МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет кібербезпеки, комп'ютерної та програмної інженерії
Кафедра комп'ютерних систем та мереж

"ДОПУСТИТИ ДО ЗАХИСТУ"
Завідувач кафедри

_____ Жуков І.А.

"_____"_____2020 р.

ДИПЛОМНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

випускника освітнього ступеня "МАГІСТР"

спеціальності 123 «Комп'ютерна інженерія»

освітньо-професійної програми «Комп'ютерні системи та мережі»

на тему: **" Дослідження та вдосконалення алгоритмів адміністрування мережевої бази даних "Navy"**

Виконавець: _____ Лебедь Н.В.

Керівник: _____ Проценко М.М.

Нормоконтролер: _____ Надточій В.І.

Засвідчую, що у дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань

Лебедь Н.В.

Київ 2020

MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
NATIONAL AVIATION UNIVERSITY
Faculty of Cybersecurity, Computer and Software Engineering
Computer Systems and Networks Department

"PERMISSION TO DEFEND GRANTED"

The Head of the Department

_____ Zhukov I.A.

"_____"_____2020

MASTER'S DEGREE THESIS

(EXPLANATORY NOTE)

Specialty: 123 Computer Engineering

Educational-Professional Program: Computer Systems and Networks

Topic: **"Research and improvement of administration algorithms of "Navi" network database"**

Completed by: _____ Lebed N.V.

Supervisor: _____ Protsenko M.M.

Standard's Inspector: _____ Nadtochii V.I.

Kyiv 2020

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії
Кафедра комп'ютерних систем та мереж
Освітній ступінь: «Магістр»
Спеціальність: 123 «Комп'ютерна інженерія»
Освітньо-професійна програма: «Комп'ютерні системи та мережі»

"ЗАТВЕРДЖУЮ"
Завідувач кафедри
_____ Жуков І.А.
"_____"_____2020 р.

ЗАВДАННЯ
на виконання дипломної роботи
Лебедь Наталії Володимирівни
(прізвище, ім'я та по-батькові випускника в родовому відмінку)

1. Тема дипломної роботи: "*Дослідження та вдосконалення алгоритмів адміністрування мережевої бази даних "Navy*"

затверджена наказом ректора від 25.09.2020 р. № 1793/ст

2. Термін виконання роботи (проекту): з 1 жовтня 2020 р. до 25 грудня 2020 р.

3. Вихідні дані до роботи (проекту*): два сервери, до 10 мережевих робочих станцій, канал доступу до супутникової системи доступу до Інтернету.*

4. Зміст пояснювальної записки: *Вступ, огляд теми, огляд існуючих методів адміністрування мережевою базою даних, розробка схематичної топології мережі, розробка структури бази даних, розробка інтерфейсу для керування базою даних "Navy", висновки по роботі.*

5. Перелік обов'язкового графічного (ілюстративного) матеріалу: *Графічні матеріали результатів дослідження надати у вигляді презентації у форматах .ppt, .pdf.*

NATIONAL AVIATION UNIVERSITY

Faculty of Cybersecurity, Computer and Software Engineering
Department: Computer Systems and Networks
Educational Degree: "Master"
Specialty: 123 "Computer Engineering"
Educational-Professional Program: "Computer Systems and Networks"

"APPROVED BY"
The Head of the Department
_____Zhukov I.A.
"_____"_____2020 p.

Graduate Student's Degree Thesis Assignment

Lebed Nataliia Volodymyrivna

1. Thesis topic: "*Research and improvement of administration algorithms of "Navi" network database***"** approved by the Rector's order of 25.09.2020 p. № 1793/ст

2. Thesis to be completed between з 1 жовтня 2020 р. до 25 грудня 2020 р.

3. Initial data for the project (thesis): *two servers, up to 10 network workstations, satellite access to the Internet access system.*

4. The content of the explanatory note (the list of problems to be considered): *Introduction, review of the topic, research of existing methods of network database administration, development of schematic network topology, development of database structure, development of interface for database management "Navy", conclusions.*

5. The list of mandatory graphic materials: *Graphic materials are given in MS Power Point presentation.*

6. Календарний план-графік

№ пор.	Завдання	Термін Виконання	Підпис керівника
1	Узгодити технічне завдання з керівником дипломної роботи	01.10.20-8.10.20	
2	Виконати пошук та вивчення науково-технічної літератури за темою роботи	9.10.20-15.10.20	
3	Опрацювати теоретичний матеріал щодо стану науково-технічної проблеми	16.10.20-19.10.20	
4	Провести аналіз існуючих алгоритмів адміністрування мережевими базами даних.	20.10.20-04.11.20	
5	Розробити схематичну модель мережі корабля та програму для його адміністрування	05.11.20-04.12.20	
6	Зробити висновки та оформити пояснювальну записку.	05.12.20-14.12.20	
7	Оформити графічні матеріали роботи та подати матеріали на антиплагіатну перевірку	13.12.20-14.12.20	
8	Отримати рецензію та відгук керівника. Надати матеріали роботи на кафедру.	15.12.20 18.12.20	

7. Дата видачі завдання: “1” жовтня 2020 р.

Керівник дипломної роботи _____ Проценко М.М.
(підпис керівника)

Завдання прийняв до виконання _____ Лебедь Н. В.
(підпис випускника)

6. TIMETABLE

#	Completion stages of Degree Project (Thesis)	Stage Completion Dates	Signature of the supervisor
1	Agree on the terms of reference with the thesis supervisor	01.10.20-8.10.20	
2	Perform search and study of scientific and technical literature on the topic of work	9.10.20-15.10.20	
3	To study the theoretical material on the state of scientific and technical problems	16.10.20-19.10.20	
4	Analyze existing network database administration algorithms	20.10.20-04.11.20	
5	Develop a schematic model of the ship's network and a program for its administration	05.11.20-04.12.20	
6	Make conclusions and prepare an explanatory note	05.12.20-14.12.20	
7	Prepare graphic materials of the work and submit materials for anti-plagiarism	13.12.20-14.12.20	
8	Get a review and feedback from the supervisor. Presentation of Graduation Project to the department	15.12.20 18.12.20	

7. Assignment issue date: 1.10.2020

Diploma Thesis Supervisor _____Protsenko M.M.
(Signature)

Assignment accepted for completion _____Lebed N. V.
(Student's Signature)

ABSTRACT

The Explanatory Note to Master's Degree Thesis – "Research and improvement of administration algorithms of "Navy" network database": 94 pages, 43 figures, 9 tables, 13 references.

COMPUTER NETWORK, TOPOLOGY, DATABASE, DBMS, RELATIONAL DATABASE MODEL, ADMINISTRATOR, SQL, INDUSTRIAL EQUIPMENT, GRAPHICAL USER INTERFACE.

Topic actuality – the existing interfaces do not allow to fully provide the requirements for military tasks on modern equipment installed on Navy ships. The introduction of new high-speed and accessible interfaces will allow to quickly perform the tasks on the ship.

The Goal of Graduation Project – to examine some of the existing methodologies of administration of network databases and design a project for administration of "Navy" database.

The Main Task of Master's Degree Thesis – to develop the most optimal algorithm for administration a network database. Provide an accessible interface for use in specific environments.

The Object of the research – introduction of a digital information processing system on board the ship.

The subject of the research – a complex providing the operation of the ship's systems.

Method of the research – structural, algorithmic and programmed implementation and the basic operations with operators of management of network databases.

CONTENTS

LIST OF SYMBOLS, ABBREVIATIONS AND TERMS

TDMS	– Time-Shared Data Management System
IMS	– Information Management System
REST	– Representational State Transfer
HTTP	– HyperText Transfer Protocol
ASCII	– American Standard Code for Information Interchange
DBMS	– DataBase Management System
SQL	– Not only SQL
JSON	– JavaScript Object Notation
Navy	– Naval force
NATO	– North Atlantic Treaty Organization
IFF	– Identification, Friend or Foe
GUI	– Graphical User Interface
JRE	– Java Runtime Environment
DDS	– Data Distribution Service

INTRODUCTION

Long before the advent of computerized databases, humanity already then had a need to store information in a structured form. Since at a time when computers either did not exist at all, or they were just entering the market and were at the stage of their formation, peculiar databases existed in written or physical form, for example, data archives, reference centers, libraries, ledgers, telephone directories etc. And since now there is a need to store huge amounts of information, while taking up as little storage space and resources as possible, databases are an integral stage in the development of opportunities to simplify the life of mankind.

The database is both a tool and the very subject of a collection of data, which shows the state of certain objects and their relationship in a certain subject area.

In order to have the look that databases have now, they have undergone quite a few changes since their introduction in the early 1960s. Initially, the original systems used to store and process data were navigation databases - for example, hierarchical databases (which had a one-to-many relationship based on a tree model) and networked databases (a more flexible model, allowing many-to-many relationships, that is, multiple relationships). Despite the simplicity and conciseness that were natural, these early systems were inflexible. More precisely, one was more flexible than the other, but still not enough to simplify the operator's work at the computer and use the final product.

Relational databases became popular in the 1980s to replace the previous ones, and object-oriented databases followed in the 1990s. Of course, other types arose, but they did not find widespread use in common areas of use. More recently, due to the growth of the Internet and technology, and the need for faster processing of unstructured data, NoSQL databases have emerged.

Offline and cloud-based databases are now opening up new possibilities in how data is collected, stored, used, and managed.

At present, the problem of ensuring the administration of databases on ships and ensuring stable and integral work in specific conditions is urgent. Such systems allow automating the processes of collecting, analyzing and presenting data in a convenient form for the entire crew.

The aim of the master's work is to study existing approaches of improving, and choosing the most appropriate, data administration algorithms in the network database of the ship's system.

When solving problems of managing structured content to increase the performance of ship control systems, when working with data, in limited environments, specific communication means are used and have a complex non-trivial structure.

To prevent errors on the complex, it is necessary at the development stage to think over all the nuances, design the necessary database data schema, and, taking into account the customer's requirements, possible risks associated with data safety, develop a software product with which the operator could easily interact.

Cisco Packet Tracer will be used to design a network topology that will simulate the ship environment. Such a program is relevant for the schematic representation of the network, which is being prepared for this implementation, and also allows to evaluate the performance of the network being created. Such a program is free and educational, which is more suitable than ever for a person receiving a master's degree.

Java was chosen as a programmable language, since in addition to the fact that this language is object-oriented, it is also quite flexible in development, has all the necessary modules to perform this task, is cross-platform, due to the JVM (Java virtual machine), which allows work stably user software on a Linux kernel system.

PART 1
RESEARCH AND REVIEW OF DATABASES

The ability to collect extensive information and data access from the Internet offers greater opportunity for the productive sector – today's companies have access to much more information than ever before.

To move from conventional storage of data and basic transactions from multiple systems to analyzing massive amounts of data, progressive organizations can nowadays use databases. Thanks to databases, various tools for computing and market and business analytics, various companies can use collected data for more effective decision making, efficient work, flexibility and scalability, as well as analyze a specific type of activity and make decisions based on it. A stand-alone database can greatly enhance these capabilities. Standalone databases automate costly and time-consuming manual procedures so business users can focus on working proactively with their data. By being able to creating and using databases, users gain control and autonomy while maintaining important security standards. At present, it is impossible to imagine the functioning of any system, organization, enterprise or firm without a developed information system that can automate the collection and processing of data [1]. Usually, a database servs for this purpose. Data that contains some information about a specific subject area can be stored and directly accessed.

In simple words, today, a database is a collection of information or data that is structured and ordered using certain rules and stored electronically in a computer system. Since at a time when computers either did not exist at all, or they were only coming into use and were at the stage of their formation, databases existed in written or physical form, for example, repositories, movie catalogs, libraries, phone books, etc.

And since now it is simply necessary to store volumes of information, databases are an integral part of our existence, indirectly or explicitly. The database is both a tool and the very subject of a collection of data, which shows the state of certain objects and their relationship in a certain subject area. In this context, the subject area is a certain area of the real world, or a product of human activity, on the basis of which the necessary database is created. But also, there is need to manage this structure. Database Management Systems (DBMS) come to the rescue in this aspect.

The concepts of databases and spreadsheets are different from each other, both provide convenient options and methods for storing and retrieving information. Microsoft Excel can be classified as spreadsheets. The main differences between them are in the way of storing and processing data, they have different access rights to data and amount of stored data [1].

Since originally spreadsheets were designed for a single users, their properties reflect that entity. They are great for this purpose – managing and manipulating a small amount of data, and do not perform super-complex data operations. But databases are intended for a much larger amount of information stored and ordered according to a certain criterion. Also, databases have more complex logic and input language, but they allow multiple users to work with data quickly and safely at the same time.

There are certain basic requirements and principles of construction for modern databases and their DBMS:

- high performance (short response time to request). Response time - the time interval from the moment of the request to the database until the actual receipt of the response data;
- ease of data updating;
- no data dependency - the ability to change the logical and physical structure of the database without changing user views;
- data sharing by many users;
- data security - protection of data from intentional or unintentional violation of integrity and secrecy, distortion or destruction;

- standardization of the construction and operation of the database (in fact, there is a DBMS);
- accuracy, adequacy and correctness of data display in the corresponding subject area;
- simple and intuitive user interface.

The first two requirements are very important, although contradictory. Improving performance requires simplifying the structure of the database, which, in turn, complicates the procedure for updating data, while increasing their redundancy.

Integrity and data protection make it safe. Data integrity – the stability of the stored data to destruction or disruption, as well as compliance with its logical structure associated with technical faults, system errors or erroneous actions of users. It suggests:

- no introduction of inaccurate data or duplication of two identical records about the same aspect;
- protection against errors during updating, both the database itself and the data stored in it;
- impossibility of deleting (or cascading deletion) of related data from different tables, although this is a controversial issue;
- absence of data distortion, or their incorrect recording, when working in multi-user mode and in distributed databases;
- safety of data in the event of equipment failures (the possibility of data recovery or backup).

Integrity is ensured by special applications, programs or its modules that work under certain conditions. Such software products can be called integrity triggers. Data protection from unauthorized access involves restricting access to confidential data and can be achieved by:

- introduction of a password system;
- obtaining the necessary rights and permissions from the database administrator (DBA);
- prohibition from the DBA for access to confidential data;

- formation of views – tables derived from the original and intended for specific users.

To record information through queries, most databases used a SQL – structured query language. This language is a programming language that is used in most relational databases to form a query, control an access and rights, as well as data processing and definition. SQL has given impetus to the release of numerous extensions from companies such as IBM, Oracle and Microsoft. And despite the fact that SQL is still widely used today, new languages for programming have begun to emerge [1].

Standardization simplifies the interaction of databases of the same generation of DBMS with the same and different data models, and, accordingly, ensures the continuity of generations of DBMS. In this case, both local and remote access to data can be carried out (client-server technology or network variant).

The process that allows solving of a wide class of problems associated with the creation of databases is called design, or building a database.

1.1. Means for constructing database

The main tasks of database design consist of:

- ensuring the necessary storage in the database of all the necessary information;
- providing the ability to quickly obtain data on all necessary requests;
- reduction of redundancy and duplication of data, that is, their uniqueness;
- ensuring data integrity (correct storage and content): elimination of contradictions in the content of data, elimination of their loss, etc.

There are three main stages in database design: conceptual, logical, and physical design.

Conceptual (infological) design consists in building a formalized model of the considered subject area. Such a model is built using standard language tools, usually graphical ones, for example ER-diagrams (entity-relationship diagrams). When building such a model, should not be guided on any specific DBMS.

The main elements of given model are:

- description of the essence of objects in the subject area and types of relationships between them;
- a description of the information needs of users (a description of the main requests for data selection from the database);
- description of algorithmic dependencies between data when searching for them;
- a description of integrity constraints, i.e., requirements for admissible data values and for relationships between them.

Logical (datalogical) design is a mapping of an infological model to a data model used in a specific DBMS, for example, a relational data model. For relational DBMSs, a datalogical model is a collection of tables, usually with key fields that describe the relationships between these tables. Datalogical design is the construction of tables according to certain formalized rules, as well as the normalization of these tables, if the infological model is built in the form of ER-diagrams (or other formalized means). This stage can be largely automated by means of modern software [2].

Physical design is the implementation of a datalogical model by means of a specific DBMS, as well as making selected decisions related to the physical environment of data storage: choosing methods for managing disk memory, methods of accessing data, methods of compressing data, etc. – these tasks are solved mainly by means of the DBMS and are hidden not only from the database developer, but also further from the user or the database operator.

Also, to simplify the third stage and reduce the time for its development, even at the stage of infological design in the course of collecting information about the subject area, it is necessary to find out:

- what objects of the subject area are the main ones (objects about which information should be stored in the database);
- attributes of objects;
- connections and their types between objects;
- basic queries to the database.

Over the course of several decades, modern database technologies have emerged as a result of the development of data processing and information management. Data processing has evolved from primitive techniques to the sophisticated integrated systems that exist today.

Currently, databases are present in all spheres of human activity and constitute practically the basis of computer support for information processes.

Databases are an effective tool for representing and manipulating data structures, since certain entities have a common nature and have a pronounced similarity in the form of a collection of data. The concept of databases presupposes the use of integrated storage facilities that allow centralized data management and maintenance for many users. In this case, the database must be supported in a computer environment by a unified software called a database management system (DBMS). The DBMS together with application programs is called a data bank [2].

Application fields using databases have overcome the limitations of file systems such as:

- redundancy of data;
- weak control;
- insufficient data management capabilities;
- high labor costs of the programmer.

Database administrators solve all the nuances associated with the effective use of the database, and also the storage of the data itself, and access to them. With the explosive growth in the use of databases, all new methods of accessing data simplified the process of linking data items, which in turn led to an increase in the possibilities of manipulating them. All of these characteristics of database management systems make programming easier and reduce the need for software support.

To create, view and modify database schemas can be used by others are very handy tools, such as tools and instruments environment Visual Studio .NET. Also, Microsoft Visio has all the necessary capabilities to automatically create a diagram for an existing

database, i.e., database reengineering. This functionality is especially useful for working with legacy databases that were created a long time ago and using very different tools.

Access DBMS includes a diverse and numerous, relatively stand-alone software tools that are focused on creating database objects and user applications.

Currently, the process of creating the most powerful database management systems is in full swing. Systems based on three basic data models, or conceptual methods for structuring data: hierarchical, network, and relational, have consistently emerged. However, these are not the only existing data models.

The database model is essentially the database schema itself, that is, it is a description of the content, structure and constraints that affect the integrity used to create and maintain the database.

1.1.1. Hierarchical model

The first information system to use databases was based on a hierarchical model. It arose, in contrast to other models, in a practical way, therefore there is no original source describing the hierarchical model. Since the hierarchical data model has no standard, its study requires consideration of the DBMS used in practice. Among the implementations of the hierarchical model, IBM's IMS (Information Management System) prevails. All descriptions of the hierarchical model invariably include the vocabulary and conventions of the IMS [3].

However, other hierarchical systems are used, including TDMS (Time-Shared Data Management System) from Development Corporation, MARK IV (Multi-Access Retrieval System) from Control Data Corporation, and System-2000 (SAS-Institute).

IMS was the result of a collaborative effort between IBM and North American Aviation (later Rockwell) to create a database management system to support the Apollo lunar project, one of the largest engineering projects of the time. The main factor in the creation of the IMS was the need to manage a huge number of parts linked together in a hierarchical manner. That is, from small parts, larger units were assembled, which were included in even larger modules, etc.

For large systems with scheduled transactions that require high performance, IMS remains quite competitive. An additional aspect of IMS survivability is that many data

structures are naturally hierarchical. For example, an organization may consist of departments (one level), the departments have employees (second level) who have certain specialties (third level). This data structure can be implemented in other models, but the more powerful representation capabilities of this model can lead to more system complexity than required.

The hierarchical data structure consists of three objects: trees, segments and segment fields (fig. 1.1).

