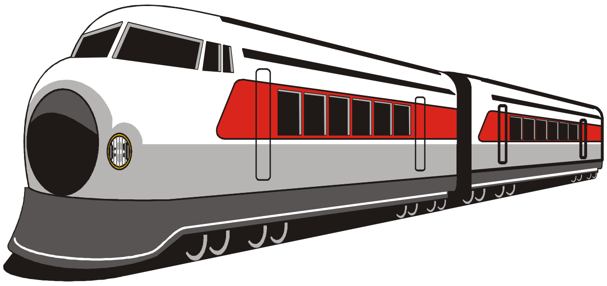
ТОШКЕНТ ТЕМИР ЙЎЛ МУҲАНДИСЛАРИ ИНСТИТУТИ



Кафедра Темир йўл транспортида ахборот тизимлари

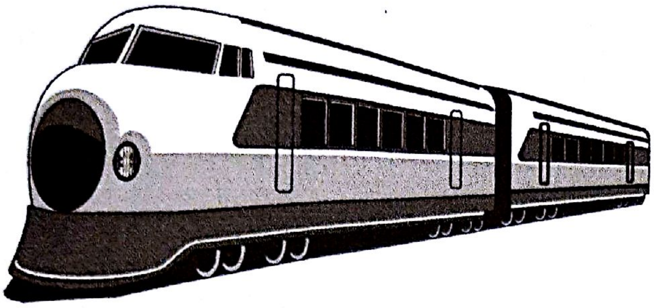
DEVELOPMENT OF INFORMATION SYSTEM OF CONTROLLING EXECUTION ORDERS мавзусидаги

БИТИРУВ МАЛАКАВИЙ ИШИ

Муаллиф: Бобожанова С. Қ.

Тошкент – 2019 й.

TOSHKENT TEMIR YO‘L MUHANDISLARI INSTITUTI



Himoya qilishga
ruhsat berilsin

Kafedra mudiri

«26» 06 2019 й.

Kafedra Temir yo‘l transportida axborot tizimlari

DEVELOPMENT OF INFORMATION SYSTEM OF CONTROLLING EXECUTION ORDERS mavzusida

BITIRUV MALAKAVIY ISH

Muallif Bobojonova S.Q.

Asosiy maslahatchi Rasulmuxamedov M.M.

Maslahatchilar:

t.f.n., dotsent Aliyev M.M.

katta o‘qituvchi Botirova M.

assistent Achilov O.

Taqrizchi:

t.f.d., professor Ravshanov N.

Toshkent – 2019 y.

Toshkent temir yo‘l muhandislari instituti

Oliy o‘quv yurti

Iqtisodiyot fakulteti Temir yo‘l transportida axborot tizimlari kafedrasi

Informatika va axborot texnologiyalari (temir yo‘l transportida) yo‘nalishi AT-26 guruhi

Tasdiqlayman

Kafedra mudiri Rasulmuxamedov M.M.

«10» 01 2019 yil

sana

BITIRUV MALAKAVIY ISHI BO‘YICHA TOPSHIRIQ

Talaba Bobojonova Saida Quvondiq qizi

(familiyasi, ismi, sharifi)

1. Bitiruv ishining mavzusi Development of Information System of Controlling Execution Orders

«19» dekabr 2018 yil 5-sonli kafedra majlisida ma’qullangan va institutning 7 yanvar 2019 yilgi T-4 buyru‘g‘i bilan tasdiqlangan.

2. Bitiruv ishini topshirish muddati 15.06.2019

3.Bitiruv ishini bajarishga doir boshlang‘ich ma’lumotlar Mavzu bo‘yicha adabiyotlar, Farmoyishlarning berilish va bajarilish tartibi, MySQL ma’lumotlar bazasini boshqarish tizimi, PHP dasturlash tili

4. Hisoblash-tushuntirish yozuvlarining tarkibi (ishlab chiqiladigan masalalar ro‘yxati) Adabiyotlar yig‘ish, mavzu bo‘yicha materiallar ajratish, mavzularni bog‘lash, MySQL ma’lumotlar bazasini boshqarish tizimi va PHP dasturlash tilining imkoniyatlaridan foydalanish, Farmoyishlar bajarilishini nazorat qilish axborot tizimini yaratish

5. Chizma ishlar ro‘yxati (chizmalar nomi aniq ko‘rsatiladi)

Poster 1 – Structure of the database Controlling the Execution of Orders ✓

Poster 2 – Home pages of the application ✓

Poster 3 – Desktop of the register ✓

6. Bitiruv bo'yicha maslahatchi (lar)

№	Bo'lim mavzusi	Maslahatchi o'qituvchi F.I.SH.	Imzo, sana	
			Topshiriq berildi	Topshiriq bajarildi
1.	Introduction	Aliyev M.M.	21.01.2019	
2.	Description of the requirements of the information system	Aliyev M.M.	18.02.2019	
3.	Description of main technologies, tools and languages	Aliyev M.M.	18.03.2019	
4.	Development of information system	Aliyev M.M.	15.04.2019	
5.	Labor safety	Botirova.M.	16.05.2019	
6.	Conclusion	Aliyev M.M.	30.05.2019	

7. Bitiruv ishini bajarish rejasi

№	Bitiruv ishi bosqichlarining nomi	Bajarish muddati (sana)	Tekshiruvdan o'tganlik belgisi
1.	Introduction	15.02.2019	bajarildi
2.	Description of the requirements of the information system	14.03.2019	bajarildi
3.	Description of main technologies, tools and languages	10.04.2019	bajarildi
4.	Development of information system	26.04.2019	bajarildi
5.	Labor safety	29.05.2019	bajarildi
6.	Conclusion	06.06.2019	bajarildi

Bitiruv ishi rahbari **Aliyev M.M.**
(F.I.SH) (imzo)

Topshiriqni bajarishga oldim **Bobojonova S.Q.**
(F.I.SH) (imzo)

Topshiriq berilgan sana «14» 01 2019 yil

ANNOTATSIYA

Bitiruv malakaviy ishi muassasalarda farmoyishlar bajarilishini nazorat qilishning axborot tizmini yaratish masalasiga bag‘ishlangan. Axborot tizimi MySQL ma’lumotlar bazasini boshqarish tizimi va PHP dasturlash tilining imkoniyatlaridan foydalangan holda tuzilgan.

Ish 60 varaqdan iborat va unda 12 ta rasmdan foydalanilgan. Bibliografiyasida 8 ta adabiyot ko‘rsatilgan.

АННОТАЦИЯ

Выпускная квалификационная работа посвящена разработке информационной системы контроля выполнения распоряжений. Информационная система создана с использованием особых возможностей системы управления базами данных MySQL и языка программирование PHP.

Работа выполнена на 60 страницах, имеет 12 рисунков. В библиографии приведено 8 перечней.

ANNOTATION

The final qualifying paper is devoted to the development of information system for controlling the execution of orders. Information system developed using the special features of MySQL Database Management System and PHP programming language.

The paper consists of 60 pages, it has 12 drawings. In bibliographies have 8 lists.

Тошкент темир йўл муҳандислари институти

Факултет Иктисодиёт Кафедра ТЙТда ахборот тизимлари
Таълим йуналиши: 5330200 «Информатика ва ахборот тизимлари (ТЙТ)»

«Тасдиқлайман»
Каф. мудири
«10» январ 2019 йил.

Талаба Бобожонова Саида Кувондик кизи га
(фамилияси, исми, шарифи)
битирув малакавий ишини «Ҳаёт фаолияти хавфсизлиги» бўлимини бажариш бўйича

ТОПШИРИК

1.Мавзу Development of Information System of Controlling execution orders

кафедра кенгашининг 2018й. «19» декабр №5 сонли баённомаси қарори билан тасдиқланган.

2.Талабанинг тугалланган бўлимни топшириш муддати

3.Битирув малакавий ишини «Ҳаёт фаолияти хавфсизлиги» бўлимини бажаришга оид дастлабки маълумотлар Computer health and safety at work; workstation health and safety: desks, chairs and posture;

4.Ҳисоб–тушунтириш қисмининг мазмуни (ишлаб чиқиладиган саволлар рўйхати)

—

5.График материаллар рўйхати (аниқ номланган зарурий чизмалар)

—

Топшириқ олди Бобожонова С. К.
(имзо, сана, талабани ф.и.ш.)

Топширик берди Ботирова М.М.
(сана, рахбар имзоси, лавозими, ф.и.ш.)

Toshkent Temir yo‘l muhandislari instituti, Iqtisodiyot fakultetining 5330200 – “Informatika va axborot texnologiyalari” ta’lim yo‘nalishi AT-26 guruhi talabasi Bobojonova Saida Quvondiq qizining “Development of Information System of Controlling Execution Orders” mavzusidagi bitiruv malakaviy ishiga ilmiy rahbar

FIKRI

Ishning dolzarbligi, yangiligi: Farmoyishlar bajarilishini nazorat qilish axborot tizimini yaratish ko‘p parametrli vazifa bo‘lib, istalgan muassasada farmoyishlar berilishini elektron ko’rinishga o’tkazish ularning ijrosini nazorat qilishda qulaylik tug‘dirishda muhim bosqichlardan biri hisoblanadi. Yartilgan dasturiy ta’minot mahsuloti berilgan farmoyishlar va ularning ijrosi ta’minlanishini nazorat qilishda muassasa rahbarlari va xodimlari o‘rtasidagi aloqani ta’minlash uchun kerak.

Ishning ijobiy tomonlari: Muassasa devonxonasiga kelib tushgan hujjatlarni registratsiyadan o‘tkazishda devonxona xodimiga tizimga soda va tushunarli interfeysli oyna orqali kiritish imkoniyati yaratilgan, tizimga kiritilgan hujjatlar rahbar tomonidan ko‘rib chiqilib, ijrochilarga farmoyishlar beriladi va farmoyishlarning ijrosi ta’minlanganini ham rahbar tasdiqlashi kerak bo’ladi. Farmoyishga asos bo’luvchi hujjatlar elektron shaklda tizimga biriktiriladi, bu esa nazorat qilishni yengilashtiradi.

