports. For this purpose, we first created the queries Q_statement_5_statistics_a and Q_statement_5_statistics_b. Here, students can learn how to work with aggregation functions and they can learn how to use these functions. We have based these queries on the T_equipped table. We turn on the aggregation functions by using the sum icon in the MS Access menu. In the queries, we will see an extra row - Total, where we have to select the function. As an example, we will show the query setup for the report of drugs sold last year. We can see this in the following figure. We have implemented the selection of last year's year with a criterion built from the Year(Date()) - 1 function under the year from the dispatch date. Using Group By, we concatenate the same drug names and use the Count function to count the number of entries.

Field:	Medicament	D issued	Price with VAT	Price with VAT	Price without VAT	Price without VAT	Year([T equipped]![D dispatch])
Table:	T equipped	T equipped	T equipped	T equipped	T equipped	T equipped	
Total:	Group By	Count	Sum	Avg	Sum	Avg	Group By
Sort:							
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> ←
Criteria:							Year(Date())-1
or:							

Figure 6: Background query for report Z_5_statistics_a. An unchecked box will cause the column to not be in the query result

From the query prepared in this way, we create the assembly that can be seen in Fig. 7. We can see that, for example, the drug Salofalk 500 tablets were sold on two invoices (2x Dates of Issue - Count) in the amount of 108 Euros including VAT with an average profit of 54 Euros. But, for example, we sold PANZYNORM FORTE-N on only one invoice (1x Dates of Issue - Count) in the amount of EUR 5.76 incl. VAT with the same average profit of EUR 5.76. Of course this is just a small sample. MS Access allows much more complex calculations, but for our needs it is enough for now.

Medicament	Count_medic	Sum_w_VAT	Avg_w_VAT	Sum_wh_VAT	Avg_wh_VAT
Salofalk 500 tablety	2	108	54	90	45
Pentasa Slow release tablety 500 mg	2	31,2	15,6	26	13
PANZYNORM FORTE-N	1	5,76	5,76	4,8	4,8
	3				

Figure 7: Created report based on the query in A4 format ready for printing

To work with macros, we will use existing queries to move records to the T_equipped table and delete records from the T_orders table. If we wanted to use them, we would have to run each query separately with a query action confirmation. But we can automate these actions using the macro we see in the sequence in Figure 8.

```
Submacro: Add_Accept
    DisplayHourglassPointer
        Hourglass On Yes
    SetWarnings
        Warnings On No
    OpenQuery
        Query Name Q_statement_3_data_connection
        View Datasheet
        Data Mode Edit
    OpenQuery
        Query Name Q_statement_4_data_deleting
        View Datasheet
        Data Mode Edit
    CloseWindow
        Object Type Form
        Object Name F_orders
        Save No
    MessageBox
        Message Všetky Vami želané akcie sa takmer dokončili. Na dokončenie stisknite OK.
        Beep Yes
        Type Information
        Title Informacia
    OpenForm
        Form Name F_orders
        View Form
        Filter Name
        Where Condition
        Data Mode
        Window Mode Normal
    SetWarnings
        Warnings On Yes
    DisplayHourglassPointer
        Hourglass On No
End Submacro
```

Figure 8: Building the Add_Accept submacro

ATC_1_level	ATC_1_level_description	ID_ATC_Glot	ATC_code_full	ATC_title	Value	Unit	ID_ATC_codt	Medicament_title	Supplement	Unit_price	Type_of_me	Medical_pre	ID_name_of	D_issued	D_due	Issued	Customer	Number_of	paid	Shipped	D_shipped	Conclude	ID_invoice	
A07EC02	Mesalazin	1,5	g	1	1,20 €	1g/18	1	Salofalk 500 tablety	tbl ent 100x500 mg (blis.AI/PVC/PVDC)	1,20 €	1g/18		1	15. 4. 2022		Anna Potocká; PhD.	Lekáreň_A	50				19. 4. 2021	Anna Potocká; PhD.	2
														18. 4. 2022	21. 4. 2021	Katarína Bezáková;	Lekáreň_A	25				20. 4. 2021	Anna Potocká; PhD.	3
														21. 4. 2023		Ingrid Huleková;	Lekáreň_ORTO	30						4
														22. 4. 2023		Martina Šípošová; Mgr. CSc.	Lekáreň_A	10						5
														24. 4. 2023	1. 5. 2022	Anna Potocká; PhD.	Lekáreň_A	11						14
														11. 3. 2023	18. 3. 2023			0						(New)
																								2
														12. 4. 2022		Iveta Štampelová; PharmDr	Lekáreň_ORTO	10				14. 4. 2021	Ondrej Takáč; Ing. PĽ	6
														12. 4. 2021		Anna Potocká; PhD.	Lekáreň_A	5						7
														14. 4. 2022		Ondrej Takáč; Ing. PhD.	Lekáreň_ORTO	3				23. 4. 2021	Martina Šípošová; M	8
														21. 4. 2023		Iveta Štampelová; PharmDr.	Lekáreň_Quatro	50						9
														21. 4. 2023		Anna Potocká; PhD.	Lekáreň_ORTO	5						10
														11. 3. 2023	18. 3. 2023			0						(New)
																								3
																								4
																								5
																								(New)