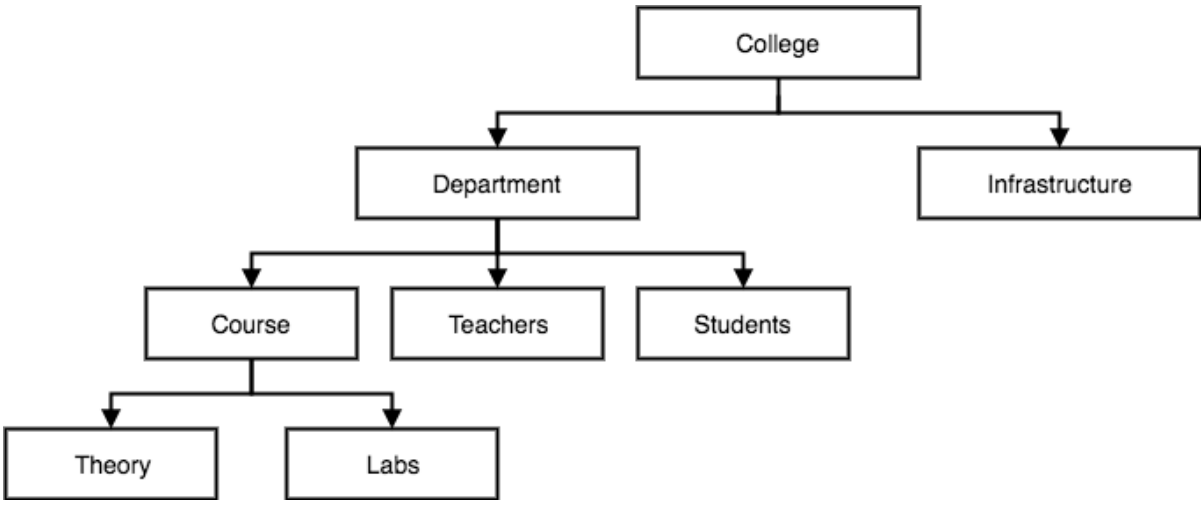


Fig.1.1 Illustration of Hierarchical database model

By the repetition of the segments can lead the transformation of any object-oriented model, the following requirements so that all database records were trees. Any situation in which a natural transformation causes a segment to have two different ancestors requires the creation of two separate trees to which those ancestors will belong. Despite this repetition, it allows to get rid of certain implementation difficulties, but leads to the following negative outcomes:

1. The amount of memory is increased and wasted inefficiently, since the segments are repeated;

2. There is a possibility of data inconsistency. If the data changes in one copy of the segment and does not change in another, the database becomes inconsistent, as a result of which the data is incorrect.

This problem can be solved by using virtual segments and pointers. The virtual segment contains no data, but instead contains a pointer to the address of the data storage. When a segment is to be duplicated in two or more trees, then valid data is stored in only

one of those trees. All other occurrences of this data segment will contain a pointer to the storage address of the valid data, preventing the second problem.

The main limitation of the hierarchical model is that for many applications tree is not a natural data structure.

For applications with a hierarchical structure and stable transaction requests, a hierarchical model may be satisfactory. The fact that about 7000 IMS have been installed supports this conclusion, although moving to relational systems.

The language interface provided for the hierarchical model is highly dependent on the system manufacturer. This means that the programmer must be aware of which relationships should be predefined in the system and which should not. The reliability of the programmer will be unsatisfactory if he does not have full knowledge of what integrated requirements are built into the system. In addition, if a hierarchical database is reorganized, it can negatively affect the quality of existing programs, or even stop their work, since the structure that supports one application, may not be the best for supporting another.

To make it easier to imagine a hierarchical model, it can be imagined the structure of government, or organization of directories on disk. That is, this model is built in the form of a hierarchical tree-like structure, in which for each “subordinate” object there can be only one master, and each master can have several “subordinate” objects.

The advantage of this model is the navigation method, which allows to provide high-speed access to the object, but subject to predefined connections. However, this model has two main disadvantages. The first makes it impossible to direct the search from the lower levels of the hierarchy to the higher ones. And the second drawback is the lack of flexibility of the model itself, which gives rise to the lack of direct access to data and makes it unsuitable for frequent queries. It is impossible for an object to have several “parents”, which follows in the absence of its flexibility.

1.1.2. Network model

Networks are a natural way of representing relationships between objects. They are widely used in such areas as mathematics, research, chemistry, physics, sociology and other areas of expertise. The network architecture applies to databases as well.

Networks can be represented using a mathematical structure called a directed graph. The graph consists of points or nodes connected by arrows or edges. In the context of these models can be represented as nodes of the types of data records, and the ribs are of a one-to-one or one-to-many relationships. Thus, the network data model represents data by network structures of record types linked by a one-to-one or one-to-many relationships. The structure of the graph makes possible simple representations of hierarchical relationships, membership relationships and many others, an example is shown in the figure 1.2. It is possible to actively perform data extraction and manipulation after the relationship between the two objects is established [4].

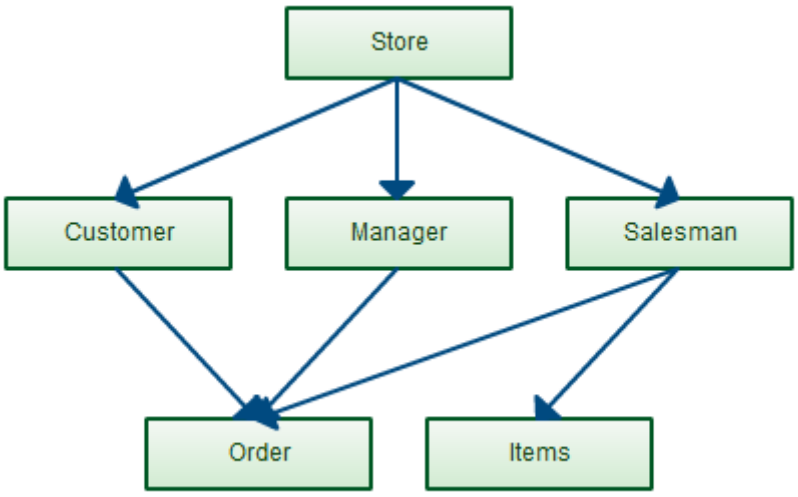


Fig.1.2 Network database model

The Conference on Data Systems Languages (CODASYL), which brings together representatives from major hardware, software and user suppliers, developed and standardized the Cobol language in the early 60s. Over time, it commissioned a subgroup called the Database Task Group (DTBG) to develop standards for database management systems. DTBG was heavily influenced by the architecture used in one of the earliest DBMSs, the Integrated Data Store (IDS), previously created by General Electric. This influence led to the recommendation of the network model in the preliminary report. This first report, published in 1969, give birth to a huge number of improvement suggestions. It has been submitted to the American National Standards Institute (ANSI) for possible adoption as a national DBMS standard.

The CODASYL DTBG model served as the basis for the development of networked database management systems from several manufacturers. IDS (Honeywell) and IDMS (Computer Associate) are the best-known commercial implementations. Although the network data model may increasingly yield positions in the DBMS market of the relational data model, it nevertheless effectively serves in many information systems.

There are two main data structures in the network model – record types and sets. Record types are generally defined as collections of logically related records. Sets in the DBTG model express a one-to-many (or one-to-one) relationship between two record types. Valid values in the data structure are called instances or entries [4].

In the DBTG model, only simple networks are possible in which all relationships have a capacity of one-to-one or one-to-many. Complex networks involving one or more many-to-many relationships cannot be directly implemented in this model. Despite this, there is a method for transforming a complex network into a simple network form.

The DBTG network model is particularly well suited for information systems that are characterized by:

- large size;
- well-defined, repetitive queries;
- well-defined transactions.

Users and designers of a system can focus their energies on writing an application in the most efficient way if all of these factors are present. The negative side is that unforeseen applications, or their modules, may work poorly in the future. At their worst, they can require a laborious reorganization of the database structure.

In the network data model, each object can consist of an arbitrary number of links. A good and simple example of such a model is the bus route map – there are routes from one city to another, etc. That is, when the value of several fields contains links to data that are in another file, then the link in this case is established explicitly.

Such structure is more suitable for modeling a wider class of problems and is more expressive and flexible, in contrast to the previous one. Hierarchical and networked databases are often referred to as navigational databases.

The principle of navigation increases the degree of dependence of programs and data, but increases the efficiency of access to data and thus shortens the response time to a request. Data processing programs are rigidly tied to the current state of the database structure. And in case of changes in the structure of the database, the program itself, or programs must be rewritten. Editing and deleting data requires re-positioning of pointers, and data manipulation remains oriented towards writing data. In addition, the principle of navigation does not allow significantly increasing the level of the data manipulation language to make it accessible to a user who is not a programmer or even a non-professional programmer. To search for a target record in a hierarchical or network structure, the programmer must first determine the access path and then, step by step, go through all the records that may happen along the way.

Since the presentation schemes of hierarchical and network databases are complex, the design of specific application systems based on them is laborious. As many years of experience show, long periods of development of application systems often lead to the fact that they are constantly at the stage of development and improvement, which leads to an endless process without a logical completion of development, and only its support. The complexity of the practical implementation of databases based on hierarchical and network models predetermined the creation of a relational data model [4].

1.1.3. Relational model

The established understanding of databases was changed by an article published by E. F. Codd in 1970. He put forward the idea that data should be linked according to their internal logical relationships, not physical pointers.

In his article, Codd proposed a simple data model in which data was represented as rows and columns, which in turn were tabulated. These tables were called relational, and the model was called relational. An example of this relationship can be understood from figure 1.3. Codd put forward a proposal to use "two languages" for working with data in tables: relational algebra and relational calculus. Both languages provide data manipulation based on logical characteristics, not physical pointers.

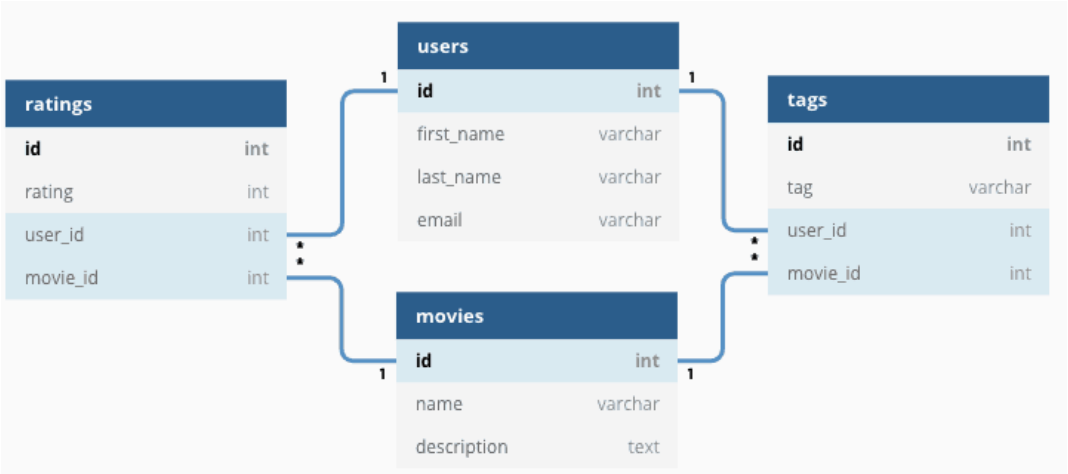


Fig.1.3 Relational database model

By looking at data from a conceptual rather than a physical point of view, Codd proposed another idea that made database programming more efficient. In traditional systems, one record was processed at a time, and in relational database systems, entire data files can be processed in a single command, which significantly increased not only efficiency, but also productivity.

This logical approach to data has given impetus to the ability to create query languages, making work more accessible to non-computer users.

Despite the challenges of creating a language that everyone can use, regardless of computer experience, relational query languages have made working with databases available to a wider range of users than ever before.

The publication of Codd's work triggered an increase in the work of scientists and developers. The result of his work was the formation of various relational database languages, such as Structured Query Language (SQL), Query Language (Quel) and Query-by-Example (QBE) and others. SQL was adopted as the ANSI standard for relational database languages in 1986. This standard was later updated in 1989 and 1992.

The degree of relations is the number of attributes in the relations. No two relational attributes cannot have the same name. Relation strings are called tuples. It is also assumed that no two tuples can have the same set of values, and there is no predefined ordering of tuples or strings [5].

An attribute scope is a collection of all possible values that an attribute can take. Two attribute scopes can only be the same if they have the same meaning. Two attributes with the similar scope may not necessarily have the identical name [5].

Any set of attributes that uniquely identifies each tuple in a relational table is called a superkey. A relational key is a minimal set of attributes. Thus, the key is the smallest superkey.

A composite key is a key that contains two or more attributes. As a key, in any relational table, it can be selected more than one set of attributes that may appear there. These are called potential keys. The primary key can be called one potentially selected key, as a relational key. Typically, a potential key is chosen as the primary key, which is the easiest to use in everyday work for data entry. A set of attributes from one table that is the key of another, or the same table, is called a foreign key. The list that gives the names of relational tables with their attributes and foreign key definitions is called the relational database schema. Thus, the relational database schema is the preliminary result of the creation of the life cycle stage of the relational database itself [5].

To maintain the integrity and verify the data in the database, as well as to make the data meaningful structure in the relational model Codd has several restrictive conditions. Constraint conditions provide a logical basis for maintaining correct data values in the database, which prevents errors when updating and processing data.

Rows in a relational table represent members of a particular collection, object, or category in the database. A relational table key uniquely identifies each row and therefore each item in a category or set. Thus, the user needs to know the key value of that row in order to retrieve the data of a particular row, or to manipulate the data. It follows that until the values of the key attribute element are fully defined, it should not be written to the database [5].

In relational tables, foreign keys are used to link rows from one table to rows from another table. A database in which all foreign keys refer to the current key values of another relational table has reference integrity if the non-empty foreign key condition is met.

Normalization is the process of bringing relational tables to a standard form, which eliminates data redundancy or repetition and violation of data integrity in the database. Data integrity refers to the absence of inconsistency between these data. The normalization is based on the partition – the separation process is a single table for a few others. In order to carry out the process of splitting tables, it can be used normal forms or rules for structuring tables [5].

Relational algebra is a procedural language for processing relational tables. This means that relational algebra takes a step-by-step approach to creating relational tables that contain the answers to queries. Since relational algebra is the source of many of the terms for data processing concepts that are often found in database languages, it can be concluded that relational algebra is of great importance [6].

Relational calculus is a non-procedural language. In relational calculus, a query is created by defining a query table in one step. Relational algebra and relational calculus are logically equivalent. This means that any query that can be formulated using relational algebra can also be formulated using logical calculus, and vice versa.

A relational algebra operation that creates a set-theoretic join between two join-compatible relational tables is called a join operation.

Intersection is a relational algebra operation that creates an intersection, in the set-theoretic sense, of two merge-compatible relational tables. As part of the join operation, the product operation is of great importance. It is identical to the mathematical operation of taking the Cartesian product of two sets.

Selection is an operation of relational algebra that selects rows from a table based on a certain condition.

By displaying only those rows that satisfy a given condition, a select operation is used to create a relational table from another relational table.

If the sampling operation can be represented as the avoidance of unnecessary lines, the operation for the projections can be represented as eliminating unnecessary columns. The table resulting from the operation of creating projections is called the projection of the original table.

Join is a relational algebra operation that joins tables. That is, this join operation is used to link data between tables.

A query that only refers to one table is called a simple query. A subquery is a query within the current query.

Relational tables have the following properties:

- all values are atomic;
- each row is unique;
- the order of rows and columns is not important;
- each column has its own unique name [6].

Thus, Codd has developed a more convenient way of storing, selecting and manipulating data than hierarchical and networked database models. An unambiguous advantage is the dual ability to refer to elements – both by rows and by columns. Based on this, the conclusion can be made that about the two-dimensionality of the table, which is obvious.

Also, the name “relational” is associated with the fact that each record that is contained in the table must refer to a specific element. In addition, even data of different types can be considered as one whole if they are related to each other, that is, they are in a relationship with each other.

The main difference between the search for data of the relational model from the hierarchical and network is the search by the values of key attributes, while in other models the connection and search is performed between different objects in structure. For example, it can be found all house number records with a value of 3, but it cannot be found the 3rd row [6].

The tabl. 1.1 shows a comparative characteristic of ways of accessing data for various database structures.

Table 1.1

Table of comparison of characteristics and ways of accessing data

Data access method	Characteristic
Sequential access files	Records must be processed in sequential order
Random access files	Support direct access to a specific record. It is difficult to access multiple records bind with one
Hierarchical database	Supports access to multiple records associated with one. Data relationships are limited to hierarchical relationships. Depends on predefined physical pointers
Network database	Supports hierarchical and non-hierarchical relationships between data. Depends on predefined physical pointers
Relational database	Supports all logical relationships between data. Logical data access independent of physical implementation

Since the relational structure is conceptually simple, it allows the implementation of small and simple databases, even personal ones, the very possibility of implementation, which has never even been considered in systems with a hierarchical or network model. Therefore, relational databases are easy to create, even for personal use.

Today, the vast majority of databases are created precisely on the basis of the relational model, and as a result, software products, since such a structure gives flexibility in data manipulation.

Naturally, in addition to the three fundamental models, there are fewer common ones that exist mainly in non-commercial projects.

1.2. Modern DBMS

A database usually requires complex software called a database management system (DBMS). A database management system is a set of language, software and hardware tools designed to create, fill, update and delete databases, as well as other actions

performed in most cases by the database system administrator. The DBMS serves as an interface between the database and users or programs, providing users with the ability to receive and update information, as well as manage its ordering and optimization. The DBMS also allows to simplify the ability to monitor and manage databases, allowing to perform deep administrative actions, such as monitoring of performance, tuning, backup and recovery.

Examples of popular database management software, or DBMS, include MySQL, Microsoft Access, Microsoft SQL Server, FileMaker Pro, and Oracle Database [7]. Some of them are shown in figure 1.4:

Fig.1.4 Modern DBMS

All the above considered database models are used in the DBMS. The ancestors are hierarchical and network models, since they preceded the appearance of the relational, and due to this they are pre-relational models [7].

Next, there will be specifically considered modern DBMS that allow working with different database models.

IBM decided not to abandon the hierarchical model in extensions to its products such as IMS and DL/I, despite the rapid change in the relational model, which offered a higher-level, declarative interface.

The Information Management System (IMS) has transactional capabilities for managing hierarchical databases. This system has a built-in transaction manager. Support

and interaction with clients are carried out via TCP/IP. This DBMS provides queuing of jobs for executing transactions in the database, as well as monitoring the execution of transactions.

Common relational database management systems are Oracle, Sybase, DB2, Access, Ingres, Informix, and MS-SQL Server [7].

Access DBMS includes a diverse and numerous, relatively stand-alone software tools that are focused on creating database objects and user applications.

SQLite is powerful enough for an embedded system. Although MySQL is the most popular and frequently used RDBMS, PostgreSQL is the more advanced and flexible.

Recently, the DBMS market has been enriched with products presented for managing the object-oriented data model, such as Gem Stone, Versant, IRS, ORION and ONTOS.

Since NoSQL DBMSs do not use a relational data structuring model, many implementations have appeared, that solve this aspect in their own way, often very unusual. In these circuit less solutions, an unlimited number of formations of records and data storage in the form of a key-value are allowed.

DBMSs such as MongoDB store data as a whole, allowing collections of data to be grouped with other databases. This data can represent a single object and at the same time correctly respond to requests for fields. And this is a distinguishing feature from traditional RDBMS. NoSQL databases do not use a common query format (like SQL in relational databases). Each solution uses its own query system. MongoDB is open source. If there is a need integration with other programming tools, it can be used CouchDB, which uses JSON for documents, JavaScript for MapReduce requests, and regular HTTP for APIs.

If there is a need dynamic scalability, high performance and reliable security, the GemFire distributed platform will help. Another example is Cassandra, which in addition provides high reliability without sacrificing performance.

Redis is a data structure server where keys can be strings, hashes, lists, sets, and sorted sets.

High scaling properties and the availability of open source give Hazelcast.

Conclusions on the First Part

In this section, the basic concepts of databases and their DBMS were studied, researches of existing database models were carried out, including hierarchical, network, relational, inverted file model, object-oriented, as well as NoSQL.

During the analysis of the information obtained, it was concluded that the best model for this particular task is a relational one. It is most effective when it is needed to ensure complete data independence. If the database structure is changed, the changes made in the application programs will be minimal.

It was found that the most important features in the implementation of a network relational database on board of a ship are:

- saving structured data in a convenient form for the user;
- accuracy (relationships give in to mathematically accurate manipulation methods);
- easy data manipulation at the level of the database language;
- simple navigation through linked tables;
- simple and logical data connectivity;
- simplicity of privacy control.

If it is necessary to process unstructured data, then the effectiveness of such a specific system worsens, due to the inappropriateness of using existing methods for processing this type of data.

The relational approach to building databases simplifies the structure of physical and logical databases, eliminates the need to predict access methods, and makes it possible to process unpredictable queries. Comparing the capabilities of the considered data models, it may be concluded about the attractiveness of the relational approach to building a database.

PART 2
NETWORK DATABASE ADMINISTRATION

Administration of a network database requires some theoretical training, both from the side of the network structure and from the side of database management.