Ishning amaliy ahamiyati va tadbiq etishga takliflar: bitiruv malakaviy ishning amaliy ahamiyati shundan iboratki, farmoyishlar bajarilishini nazorat qilish axborot tizmini rahbariyat uchun juda katta yengillik tug‘diradi va rahbarlar istalgan vaqtda farmoyishlar bo’yicha o‘ziga kerakli ma’lumotlarga ega bo‘ladi.

BMI uchun qo‘shimcha ma’lumot: Bitiruv malakaviy ishni bajarish vaqtida talaba o‘zini ijobiy tomondan ko‘rsatdi va mavzuning yechimining topishida ijodiy yondashdi hamda yangi yechimlarni taklif qila bildi.

Talaba topshiriq bilan muvaffaqiyatli mehnat qilib, mustaqil ravishda Farmoyishlar bajarilishini nazorat qilish axborot tizimini yaratdi, nazariy va amaliy bilimga ega, unga qo‘yilgan vazifalarni bajara oldi.

Xulosa qilib aytganda, S.Q. Bobojonovaning bitiruv malakaviy ishi yetarlicha saviyada bajarilgan va unga qo‘yiladigan talablarga to‘liq mos keladi, ish himoya talablariga javob beradi va uni bitiruv malakaviy ishi deb taqdim etaman, a’lo darajada bajarilgan va himoya qilishga tavsiya qilaman hamda uning muallifiga 5330200 – “Informatika va axborot texnologiyalari (temir yo’l transportida)” yo‘nalishi bo‘yicha bakalavr darajasini berish mumkin, deb hisoblayman.

Ilmiy rahbar,

“Temir yo‘l transportida axborot

tizimlari” kafedrasi dotsenti, t.f.n. **Aliyev M.M**

“O‘ZBEKISTON TEMIR YO‘LLARI” AKSIYADORLIK JAMIYATI
...KENT TEMIR YO‘L MUHANDI... INSTITUTI
DEVONXONA

Toshkent Temir yo‘l muhandislari instituti, Iqtisodiyot fakultetining 5330200 – “Informatika va axborot texnologiyalari” ta’lim yo‘nalishi AT-26 guruhi talabasi Bobojonova Saida Quvondiq qizining “Development of Information System of Controlling Execution Orders” mavzusidagi bitiruv malakaviy ishiga

TAQRIZ

Ishning dolzarbligi, yangiligi: Farmoyishlarni bajarilishi nazorat qilish har bir muassasa uchun muhim hisoblanadi, shuningdek bunday nazoratni axborot tizimi orqali amalga oshirganda tezkorlik va qulaylik ta’minlanadi hamda hozirgi kunda har bir sohani takomillashtirishda axborot tizimlaridan foydalanilmoqda.

Ishning mazmunini baholash: Bitiruv malakaviy ishi kirish, asosiy qism, xulosa, foydalanilgan adabiyotlar ro‘yxatidan tashkil topgan. Birinchi qismda axborot tizmiga qo’yiladigan talablar va farmoyishlar bajarilishini nazorat qilish ma’lumotlar bazasining infologik modeli berilgan. Ikkinchi qismda axborot tizimini yaratishda foydalanilgan asosiy vositalar va tillar haqida ma’lumot berilgan. Uchinchi qismda ma’lumotlar bazasini va veb-ilova ko’rinishidagi axborot tizimini yaratish bosqichlari keltirilgan. To‘rtinchi qismda hayot faoliyati xavfsizligiga oid ma’lumotlar keltirilgan.

Ishning ijobiy tomonlari: Talaba rahbariyat uchun, ijrochilar uchun, devonxona xodimlari uchun va ma’murlar uchun maxsus oynalar yaratdi va axborot xavsizligini ta’minlash chora-tadbirlarini ko‘rdi.

Talaba S.Q. Bobojonova ma’lumotlar bazasini yaratish uchun MySQL ma’lumotlar bazasini boshqarish tizimini tanlagan. Ma’lumotlar bazasini ishga tushirish uchun eng qulay vosita sifatida PHP tilidan foydalangan.

Ishning amaliy ahamiyati va tadbiq etishga takliflar: bitiruv malakaviy ishning amaliy ahamiyati sifatida farmoyishlar bajarilishini nazorat qilish axborot tizmini rahbariyat uchun juda katta yengillik tug‘diradi va rahbarlar istalgan vaqtda farmoyishlar bo‘yicha o‘ziga kerakli ma’lumotlarga ega bo‘lishini aytishi mumkin.

<u>Ishning kamchiligi</u> sifatida quyidagini aytib o‘tish mumkin: ijrochilarga yangi farmoyish biriktizilganda matnli va ovozli signallar berilishini ham kiritishni ko‘rib chiqish maqsadga muvofiq bo‘lar edi.

Talaba S.Q. Bobojonova ma’lumotlar bazasini yaratish uchun MySQL ma’lumotlar bazasini boshqarish tizimini tanlagan. Ma’lumotlar bazasini ishga tushirish uchun eng qulay vosita sifatida PHP tilidan foydalangan.

<u>Bajarilgan ishga taklif etilyotgan baho</u>: Xulosa qilib aytganda, S.Q. Bobojonovaning bitiruv malakaviy ishini o‘rganishdan so‘ng, talaba topshiriqni muvaffaqiyatli bajarganligi, mustaqil ravishda Farmoyishlar bajarilishini nazorat qilish axborot tizimini yaratilgani, nazariy bilimga ega bo‘lganligi, unga qo‘yilgan vazifalarni hal qilishi mumkinligi ko‘rinadi. Ish bitiruv malakaviy ishi talablariga javob beradi va himoya qilish uchun tavsiya etiladi hamda uning muallifiga 5330200 – “Informatika va axborot texnologiyalari (temir yo‘l transportida)” yo‘nalishi bo‘yicha bakalavr darajasini berish mumkin deb hisoblayman.

Taqrizchi:

Muhammad al-Xorazmiy nomli **N. Ravshanov**
TATU huzuridagi “Axborot-kommunikatsiya
texnologiyalari ilmiy innovatsion markazi”
yetakchi ilmiy xodimi, t.f.d., professor

MUHAMMAD AL-XORAZMIY NOMIDAGI TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI * AXBOROT-KOMMUNIKATSIYA TEXNOLOGIYALARI ILMIY-INNOVATSION MARKAZI *
XODIMLAR BO'LIMI

INTRODUCTION

In pursuance of the Decree of the President of the Republic of Uzbekistan № PQ-2909 of 20.04.2017 "On measures for further development of higher education system" in the further perfection and complex development of higher education system, further improvement of the educational process, the introduction of new curriculums and curricula of higher education, introduction of new pedagogical technologies and teaching methods, qualitative renewal of the master's scientific process and introduction of modern organizational forms defined.

Information systems

Information systems (IS) are formal, sociotechnical, organizational systems designed to collect, process, store, and distribute information. In a sociotechnical perspective, information systems are composed by four components: task, people, structure (or roles), and technology.

A computer information system is a system composed of people and computers that processes or interprets information. The term is also sometimes used in more restricted senses to refer to only the software used to run a computerized database or to refer to only a computer system.[1]

Information system development

A series of methodologies and processes can be used to develop and use an information system. Many developers use a systems engineering approach such as the system development life cycle (SDLC), to systematically develop an information system in stages. The stages of the system development lifecycle are planning, system analysis and requirements, system design, development, integration and testing, implementation and operations and maintenance.

A computer-based information system, following a definition of Langefors, is a technologically implemented medium for:

- recording, storing, and disseminating linguistic expressions,
- as well as for drawing conclusions from such expressions.

System development is done in stages which include:

- Problem recognition and specification
- Information gathering
- Requirements specification for the new system
- System design
- System construction
- System implementation
- Review and maintenance. [2]

Web application

In computing, a web application or web app is a client–server computer program which the client (including the user interface and client-side logic) runs in a web browser. Common web applications include webmail, online retail sales, and online auction.

The general distinction between a dynamic web page of any kind and a "web application" is unclear. Web sites most likely to be referred to as "web applications" are those which have similar functionality to a desktop software application, or to a mobile app. HTML5introduced explicit language support for making applications that are loaded as web pages, but can store data locally and continue to function while offline.

Single-page applications are more application-like because they reject the more typical web paradigm of moving between distinct pages with different URLs. Single-page frameworks might be used to speed development of such a web app for a mobile platform.

What is Data?

Data is nothing but facts and statistics stored or free flowing over a network, generally it's raw and unprocessed. For example: When you visit any website, they might store you IP address, that is data, in return they might add a cookie in your browser, marking you that you visited the website, that is data, your name, it's data, your age, it's data.

Data becomes information when it is processed, turning it into something meaningful. Like, based on the cookie data saved on user's browser, if a website can analyze that generally men of age 20-25 visit us more, that is information, derived from the data collected.

What is a Database?

A Database is a collection of related data organized in a way that data can be easily accessed, managed and updated. Database can be software based or hardware based, with one sole purpose, storing data.

During early computer days, data was collected and stored on tapes, which were mostly write-only, which means once data is stored on it, it can never be read again. They were slow and bulky, and soon computer scientists realized that they needed a better solution to this problem.

Larry Ellison, the co-founder of Oracle was amongst the first few, who realized the need for a software based Database Management System.

What is DBMS?

A DBMS is a software that allows creation, definition and manipulation of database, allowing users to store, process and analyze data easily. DBMS provides us with an interface or a tool, to perform various operations like creating database, storing data in it, updating data, creating tables in the database and a lot more.

DBMS also provides protection and security to the databases. It also maintains data consistency in case of multiple users.