Figure 9: Preview of the tables

Finally, we would like to make a few comments on the practical side. In Figure 9 we can see a preview of the tables in the data view. Also from this view, we can see that from a functional point of view the database is fully functional and

shows no deficiencies during use. When working with the database, we can also notice that if we create a new order, the current date is automatically entered in the creation date field and the due date 7 days after the creation date in the due date field - see the red arrow in Fig. 9. We achieved this by inserting the *date()* and *date() + 7* functions respectively into the default field value properties - in the design view of the table. Of course, these can be modified at any time if necessary. Notice. We have divided the table design into two parts. The actual design of those fields that will reflect what we want to store in the tables, and the refinement of those fields that are needed to create the relationships between the tables. Those fields that we created in the first step we as users enter and if we have done everything correctly, those fields that were created for the purpose of relationships between tables are taken care of by MS Access and the user does not fill those in or interfere with them.

3. CONCLUSION

The data in the tables as well as the chosen queries could certainly be further elaborated. E.g. in the table T_employees there is missing information of a personal nature such as account number, social security number, etc. Here we have to mention that our goal was not to prepare the final application, but to design the learning of the material and for this purpose we had to reduce the number of fields in the tables to make them acceptable for the students and not to demotivate them. As can be seen, even from a relatively small amount of necessary information we were able to create a well working database with a stable background. Thus, we can say that we have achieved our goals.

We propose to implement all these procedures in three lessons - in the computer room, of course. We propose to create tables in the first lesson, queries and some forms in the second, and reports and macros in the third. Working in the computer lab is just an introduction that students will encounter. The real knowledge will only be gained when they prepare their own project. We are aware that pharmacy students are not computer science students. Therefore, we consider three 90-minute classes to be sufficient for the acquisition of basic knowledge. The database is available at: <https://cloud.ujs.sk/index.php/s/flUxiq7uc9vOfLm>.

4. REFERENCES

- [1] Belko, P. Access 2013. (2014). Praha: Computer Press. ISBN 978-80-251-4125-0
- [2] Kruczek, A. Microsoft Access. (2011). Brno: Press. 408 str. ISBN 9788025135075
- [3] Microsoft. Základy databáz. (22. 4 2022). Dostupné na Internete: Support.Microsoft: <https://support.microsoft.com/sk-sk/office/z%C3%A1klady-datab%C3%A1z-a849ac16-07c7-4a31-9948-3c8c94a7c204>
- [4] Palková, Z. Informatika a informačné technológie. (2008). Nitra: Slovenská poľnohospodárska univerzita v Nitre. 252 str. ISBN 978-80-552-0113-9
- [5] Palková, Z. Projects oriented on e-Learning and learning supported by ICT. (2014). Sieťové a informačné technológie 2014. Cit. 22. 04 2022. Dostupné na Internete: <http://spu.fem.uniag.sk/sit2014/zbornik/zbornik.pdf>
- [6] Pribilová, K. Úvod do databázových systémov. (22. 4 2022). Dostupné na Internete: Trnavská univerzita: <https://pdf.truni.sk/e-ucebnice/databazove-systemy1/>
- [7] Sziray, J., & Gaul, G. Vállalati információs rendszerek I. (2006). Cit. 22. 04 2022. Dostupné na Internete: Széchenyi István Egyetem: <http://jegyzet.sze.hu/index.php?fajl=jegyzett&tsz=in&intz=ivi&kr=mtk&PHPSESSID=62de2a556608951fcb389678c875acc3>
- [8] UDVAROS, J. – CZAKÓOVÁ, K.: Developing of computational thinking using microcontrollers and simulations. In. EDULEARN21 : Proceedings of the 13th International Conference on Education and New Learning Technologies. DOI: 10.21125/edulearn.2021.1619. , p. 7945-7951, Valencia : IATED Academy, 2021. ISBN 978-84-09-31267-2. ISSN 2340-1117.
- [9] Stoffová, V. – Zborna, M. – Hyksová, H.: APPLICATION OF SIMULATION TOOLS IN EDUCATIONAL ROBOTICS. In. EDULEARN21 : Proceedings of the 13th International Conference on Education and New Learning Technologies. Valencia : IATED Academy, 2021, p. 9214-9221. ISBN 978-84-09-31267-2. ISSN 2340-1117.
- [10] CZAKÓOVÁ, K. - UDVAROS, J.: Applications and games for the development of algorithmic thinking in favor of experiential learning. In. EDULEARN21 : Proceedings of the 13th International Conference on Education and New Learning Technologies. DOI: 10.21125/edulearn.2021.1389, p. 6873-6879, Valencia : IATED Academy, 2021. ISBN 978-84-09-31267-2. ISSN 2340-1117.