Correctness, accuracy, property and speed of administration work in certain areas of activity is extremely necessary. For example, when our ship meets a potentially enemy one, it is necessary to know the exact characteristics of its armament, the speed of movement at a certain draft of the side, for an accurate calculation of the angle and speed of approach, by other modules of the system.

Thus, the issues of ensuring simultaneous access to the database, sharing information and the speed of requests response become one of the critical ones when planning the network topology and the database running on it. It is also worth taking into account the questions about the clarity of the interface and the ease of interaction with it. The database should be accessible to a user who is superficially versed in database administration, and needs shortness and conciseness of actions so that it is not necessary to go into details, namely, writing queries and subqueries.

Administrative control ensures proper sharing. To perform the role of a database administrator, rights to it can be granted to one or several employees performing these duties.

In small companies there is an opinion that due to the use of the database by a small number of people, no third-party service is required. Such allocation of special means for admin-support of the application is considered inappropriate, while putting forward the lack of time and the lack of a free staff member to perform all duties, as an argument. However, if fall under such opinion and do not make the right decision, then in the future, this can lead to undesirable consequences.

Possible outcome of events such as, deliberate or not, data corruption, lack of a backup copy of the database. Also, specific issues and tasks that require certain administrator qualifications will not be resolved, and ordinary users do not have one. It is also necessary to conclude an agreement on the maintenance of the database, if it was simply purchased from the developers, or there is no way to immediately call the necessary specialist.

To the main tasks of the administrator can be attributed:

- backup;
- periodic compression of database files;
- protection of files and data by means of encryption;
- granting access rights for users and managing their accounts;
- installation of the application, if available, to work with data;
- installation of a client application on a personal computer;
- correct connection of a new user to the database installed on the server.

When in a company a database is used by a team of more than a couple of dozen people, such tasks can arise quite often. Since the range of such tasks can be very wide, in such cases it would be a good recommendation to divide the admin responsibilities among several specialists. If there may be none, it is worth introducing new ones.

2.1. Computer networks and types of topologies

A computer network is a set of some computers that are controlled by special software and are interconnected by special communication lines. Typically, the purpose of such connection of computers is to solve problems associated with computing, operating information flows and also providing educational opportunities. One of the main purposes is to provide shared access to resources that is also shared. Resources can be divided to hardware, software and information.

Usually, the complex of the physical mediums and technical devices providing the transmission of signals from the transmitter to the receiver is called a communication line. An example of communication lines can be sections of different types of cables that

provide signal transmission between computers, switches, routers located in a computer network. Whole communication channels are built on the basis of such communication lines [8].

A communication channel is a system consisting of technical devices and communication lines that provides information transfer between clients. Several communication channels can use the same communication line, and include several communication lines of different types - in this way, the relationship between the lines and communication channels can be described.

Accordingly, to the criterion of scalability, the main types of networks can be distinguished by global and local networks.

Wide area networks (WANs) are networks located over large areas (many kilometers) and can combine not only single remote computers, but also local networks. Often, in global networks, communication channels and cable lines (for example, telephone or telegraph) that were not originally intended for the construction of computer networks are often used, since the laying of specialized high-quality communication channels, for example, between megacities is long and expensive. For this reason, the speed on wide area networks is much lower than on local ones [8]. A schematic representation of the WAN is shown in figure 2.1:

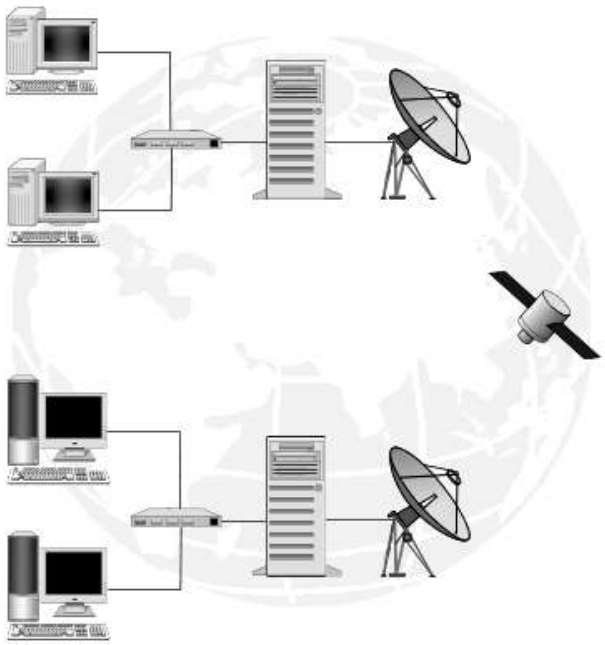


Fig.2.1 Wide Area Network

Local area networks (LANs) are computers located in the network of one company or enterprise, located in relatively small areas within a radius of 1-2 km. Such companies can be located in the offices of one or more buildings, which are located close to each other. A small area allows for laying more expensive cables and installing high-quality technologies that would provide high-speed information transfer between computers. An elementary example of a local network is shown in figure 2.2:

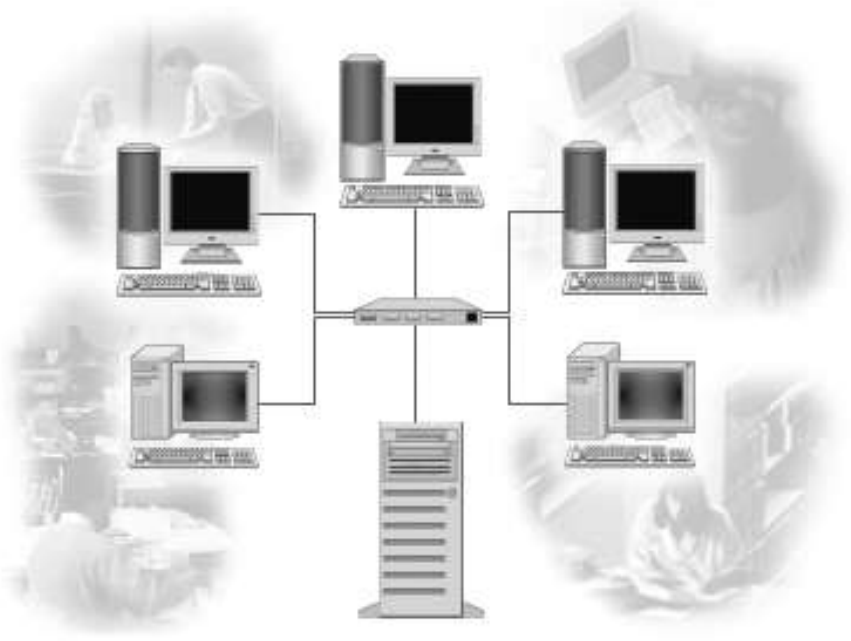


Fig.2.2 Local Area Network

Network topology, or topology of the network, is a way of connection of computers and equipment for communication by routing cables in a special way between them. Another network topology can be represented as a graph, the vertices of which are the technical equipment of the network, and the edges are the communication line [8].

Network topology can be of three types:

- physical – represents the actual location and connections between network nodes;
- logical – describes the signal flow within the physical topology;
- informational – describes the direction of information flows transmitted over the network.

Also, networks can be fully-connected and not fully-connected. Each computer is connected to all the others. It is a fully-connected topology. This communication option is

rather bulky and ineffective, because every computer in the network must have a sufficient number of ports (if they are not enough, it is necessary to provide them with a shortage) for communication with every available computer [8]. An example of such a relationship is shown in figure 2.3:

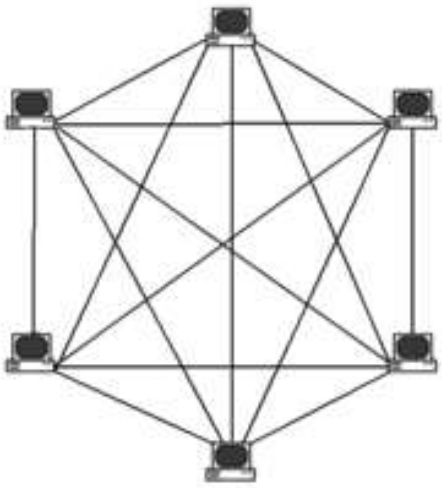


Fig.2.3 Fully-connected Topology

Accordingly, in a loosely-connected (not fully-connected) topology, communication between each with each is not required, and can be provided through additional nodes such as hubs, switches and routers. Non-fully connected topologies include bus, star, ring, mesh and also mixed, which can combine nodes designed on a different topology [8].

A bus topology (fig. 2.4) represents a common cable, sometimes called a backbone, that connects all the working computers on the network. Terminators are installed at each end of the cable to prevent the signal from being reflected.

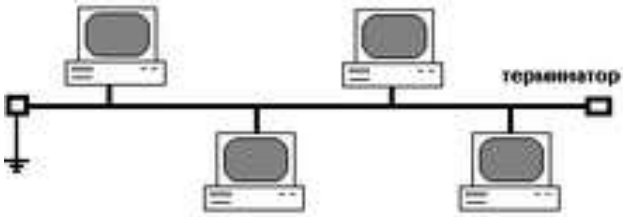


Fig.2.4 Bus topology

Benefits of bus topology networks are:

- low cable consumption;
- failure of one computer does not greatly affect the performance of the network;
- ease of network configuration and its settings [8].

The disadvantages of bus topology networks are:

- the impact of a cable break on the operation of the entire network;
- limited number of workplaces and cable length;
- special cable connectors;
- low performance generated by the channel sharing between all network users.

In a network with a star topology (fig. 2.5), all personal computers are connected to a concentrator or hub, using a twisted pair cable, forming a figure similar to a star. All workstations can communicate with each other thanks to the hub, which allows parallel connection.

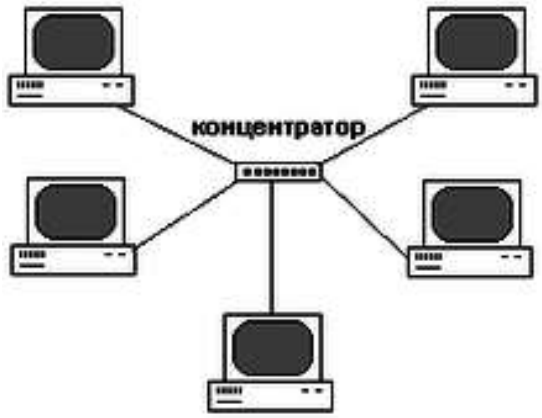


Fig.2.5 Star topology

Data transmitted from one device is transmitted over all communication lines through the hub. The data comes to all stations, but only those to which they addressed are accepted. This topology is actually a bus, since the signal simultaneously propagates to all network nodes, which is a sign of broadcasting. This topology can be used in local area networks with 10Base-T Ethernet architecture [8].

Benefits of star topology networks:

- simplicity of connecting a new network node;
- availability of centralized management;
- high fault tolerance – malfunctions of specific PCs and disconnections in the connection are easily eliminated.

Disadvantages of star topology networks:

- significant consumption during cable laying;

- dependence on the performance of the hub – its failure entails a failure of the entire network.

The name of the ring topology speaks for itself – it is the creation of an unbreakable ring of all nodes, through which data is transmitted, as shown in the figure 2.6. The input of one computer is connected to the output of another, and vice versa. Coming out of one point, data always moves along the ring connection in one direction, bypassing all nodes, and arrives at the starting point of transmission.

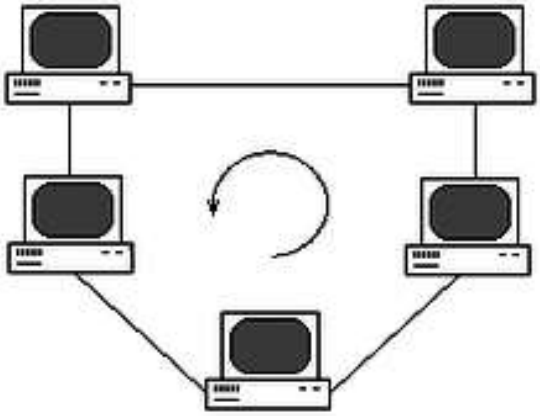


Fig.2.6 Ring topology

The node that receives the data, recognizes it and receives only those addressed to it. This work is organized through the use of token access – giving the computer the right to use the ring in a specific order. The advantage of this ring topology is that it is easy and simple to create and configure. Its global disadvantage is the high probability of damage to the cable or a malfunction of the computer on any part of the network, which causes the entire network to malfunction. Due to the lack of reliability, this topology is not applied in practice, with the exception of various modifications in some parts of the network [8].

A mesh topology (fig. 2.7) can be obtained by removing some of the links in a fully connected topology, thus creating a kind of cells. It is typical for large networks with a large number of personal computers. The home network is a prime example. When a user goes online from home, he does not have a direct cable to all nodes. Instead, he sends data to his provider, and he in turn sends this data where needed [8].

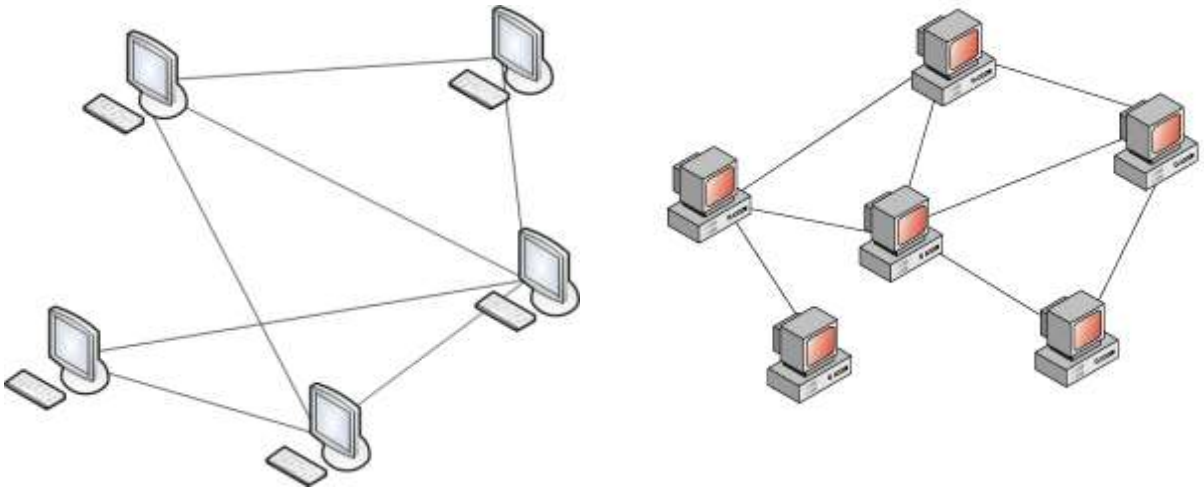


Fig.2.7 Examples of mesh and partial-mesh topology

There are also topologies such as double ring, grade, tree, thick tree, which is a complicated version of a simple tree, and Clos network. They are complementary topologies, making up combinations of basic ones. Another name for them is mixed or hybrid. An example of a mixed topology is illustrated in figure 2.8:

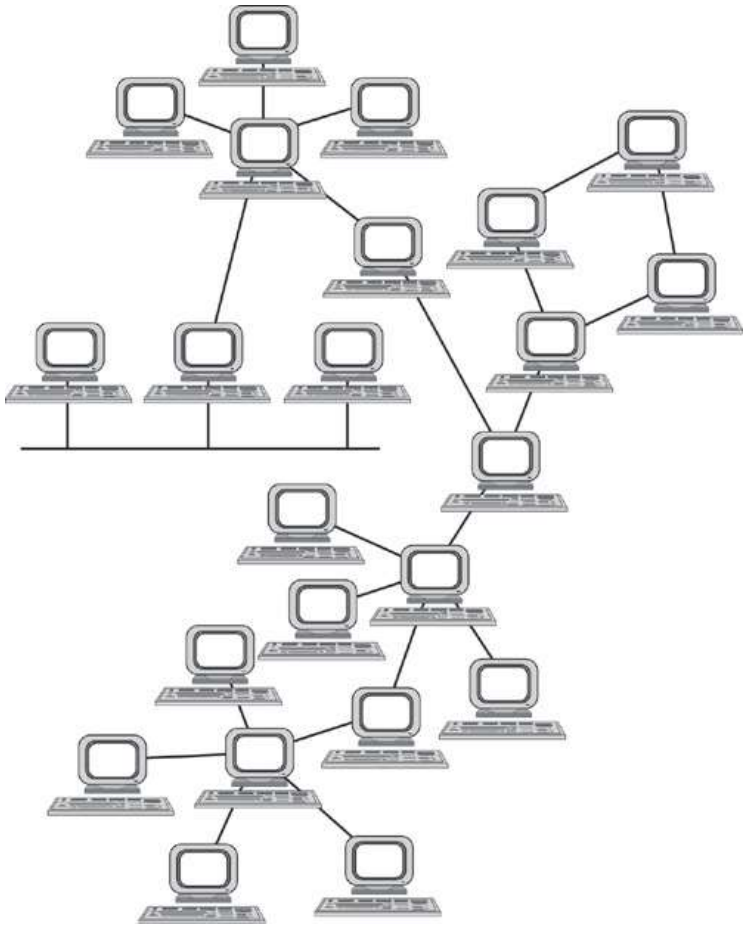


Fig.2.8 Network example, based on mixed topology

Mixed topology is the type of topology prevalent in most large companies with arbitrary connections between network fragments. However, there are several of the most common hybrid topologies:

- star-bus, where nodes with star topology are connected by a backbone bus (fig. 2.9);
- tree structure (fig. 2.10);
- intersecting rings (fig. 2.11).

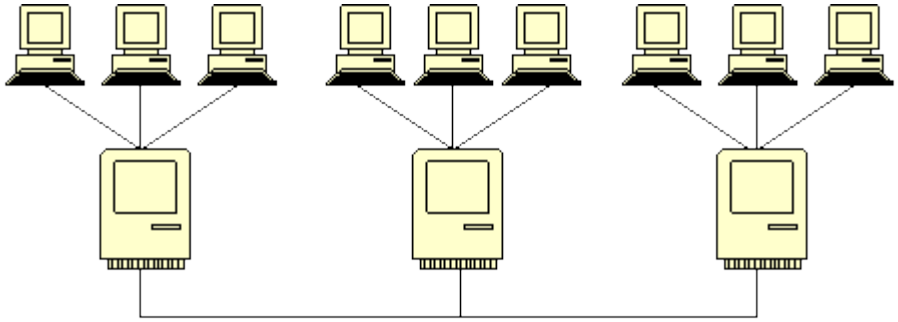


Fig.2.9 Star-bus topology

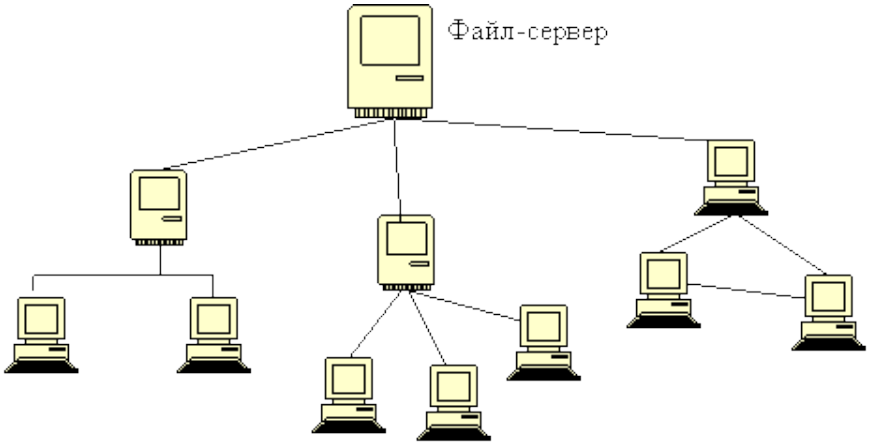


Fig.2.10 Tree topology

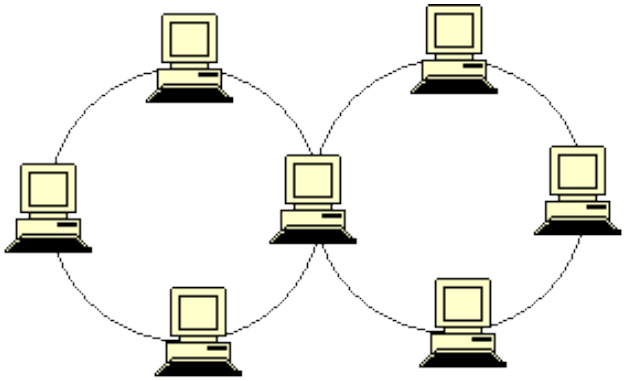


Fig.2.11 Topology of intersecting rings

Typically, local area networks can have symmetric topology, while global networks cannot.