Here are some examples of popular DBMS used these days:

- MySql
- Oracle
- SQL Server
- IBM DB2
- PostgreSQL
- Amazon SimpleDB (cloud based) etc.

1. DESCRIPTION OF THE REQUIREMENTS OF THE INFORMATION SYSTEM

1.1. Statement of the problem of the information system for controlling the execution of orders

Purpose of the system

The information system for controlling the execution of orders must be designed to automate the management of documents of an organization and must contain all the necessary tools for organizing electronic document management.

The Information system for controlling the execution of orders must allow one to create a single information space for an organization. The interaction between objects must be fully automated. This increases the transparency of stages of the workflow process and must contribute to the effective analysis of activities of the organization.

Main goals

As a result of the implementation of the information system, the following main capabilities must be achieved:

- Improving the control process of a document;
- Ensuring optimal user-friendly database search;
- Ability to use and transfer to the system
- Ability to utilize documents of any formats
- Improved performance discipline.

The modules of the Information system for controlling the execution of orders must provide optimal document management for the entire organization (an object to be automated), as they must contain all the necessary tools for successful organization of electronic document management for any company, regardless of the number and type of ownership. The system must be designed to work both within a small department, as well as organization with a complex scheme of information flow.

The information system for controlling the execution of orders must include software that has the following functionalities:

- Registration of documents in the system;
- Creation of registration cards for documents;
- Possibility to attach an electronic document to the registration card;
- Creating a connection between documents registered in the system;
- Assignment of access rights to the document;
- Work with documents in on-line mode;
- Attaching files of any type to the document;
- Creating routes for documents;
- Automatic creation of a control card of documents when registering the RCM in the system, when setting up a document or an order for control.
- Control of execution of documents, installation on control and removal from control;
- Creation of orders and sub-orders on the document;
- Recording of documents in the system.

1.2. Controlling the Execution of Orders database's infological model

The analysis of the described subject area and the tasks to be solved allows to distinguish the following entities:

1. Senders
2. Type_Doc
3. Departments
4. Employees
5. Documents_Come
6. Files_Come
7. Orders

Entity classification

Associative entity - 1

Core entity - 3

Denoted entity - 2

Characteristic entity – 1

Below is a description of the infological model of the database of the Controlling the Execution of Orders in the language of infological designing:

1. **Senders** (Code_Sender, Name)
2. **Type_Doc** (Code_Type, Name)
3. **Departments** (Code_Dep, Name)
4. **Employees** (Code_Emp, Name, Dep_Code, Login, Password, Post)
5. **Documents_Come** (Code_Doc, Reg_num, Reg_Date, Doc_Num, Send_Date, Sender_Code, Type_Code, Theme, Whom, Resolutor)
6. **Orders** (Code_Ord, Doc_Code, Author, Emp_Code, Date_Beg, Date_End, Status)
7. **Files_Come** (Code_File, Order_Code, Name, Date_Add, Emp_Code)

2. DESCRIPTION OF MAIN TECHNOLOGIES, TOOLS AND LANGUAGES

2.1. Web server XAMPP

Xampp is a free and open source cross-platform web server solution stackpackage developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perlprogramming languages.

XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. Everything needed to set up a web server – server application (Apache), database (MariaDB), and scripting language (PHP) – is included in an extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy as well.

Prerequisites

XAMPP requires only one zip, tar, 7z, or exe file to be downloaded and run, and little or no configuration of the various components that make up the web server is required. The Windows' version of XAMPP requires Microsoft Visual C++ 2017 Redistributable.

Features

XAMPP is regularly updated to the latest releases of Apache, MariaDB, PHP and Perl. It also comes with a number of other modules including OpenSSL, phpMyAdmin, MediaWiki, Joomla, WordPress and more. Self-contained, multiple instances of XAMPP can exist on a single computer, and any given instance can be copied from one computer to another. XAMPP is offered in both a full and a standard version (Smaller version).

Usages:

The most obvious characteristic of XAMPP is the ease at which a WAMP webserver stack can be deployed and instantiated. Later some common packaged applications that could be easily installed were provided by Bitnami.

Officially, XAMPP's designers intended it for use only as a development tool, to allow website designers and programmers to test their work on their own computers without any access to the Internet. To make this as easy as possible, many important security features are disabled by default.

XAMPP has the ability to serve web pages on the World Wide Web.

A special tool is provided to password-protect the most important parts of the package.

XAMPP also provides support for creating and manipulating databases in MariaDB and SQLite among others.

Once XAMPP is installed, it is possible to treat a localhost like a remote host by connecting using an FTP client. Using a program like FileZilla has many advantages when installing a content management system (CMS) like Joomla or WordPress. It is also possible to connect to localhost via FTP with an HTML editor.

What are the Main Tools of XAMPP and its definition?

XAMPP contains tools such as Apache, MYSQL, PHP, and Perl. We will see these tools.

2.1.1. Apache server

Apache server is an open source free software which is initially developed by a group of software developers and now it is maintained by Apache software foundation. Apache HTTP is a remote server(computer) if someone request files, images or documents using their browser they will serve those files to clients using HTTP servers. Mainly hosting companies use this application to create a VPS server and shared hosting for their clients.

If you plan to move your testing website to web hosting, we recommend FastComet Cloud Hosting because they provide you the best service compared to other big companies. We have an Exclusive 50% Offer on FastComet Coupon for our WPBlogX visitors.

The Apache HTTP Server, colloquially called Apache is free and open-source cross-platform web server software, released under the terms of Apache License 2.0. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation.

The vast majority of Apache HTTP Server instances run on a Linux distribution, but current versions also run on Windows and a wide variety of Unix-like systems. Past versions also ran on OpenVMS, NetWare, OS/2and other operating systems.

Originally based on the NCSA HTTPd server, development of Apache began in early 1995 after work on the NCSA code stalled. Apache played a key role in the initial growth of the World Wide Web, quickly overtaking NCSA HTTPd as the dominant HTTP server, and has remained most popular since April 1996. In 2009, it became the first web server software to serve more than 100 million websites. As of August 2018, it was estimated to serve 39% of all active websites and 35% of the top million websites.

Feature overview

Apache supports a variety of features, many implemented as compiled modules which extend the core functionality. These can range from authentication schemes to supporting server-side programming languages such as Perl, Python, Tcl and PHP. Popular authentication modules include mod_access, mod_auth, mod_digest, and mod_auth_digest, the successor to mod_digest. A sample of other features include Secure Sockets Layer and Transport Layer Security support (mod_ssl), a proxy module (mod_proxy), a URL rewriting module (mod_rewrite), custom log files (mod_log_config), and filtering support (mod_include and mod_ext_filter).

Popular compression methods on Apache include the external extension module, mod_gzip, implemented to help with reduction of the size (weight) of web pages served over HTTP. ModSecurity is an open source intrusion detection and prevention engine for Web applications. Apache logs can be analyzed through a Web browser using free scripts, such as AWStats/W3Perl or Visitors.

Virtual hosting allows one Apache installation to serve many different websites. For example, one computer with one Apache installation could simultaneously serve example.com, example.org, test47.test-server.example.edu, etc.

Apache features configurable error messages, DBMS-based authentication databases, content negotiation and supports several graphical user interfaces (GUIs).

It supports password authentication and digital certificate authentication. Because the source code is freely available, anyone can adapt the server for specific needs, and there is a large public library of Apache add-ons.

A more detailed list of features is provided below:

Loadable Dynamic Modules

Multiple Request Processing modes (MPMs) including Event-based/Async, Threaded and Prefork.

- Highly scalable (easily handles more than 10,000 simultaneous connections)
- Handling of static files, index files, auto-indexing and content negotiation
- .htaccess per-directory configuration support
- Reverse proxy with caching
- Load balancing with in-band health checks
- Multiple load balancing mechanisms
- Fault tolerance and Failover with automatic recovery
- WebSocket, FastCGI, SCGI, AJP and uWSGI support with caching

Dynamic configuration

TLS/SSL with SNI and OCSP stapling support, via OpenSSL or wolfSSL.

- Name- and IP address-based virtual servers
- IPv6-compatible
- HTTP/2 support
- Fine-grained authentication and authorization access control
- gzip compression and decompression
- URL rewriting
- Headers and content rewriting
- Custom logging with rotation
- Concurrent connection limiting
- Request processing rate limiting
- Bandwidth throttling
- Server Side Includes
- IP address-based geolocation
- User and Session tracking
- WebDAV
- Embedded Perl, PHP and Lua scripting
- CGI support
- public_html per-user web-pages
- Generic expression parser
- Real-time status views
- XML support
- FTP support (by a separate module)

2.1.2. MySQL Database Management System

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.

MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress. MySQL is also used by many popular websites,including Facebook, Twitter, Flickr, and YouTube.

Overview

MySQL is written in C and C++. Its SQL parser is written in yacc, but it uses a home-brewed lexical analyzer. MySQL works on many system platforms, including AIX, BSDi, FreeBSD, HP-
UX, eComStation, i5/OS, IRIX, Linux, macOS, Microsoft
Windows, NetBSD, Novell
NetWare, OpenBSD, OpenSolaris, OS/2 Warp, QNX, Oracle
Solaris, Symbian, SunOS, SCO OpenServer, SCO UnixWare, Sanos and Tru64. A port of MySQL to OpenVMS also exists.

The MySQL server software itself and the client libraries use dual-licensing distribution. They are offered under GPL version 2, or a proprietary license.

Support can be obtained from the official manual. Free support additionally is available in different IRC channels and forums. Oracle offers paid support via its MySQL Enterprise products. They differ in the scope of services and in price. Additionally, a number of third party organisations exist to provide support and services, including MariaDB and Percona.