Having considered all types of topologies, it can be concluded that the latter is the most fault-tolerant, taking over most of the advantages from its predecessors.

It is also worth mentioning about the classification into centralized and decentralized topologies.

The star topology is centralized. When a physical star topology is applied to a logical bus network such as Ethernet, a central node (usually a hub) retransmit all transmissions received from any peripheral node to all peripheral nodes in the network, sometimes including towards the initiating node. Thus, all peripheral nodes can communicate with all others by transmitting and receiving from the central node only. Failure of the transmission line connecting any peripheral node to the central node will result in that peripheral node being isolated from all others, and the remaining peripheral nodes will not be affected. However, the disadvantage is that failure of the central node will result in the failure of all peripheral nodes. To reduce the amount of network traffic coming in broadcast mode, more advanced central nodes have been developed that are able to track the uniqueness of nodes connected to the network. These network switches learn the layout of the network by listening to each port during normal data transfer, examining the data packets and writing down the identifier of each connected node and the port to which it is connected in an internal look-up table. This lookup table, stored in dedicated CAM-memory, allows future transmissions to be redirected only to port of their destination [8].

2.2. Roles of network database management

The database management system consists of many concepts, but the concepts of account, role, group, user refers to the system security. When trying to enter the database or access data, each user goes through the stages of checking this security system:

- authentication;

- obtaining appropriate access to information.

At the first stage, the entire DBMS server is involved. At this stage, the user is authenticated using a logical name (login) and password.

Login and password are stored on the DBMS server as an account. The DBMS server allows an attempt to access a specific database, if the credentials were entered correctly, then the authentication procedure is passed.

However, authentication by itself does not give the user access to any data. The specific database user must match the user account. This is necessary to gain access to data. A database user is a set of issued permissions and prohibitions for actions with data in the database. Since there can be quite a lot of users with different rights and prohibitions, to simplify the management of the database security system, they can be combined into roles or groups. Database roles combine multiple users into a single collection. They allow to grant access rights to each member of this role.

Roles can be built-in or custom. The built-in roles cannot be changed because they are created automatically during the installation of the database server. There are built-in roles at the DBMS level and at the level of a specific database [9]. The SQL Server roles are shown in the tabl. 2.1 below:

Table 2.1

Table of SQL server roles

Built-in server role	Appointment
Sysadmin	Can perform any action in SQL Server
Serveradmin	Performs server shutdown and its configuration
Setupadmin	Manages connected servers and procedures that start automatically
Securityadmin	Manages accounts and database creation rights.
Processadmin	Controls processes running in SQL Server
Dbcreator	Can create and modify databases
Diskadmin	Manages SQL Server files
Bukladmin	Can paste data using mass means of copying without having direct access to the table

SQL Server also supports another group of built-in roles of database level listed in tabl. 2.2 below.

Table 2.2

Table of another SQL server roles

Built-in server role	Appointment
Db_owner	Has all the rights in the database
Db_accessadmin	Can add or remove users
Db_securityadmin	Manages all extensions, objects, roles, and role members
Db_ddladmin	Can execute any DDL commands except GRANT, DENY, REVOKE
Db_backupoperator	Can execute commands DBCC, CHECKPOINT, BACKUP
Db_datareader	Can view any data in any table
Db_datawriter	Can modify any data in any table
Db_denydatareader	It is forbidden to view data in any table
Db_denydatawriter	It is forbidden to modify any data in the table

All of these SQL server roles cannot be removed. In addition to the above roles, there is one more role that cannot be removed – public. All users who have access to the database are a priori members of it. As all users are already included in it, then it cannot be explicitly specified its members [9].

2.3. Data integrity and encryption

The need to encrypt files is driven by the desire and need to keep data safe and sound. In MS SQL Server, a certain method is used to change data that has an unreadable form, and has a name – encryption. Encryption ensures that valuable and confidential information cannot be viewed by other users to whom it is not intended. The user can copy this data, but he cannot use it. Authorized users use decryption when viewing data. MS SQL Server encryption is subject to:

- all data that can be transferred between the server and the client over the network;
- roles or passwords of SQL Server accounts;
- code of triggers, stored procedures, as well as views, that is, everything all, that is used to create database objects.

In SQL Server system tables, passwords for accounts and roles are stored encrypted. This protects them from being viewed by any user, even the administrator cannot view them. Before sending application passwords to the server, they may already be encrypted.

It is also worth encrypting the trigger, stored procedure, or view code if they contain data or an algorithm. It can be done without it, but encryption allows to be sure that the data of various stored procedures and others will not be leaked.

Encrypting data as it travels over the network ensures that no application or device can read it, even if it intercepts it. Encrypting the connection also prevents interception of user passwords [10].

The Multiprotocol Net Library is a network library that should be used if data needs to be transferred over a network.

Restricting access to MS SQL Server files is necessary to avoid their, intentional or not, damage.

Databases, their backups, lag and error logs, files for importing and exporting data, and much more, all this is created and used by MS SQL Server in its work. All these data are subject to the possibility of changes in one way or another. SQL Server services run as processes on the level of operating system. That is why they can be found in the task manager processes of any OS. For SQL Server to function properly, these processes must have full access to all of the above files in the file system. For this, it is needed to set the corresponding rights of the accounts used to run MS SQL Server at the operating system level. It is still better to manage access rights at the file and folder level. To do this, the server must run under the Windows NT operating system and have the NTFS file system [10]. The Windows operating system requirement is not critical as some Linux kernel-based operating systems also support the NTFS file system.

It must be granted full access rights for the accounts used to start the services if the server is started as a service. The server may have access rights for the user account that started SQL Server from the command line. In the case of using the local system account to start the server, access must be granted to the SYSTEM user.

It is necessary to prohibit all users, except for SQL Server, to restrict unauthorized access to files. The prohibition should be set for use by any users, namely reading, deleting and any modifications to the data.

To restrict unauthorized access to SQL Server files, you must set the read, delete, modify, and execute prohibition to all users except SQL Server itself.

In order to restrict unauthorized access to SQL Server files, it is necessary to prohibit reading, deleting, modifying and executing all users except SQL Server itself.

2.4. Improving the algorithm for obtaining access rights to the database operator

After examining information from the information sources, it can be found that a lot of information about granting, restricting and denying access to data, but there is no clear sequence of assignment. From this, it can be concluded that each system administrator or person in a similar position does it in his own way. However, it is worth considering all kinds of rights and highlighting the best procedure for granting them.

Some users may only see data partially, have limited ability to change this data, and some users have full access to all data stored in the database. In order for the user to be able to see or change the data, person must be given this right. This right is called a privilege. Privileges could be system and object. The syntax for granting these privileges differs depending on the selected DBMS. Some users can create new users, can create procedures or delete them, create indexes or tables. That is, to perform manipulations with various database objects.

It also can be granted to the user system privileges with the administrator option. The database administrator can initially issue access rights. There is a command to revoke

previously granted privileges. To do this, it is needed to run this command and specify what and from whom it is necessary to delete.

The database can simultaneously work multiple users, and each user can have its own set of privileges. It often happens that it is required to provide granting the same privileges to multiple users. Doing this every time is quite laborious. To reduce effort, it can be created a role. A role is a collection of privileges combined under one name. It also can be assigned roles to a user and thereby give him all the privileges formed in this role. This enables efficient administration. In addition, it can be dynamically managing user rights. For example, it is needed to add the ability to change a table for many users. This option is added to the role, and all users are immediately granted this privilege. As already mentioned, roles can be created by a database administrator who has system privilege. Roles are very common in many DBMSs, but unfortunately in MySQL, roles are not supported.

In order to derive a certain algorithm for granting access rights to the database user, it is necessary to consider what access rights exist, types of transactions, what are explicit and implicit rights, as well as prohibition and their rejection.

2.4.1. Rights and access to database objects

Users connected to MS SQL Server can perform actions that are determined by their rights. As mentioned earlier, they are issued to their account, or the group or role in which they belong. Rights in SQL Server can be divided into three categories [11]:

1. Rights to access database objects;
2. Rights to execute Transact SQL commands;
3. Implicit rights (permissions).

As soon as the user was created, he was immediately given the access rights of the special database role – public, and nothing else. Such rights are available to all users in the database.

The rights are granted by the database administrator or their owner. As already know, the user can be awarded with a certain set of rights, and this is what roles serves. The administrator can create these roles in advance and grant them the necessary access rights. Then users can simply be added to the appropriate roles, or rather assign them a

certain role. The roles already in the database are standard and already have a certain set of rights. For example, a user with the “db_datareader” role can view any data in any table. Standard DB types and their purpose are described in the table [11].

It is important with carefulness granting permissions to data access. There is a need to control all access rights carefully. This is especially refers for large companies with a staff of thousands of users, with serious security systems, where dozens of groups and roles are created to prevent unpleasant situations. It is necessary to wisely approach about creating roles so that the existing security system correctly gives the user the ability to perform the necessary actions with the data, but would restrict access to unnecessary information. For even more flexible security management, it can be specified an access rights to a specific column, not just at the level of the table.

These guidelines apply to Transact SQL commands, where the database design will strictly define users to create tables, rules, views, and even backups.

Working with data and executing stored procedures requires an access class called database object permissions. Objects should be understood as tables, table columns, views, stored procedures. Database object authority controls the ability of users to execute, for example, SELECT, INSERT, UPDATE, and DELETE commands on tables and views. Thus, if the user needs to add new data to the table, he should be granted the INSERT (insert records into the table) right. To execute any stored procedure, the user must be granted EXECUTE privilege [11].

For different objects it can applied a different set of rights of access to them:

- SELECT, INSERT, UPDATE, DELETE and REFERENCES – these rights can be used to operate on tables;
- SELECT and UPDATE – these rights can be applied in relation to a specific column of the table;
- EXECUTE – this right is used to save procedures and functions.

The INSERT privilege allows new data to be inserted as rows not only into a table, but also into a view. It is only issued at the table or view level and cannot be issued at the column level [11].

UPDATE privileges are issued at either the table or separate column level. At the table level, UPDATE allows to change all the data in a table, and at the level of an individual column, it allows to change data within one column [11].

To delete rows from tables, use the DELETE privilege. This right, just like INSERT is issued at the table or view level, it is not at the column level.

To select data, there exists a SELECT privilege, which is issued both at the table level and at the level of an individual column.

The REFERENCES privilege enables to refer to the specified object. It allows the user to create a foreign key that is sent to the primary with respect to the tables. And in relation to views, it allows them to be linked. This allows to track changes in the structure of the source tables, which may affect the operation of the view [11].

As mentioned above, access can be provided both at the level of an individual column and at the level of the entire table or view. However, column-specific permission management is rarely used. The view is best used when users need to restrict access to one or more columns.

For enabling the reading data from all tables and views of the database, it is enough to include the user in the fixed role “db_datareader”, and not to change the user's access rights to each table and view separately [11].

For user permissions control access to different database objects used by the GRANT command. The complete SQL syntax is as follows:

```
GRANT
{ ALL [ PRIVILEGES ] | permission [,... n ]}
{
[ ( column [,... n ] )] ON { table | view }
| ON { table | view } [ ( column [,... n ] )]
| ON { stored_procedure | extended_procedure }
}
TO security_account [,... n ]
[ WITH GRANT OPTION ]
[ AS { group | role }] .
```

It should be considered the meaning of the basic parameters of the GRANT command. The ALL parameter gives the user all available rights and accesses. It can only be assigned by members with the sysadmin role. Permission is a list of the available permissions granted to the user. Several accesses can be granted at the same time, separating them with a comma (For example: SELECT, INSERT, UPDATE, DELETE, EXECUTE).

Name of the object security, which must be included in part – is security_account. Such objects can be both SQL Server accounts and Windows NT users and groups of users who are granted access to the database server. The parameters table, view, column, stored_procedure, extended_procedure are responsible for the names of objects in the database to which it is needed to grant access [11].

Ensure the right of access to the object to other users allows the use of option WITH GRANT OPTION.

And here is the parameter [AS {group | role}] is optional, but allows to specify the user's participation in a role that has the ability to grant rights to other users [11].

For example, it must be provided granting of INSERT and SELECT commands to the “Students” group in the “Exams” table. At the same time, it is necessary that in the future the users of this group can themselves grant similar rights. To do this, next command should be run: GRANT SELECT, INSERT ON Exams TO Students WITH GRANT OPTION. As a result, the user “Volodymyr” who is a member of the “Department” group can grant similar rights to another “Teacher” user: GRANT SELECT, INSERT ON Exams TO Teacher AS Department. In this case, it is necessary to confirm on what basis “Volodymyr” grants the rights, therefore the AS parameter is applied [11].

Using the WITH GRANT OPTION parameter should be well thought out, because control over the presentation of the rights of access to other users may be lost. The range of people who have the ability to manage the assignment of rights to other users should be minimized.

The most necessary and practically the only access right for a stored procedure is the EXECUTE right – the right to execute it. The owner of such a stored procedure can not only view the code, but also modify it. Functions can also be given execution rights. But

the REFERENCES right will provide the ability to bind a function with the objects that this function refers to. Such binding allows to save changes to the structure of objects, which could lead to disruption of the function [11].

2.4.2. Transact SQL command execution rights

Transact SQL has such features as control statements, additional functions for processing dates, strings and other things, the presence of local and global variables. All applications that interact with an instance of SQL Server can send statements of Transact SQL. Thus, the key to using SQL Server is Transact SQL.

As the name suggests, Transact SQL is based on a transactions. Transactions are a series of operations that are combined into one logical unit. Under the module it may be referred a request for data retrieval, change of data structure or table.

After the start of the execution of transactions, all data involved in a particular scenario is blocked for the duration. This occurs in order to prevent data inconsistency on the finalization of the scenario with the initial data table.

The Transact SQL rights class takes control of the creation of not only the database itself, but also its objects. Also, the class of such commands performs control over the creation of a database backup. For example, the administrator grants some user with the right to execute the CREATE VIEW command in case the user needs to create a view.

Command rights are shown in tabl. 2.3.

Table 2.3

Table of rights for Transact SQL commands

Transact SQL command	Appointment
CREATE DATABASE	Creating databases, permission is applied to the command itself
CREATE TABLE	Creating tables
CREATE VIEW	Creating views
CREATE DEFAULT	Creating defaults
CREATE RULE	Creating rules

Continuation of tabl. 2.3

CREATE PROCEDURE	Creating saved procedures
BACKUP DATABASE	Creating database backup
BACKUP LOG	Creating of transaction backup log
AL	All rights listed above are granted by this right

Automatically members of the server role have ALL right– sysadmin and the role of the database – db_owner. All users of other fixed roles have a set of rights that correspond to the functions of the role. For providing the right to execution of the Transact SQL command it can be used command: GRANT {ALL | statement [... n]} TO security_account [... n]. Permissions to execution of the Transact SQL commands that are listed in table above can be passed via the statement parameter [12].

In tabl. 2.4 below is a list of stored Transact SQL commands and procedures used to manage security.

Table 2.4

Table of stored Transact SQL commands and procedures

Transact SQL command	Assignment
Sp_addapprole	Create a role for the program
Sp_addlogin	Create a new SQL Server account
Sp_addrole	Create a new database role
Sp_addrolemember	Add a member to the database role
Sp_addsrvrolemember	Add a new member to a fixed server role
Sp_approlepassword	Change the password for the application role
Sp_defaultdb	Change the default database for the account
Sp_defaultlanguage	Change the default language for the account
Sp_denylogin	Restrict access to a Windows NT user or group
Sp_dropapprole	Delete the application role
Sp_droplinkedserver	Delete account display from another server

Continuation of tabl. 2.4

Sp_droplogin	Delete the SQL Server account
Sp_droprole	Delete the database role
Sp_droprolemember	Remove the user from the database role
Sp_dropsrvrolemember	Remove a member from the server role
Sp_dropuser	Delete user from database
Sp_grantdbaccess	Allow access to the account database for Windows NT users and groups
Sp_grantlogin	Allow access to SQL Server
Sp_helpdbfixedrole	Issue a list of fixed roles in the database
Sp_helplogins	View account information
Sp_helpntgroup	View information about Windows NT groups that have been granted access to SQL Server
Sp_helprole	View the roles defined for the database
Sp_helpsrvrole	Provide a list of fixed server roles
Sp_helpsrvrolemember	Provide information about the member of the server role
Sp_helpuser	View user information
Sp_password	Change the password of the SQL Server account
Sp_revokelogin	Delete a Windows NT user or group
Sp_setapprole	Initialize the application role
GRANT	Grant access
DENY	Deny access
REVOKE	Implicitly deny access

Stored procedures are the set statements of Transact SQL, stored on the server. Stored procedures are more powerful than view objects because they are a method for performing repetitive tasks. Preserving them, in other words, prescribing them in advance and their further use, allows to significantly optimize the operation of the system by reducing the time for their creation.

Performing such a procedure involves creating it, executing it, and then placing it in the cache. A procedure runs much faster if its execution plan is already in the cache. Thus, the server does not need to parse the request and develop the necessary actions to solve the problem, due to which the procedure is performed faster. Creating a procedure is very similar to creating an object.

Tables, functions, view objects, temporary tables, and other procedures can be used as objects in the code when creating a procedure. Stored procedures have the following benefits:

- sharing access and modification – separation of logic;
- protection mechanism – users can get the right to perform those procedures that contain tables to which they have no rights;
- increased productivity – provides a variety of tasks;
- lack of direct access to tables allows to avoid deliberate or not, damage to information. All necessary actions can be performed using procedures that act as a guarantor of the integrity and security of database data.

2.4.3. Implicit permission rights

By default, there is an access to perform some actions. Such a class of rights does not require explicit permissions and provides such actions. Only owners of database objects or members of a server role can perform this action.

Implicit rights cannot be granted directly to the database user. User can get them only under certain circumstances. For example, a user can create an object himself, and thus become the owner of the database. Another way of becoming an owner occurs through the transfer, by another user, of such rights of ownership of an object. Thus, the new owner of the object automatically gets the rights to perform any actions with the object, including the right to grant access to the object to other users. The fact of ownership of an object allows the user to perform any action, and at the same time, the rights are not explicitly indicated anywhere.

The situation with server roles is similar. For example, rights to control SQL Server processes can only be granted to a user by including him in the processadmin role. There is no way to give this user control over the SQL Server processes.

There are three access options for a specific action that is controlled by permission – grant, deny, and implicit denial.

For a specific action controlled by permission, there are three options for the access state: grant, denial and implicit rejection (GRANT, DENY, REVOKE) (fig. 2.12).

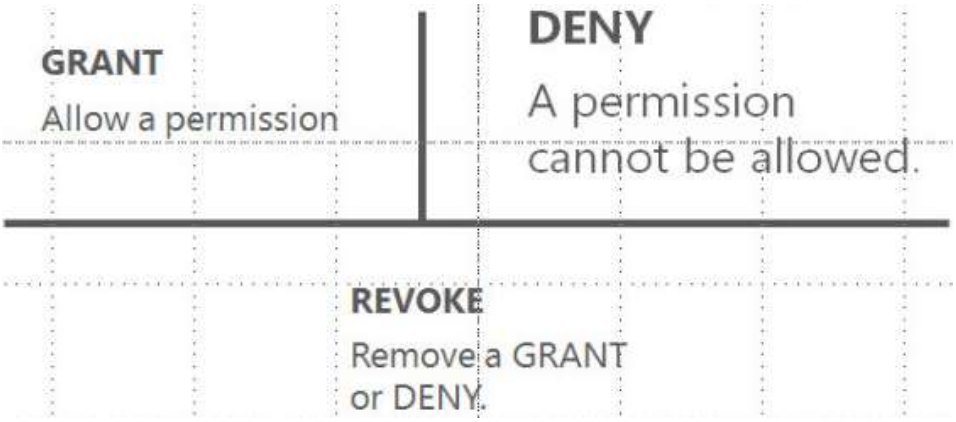


Fig.2.12 Access states in SQL

Implicit denial is similar to implicit denial of access. However, the main difference is its action at the level at which it is defined. User access to data is implicitly denied by default. Suppose a user at a certain level can be implicitly denied access, but he can get it at a different level of the hierarchy. To do this, he must be in a role that has this access right.

2.4.4. Prohibition and denial of access

As mentioned above, since the database allows to include the roles of SQL Server accounts and Windows NT groups, the SQL Server security system is hierarchical. User participation in multiple roles is allowed.

A signal for a possible security disturbance is a consequence of the hierarchical structure: a user can simultaneously have different access rights for different roles.

If a role has permission for certain access to data, then the user in this role has similar rights automatically.

Sometimes it is required to deny the ability to access data. In this case, if the user is denied access to data or Transact SQL commands, all permissions granted by the user at any level of the hierarchy to that access are revoked.

At the same time, a guarantee is given that access to prohibition will continue to be independent, that is, the permissions granted at the levels above remain.

A good example is creating a regular table. Access to such a table is provided by default to all database users; it is necessary to temporarily restrict access to specific users of this table. This can be done in two ways: revoke access on permissions and creation of a restrictive role. In such a role, access will be limited to this table. In the latter case, it must be included the user in such a role. With this method, it is easier to control multiple entries in a denied role than to manage and control a number of disparate accounts. With such actions, of course, it will be more difficult to monitor whether access rights have been returned to the user back, which can affect the decrease in operator productivity and postpone the launch of the product, since the administrator will be forced to redo a bunch of unnecessary work. Consequently, with a large number of users, this approach can simplify security administration.

To deny a user access to database objects, it is necessary to use the DENY command:

```
DENY
{ ALL [PRIVILEGES] | permission[,...n] }
{
[(column[,...n])] ON {table|view}
| ON {table|view} [(column[,...n])]
| ON {stored_procedure | extended_procedure}
}
TO security_account[,...n]
[CASCADE].
```

To disallow the execution of Transact SQL commands, use another command: DENY {ALL [PRIVILEGES] | permission [, ... n]} TO security_account [, ... n].

The parameters of this command are the same as those of the GRANT command. The CASCADE parameter speaks for itself. It allows to cascade the revocation of all users who were granted access by the first user. To understand the above, consideration of an example is obligatory.

The CASCADE parameter allows to revoke rights not only from this user, but also from all those users to whom he has granted these rights. Explanation of the meaning of the above will be with an example. Let the “ChiefTeacher” user be granted specific rights: GRANT CREATE TABLE To ChiefTeacher WITH GRANT OPTION.

Then, it is assumed that “ChiefTeacher” will grant similar rights to certain users, says “Teacher”. To revoke access from “ChiefTeacher” in the future, it must be running the command:

```
DENY CREATE TABLE
TO ChiefTeacher
CASCADE.
```

With this cancellation of access rights to create a table in “ChiefTeacher”, they will revoke all permits for access rights from those granted this user.

As discussed earlier, denying access is similar to implicitly denying access. What is distinctive is that the implicit deviation operates at the level at which it is defined. This can be understood as the absence of any permissions, that is, the “middle state” between granting and denying access.

To implicitly deny access to database objects, used the REVOKE command:

```
REVOKE [GRANT OPTION FOR]
{ ALL [PRIVILEGES] | permission[,...n] }
{
[(column[,...n])] ON {table|view}
| ON {table|view} [(column[,...n])]
| ON {stored_procedure | extended_procedure}
}
TO security_account[,...n]
[CASCADE]
```

[AS { group | role }] [13].

For an implicit rejection of access to perform the following Transact SQL command structure can be used:

REVOKE { ALL | statement[,...n] } FROM security_account[,...n].

The essence of applying parameters is the same as for the GRANT and DENY parameters. When it is necessary to revoke the privilege that was granted with the WITH GRANT OPTION parameter of the GRANT command, it must be used the GRANT OPTION FOR parameter. In this situation, the user loses the ability to grant this permission to other users, but remains the permission to access the object.

For example, all members of the “Students” role have access rights to the “Subjects” table. The “Student” user belongs to the “Students” role. Even when using the REVOKE command to deny access for the “Students” role to the “Subjects” table, “Student1” can still access the table. This will happen because the rights for such a user are implicitly defined. Such a user will lose the right to access this table only when the REVOKE command is applied specifically to him.

Configuring the security system becomes more flexible when applying the implicit rejection of access rights. However, the potential for confusion with roles, their rights and permissions is increases.

Members of roles or groups inherit the permissions that are granted to them. Since a user can have multiple roles, access conflicts can occur [18]. They can arise by granting membership in one role (where certain actions with the object are allowed), and at the same time in another role that has access of another, much lower level, where there is a restriction on actions with the object.

In SQL Server, there is a principle for resolving conflicts: permission to deny access has the highest priority, and permission to grant access is the lowest. This means that there should be no denial of access to data at any other level of the security hierarchy, and this access can only be obtained by explicitly granting it. If access is not explicitly granted, the user will not be able to work with the data.

Conclusions on the Second Part

Regardless of the size of the company's staff, it still needs an employee with suitable qualifications.

It can be distinguished the most important tasks that a database administrator must perform:

- user account management;
- granting permissions and restrictions to users;
- installation of software accompanying the work for all users of the system;
- training of employees, if it is necessary;
- create a backup copies from time to time.

Having studied the above material for the best approach to solving the issue of issuing and rejecting the access rights of the DB user, it can be concluded that a certain case requires a certain order of granting certain rights to the user.

It was also determined that the creation of roles and the inclusion of users in them, requires an initial analysis of the subject area. It is that who determines the need to create database roles, determine the need for redundant data encryption, the need to create stored procedures, assign appropriate privileges, the right to execute Transact SQL commands, resolve access conflicts, etc.

In SQL Server, there is a principle for resolving conflicts: permission to deny access has the highest priority, and permission to grant access is the lowest. This means that there should be no denial of access to data at any other level of the security hierarchy, and this access can only be obtained by explicitly granting it. Members of roles or groups inherit the permissions that are granted to them. Since a user can have multiple roles, access conflicts can occur.

PART 3

DEVELOPMENT OF A COMPLEX FOR THE ADMINISTRATION OF NETWORK DATABASE “NAVY”

The purpose of the development of this complex is to create an effective interaction between the system and the person.

To develop holistic complex for the administration of the network database "Navy", it is necessary to develop topology of computer network, a database of ships and submarines, and applications for the administration of this database.

Since the development of a computer network on board the ship is impossible due to the absence of scheme of the structure of the ship, this network will be developed for the stand hall, in which the location of all equipment and the laying of connection cables (cabling) will be simulated. A stand hall is a room similar in size to real data, in which a computer network will be laid.

Since building a network is not the easiest thing, first of all it is necessary to decide on the choice of topology that will provide the best connections that will provide the fastest and highest quality interaction. In the future, this equipment will be configured by system administrators.

To create a successful database, it is necessary to analyze the data that will be stored in it.

When writing an application for administration, it is necessary to conduct a conversation with the customer to find out the basic requirements that need to be displayed.

3.1. Development of a network topology for the “Navy” database

Network topologies for companies and firms are unique and are created specifically for each case. This network is very specific, since there is not much space where all the computer equipment on the ship is located. Computers, hubs, switches, routers and servers will be located close to each other, for better interaction between the officers, who need to communicate freely with each other. The amount of information passing between computers is not large. Moreover, other technologies are used for data transmission, such as Vortex Opensplice DDS of the ADLINK company.

In this project, the required number of automated work places, namely computers, is determined by the number of crew members – ship commander, senior assistant of the ship commander, combat control assistant, underwater situation control officer, surface situation control officer, officer responsible for performing tactical and logical tasks.

A star topology was chosen as the base topology to build the entire LAN topology of the room. Thus, it is necessary to allocate one switch for transmitting data between the ship's commander, officer responsible for performing tactical and logical tasks, combat control assistant, and one switch for communicating the senior assistant to the ship's commander, the rest of the officers and ordinary crew members.

Since all staff are distributed over switches (“Switch0” and “Switch1”), it is necessary to connect them and the server to the router. It should also be said that there will be two server racks for storing information. One is full-time, and the second “Server R” will serve as a backup, in case of failure of the first. For this, a “star” topology will also be used, the center of which will be a router “OutRouter”, which, in turn, is connected by a communication line with an external provider.

DNS and DHCP servers are optional as the main purpose of DHCP is to allow a device on the network to dynamically obtain an IP address. And if there are about 10 computers in the room, it is possible to set up all the addresses manually, since it is known that the number of work places (AWP) will not be increased. At the same time, the DNS service makes a request to translate domain names into IP addresses. For example, when

accessing Internet resources in the URL field, the user enters the readable name “google.com”, and the resource is accessed by IP addresses.

To build the network topology necessary for the correct operation of the entire ship complex, the Cisco Packet Tracer program was chosen. Having considered all the features and needs of the network, the construction result is shown in figure 3.1. All connections were shown symbolically, using the designations of all devices, since the space for placement is not large.

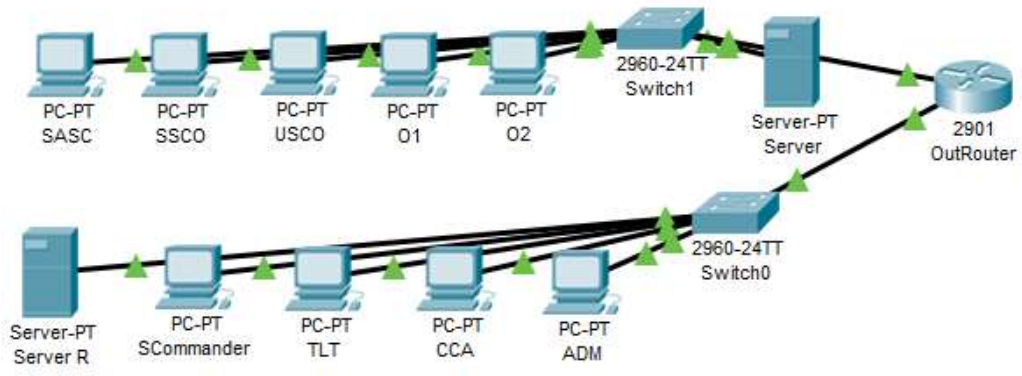


Fig.3.1 Screenshot of topology designed in Cisco Packet Tracer

As it can be seen from the screenshot of the constructed topology, all devices have been configured. For example, it is worth showing the router configuration:

```
Router>en
Router#conf t
Router(config)#int fa0/0
Router(config-if)#ip address 10.1.1.1 255.0.0.0
Router(config-if)#no shut
Router(config-if)#end
Router#wr mem
```

The “en” command means to enable advanced customized settings, the “conf t” command opens the configurational terminal, and the “no shut” command – enable interface, that were set up.

3.2. Equipment selection

As already mentioned, this system is very specific – the equipment will be installed on board the ship. Thus, the system must be water resistant, have increased shock resistance, resistance to temperature changes, salt resistant, and the input device must be

protected with a special coating. From this it can be concluded that the choice of equipment should be made among industrial equipment, namely computers.

An industrial computer is a computer adapted to work in unfavorable climatic or specific, natural or artificial conditions. Such computers are developed and used in the military-industrial complex, as well as in systems that require a continuous and long-termed period of operation. At the same time, they must ensure high quality of processed information, uninterrupted operation and high reliability in operation. All characteristics are individual and are determined by the requirements of the specific purpose of the customer. Devices for input and output of information, system unit, monitor – all these parts are united by a special case.

By researching the existing product market, an analysis was made and the most suitable equipment was selected. Thus, WS-612GS/A225A/R-R20 6U 10.4 "400cd/m2 SVGA LCD 8-Slot Industrial Workstation was chosen to design the workplace to the senior assistant of the ship commander, combat control assistant, underwater situation control officer, surface situation control officer and ordinary staff. This is a panel with a built-in LCD display, robust casing, membrane keypad. Specifications are presented in tabl. 3.1, and the device itself is shown in the figure 3.2 below.

Such equipment maintains its ability to work 24/7. All of its components have increased wear resistance. For example, industrial SSDs may exceed those in the 5-10 times longer, than those used in consumer computers. Random access memory for industrial computers is practically the same, except that it can use components with increased reliability, extended temperature range and vibration resistance.

Fig.3.2 Industrial Workstation WS-612GS

Table 3.1

Table of technical specifications of WS-612GS

Model	WS-612GS
LCD panel	10.4”
Resolution	800 x 600
Brightness, (cd / m²)	400
Contrast	700:1
Number of colors	262K
Backlight, h	50000 (LED Backlight)
Input signal	Analog VGA (DB-15)
Mounting	19”Rack
Front panel material	Aluminum
Degree of protection	IP 64
Number of keys keyboard	63 Keys
Function Keys	20 Keys
Touchpad	Yes
Front USB	4
Storage devices	1 x 3.5” HDD
System fan	12 cmball bearing 99.6 CFM fan

Continuation of tabl. 3.1

Touch screen	5-wire resistive touch screen
Vibration resistance	5 ~ 17 Hz, 0.1” double amplitude displacement 17~640 Hz, 1.5G acceleration peak to peak
Shock resistance	10G acceleration peak to peak (11ms)
Relative humidity	5 ~ 95%, non-condensing
Working temperature	0°C~ 50°C
Dimensions,	483 x 266 x 248
(W x H x D) (mm)	9.2 / 14.4 kg

By researching the existing product market, an analysis was made and the most suitable equipment was selected. Thus, fanless panel computer GOT812L(H)-880 was chosen to design the workplace of the ship commander and officer responsible for performing tactical and logical tasks. It has Intel Core i5-4300U base. Specifications are presented in tabl. 3.2, and the device itself is shown in the figure 3.3 below.

Fig.3.3 Fanless panel computer GOT812L(H)-880

The fanless panel computer was chosen specifically for the ship's commander and the officer responsible for tactical and logical tasks, because their specific work does not obligate them to constantly use additional input means. Despite the fact, that the computer panel has touch-sensitive screen and data entry is present, it is still necessary to select an additional tool for entering data to prevent the possibility of failure.

Table 3.2

Table of technical specifications of GOT812L(H)-880

LCD Panel	12.1" XGA TFT LCD; 1024 x 768 pixels
CPU	Intel® Core™ i5-4300U 1.9 GHz (Haswell ULT)
Chipset	SoC integrated
System Memory	1 x 204-pin SO-DIMM DDR3L-1600, up to 8 GB
COM	2 x RS-232/422/485 (default RS-232, A-Coded)
USB	4 x USB 2.0 (A-Coded)
Ethernet	1 x 10/100/1000 Mbps (X-Coded; Intel® i218LM)
Storage	1 x 2.5" SSD
Expansion	2 x PCI Express Mini Card slot
Touchscreen	5-wire resistive touch or projected capacitive multi-touch
Front Bezel	Stainless type 316
Power Supply	9 to 36 VDC
Power Comsumption	56W
Watchdog Timer	255 levels, 1 ~ 255 sec.
Operating Temperature	-20°C to +50°C (-4°F to +122°F)
Relative Humidity	10% to 95% @ +40°C, non-condensing
Vibration	2G, 5 ~ 500Hz, random for 2.5" SSD
Dimensions (W x D x H)	318 x 248.5 x 75.3 mm (12.52" x 9.78" x 2.96")
Weight (net/gross)	4.27 kg (9.41 lb)/5.77 kg (12.72 lb)
Mounting	Suspension; VESA mount

This workstation has a built-in keyboard, however, to reduce the probability of its failure, an additional one is required. A pull-out keyboard AX7042 is suitable for this case shown on the figure 3.4.

Fig.3.4 Pull-out keyboard AX7042

From the technical characteristics can be distinguished the presence of the touchpad, 1U height, the keyboard connection - USB, mouse - PS / 2, possibility of mounting in a 19"rack.

In addition to production computers, there are production servers. An example of this would be the ASR-5200E server (figure 3.5).

Fig.3.5 Production server ASR-5200E

Main technical characteristics are shown in the tabl. 3.3:

Table 3.3

Table of technical specifications of ASR-5200E

Form Factor	2U 12-Bay (LFF) / 24-Bay (SFF)
Configuration	Dual Controller
Number of Drives	12-Bay (3.5") / 24-Bay (2.5")
Drive Type	SAS 6Gb/s
Cache Memory	8 GB
Max Drives	60 x 3.5" drives / 120 x 2.5" drives
Default Host	4 x 1Gb/s iSCSI RJ45 port; 6 x 6Gb/s SAS wide-port

Continuation of tabl. 3.3

Advanced Functions	Thin Provisioning; FlashCopy (Default 64 Targets); Easy Tier	
Optional Features	FlashCopy upgrade (up to 2048 Targets); Remote Mirror	
Power Output Wattage	800W Redundant Power	
InputRange	AC 100-240V	
Dimensions	556 x 483 x 87 (mm)	
Weight	18 kg (without hard drives)	
	Operating	Non-operating
Temperature	10°C ~ 35°C (50°F ~ 95°F)	-10°C ~ 50°C (14°F ~ 125°F)
Humidity	20%~80%, 35°C	10% ~ 90%, 35°C

When developing the network topology, a wired connection with a router was indicated, however, in a ship environment, the AWK-3131A-EU-T Mesh wireless access point would be the best device (fig. 3.6). The basic characteristics are shown in tabl 3.4.

Fig.3.6 Wireless Access Point AWK-3131A-EU-T

Table 3.4

Table of basic technical specifications of AWK-3131A-EU-T

Number of Ethernet connectors:	1
Number of 10/100/1000 Mb Ethernet:	1
Connector type 10/100/1000 Mb Ethernet:	1xRJ-45
Wi-Fi type:	802.11a/b/g/n
Wi-Fi Bandwidth:	300 Mbps
Number of antennas:	2
Standard protocols:	SNTP, HTTP, HTTPS, PPPoE, RADIUS, ICMP, TCP, UDP, DHCP, DNS, SNMP, Proxy ARP, VLAN, STP/RSTP
Cybersecurity:	NAT, Firewall, IP Address Denying, IEEE 802.1x Access Control

To connect workstations with a router, it must be used Moxa EDS-208A switch. It is a compact industrial Ethernet switch (fig. 3.7) that supports 8/5 Ethernet ports with 10/100M full/half-duplex and MDI/MDI-X auto-sensing. The MOXA EDS-208A/205A can be used in rough environments, including marine (DNV/GL) and comply with FCC, TUV, UL and CE standards. Wide operating temperature range from -10 to 60°C or -40 to 75°C. Protected case allows the switches to easily integrate into industrial networks and provide the most optimal solutions for industrial Ethernet applications.

Fig.3.7 Compact industrial Ethernet switch Moxa EDS-208A

This topology does not require high-speed cable data transmission, so a standard cable provided by the provider itself is suitable – a 100BASE-TX twisted pair with a data transfer rate of 100 Mbit/s over a cable consisting of two twisted pairs of Category 5. Typically, data transmission in each direction is carried out on one twisted pair, providing up to 100 Mbps of total throughput.

3.3. Development of system database architecture

When developing the architecture of the database system, it is necessary to do a thorough analysis of the equipment, both military and technical, of the ships of the naval forces. Since not only domestic ships enter the waters of the Black Sea, but also ships of other countries. After the analysis, it is already possible to think over the creation of the database structure. In this case, it is necessary to take into account the further development of software and its logic of work. It is also needed to pay attention to the professional skills of the person who will further maintain the correct operation of the database. Since this is a board of a ship on which a sailor a priori does not have deep knowledge in this area, the number of people working and having access to the base is small, then the

structure of the database should not be tricky and twisted. On the contrary, it should be distinguished by the simplicity of its logic and the possibility of easy maneuvering.

3.3.1. Analysis of equipment (military and technical) of ships

As mentioned above, the Black Sea basin receives ships from many countries into its waters.

Ships should be considered such types as frigates, aircraft carriers, cruisers, destroyers, battleships, corvettes, tugs, patrol boats, anti-submarine ships, patrol ships, minesweepers, all kinds of auxiliary ships, hovercraft. Other types are possible, but all of the above are basic.

The naval forces of Ukraine also have submarines in service. These include submarines, nuclear powered, diesel-electric and flying submarines.

By comparing general information about ships of different countries, it can be determined the following characteristics:

- name;
- class;
- hull number (preferably in NATO classification);
- displacement and weights;
- crew;
- performance (speeds and ranges);
- history dates (laid down and launched dates, commissioned and recommissioned dates);
- status of the ship and its builder;
- machinery;
- IFF (state recognition) - radar-based identification system designed for command and control;
- physical and electronic countermeasures – optional;
- combat data systems;
- weapon control systems;
- air group or presence of helicopters;

- image;
- additional information (information that does not fit into the existing categories).

On the basis of open data sources (reference books, Internet resources), it is possible to identify the main objects inherent in ships. Namely: rockets, torpedoes, radars, sonars, guns and mortars. Looking at the equipment of submarines, it may be distinguished the presence of various types of missiles, torpedoes, radars and sonars.

The main characteristics of missiles:

- type;
- name;
- homing type;
- homing range;
- altitude;
- velocity;
- warhead weight.

The main characteristics of torpedoes:

- type;
- name;
- diameter;
- homing type;
- homing range;
- speed;
- warhead weight.

The main characteristics of guns:

- name;
- caliber;
- rate of fire;
- firing range;
- weight of shell.

The main characteristics of mortars:

- name;
- number of tubes;
- aiming range;
- warhead weight.