MySQL has received positive reviews, and reviewers noticed it "performs extremely well in the average case" and that the "developer interfaces are there, and the documentation (not to mention feedback in the real world via Web sites and the like) is very, very good". It has also been tested to be a "fast, stable and true multi-user, multi-threaded sql database server".[3]

History

MySQL was created by a Swedish company, MySQL AB, founded by David Axmark, Allan Larsson and Michael "Monty" Widenius. Original development of MySQL by Widenius and Axmark began in 1994. The first version of MySQL appeared on 23 May 1995. It was initially created for personal usage from mSQL based on the low-level language ISAM, which the creators considered too slow and inflexible. They created a new SQL interface, while keeping the same API as mSQL. By keeping the API consistent with the mSQL system, many developers were able to use MySQL instead of the (proprietarily licensed) mSQL antecedent.

Features

MySQL is offered under two different editions: the open source MySQL Community Server and the proprietary Enterprise Server. MySQL Enterprise Server is differentiated by a series of proprietary extensions which install as server plugins, but otherwise shares the version numbering system and is built from the same code base.

Major features as available in MySQL 5.6:

- A broad subset of ANSI SQL 99, as well as extensions
- Cross-platform support

- Stored procedures, using a procedural language that closely adheres to SQL/PSM
- Triggers
- Cursors
- Updatable views
- Online Data Definition Language (DDL) when using the InnoDB Storage Engine.
- Information schema
- Performance Schema that collects and aggregates statistics about server execution and query performance for monitoring purposes.
- A set of SQL Mode options to control runtime behavior, including a strict mode to better adhere to SQL standards.
- X/Open XA distributed transaction processing (DTP) support; two phase commit as part of this, using the default InnoDB storage engine
- Transactions with savepoints when using the default InnoDB Storage Engine. The NDB Cluster Storage Engine also supports transactions.
- ACID compliance when using InnoDB and NDB Cluster Storage Engines
- SSL support
- Query caching
- Sub-SELECTs (i.e. nested SELECTs)
- Built-in replication support
- Asynchronous replication: master-slave from one master to many slaves or many masters to one slave
- Semi synchronous replication: Master to slave replication where the master waits on replication
- Synchronous replication: Multi-master replication is provided in MySQL Cluster .
- Virtual Synchronous: Self managed groups of MySQL servers with multi

master support can be done using: Galera Cluster or the built in Group Replication plugin

- Full-text indexing and searching
- Embedded database library
- Unicode support
- Partitioned tables with pruning of partitions in optimizer
- Shared-nothing clustering through MySQL Cluster
- Multiple storage engines, allowing one to choose the one that is most effective for each table in the application.
- Native storage engines InnoDB, MyISAM, Merge, Memory (heap), Federated, Archive, CSV, Blackhole, NDB Cluster.
- Commit grouping, gathering multiple transactions from multiple connections together to increase the number of commits per second.

The developers release minor updates of the MySQL Server approximately every two months. The sources can be obtained from MySQL's website or from MySQL's GitHub repository, both under the GPL license.

Limitations

When using some storage engines other than the default of InnoDB, MySQL does not comply with the full SQL standard for some of the implemented functionality, including foreign key references. Check constraints are parsed but ignored by all storage engines before MySQL version 8.0.15.

Up until MySQL 5.7, triggers are limited to one per action / timing, meaning that at most one trigger can be defined to be executed after an INSERT operation, and one before INSERT on the same table. No triggers can be defined on views.

MySQL database's inbuilt functions like UNIX_TIMESTAMP() will return 0 after 03:14:07 UTC on 19 January 2038. Recently, there had been an attempt to solve the problem which had been assigned to the internal queue.

2.1.3. PHP programming language

PHP: Hypertext Preprocessor (or simply PHP) is a general-purpose programming language originally designed for web development. It was originally created by Rasmus Lerdorf in 1994; the PHP reference implementationis now produced by The PHP Group. PHP originally stood for Personal Home Page, but it now stands for the recursive initialism PHP: Hypertext Preprocessor.

PHP code may be executed with a command line interface (CLI), embedded into HTML code, or it can be used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in a web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP can be used for many programming tasks outside of the web context, such as standalone graphical applications and robotic dronecontrol.

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

The PHP language evolved without a written formal specification or standard until 2014, with the original implementation acting as the de facto standard which other implementations aimed to follow. Since 2014, work has gone on to create a formal PHP specification. [4]

Use

PHP is a general-purpose scripting language that is especially suited to server-side web development, in which case PHP generally runs on a web server. Any PHP code in a requested file is executedby the PHP runtime, usually to create dynamic web page content or dynamic images used on websites or elsewhere. It can also be used for command-line scripting and client-side graphical user interface (GUI) applications. PHP can be deployed on most web servers, many operating systems and platforms, and can be used with many relational database management systems (RDBMS). Most web hosting providers support PHP for use by their clients. It is available free of charge, and the PHP Group provides the complete source code for users to build, customize and extend for their own use.

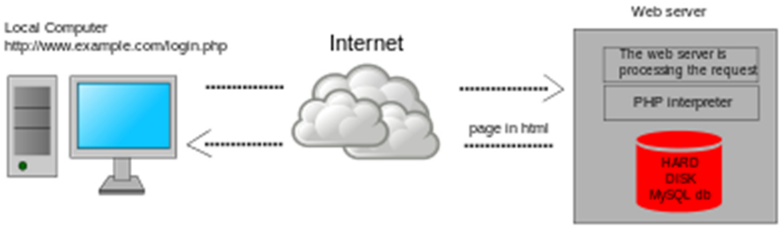


Figure.1. Dynamic web page: example of server-side scripting (PHP and MySQL)

PHP acts primarily as a filter, taking input from a file or stream containing text and/or PHP instructions and outputting another stream of data. Most commonly the output will be HTML, although it could be JSON, XML or binary data such as image or audio formats. Since PHP 4, the PHP parser compiles input to produce bytecode for processing by the Zend Engine, giving improved performance over its interpreter predecessor.

Originally designed to create dynamic web pages, PHP now focuses mainly on server-side scripting, and it is similar to other server-side scripting languages that provide dynamic content from a web server to a client, such as Microsoft's ASP.NET, Sun Microsystems' JavaServer Pages, and mod_perl. PHP has also attracted the development of many software frameworks that provide building blocks and a design structure to promote rapid application development (RAD). Some of these include PRADO, CakePHP, Symfony, CodeIgniter, Laravel, Yii Framework, Phalcon and Zend Framework, offering features similar to other web frameworks.

The LAMP architecture has become popular in the web industry as a way of deploying web applications. PHP is commonly used as the P in this bundle alongside Linux, Apache and MySQL, although the P may also refer to Python, Perl, or some mix of the three. Similar packages, WAMP and MAMP, are also available for Windows and macOS, with the first letter standing for the respective operating system. Although both PHP and Apache are provided as part of the macOS base install, users of these packages seek a simpler installation mechanism that can be more easily kept up to date.

As of April 2007, over 20 million Internet domains had web services hosted on servers with PHP installed and mod_php was recorded as the most popular Apache HTTP Server module. As of June 2018, PHP was used as the server-side programming language on 83.5% of websites where the language could be determined. Web content management systems written in PHP include MediaWiki, Joomla, eZ Publish, eZ Platform, SilverStripe, WordPress, Drupal, and Moodle. Websites written in PHP, in back-end and/or user-facing portion, include Facebook, Digg, Tumblr, Dailymotion, and Slack.

For specific and more advanced usage scenarios, PHP offers a well defined and documented way for writing custom extensions in C or C++. Besides extending the language itself in form of additional libraries, extensions are providing a way for improving execution speed where it is critical and there is room for improvements by using a true compiled language. PHP also offers well defined ways for embedding itself into other software projects. That way PHP can be easily used as an internal scripting language for another project, also providing tight interfacing with the project's specific internal data structures.

PHP received mixed reviews due to lacking support for multithreading at the core language level, though using threads is made possible by the "pthreads" PECL extension.

As of January 2013, PHP was used in more than 240 million websites (39% of those sampled) and was installed on 2.1 million web servers.

A command line interface, php-cli, and two ActiveX Windows Script Host scripting engines for PHP have been produced.

2.2. HTML markup language

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as `<img/>` and `<input/>` directly introduce content into the page. Other tags such as `<p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium(W3C), former maintainer of the HTML and current maintainer of the

CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.[5]

Markup

HTML markup consists of several key components, including those called tags (and their attributes), character-based data types, character references and entity references. HTML tags most commonly come in pairs like `<h1>` and `</h1>`, although some represent empty elements and so are unpaired, for example `<img>`. The first tag in such a pair is the start tag, and the second is the end tag (they are also called opening tags and closing tags).

Another important component is the HTML document type declaration, which triggers standards mode rendering.

The following is an example of the classic "Hello, World!" program:

```
<!DOCTYPE html>
<html>
  <head>
    <title>This is a title</title>
  </head>
  <body>
    <p>Hello world!</p>
  </body>
</html>
```

The text between `<html>` and `</html>` describes the web page, and the text between `<body>` and `</body>` is the visible page content. The markup text `<title>`This is a title`</title>` defines the browser page title.

The Document Type Declaration `<!DOCTYPE html>` is for HTML5. If a declaration is not included, various browsers will revert to "quirks mode" for rendering.

Elements

HTML documents imply a structure of nested HTML elements. These are indicated in the document by HTML tags, enclosed in angle brackets thus: `<p>`.

In the simple, general case, the extent of an element is indicated by a pair of tags: a "start tag" `<p>` and "end tag" `</p>`. The text content of the element, if any, is placed between these tags.

Tags may also enclose further tag markup between the start and end, including a mixture of tags and text. This indicates further (nested) elements, as children of the parent element.

The start tag may also include attributes within the tag. These indicate other information, such as identifiers for sections within the document, identifiers used to bind style information to the presentation of the document, and for some tags such as the `<img>` used to embed images, the reference to the image resource.

Some elements, such as the line break `<br>`, do not permit any embedded content, either text or further tags. These require only a single empty tag (akin to a start tag) and do not use an end tag.