The main characteristics of radars:

- type;
- name;
- band.

The main characteristics of sonars:

- type;
- name;
- placement;
- search and attack type;
- frequency range.

All the same objects can be located on the submarine, with the exception of mortars and guns. Of course, one common parameter of all objects installed on ships and submarines is their number. Naturally, all information should be stored in three languages: English, Ukrainian and Russian. And of course, different fields are needed. It is also necessary to consider all units of measurement applicable to objects. Basically, these are tons, nautical miles, knots, meters, Mach, kilometers, kilograms, rds/min and also millimeters.

3.3.2. Structural links and logic of the database

To achieve the greatest efficiency when executing queries, it must be used standard SQL statements, both individually and in combination.

Logical operators, comparison operators, data descriptions, data manipulations, and access control operators have been applied when writing queries.

Data description operators allow:

- create new databases, tables, users;
- delete databases, tables, users;

- edit the structure of tables and fields.

Data manipulation operators allow:

- add new records to tables;
- delete records from tables;
- view data from tables;
- update, that is, change already existing data.

Operators of setting access rights allow to grant users with certain rights, or to deny them.

Comparison operators are used to compare data of the same numeric or boolean type. For expressions of a text or signed type, such operators do not work.

The truth of a certain condition is checked by logical operators. Information will be displayed only if the conditions in the query satisfy the predicate.

The database structure consists of tables and relationships between them. Since this database is relational, it is a collection of interconnected tables (APPENDIX A), each of which contains information about objects of a certain type.

Records, that is, rows of a table, have the same structure – they consist of fields that store the attributes of an object. Each field, i.e., column, describes only one characteristic of the object and has a strictly defined data type. All records have the same fields, only they display different information properties of the object.

Each database has a field that uniquely identifies each row in the table and this is the primary key. It must be unique and uniquely identify a single record. To search for a specific and separate record, it can be used this key value. However, the main purpose of such keys is to organize the data in the database. This field is often called "ID" (fig. 3.8). The unique number is generated by automatic incrementing. This happens when adding a new record to the table. When a field is deleted, the identifier will be irretrievably lost, and the next added record will have a new, next number.

Browse | Structure | SQL | Search | Insert | Export | Import | Operations

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	surface_id	int(11)			No	None	AUTO_INCREMENT
2	Name	varchar(255)	utf8_general_ci		Yes	NULL	
3	Name_ukr	varchar(255)	utf8_general_ci		Yes	NULL	

Fig.3.8 Screenshot of primary key “Surface_id” and auto-increment property

There are three types of relationships between database tables:

- “one-to-many”;
- “one-to-one”;
- “many-to-many”.

The main entities of the database are “ship” and “submarine” and they, in turn, can have such objects as “missiles”, “guns”, “mortars”, “radars”, “sonars” and “torpedoes”. The relationship between them is implemented as “many-to-many” using the linking index tables in which they are linked (figure 3.9).

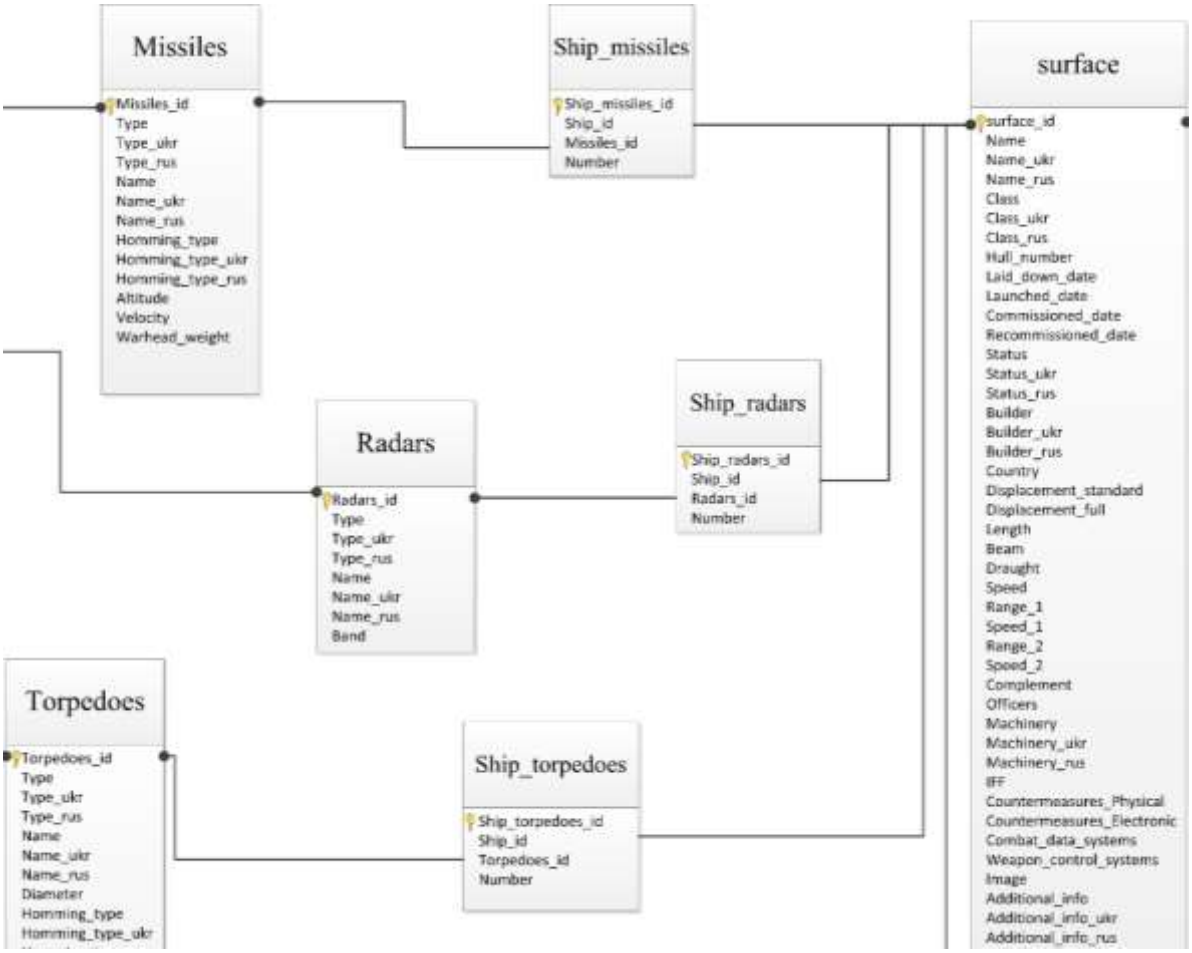
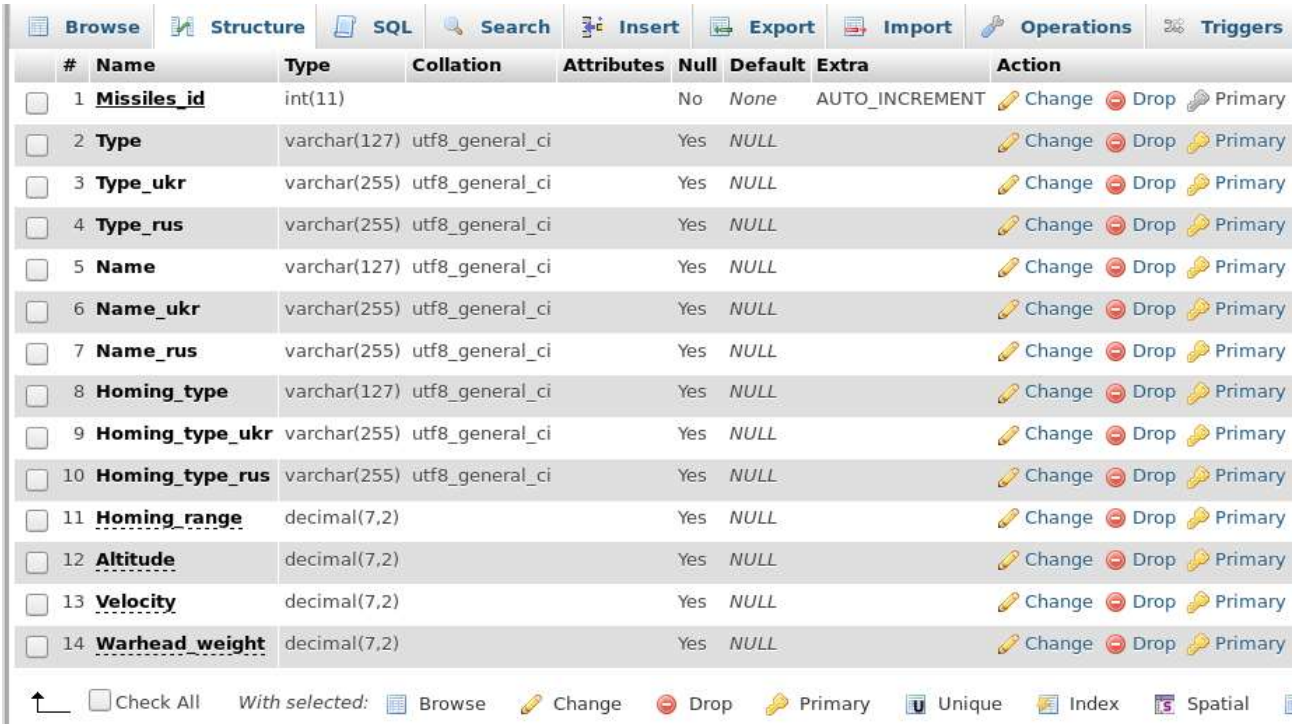


Fig.3.9 Example of a many-to-many relationship using link index tables

The “Missiles” table stores data about missiles, the “surface” table stores data about the ship, and the “Ship_missiles” table links these tables, storing the values of the primary keys of both tables: “Ship_missiles_id” stores the sequence number of adding records, “Ship_id” stores the primary key of the “surface” table, “Missiles_id” stores the primary key of the “Missiles” table, and “Number” contains the quantity data.

Relational database tables meet the requirements of normalization of relations, which allows eliminating duplication, ensures the consistency of those stored in the database, and reduces the labor costs of maintaining the database.

As already mentioned, a table row contains data about one object, and the table columns describe various characteristics of these objects - attributes. So, the figure 3.10 shows the structure of the “Missiles” table.



#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	Missiles_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary
2	Type	varchar(127)	utf8_general_ci		Yes	NULL		Change Drop Primary
3	Type_ukr	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop Primary
4	Type_rus	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop Primary
5	Name	varchar(127)	utf8_general_ci		Yes	NULL		Change Drop Primary
6	Name_ukr	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop Primary
7	Name_rus	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop Primary
8	Homing_type	varchar(127)	utf8_general_ci		Yes	NULL		Change Drop Primary
9	Homing_type_ukr	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop Primary
10	Homing_type_rus	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop Primary
11	Homing_range	decimal(7,2)			Yes	NULL		Change Drop Primary
12	Altitude	decimal(7,2)			Yes	NULL		Change Drop Primary
13	Velocity	decimal(7,2)			Yes	NULL		Change Drop Primary
14	Warhead_weight	decimal(7,2)			Yes	NULL		Change Drop Primary

Fig.3.10 Screenshot of “Missiles” table structure

The main attributes of the "Missiles" table:

- “Name” – contains the name of the lines;
- “Type” – contains the type of the string;
- “Null” – supports a boolean value about the possibility of writing "null" value in the field;
- “Default” – contains the default value, if there is no direct recording;

- “Extra” – contains the "AUTO_INCREMENT" property, which allows to assign an automatic increment to the field.

Data types that were used to store the data:

- string;
- date;
- integer and fractional (floating point) numbers.

The command for creating a table of storing main characteristics of missiles in the SQL language is shown below:

```
CREATE TABLE IF NOT EXISTS `Missiles` (
  `Missiles_id` int(11) NOT NULL AUTO_INCREMENT,
  `Type` varchar(127) DEFAULT NULL,
  `Type_ukr` varchar(255) DEFAULT NULL,
  `Type_rus` varchar(255) DEFAULT NULL,
  `Name` varchar(127) DEFAULT NULL,
  `Name_ukr` varchar(255) DEFAULT NULL,
  `Name_rus` varchar(255) DEFAULT NULL,
  `Homing_type` varchar(127) DEFAULT NULL,
  `Homing_type_ukr` varchar(255) DEFAULT NULL,
  `Homing_type_rus` varchar(255) DEFAULT NULL,
  `Homing_range` decimal(7,2) DEFAULT NULL COMMENT '(km)',
  `Altitude` decimal(7,2) DEFAULT NULL COMMENT '(metres)',
  `Velocity` decimal(7,2) DEFAULT NULL COMMENT '(Mach)',
  `Warhead_weight` decimal(7,2) DEFAULT NULL COMMENT '(kg)',
  PRIMARY KEY (`Missiles_id`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=38 ;
```

It is worth using the varchar (size) type, which stores no more than 2^{31}-1bytes for storing string data. The date is saved in the “YYYY-MM-DD” format. To store integer numeric data, the int type (size) was used, and to store fractional numeric data, the decimal type (precision, scale). The “precision” parameter is the maximum number of digits that a number can store and must be in the range from 1 to 38. Parameter “scale” indicates the maximum number of digits a decimal point can contain. The default scale value is 0. All values are in bytes.

The structures of the “Guns”, “Mortars”, “Radars”, “Sonars” and “Torpedoes” tables are shown in Figures 3.11-3.15.

Fig.3.11 Screenshot of structure of the “Guns” table

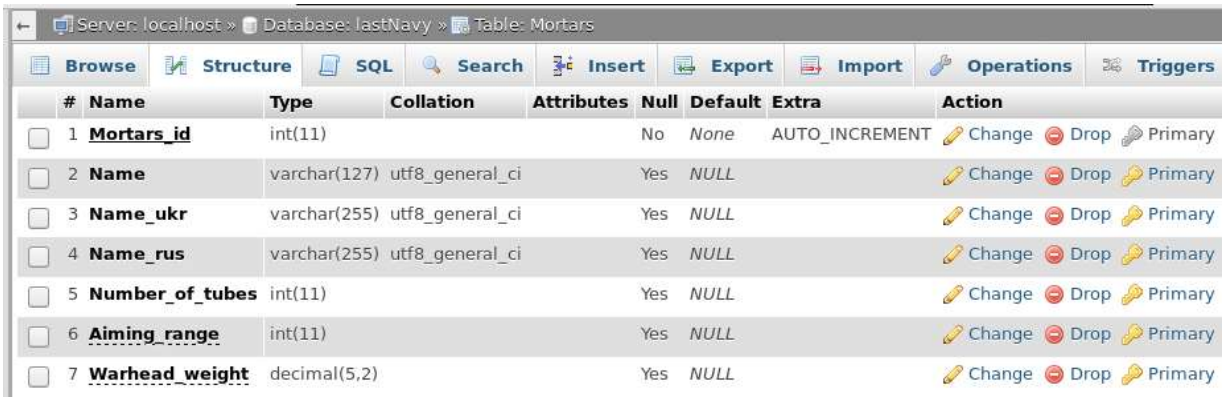
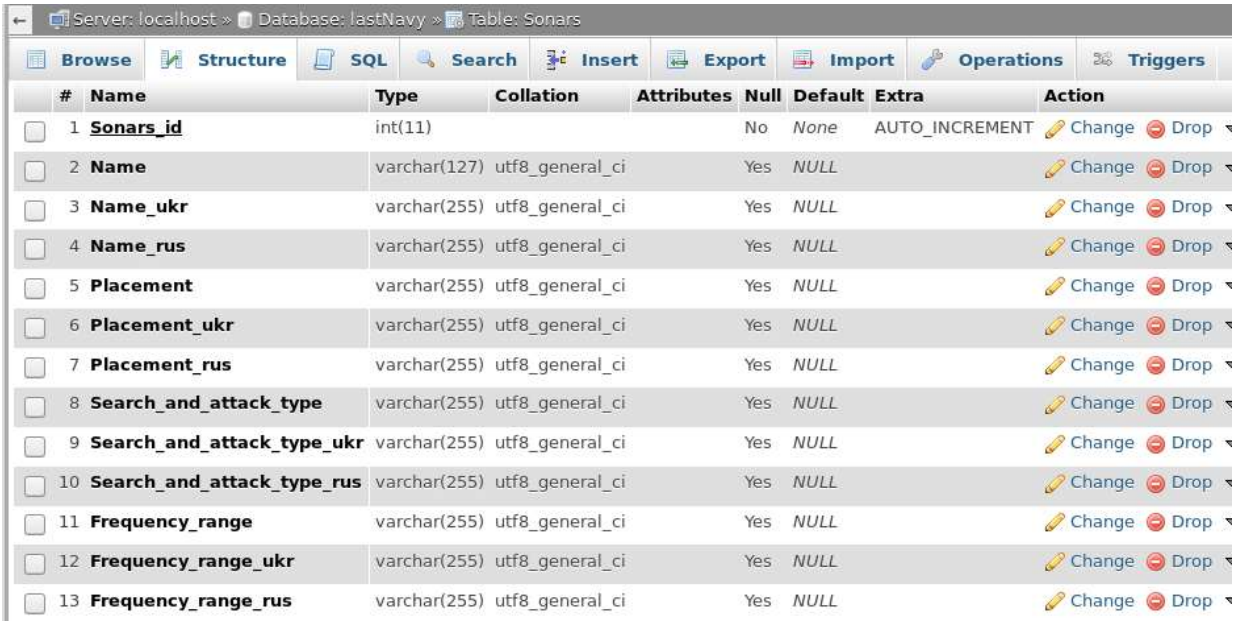


Fig.3.12 Screenshot of structure of the “Mortars” table

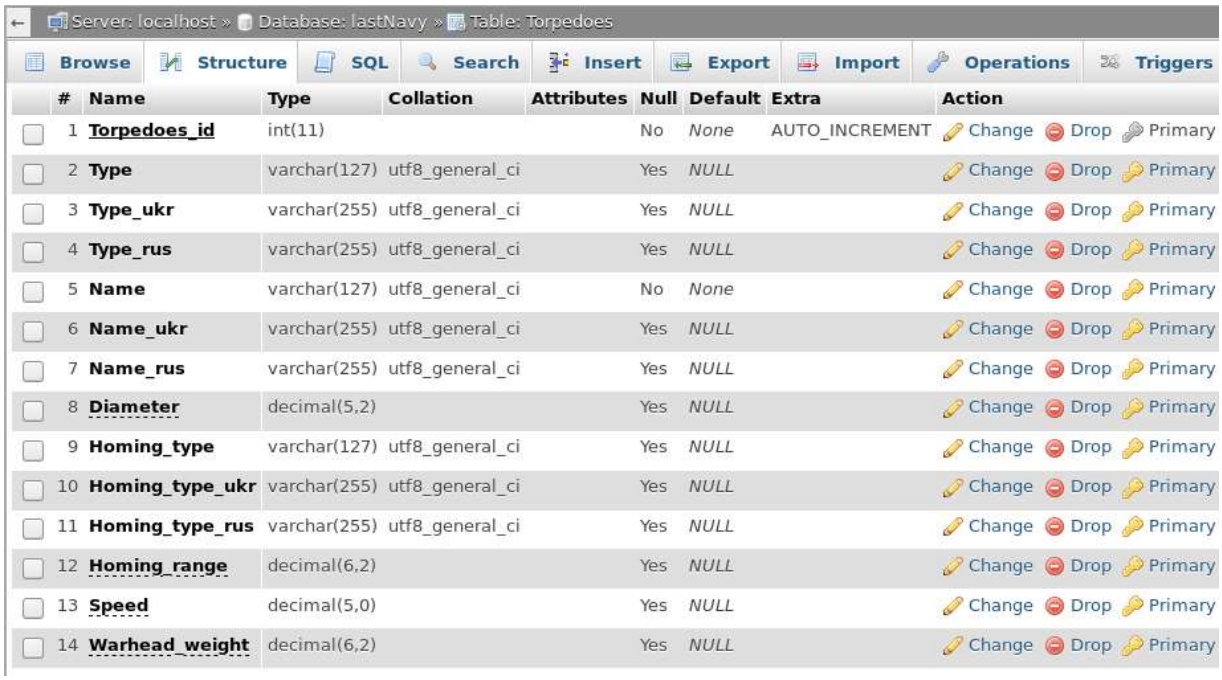
Fig.3.13 Screenshot of structure of the “Radars” table


Server: localhost » Database: lastNavy » Table: Sonars

Browse | Structure | SQL | Search | Insert | Export | Import | Operations | Triggers

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	Sonars_id	int(11)			No	None	AUTO_INCREMENT	Change Drop
2	Name	varchar(127)	utf8_general_ci		Yes	NULL		Change Drop
3	Name_ukr	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop
4	Name_rus	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop
5	Placement	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop
6	Placement_ukr	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop
7	Placement_rus	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop
8	Search_and_attack_type	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop
9	Search_and_attack_type_ukr	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop
10	Search_and_attack_type_rus	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop
11	Frequency_range	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop
12	Frequency_range_ukr	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop
13	Frequency_range_rus	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop

Fig.3.14 Screenshot of structure of the “Sonars” table


Server: localhost » Database: lastNavy » Table: Torpedoes

Browse | Structure | SQL | Search | Insert | Export | Import | Operations | Triggers

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	Torpedoes_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary
2	Type	varchar(127)	utf8_general_ci		Yes	NULL		Change Drop Primary
3	Type_ukr	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop Primary
4	Type_rus	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop Primary
5	Name	varchar(127)	utf8_general_ci		No	None		Change Drop Primary
6	Name_ukr	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop Primary
7	Name_rus	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop Primary
8	Diameter	decimal(5,2)			Yes	NULL		Change Drop Primary
9	Homing_type	varchar(127)	utf8_general_ci		Yes	NULL		Change Drop Primary
10	Homing_type_ukr	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop Primary
11	Homing_type_rus	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop Primary
12	Homing_range	decimal(6,2)			Yes	NULL		Change Drop Primary
13	Speed	decimal(5,0)			Yes	NULL		Change Drop Primary
14	Warhead_weight	decimal(6,2)			Yes	NULL		Change Drop Primary

Fig.3.15 Screenshot of structure of the “Torpedoes” table

All the above screenshots were made in the PhpMyAdmin DBMS web interface.