Many tags, particularly the closing end tag for the very commonly used paragraph element `<p>`, are optional. An HTML browser or other agent can infer the closure for the end of an element from the context and the structural rules defined by the HTML standard. These rules are complex and not widely understood by most HTML coders.

The general form of an HTML element is therefore: `<tag attribute1="value1" attribute2="value2">''content''</tag>`. Some HTML elements are defined as empty elements and take the form `<tag attribute1="value1" attribute2="value2">`. Empty elements may enclose no content, for instance, the `<br>` tag or the inline `<img>` tag. The name of an HTML element is the name used in the tags. Note that the end tag's name is preceded by a slash character, /, and that in empty elements the end tag is neither required nor allowed. If attributes are not mentioned, default values are used in each case. [6]

Data types

HTML defines several data types for element content, such as script data and stylesheet data, and a plethora of types for attribute values, including IDs, names, URIs, numbers, units of length, languages, media descriptors, colors, character encodings, dates and times, and so on. All of these data types are specializations of character data.

2.3. CSS style sheet language

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

In addition to HTML, other markup languages support the use of CSS including XHTML, plain XML, SVG, and XUL. [7]

Use

Before CSS, nearly all presentational attributes of HTML documents were contained within the HTML markup. All font colors, background styles, element alignments, borders and sizes had to be explicitly described, often repeatedly, within the HTML. CSS lets authors move much of that information to another file, the style sheet, resulting in considerably simpler HTML.

For example, headings (h1 elements), sub-headings (h2), sub-sub-headings (h3), etc., are defined structurally using HTML. In print and on the screen, choice of font, size, color and emphasis for these elements is presentational.

Before CSS, document authors who wanted to assign such typographic characteristics to, say, all h2 headings had to repeat HTML presentational markup for each occurrence of that heading type. This made documents more complex, larger, and more error-prone and difficult to maintain. CSS allows the separation of presentation from structure. CSS can define color, font, text alignment, size, borders, spacing, layout and many other typographic characteristics, and can do so independently for on-screen and printed views. CSS also defines non-visual styles, such as reading speed and emphasis for aural text readers. The W3C has now deprecated the use of all presentational HTML markup.

For example, under pre-CSS HTML, a heading element defined with red text would be written as:

```
<h1><font color="red"> Chapter 1. </font></h1>
```

Using CSS, the same element can be coded using style properties instead of HTML presentational attributes:

```
<h1 style="color: red;"> Chapter 1. </h1>
```

The advantages of this may not be immediately clear (since the second form is actually more verbose), but the power of CSS becomes more apparent when the style properties are placed in an internal style element or, even better, an external CSS file. For example, suppose the document contains the style element:

```
<style>
h1 {
    color: red;
}
</style>
```

All h1 elements in the document will then automatically become red without requiring any explicit code. If the author later wanted to make h1 elements blue instead, this could be done by changing the style element to:

```
<style>
h1 {
    color: blue;
}
</style>
```

rather than by laboriously going through the document and changing the color for each individual h1 element.

The styles can also be placed in an external CSS file, as described below, and loaded using syntax similar to:

```
<link href="path/to/file.css" rel="stylesheet" type="text/css">
```

This further decouples the styling from the HTML document, and makes it possible to restyle multiple documents by simply editing a shared external CSS file.

Advantages

CSS facilitates publication of content in multiple presentation formats based on nominal parameters. Nominal parameters include explicit user preferences, different web browsers, the type of device being used to view the content (a desktop computer or mobile device), the geographic location of the user and many other variables.

Site-wide consistency

When CSS is used effectively, in terms of inheritance and "cascading", a global style sheet can be used to affect and style elements site-wide. If the situation arises that the styling of the elements should be changed or adjusted, these changes can be made by editing rules in the global style sheet. Before CSS, this sort of maintenance was more difficult, expensive and time-consuming.

Bandwidth

A stylesheet, internal or external, specifies the style once for a range of HTML elements selected by class, type or relationship to others. This is much more efficient than repeating style information inline for each occurrence of the element. An external stylesheet is usually stored in the browser cache, and can therefore be used on multiple pages without being reloaded, further reducing data transfer over a network.

Page reformatting

With a simple change of one line, a different style sheet can be used for the same page. This has advantages for accessibility, as well as providing the ability to tailor a page or site to different target devices. Furthermore, devices not able to understand the styling still display the content.

Accessibility

Without CSS, web designers must typically lay out their pages with techniques such as HTML tables that hinder accessibility for vision-impaired users.

2.4. SQL language

SQL (Structured Query Language) is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS). It is particularly useful in handling structured data where there are relations between different entities/variables of the data. SQL offers two main advantages over older read/write APIs like ISAM or VSAM. First, it introduced the concept of accessing many records with one single command; and second, it eliminates the need to specify how to reach a record, e.g. with or without an index.

Originally based upon relational algebra and tuple relational calculus, SQL consists of many types of statements, which may be informally classed as sublanguages, commonly: a data query language (DQL), a data definition language (DDL), a data control language (DCL), and a data manipulation language (DML). The scope of SQL includes data query, data manipulation (insert, update and delete), data definition (schemacreation and modification), and data access control. Although SQL is often described as, and to a great extent is, a declarative language (4GL), it also includes procedural elements.

SQL was one of the first commercial languages for Edgar F. Codd's relational model. The model was described in his influential 1970 paper, "A Relational Model of Data for Large Shared Data Banks". Despite not entirely adhering to the relational model as described by Codd, it became the most widely used database language.

SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987. Since then, the standard has been revised to include a larger set of features. Despite the existence of such standards, most SQL code is not completely portable among different database systems without adjustments. [8]

History

SQL was initially developed at IBM by Donald D. Chamberlin and Raymond F. Boyce after learning about the relational model from Ted Codd in the early 1970s. This version, initially called SEQUEL (Structured English Query Language), was designed to manipulate and retrieve data stored in IBM's original quasi-relational database management system, System R, which a group at IBM San Jose Research Laboratory had developed during the 1970s.

Chamberlin and Boyce's first attempt of a relational database language was Square, but it was difficult to use due to subscript notation. After moving to the San Jose Research Laboratory in 1973, they began work on SEQUEL. The acronym SEQUEL was later changed to SQL because "SEQUEL" was a trademark of the UK-based Hawker Siddeley Dynamics Engineering Limited company.

After testing SQL at customer test sites to determine the usefulness and practicality of the system, IBM began developing commercial products based on their System R prototype including System/38, SQL/DS, and DB2, which were commercially available in 1979, 1981, and 1983, respectively.

In the late 1970s, Relational Software, Inc. (now Oracle Corporation) saw the potential of the concepts described by Codd, Chamberlin, and Boyce, and developed their own SQL-based RDBMS with aspirations of selling it to the U.S. Navy, Central Intelligence Agency, and other U.S. government agencies. In June 1979, Relational Software, Inc. introduced the first commercially available implementation of SQL, Oracle V2 (Version2) for VAXcomputers.

By 1986, ANSI and ISO standard groups officially adopted the standard "Database Language SQL" language definition. New versions of the standard were published in 1989, 1992, 1996, 1999, 2003, 2006, 2008, 2011, and most recently, 2016.

Design

SQL deviates in several ways from its theoretical foundation, the relational model and its tuple calculus. In that model, a table is a set of tuples, while in SQL, tables and query results are lists of rows: the same row may occur multiple times, and the order of rows can be employed in queries (e.g. in the LIMIT clause).

Critics argue that SQL should be replaced with a language that strictly returns to the original foundation: for example, see The Third Manifesto. However, no known proof exists that such uniqueness cannot be added to SQL itself[citation needed], or at least a variation of SQL. In other words, it's quite possible that SQL can be "fixed" or at least improved in this regard such that the industry may not have to switch to a completely different query language to obtain uniqueness. Debate on this remains open.

Syntax

The SQL language is subdivided into several language elements, including:

Clauses, which are constituent components of statements and queries. (In some cases, these are optional.)

Expressions, which can produce either scalar values, or tables consisting of columns and rows of data

Predicates, which specify conditions that can be evaluated to SQL three-valued logic (3VL) (true/false/unknown) or Boolean truth values and are used to limit the effects of statements and queries, or to change program flow.

Queries, which retrieve the data based on specific criteria. This is an important element of SQL.

Statements, which may have a persistent effect on schemata and data, or may control transactions, program flow, connections, sessions, or diagnostics.

SQL statements also include the semicolon (";") statement terminator. Though not required on every platform, it is defined as a standard part of the SQL grammar.

Insignificant whitespace is generally ignored in SQL statements and queries, making it easier to format SQL code for readability.

2.5. Description of *phpMyAdmin*

phpMyAdmin is a free web application that provides a convenient GUI for working with the MySQL database management system. It is the most popular MySQL administration tool that is used by millions of users worldwide and has won numerous awards and honors.

Written in PHP, it has all common functions that you may need when developing a MySQL-based application or website. It also gave birth to several similar products, for example phpPgAdmin which provides similar functionality for the PostgreSQL DBMS.

Overview of phpMyAdmin Capabilities

You can use phpMyAdmin for Windows or almost any other operating system ever existed. It can export and import database created and managed by MySQL DBMS, as well as work with some other data formats.

The list of supported formats includes SQL, CVS, CodeGen, CVS for Microsoft Excel, Microsoft Word 2000, JSON, LaTeX, MediaWiki Table, OpenDocument Spreadsheet, OpenDocument Text, PDF, PHP Array, Texy! Text and YAML data types.

General Features

With phpMyAdmin you can:

- Create and remove users, manage user permissions
- Create, alter and drop databases, tables, fields and rows
- Search objects in the entire database or any particular tables
- Import and export data in different formats, including SQL, XML and CSV
- Monitor processes and track performance of different queries

- Execute custom SQL queries
- Back up your MySQL databases in manual mode

The tool is very simple to set up and use. It is usually installed on the same computer with MySQL database, so all commands are executed very quickly and don’t rely on network connectivity.

Administering MySQL DBMS

While phpMyAdmin is a perfect tool for browsing the database, managing user privileges and executing SQL queries, it can’t be considered a full-featured administration tool.

Administration expects you to not only be able to work with an object, but also deal with unpredictable situations, such as SQL injections, user mistakes and other cases of database corruption. Especially it touches the matter of automatic MySQL backup.

Data Backup Problems of phpMyAdmin

The export/import functionality of phpMyAdmin lacks a lot of the features that you would expect to find in a backup software solution:

Scheduling. With phpMyAdmin there is no way to export database data automatically.

Storage media support. Since phpMyAdmin is a web-based software, you can work with it via browser only. This means that you can save backups only to local drives available on your system, via the Save As... dialog of your browser.

Compression, encryption, and other options. The files exported with phpMyAdmin are saved as common text files, without any additional processing. Storing them in the original form usually occupies a lot of disk space, and is not secure.

Dumping MySQL Databases with phpMyAdmin

The good news about backing up MySQL tables with phpMyAdmin is that it saves data as database dumps, similarly to how Handy Backup and mysqldump work.

Resulting files can be used not only for disaster recovery, but also to conduct different experiments and migrate from MySQL to other database management systems. You can learn more about it on the MySQL Recovery feature page.

3. DEVELOPMENT OF INFORMATION SYSTEM

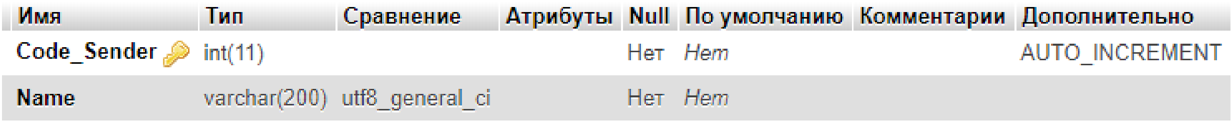
3.1. Creating database Controlling the Execution of Orders

Creation of tables

Based on the problem statement, created the following tables. Figures 2-8 show the corresponding tables.

Table SENDERS. This table stores information about senders:

```
CREATE TABLE `senders` (
  `Code_Sender` int(11) NOT NULL,
  `Name` varchar(200) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

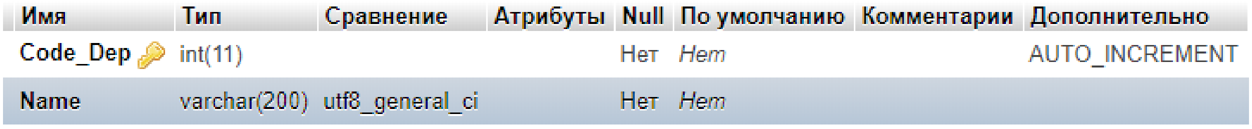


Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
Code_Sender 🔑	int(11)			Нет	*Нет*		AUTO_INCREMENT
Name	varchar(200)	utf8_general_ci		Нет	*Нет*		

Figure.2. Table SENDERS

Table DEPARTMENTS. This table stores information about departments:

```
CREATE TABLE `department` (
  `Code_Dep` int(11) NOT NULL,
  `Name` varchar(200) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

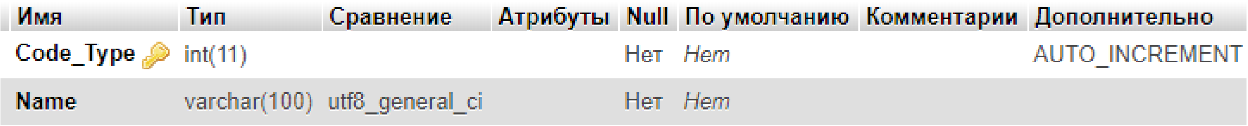


Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
Code_Dep 🔑	int(11)			Нет	*Нет*		AUTO_INCREMENT
Name	varchar(200)	utf8_general_ci		Нет	*Нет*		

Figure.3. Table DEPARTMENTS

Table TYPE_DOCS. This table stores information about types of the incoming documents:

```
CREATE TABLE `type_docs` (
  `Code_Type` int(11) NOT NULL,
  `Name` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

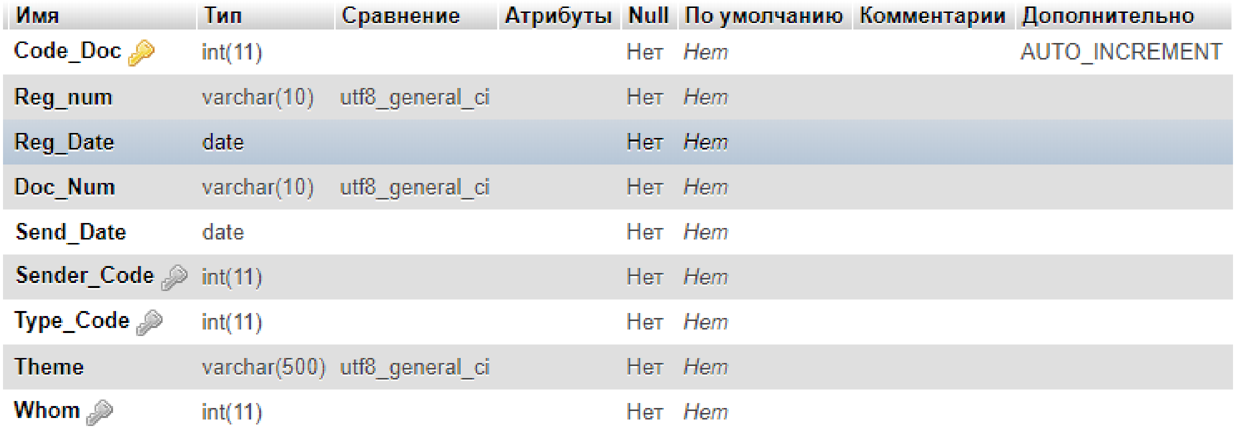


Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
Code_Type	int(11)			Нет	*Нет*		AUTO_INCREMENT
Name	varchar(100)	utf8_general_ci		Нет	*Нет*		

Figure.4. Table TYPE_DOCS

Table DOCUMENTS_COME. This table stores all the information about incoming documents:

```
CREATE TABLE `documents_come` (
  `Code_Doc` int(11) NOT NULL,
  `Reg_num` varchar(10) NOT NULL,
  `Reg_Date` date NOT NULL,
  `Doc_Num` varchar(10) NOT NULL,
  `Send_Date` date NOT NULL,
  `Sender_Code` int(11) NOT NULL,
  `Type_Code` int(11) NOT NULL,
  `Theme` varchar(500) NOT NULL,
  `Whom` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

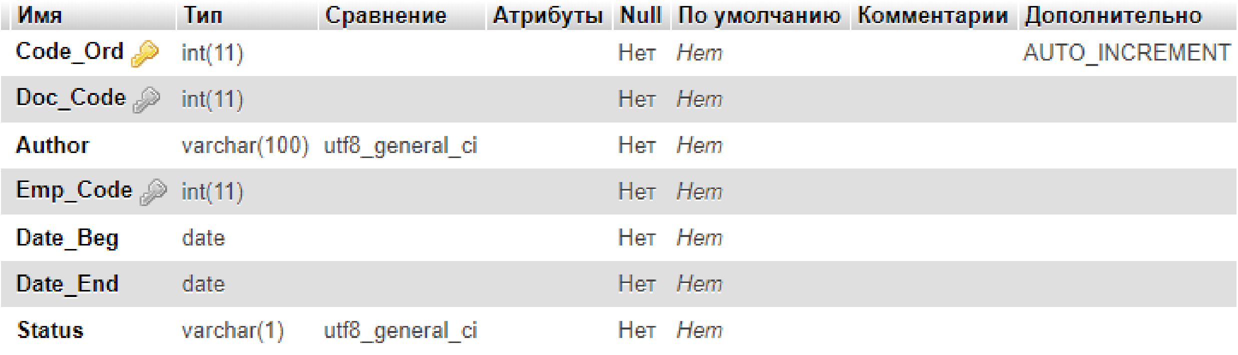


Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
Code_Doc	int(11)			Нет	*Нет*		AUTO_INCREMENT
Reg_num	varchar(10)	utf8_general_ci		Нет	*Нет*		
Reg_Date	date			Нет	*Нет*		
Doc_Num	varchar(10)	utf8_general_ci		Нет	*Нет*		
Send_Date	date			Нет	*Нет*		
Sender_Code	int(11)			Нет	*Нет*		
Type_Code	int(11)			Нет	*Нет*		
Theme	varchar(500)	utf8_general_ci		Нет	*Нет*		
Whom	int(11)			Нет	*Нет*		