The relationship between “subsurface” table and its objects is implemented as “many-to-many” using linking index tables in which they are linked (figure 3.16).

The “Missiles” table stores missile data, the “subsurface” table stores the submarine data, and the “Submarine_missiles” table links these tables, storing the primary key values of both tables: “Submarine_missiles_id” stores the sequence number of adding records,

“Submarine_id” stores the primary key of the subsurface table, “Missiles_id” stores the primary key of the “Missiles” table, and “Number” contains the quantity data.

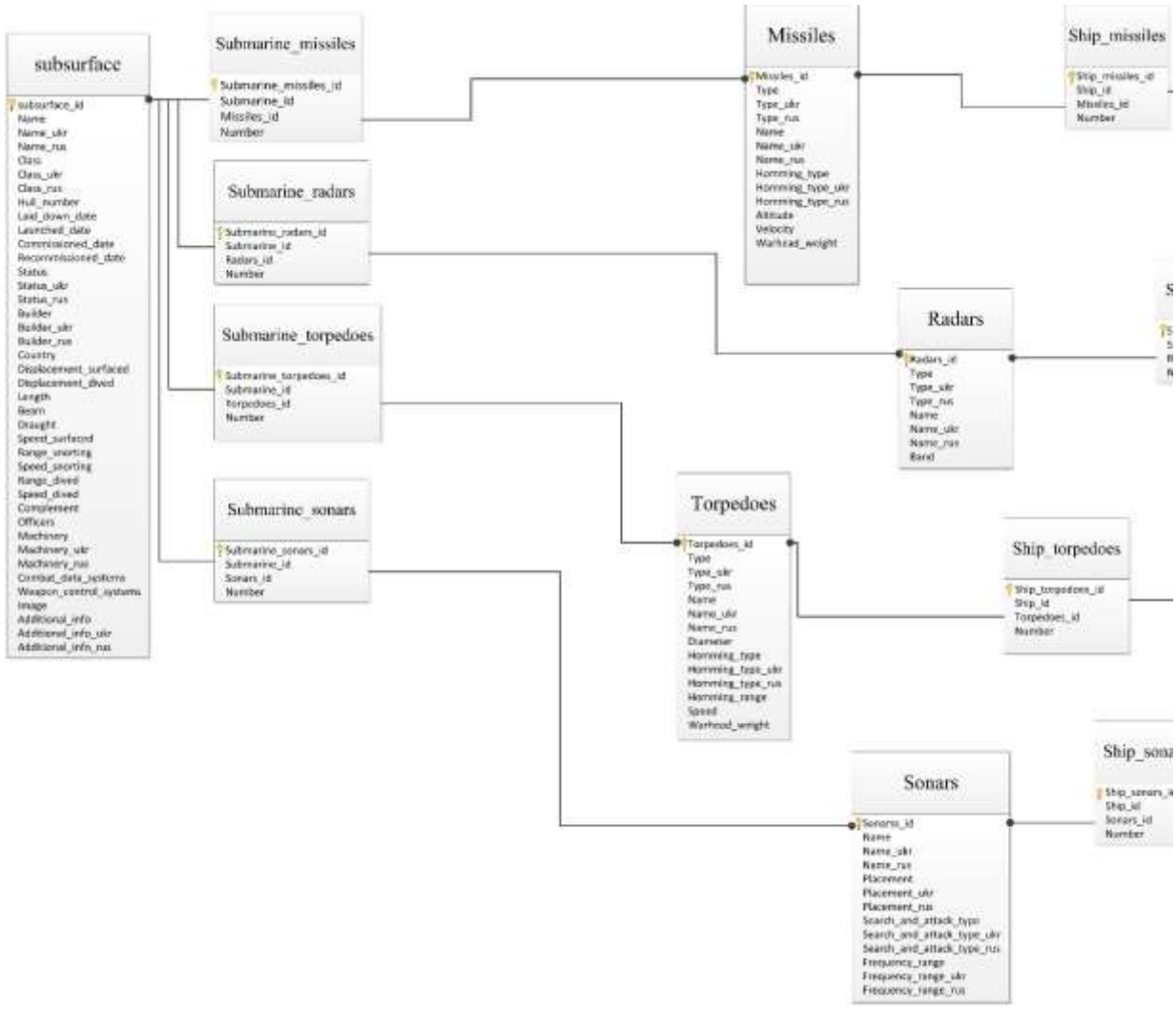


Fig.3.16 Many-to-many relationships using link index tables

This database is directly linked to the client application for the administration of the surface ships and submarines database.

3.4. Development of user interface

Before developing the application, it was decided to use the Java programming language. This language has all necessary technical features to create a desktop interface that would be able to communicate with a database.

Such an application would require for the client to use an installed runtime environment for Java (JRE).

The creation of a desktop interface is determined by the system, with which it will interact. At this moment, this system (a complex for managing the flagship of the fleet) is under development, which is why it is necessary to ensure competent interaction with it. The Swing library was chosen to work with the graphical shell. This is a Java library for creating a graphical interface that will allow to create and program graphical components (input fields, buttons, tables, sliders, etc.).

When developing a client application for administration of a network database, it is worth considering the needs of all users, namely:

- a user with administrator rights, usually a technical system support operator. Such a user will be able to view all the data, edit them, and of course has the main duties of an administrator (create accounts for another user, manage their rights, delete users);
- user with view-only rights – such a user can view data, make a selection according to various criteria, and also save information in file form;
- user with rights to edit data. Since editing is a deeper range of tasks, accordingly, such a user can view the data.

The interface should be simple and intuitive. Then the decision should be made that the first window will be “Log in”, where the user will enter his username and password. After that, the user's role will be determined, and the corresponding dialog box opens. The login window is shown in the figure 3.17:

This interface, as well as all existing windows, has the ability to change the language, which is carried out using the drop-down list. When initially displayed, the interface displays information in English. The ability to change the language is presented in three languages: English, Russian and Ukrainian. The search result for an object by parameters, in the search window, will be presented in the language that was selected earlier.

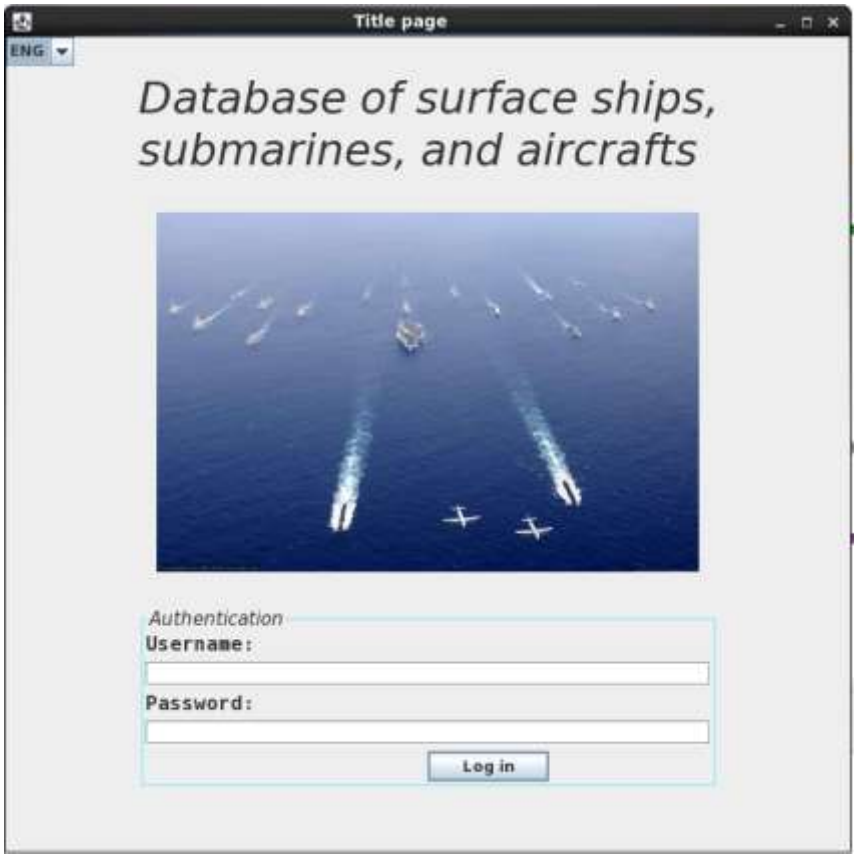


Fig.3.17 Screenshot of GUI at application startup

An example of code for connecting to a database in Java language is provided in APPENDIX B.

To avoid endless login attempts (which would litter a possible data log), the user must be given three login attempts (as shown in the figure 3.18), then deny access and close the dialog box. It is also possible to exclude this user from the database, or change the data of his account by the administrator.

Fig.3.18 Screenshot of user authentication error warning

Upon successful authentication of a user with administrator rights (pressing the “Login” button), authorization occurs, that is, the access rights are checked and the administrator's dialog box is shown, which is shown in the figure 3.19:

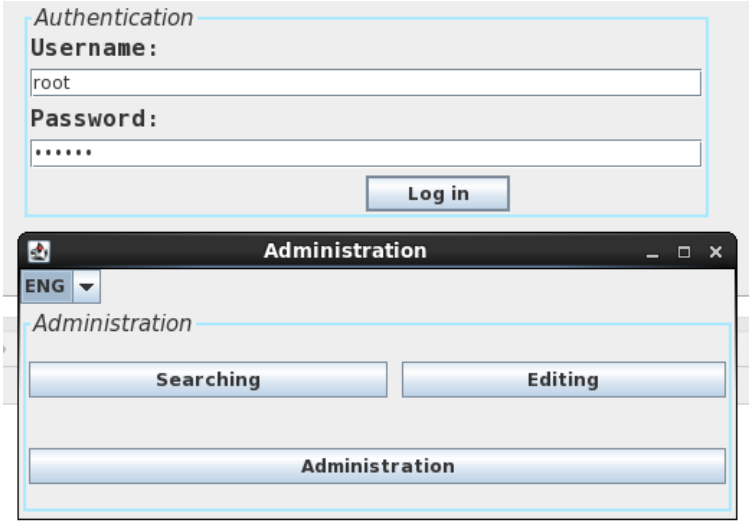


Fig.3.19 Screenshot of the administration window

When a user with access rights for viewing only passes authorization, he sees a dialog box (figure 3.20), in which he can select the search criteria he needs, by checking the checkboxes, enter the required parameters and then make a search.

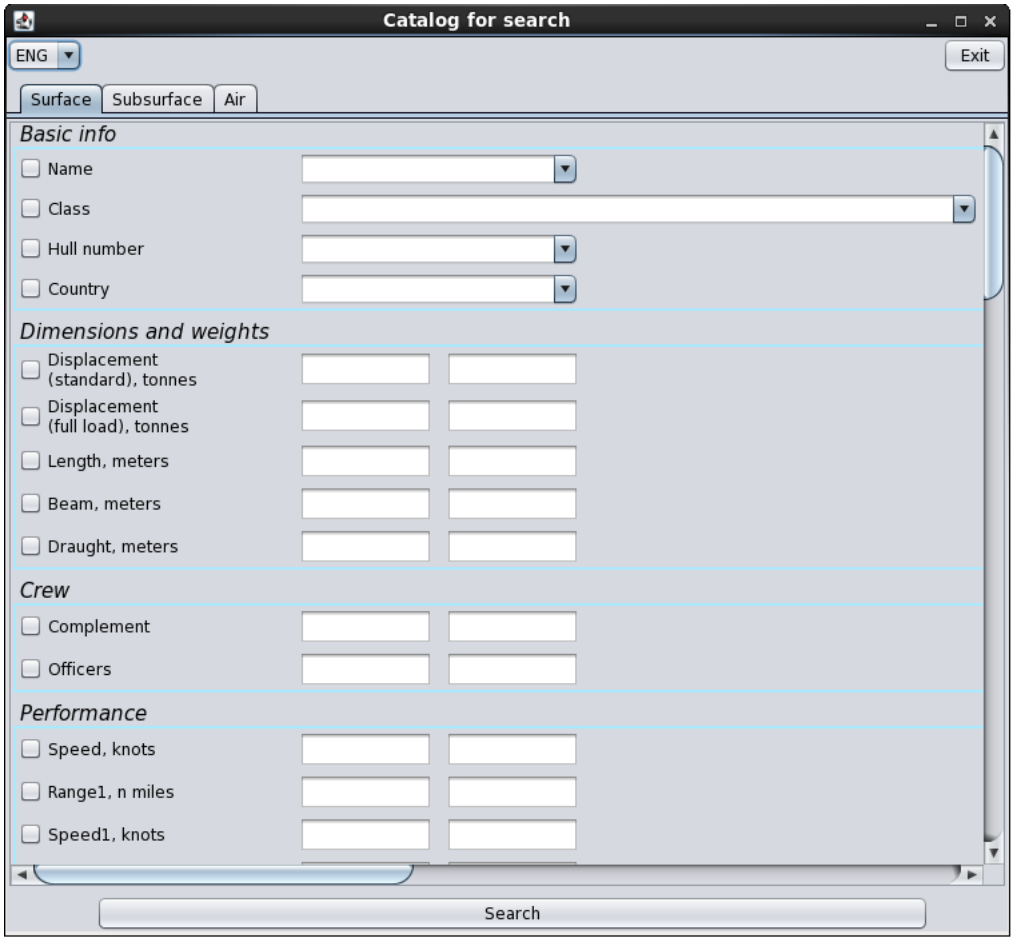


Fig.3.20 Screenshot of the searching window

If the data is entered incorrectly, an error message will be displayed when pressing the "Search" button. Otherwise, the request will be executed, and a new window will be displayed with the search result (figure 3.21).

As a result of the search, the data on the ship/submarine and its object units will be displayed in string and tabular form, respectively, as well as the image of the ships/submarines, if it exists on the current computer.

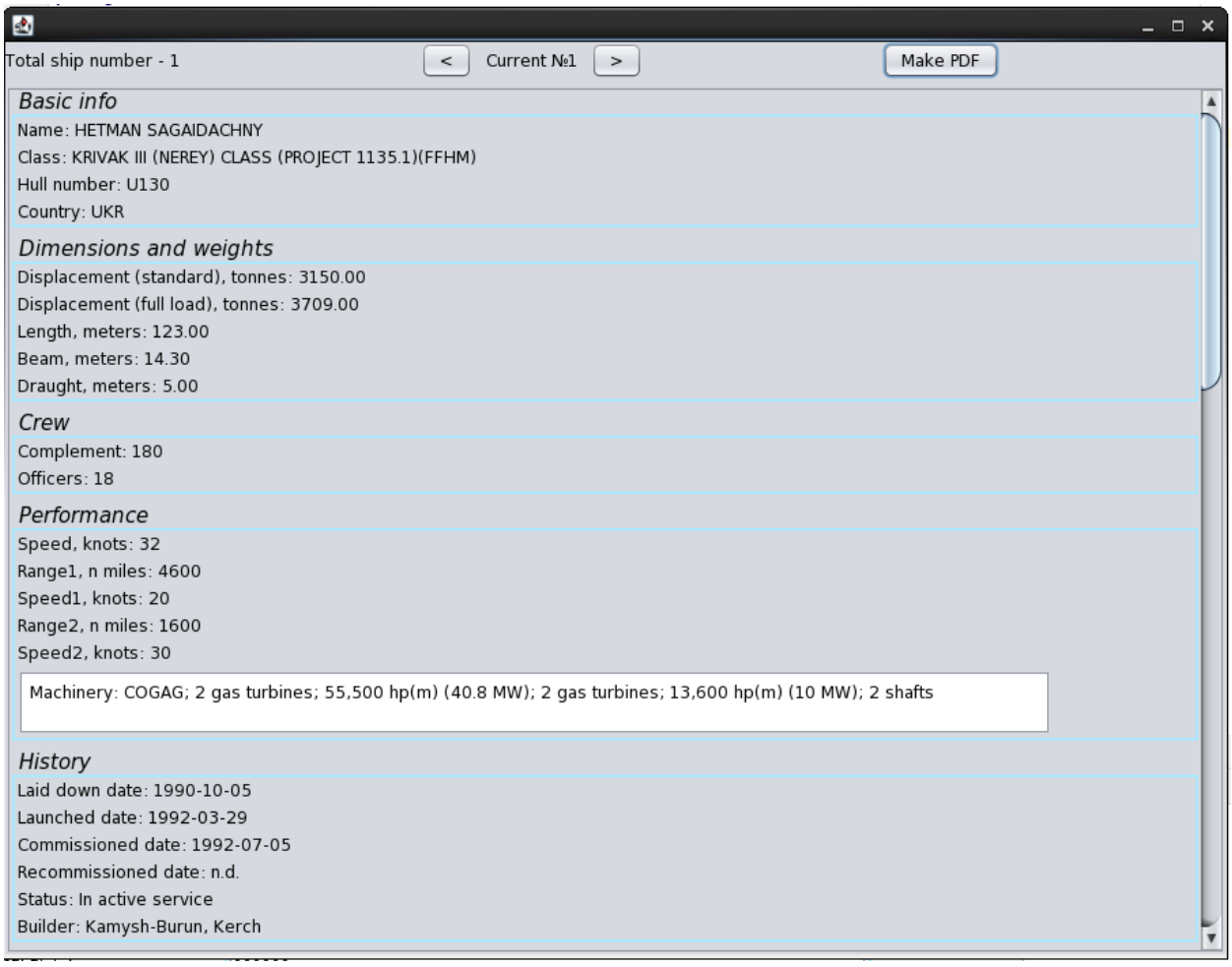


Fig.3.21 Screenshot of the window with a search result for the query: "Name – Getman Sagaidachny"

If several ships/submarines were found in the search result (the inscription is displayed in the upper left corner), they can be viewed by pressing the buttons marked “<” and “>” (the current serial number is indicated between these buttons).

On the search result window there is a button “Make PDF”. This button allows to create a document with the extension “.pdf” that stores the search result of the query. If

several objects were found as a result, then when saving, one document will be created. The PDF format was chosen at the request of the customer.

In the case when a user with rights to view and edit data passes the check, it is necessary to maintain an additional dialog box (figure 3.22) where he can choose further actions.

Fig.3.22 Screenshot of the window for selecting actions for a user with rights to view and edit data

The code responsible for displaying the above dialog box is shown in the figure 3.23 below. In this screenshot, it can be also seen the connection of button listeners, which call the classes responsible for them.

Fig.3.23 Screenshot of the code for the window for selecting actions to view and edit data

After clicking the “Editing” button, the user will open a window for editing data (fig. 3.24).