Figure.5. Table DOCUMENTS_COME

Table ORDERS. This table stores information about orders:

```
CREATE TABLE `orders` (
  `Code_Ord` int(11) NOT NULL,
  `Doc_Code` int(11) NOT NULL,
  `Author` varchar(100) NOT NULL,
  `Emp_Code` int(11) NOT NULL,
  `Date_Beg` date NOT NULL,
  `Date_End` date NOT NULL,
  `Status` varchar(1) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

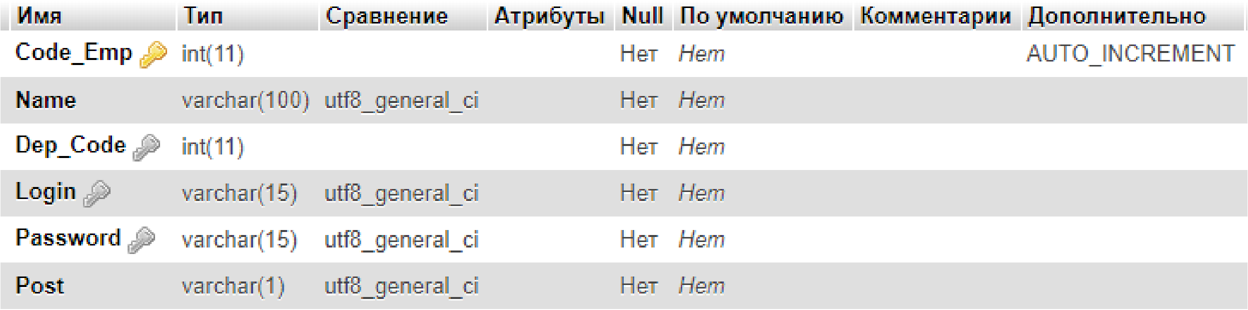


Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
Code_Ord 🔑	int(11)			Нет	*Нет*		AUTO_INCREMENT
Doc_Code 🔑	int(11)			Нет	*Нет*		
Author	varchar(100)	utf8_general_ci		Нет	*Нет*		
Emp_Code 🔑	int(11)			Нет	*Нет*		
Date_Beg	date			Нет	*Нет*		
Date_End	date			Нет	*Нет*		
Status	varchar(1)	utf8_general_ci		Нет	*Нет*		

Figure.6. Table ORDERS

Table EMPLOYEES. This table stores information about employees of the organization:

```
CREATE TABLE `employees` (
  `Code_Emp` int(11) NOT NULL,
  `Name` varchar(100) NOT NULL,
  `Dep_Code` int(11) NOT NULL,
  `Login` varchar(15) NOT NULL,
  `Password` varchar(15) NOT NULL,
  `Post` varchar(1) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

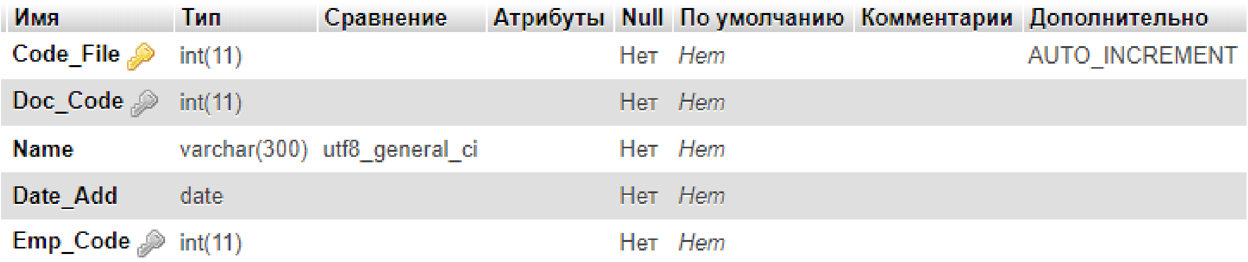


Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
Code_Emp 🔑	int(11)			Нет	*Нет*		AUTO_INCREMENT
Name	varchar(100)	utf8_general_ci		Нет	*Нет*		
Dep_Code 🔑	int(11)			Нет	*Нет*		
Login 🔑	varchar(15)	utf8_general_ci		Нет	*Нет*		
Password 🔑	varchar(15)	utf8_general_ci		Нет	*Нет*		
Post	varchar(1)	utf8_general_ci		Нет	*Нет*		

Figure.7. EMPLOYEES

Table FILES. This table stores information about senders:

```
CREATE TABLE `files` (
  `Code_File` int(11) NOT NULL,
  `Doc_Code` int(11) NOT NULL,
  `Name` varchar(300) NOT NULL,
  `Date_Add` date NOT NULL,
  `Emp_Code` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```



Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
Code_File 🔑	int(11)			Нет	*Нет*		AUTO_INCREMENT
Doc_Code 🔑	int(11)			Нет	*Нет*		
Name	varchar(300)	utf8_general_ci		Нет	*Нет*		
Date_Add	date			Нет	*Нет*		
Emp_Code 🔑	int(11)			Нет	*Нет*		

Figure.8. Table FILES

Adding indexes to the tables

Indexes of the table `department`:

```
ALTER TABLE `department`
  ADD PRIMARY KEY (`Code_Dep`);
```

Indexes of the table `documents_come`:

```
ALTER TABLE `documents_come`
  ADD PRIMARY KEY (`Code_Doc`),
  ADD KEY `Sender_Code` (`Sender_Code`),
  ADD KEY `Type_Code` (`Type_Code`),
  ADD KEY `Whom` (`Whom`);
LTER TABLE `documents_come`
  ADD CONSTRAINT `documents_come_ibfk_1` FOREIGN
KEY (`Type_Code`) REFERENCES `type_docs` (`Code_Type`),
      ADD CONSTRAINT `documents_come_ibfk_2` FOREIGN
KEY (`Sender_Code`) REFERENCES `senders`
(`Code_Sender`),
      ADD CONSTRAINT `documents_come_ibfk_3` FOREIGN
KEY (`Whom`) REFERENCES `employees` (`Code_Emp`);
```

Indexes of the table `employees`:

```
    ALTER TABLE `employees`
      ADD PRIMARY KEY (`Code_Emp`),
      ADD UNIQUE KEY `Login` (`Login`),
      ADD UNIQUE KEY `Password` (`Password`),
      ADD KEY `Dep_Code` (`Dep_Code`);
    ALTER TABLE `employees`
      ADD CONSTRAINT `employees_ibfk_1` FOREIGN KEY
(`Dep_Code`) REFERENCES `department` (`Code_Dep`);
```

Indexes of the table `files`:

```
    ALTER TABLE `files`
      ADD PRIMARY KEY (`Code_File`),
      ADD KEY `Order_Code` (`Doc_Code`),
      ADD KEY `Emp_Code` (`Emp_Code`);
    ALTER TABLE `files`
      ADD CONSTRAINT `files_ibfk_1` FOREIGN KEY
(`Doc_Code`) REFERENCES `documents_come` (`Code_Doc`),
      ADD CONSTRAINT `files_ibfk_2` FOREIGN KEY
(`Emp_Code`) REFERENCES `employees` (`Code_Emp`);
```

Indexes of the table `orders`:

```
    ALTER TABLE `orders`
```

```
  ADD PRIMARY KEY (`Code_Ord`),
  ADD KEY `Emp_Code` (`Emp_Code`),
  ADD KEY `Doc_Code` (`Doc_Code`);
ALTER TABLE `orders`
  ADD CONSTRAINT `orders_ibfk_2` FOREIGN KEY (`Emp_Code`) REFERENCES `employees` (`Code_Emp`),
  ADD CONSTRAINT `orders_ibfk_3` FOREIGN KEY (`Doc_Code`) REFERENCES `documents_come` (`Code_Doc`);
```

Indexes of the table `senders`:

```
ALTER TABLE `senders`
  ADD PRIMARY KEY (`Code_Sender`);
```

Indexes of the table `type_docs`:

```
ALTER TABLE `type_docs`
  ADD PRIMARY KEY (`Code_Type`);
```

After adding indexes to the tables, structure of the database Controlling Execution of Orders would be seen like this:

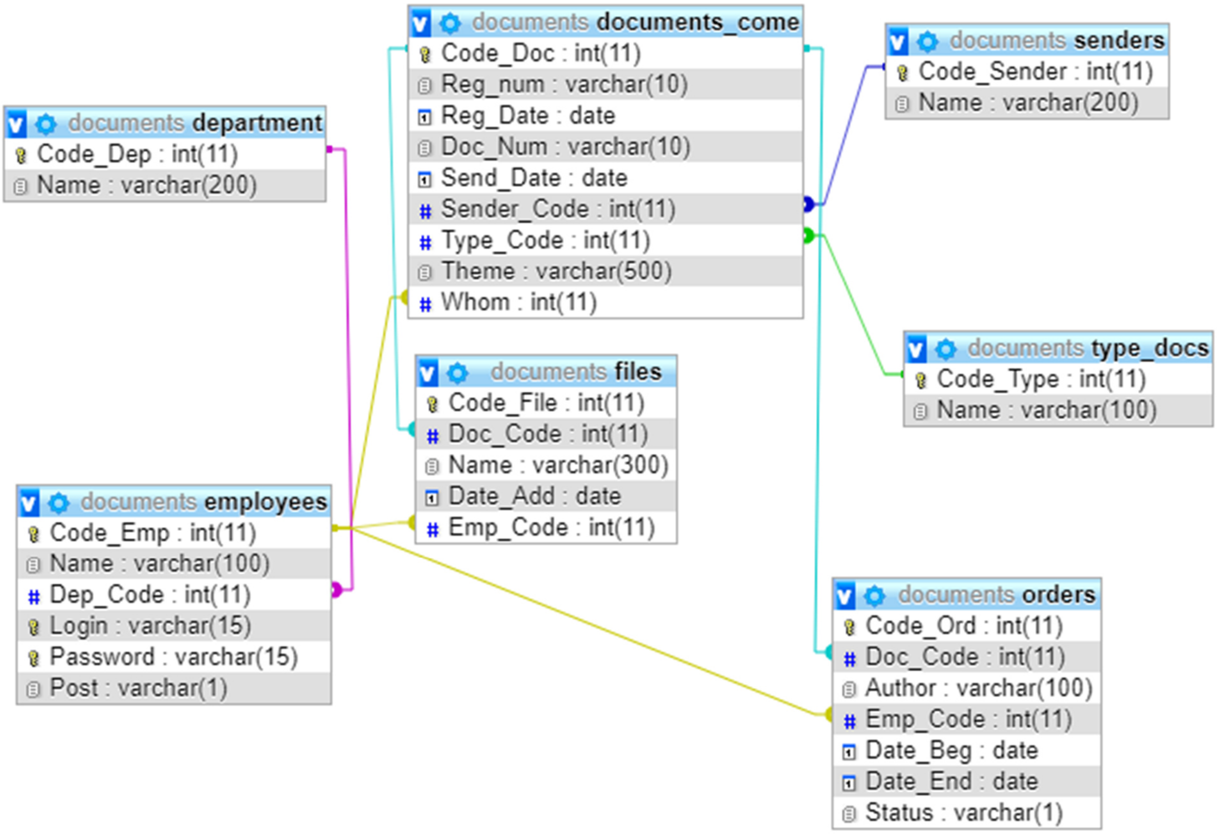
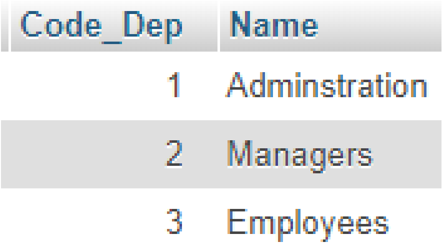


Figure.9. Structure of the database Controlling Execution of Orders

Inserting data to the tables

Figures 10-11 show the corresponding tables.

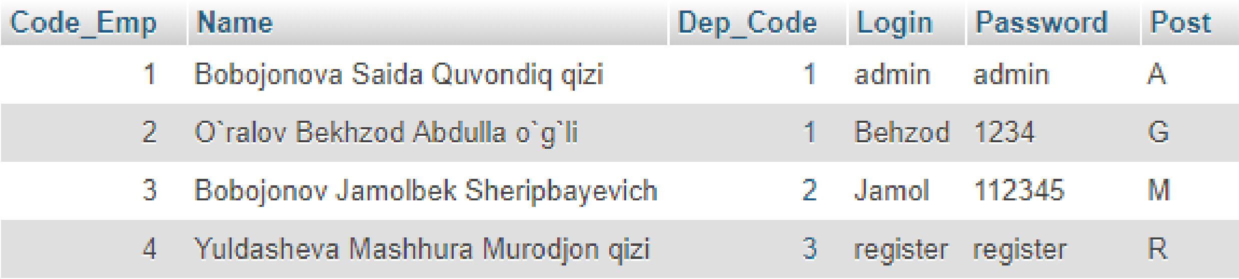
```
    INSERT    INTO    `department`    (`Code_Dep`,    `Name`)
VALUES
    (1, 'Adminstration'),
    (2, 'Managers'),
    (3, 'Employees');
```



Code_Dep	Name
1	Adminstration
2	Managers
3	Employees

Figure.10. Table Departments