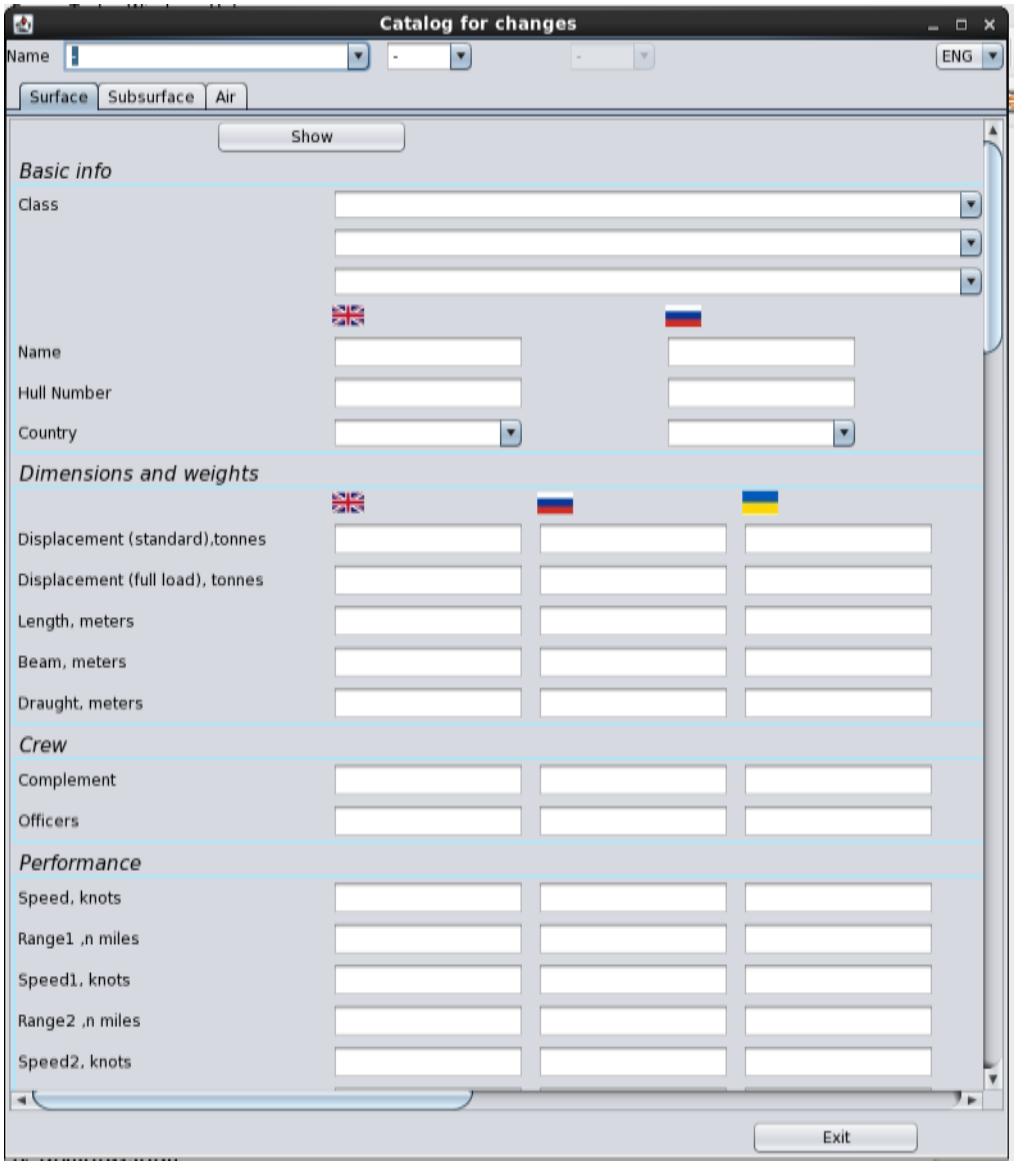


Fig.3.24 Screenshot of the editing window

As seen in the figure, a questionnaire for entering and editing data has been developed. The form also has the ability to change the interface language through the drop-down list. Data entry can be carried out simultaneously both in 3 languages, and in one of them. To define the language of the input field, a flag-icon image of the country representing the current language is presented.

In this application, it is impossible to add objects of ships/submarines without selecting/adding a new object.

To edit a ship/submarine or their objects, it must be first selected the ship/submarine using the “Show” button or add a new ship/submarine. Next, it is needed to select the editable object.

For any action with an existing ship/submarine, it must be selected/shown it in the questionnaire. To do this, select the name of the ship/submarine in the drop-down list located in the upper left corner opposite the inscription “Name”. If there is a likelihood that the names of ships/submarines in different countries coincide, then it is needed to select the country code from the adjacent drop-down list. And then confirm all actions by clicking the “Show” button. As a result of the above actions, information will be displayed in the fields and drop-down lists of the current interface.

Fields “Class”, “Name”, “Hull number” and “Country” cannot be left blank (an error message will be displayed). Otherwise, a certain error message will be displayed (figure 3.25).

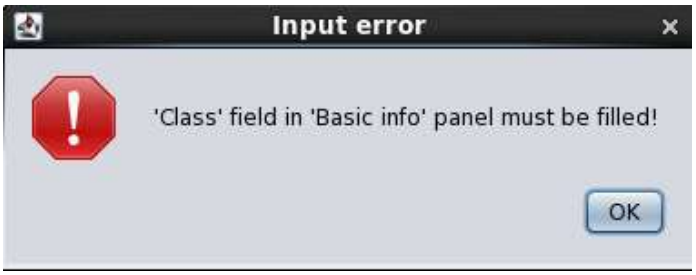


Fig.3.25 Screenshot of error message

When entering these objects of a ship or submarine, it is necessary to indicate their number. If the data is not entered in the “Quantity” field, an error message will be displayed (figure 3.26).

Fig.3.26 Screenshot of the message about an unspecified number of units of this object

On the “Administration” window there are three buttons labeled “Search”, “Edit” and “Administration”. By clicking the “Administration” button, the administrator window opens with two tabs “Users” and “Settings”. On the “Settings” tab there are two fields “Database name” and “Image path”. By default, information about the name of the database and the path to images is stored in the files admin.txt, search.txt and input.txt. It

is assumed that only the administrator or root user has access to such actions. The results of the changes made in the “Settings” tab will be added to the files search.txt and input.txt. To change the data in the admin.txt file, it must be directly made changes in the file itself and save the changes.

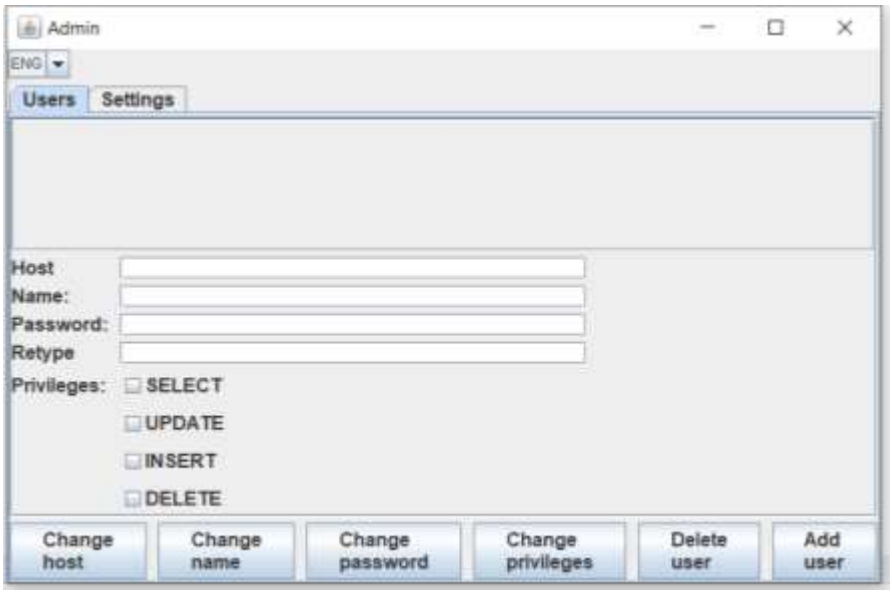


Fig.3.27 Screenshot of the “Users” tab in the admin window

On the “Users” tab (figure 3.27) of the administrator window there is a table that displays all users of the current database, fields for changing user data and buttons for confirming actions. The user table also contains a user named “root”. Such a user cannot be deleted, and it is also impossible to rename or create another user “root”. These restrictions were introduced on purpose to avoid mistakes. A user with administrator rights can change the user's host (make it possible to log into the system from any workplace), his login (name), password, add and delete a user, and change user privileges.

The user table remains empty for now, because at this stage this module has no connection to the database.

Conclusions on the Third Part

This application is being adjusted in accordance with customer requirements. In this regard, there may be some inaccuracies and improvements.

However, it can be concluded that a network database administration complex has been developed that adequately copes with its tasks.

As a result, a network topology was created to simulate the equipment on board the ship, since, unfortunately, there is no way to develop a network topology for a real space of a ship. Then the selection of the best equipment for efficient work was made. This was one of the most important tasks, since the equipment had to be highly efficient and industrial, taking into account the specifics of the premises.

The network was configured using the Cisco Packet Tracer program. On the Internet, there are many video lessons, manuals and tutorials for performing a variety of tasks and actions with specific equipment. This local area network has been built in accordance with all structured cabling standards.

Cisco Packet Tracer is a free application that does not require any costs, but its functionality does not fully reflect the essence and principle of all settings of real equipment, and the availability of all relevant tools.

A thorough analysis of the subject area was also carried out and, on its basis, a scheme for the interaction of tables was created. It has successfully combined the intelligent storage of ships and submarines data.

CONCLUSIONS

During the implementation of the master's degree project, many works were carried out, both theoretical and research, and practical.

Theoretical researches were carried out in the direction of the responsibilities of the database administrator, finding the best database administration algorithm.

The type of used database was chosen relational, since the objects of this subject area have a relationship with each other. The tables of such a database store data about interconnected objects.

Also, during the research, it was determined that each database administrator in his own way creates new users, grants rights or prohibits any actions to users, creates tables and links (shows their relationship) them using keys. And the order of all these actions is not strictly regulated. That is, it can be concluded that the best algorithm for administration of a network database is to create a user with system rights, create several users with the ability to view and edit data, create tables and link them, then fill them in, and finally, create additional users to speed up filling tables.

Practical work was carried out by both the SQL developer and the Java developer. Also, practical work on modeling the network topology was carried out using Cisco Packet Tracer.

During the creation of the network topology, difficulties arose with the schematic installation of the necessary equipment - it was not there. The reason for this is the need for specialized industrial computers. In general, this aspect is not so important. The most important task in building a network topology was to show the location of workstations, their way of connecting to the network, and the placement of server racks.

During the creation of the database schema and its implementation, there were some disagreements regarding the creation of unified tables for objects of ships and submarines. The fact is that such objects have a different configuration and can be installed both on ships and submarines. This means that somewhere information will not be fully displayed, although it will be stored in the database.

Java gave rise to difficulties with the implementation of the GUI. This is due to the behavior of the Swing graphic library in the chosen development environment - IntelliJ IDEA. When choosing another environment, for example, Eclipse, there would be no such nuances, but the chosen environment provided a flexible development interface, which significantly reduced development time.

The task was fully accomplished, despite the difficulties encountered.

The need for the presence of a well-built local network makes the work of the entire complex effective: fast transfer of data and messages between team members, interchange of urgent and emergency information. Examples of such activities are the exchange of instructions between team members, reporting, writing programming instructions, sharing files and data.

It is difficult to achieve both ease of use and ensuring of data integrity. Indeed, almost directly, data integrity depends on proper control and management of the system. And this affects the complication of the system as a whole.

As ideas for improving the project, it can be taken the encryption of the data entered into the database, and decryption when it is displayed. Such mean of protection will be relevant when intercepting data during a convergence with other ships.

At the request of the customer, it is worth taking to the attention the changing of some fields in the client application.

It's also a good idea to think over own logging mechanism, and not the one already in the system. Such an addition will allow keeping a log of logs not of the entire system as a whole, but of a specific module for interacting with the database.

REFERENCES

1. What is structured query language: [Electronic resource] – 2000. – Access mode to the resource: https://www.oracle.com/database/what-is-database/.
2. Basic principles of building databases: [Electronic resource] – 2014. – Access mode to the resource: https://sites.google.com/site/gosyvmkss12/bazy-dannyh/1-osnovnye-principy-postroenia-baz-dannyh-problemy-hranenia-bolsih-obemov-informacii.
3. Basic principles of building databases: [Electronic resource] – 2014. – Access mode to the resource: https://sites.google.com/site/gosyvmkss12/bazy-dannyh/1-osnovnye-principy-postroenia-baz-dannyh-problemy-hranenia-bolsih-obemov-informacii?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F.
4. Network model in DBMS: [Electronic resource] – 2003. – Access mode to the resource: http://petrelluzzi.com.br/masquerade-masks-dnr/network-model-in-dbms-10d02b
5. Databases: [Electronic resource] – 2005. – Access mode to the resource: https://www.stat.auckland.ac.nz/~paul/ItDT/HTML/node42.html.
6. Keys in DBMS: [Electronic resource] – 2018. – Access mode to the resource: https://www.cseworldonline.com/dbms-tutorial/keys-in-dbms.php.
7. Functionality of the DBMS: [Electronic resource] – 2018. – Access mode to the resource: http://www.bseu.by/it/tohod/sdo4.htm.
8. Олифер В. Компьютерные сети. Принципы, технологии, протоколы / В. Олифер, Н. Олифер. – Питер, 2013. – 944 с.
9. Login creation: [Electronic resource] – 2020. – Access mode to the resource: https://github.com/MicrosoftDocs/sql-docs/blob/live/docs/relational-databases/security/authentication-access/create-a-login.md.
10. Login creation: [Electronic resource] – 2000. – Access mode to the resource: http://www.its.kpi.ua/subjects/21/Documents/Адміністрування.pdf.

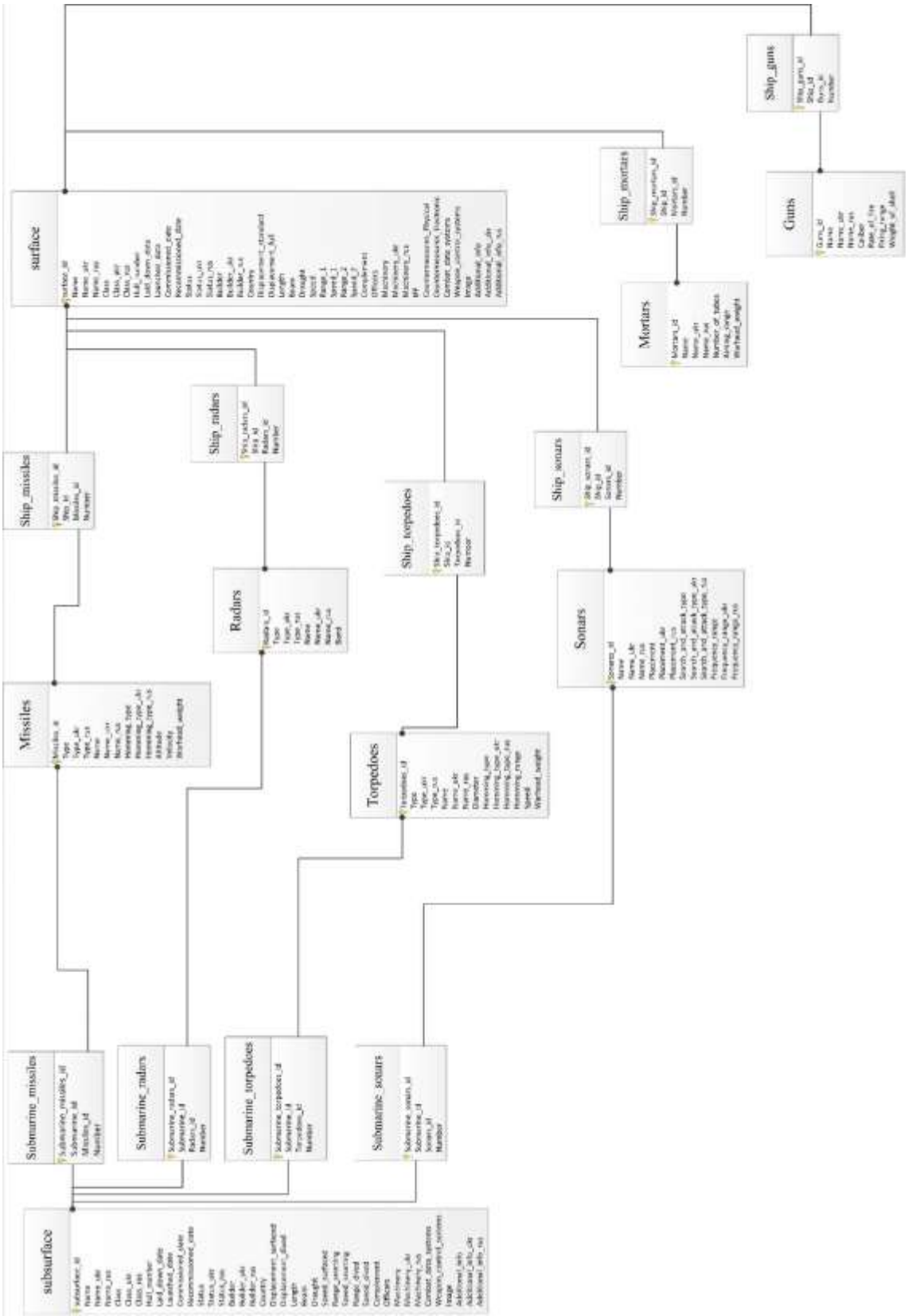
11. Database user management: [Electronic resource] – 2019. – Access mode to the resource: https://crabo.ru/en/useful-programs/dobavlenie-informacionnoi-bazy-iz-okna-zapuska-1s-predpriyatie-predostavlenie.html.

12. Stored procedures of SQL system: [Electronic resource] – 2020. – Access mode to the resource: https://titanwolf.org/Network/Articles/Article?AID=99c709a4-89a7-4a11-9e94-6fb2faff9f44#gsc.tab=0.

13. Глушаков С. В. Базы данных. Учебный курс / С. В. Глушаков, Д. В. Ломотьков. – К.: Абрис, 2000. – 102 с.

APPENDIX A

Database structure



Code for connection to database

```
package base.connect;
import com.mysql.jdbc.exceptions.jdbc4.MySQLNonTransientConnectionException;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;
import java.util.Scanner;
import java.util.Timer;

public class MysqlConnect {
    // init CONNECTION object
    private Connection connection;
    // init properties object
    private Properties properties;

    private static String url= "jdbc:mysql://", user, psd, databaseName, filePath;
    //private static String host= "localhost";//"192.168.1.200"; ////"82.207.124.148";

    private static String DATABASE_URL;//
"jdbc:mysql://"+host+":3306/"+db+"?zeroDateTimeBehavior=convertToNull&autoReconnect=true&ch
aracterEncoding=UTF-8&characterSetResults=UTF-8";
    public static String USERNAME;
    public static String PASSWORD;
    public static String ERROR;
    public static String FILEPATH;
    private static final String MAX_POOL = "250"; // set your own limit
    private Timer timer =  new Timer();

    public MysqlConnect() throws IOException {
        readSettings();
        setSettings();
    }

    private void setSettings() {
        DATABASE_URL=url;
        FILEPATH=filePath;
    }

    public static void readSettings() throws IOException {      // method read settings from file
        FileReader fr= new FileReader("admin.txt");
        Scanner scan = new Scanner(fr);
        while (scan.hasNextLine()) {
            databaseName =scan.nextLine();                  //database name
            url=url+ scan.nextLine()+ ":3306/" + databaseName +
"?zeroDateTimeBehavior=convertToNull&autoReconnect=true&characterEncoding=UTF-
```

```
8&characterSetResults=UTF-8";
                                        //database url
            filePath=scan.nextLine();                   //filePath for finding images
        }
        fr.close();
    }

    // create properties
    private Properties getProperties() {            //getting special properties for connection to database
        if (properties == null) {
            properties = new Properties();
            properties.setProperty("user", USERNAME);
            properties.setProperty("password", PASSWORD);
            properties.setProperty("MaxPooledStatements", MAX_POOL);
        }
        return properties;
    }

    public Connection connect() {               //method calls - establishConnection();
        establishConnection();
        timer.schedule(new TimerTask() {
            @Override
            public void run() {
                establishConnection();
                System.out.println(" ");
                System.out.println("------------------> reconnect is occured");
                System.out.println(" ");
            }             }, 0, 1440000); //1440000 ms
        return connection;
    }

    private Connection establishConnection(){           //method that establish connection
        if (connection == null) {
            try {
                connection = DriverManager.getConnection(DATABASE_URL, getProperties());
                ERROR="";
                System.out.println("connect.MysqlConnect| Connection established");
            } catch (MySQLNonTransientConnectionException q){
                ERROR="ERROR";
            } catch ( SQLException e) {
                e.printStackTrace();
            }
        }
        return connection;
    }

    public Connection getConnection() {             //getter
        if (connection == null)
            connect();
        return connection;
    }
}
```