```
    INSERT   INTO   `employees`   (`Code_Emp`,   `Name`,
`Dep_Code`, `login`, `password`, `Post`) VALUES
    (1,  'Bobojonova  Saida  Quvondiq  qizi',  1,  'admin',
'admin', 'A'),
    (2, 'O`ralov Bekhzod Abdulla o`g`li', 1, 'Behzod',
'1234', 'G'),
    (3,   'Bobojonov   Jamolbek   Sheripbayevich',   2,
'Jamol', '12345', 'M'),
    (4,   'Yuldasheva   Mashhura   Murodjon   qizi',   3,
'register', 'register', 'R');
```



Code_Emp	Name	Dep_Code	Login	Password	Post
1	Bobojonova Saida Quvondiq qizi	1	admin	admin	A
2	O`ralov Bekhzod Abdulla o`g`li	1	Behzod	1234	G
3	Bobojonov Jamolbek Sheripbayevich	2	Jamol	112345	M
4	Yuldasheva Mashhura Murodjon qizi	3	register	register	R

Figure.11. Table Employees

3.2. Creating web-application Controlling the Execution of Orders

3.2.1. Web-application interface development

Main.css file contains all the styles of the web-application.

Main.css:

```
header {
  padding-bottom: 210px; }
  header img {
    width: 1110px;
    height: 600px; }
  header .rasm {
    width: 100%;
```

```
        height: 140px;
        background-color: #4b3aff;
        opacity: 0.5;
        position: relative;
        margin-top: -350px;
        padding-top: 50px; }

    #asosiy {
      transition: all 0.5s linear; }

    #kafedra {
      transition: all 0.5s linear; }

    #yangilik {
      transition: all 0.5s linear; }

    #kantakt {
      transition: all 0.5s linear; }

    main section:nth-child(1) .container {
      background-color: cornflowerblue; }
      main section:nth-child(1) .container .row .col-
md-4 {
        border: 1px solid darkmagenta; }
        main section:nth-child(1) .container .row .col-
md-4 img {
          width: 150px;
          height: 150px;
          margin: 0 auto;
          border-radius: 300px;
          border: 10px solid #312aee; }
          main section:nth-child(1) .container .row
.col-md-4 img img:hover {
            background-color: greenyellow; }
    main section:nth-child(3) .container {
      background-color: #4b3aff; }
    main section:nth-child(4) .container {
      background-color: #679af4; }

    footer .container {
      background-color: #6b21af; }

    @media screen and (max-width: 600px) {
      section:nth-child(1) .container .row .col-md-12
.row .col-4 img {
        width: 80%;
        height: 30%;
        border-radius: 50%;
        border: 10px solid #6b21af; } }
```

3.2.2. Dynamic part of the web-application

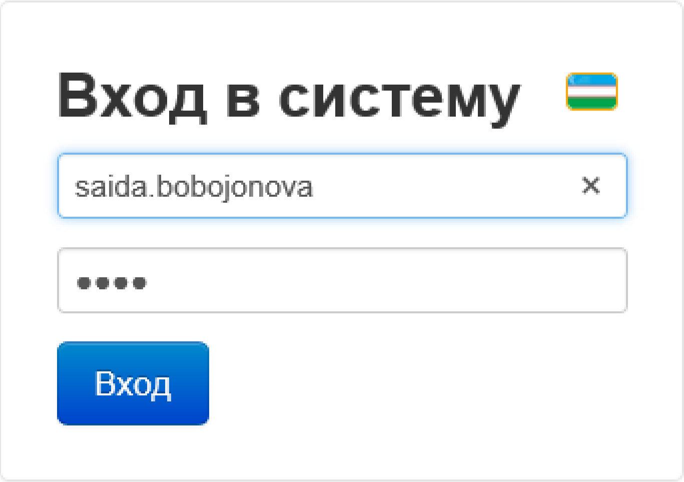
Main page of the web-application, which provides entering to the system:

Index.php:

```
<?php
$servername = "localhost";
$dBUsername = "Saida";
$dBPassword = "1123";
```

```
    $dBName = "documents";
    try {
    $conn = new PDO("mysql:host=$servername;
dbname=$dBName", $dBUsername, $dBPassword);
    $conn -> setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
    $name = $password = $nameErr = $passwordErr =
$error = "";
    if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty(test_input($_POST['name']))) {
    $nameErr = "Enter Username";
    }
    else {
    $name = test_input($_POST['name']);
    }
    if (empty(test_input($_POST['password']))) {
    $passwordErr = "Enter Password";
    }
    else {
    $password = test_input($_POST['password']);
    }
    }
    if (empty($nameErr)&&empty($passwordErr)) {
    $stmt = $conn -> query("SELECT * FROM Employees
WHERE login = '$name' and password = '$password';");
    if ($stmt -> execute()) {
    if ($stmt -> rowcount() == 1) {
    session_start();
    $_SESSION["login"] = $name;
$row = mysqli_fetch_assoc($stmt);
$post = $row['post'];
$nameEmp = $row['name'];
if ($post == 'A') {
header("Location: adm-page.php");
}
else if ($post == 'G') {
header("Location: gen-dir.php");
}
else if ($post == 'R') {
header("Location: registrator.php");
}
else {
header("Location: emp-page.php");
}
} else {
$error = "Invalid login or password";
}
}
}
}
catch(PDOException $e) {
echo "Error: ".$e->getMessage();
}
function test_input($data) {
$data = trim($data);
$data = stripcslashes($data);
$data = htmlspecialchars($data);
return $data;
```

```
    }
    ?>
    <!DOCTYPE html>
    <html>
    <head>
    <title></title>
    </head>
    <body align="center">
    <form action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]) ?>"
method="POST">
    <h1>Вход в систему h1>
    <input type="text" name="name"><br><br>
    <input type="password" name="password"><br><br>
    <input type="submit" name="submit" value="Вход">
    </form>
    <?php $conn=null; ?>
    </body>
    </html>
```



Вход в систему

saida.bobojonova

••••

Figure.12. Web-page index.php

4. Labor safety

4.1. Computer health and safety at work

Employees who regularly work with display screen equipment (DSE) such as computers - also known as visual display units (VDUs) - may be at risk of getting aches and pains. These injuries normally affect the hands, arms, neck or back. They are especially at risk if they work for long uninterrupted periods. These upper limb disorders may become persistent or even disabling if they are not reported and treated early.

Workers using DSE for long periods are also at risk of temporary eyestrain and related problems such as headaches, stress and back pain. This could lead to time off work and lost productivity.

Under health and safety law, employers have a duty to reduce risks to staff using DSE.

This guide outlines how to create comfortable working conditions for users through good workplace design. It will tell you how to set up good working practices for DSE work. It also covers how to train staff to use DSE safely.

Display screen equipment health and safety

Under health and safety law, you must ensure that computer screens or monitors, keyboards, pointing devices, furniture and the working environment meet certain minimum requirements. They must be easy to adjust to each person's needs.

Computer screens

In most cases computer screens should:

- tilt and swivel to suit the needs of the user
- be adjusted to the correct height for the user - either using a separate base or an adjustable table
- show clear characters with enough spacing
- show a stable image with no flickering
- have easily adjustable brightness and contrast controls

- be free of reflected glare
- have a screen size suitable for its intended use

You should also ensure that users:

- know how to adjust brightness and contrast controls to suit lighting conditions
- keep the screen surface clean
- know how to adjust software settings to make text large enough to read easily
- take frequent breaks

Keyboards

Keyboards should:

- be tilt-adjustable and separate from the screen to allow the user to work comfortably
- have enough space in front of them to allow the user to rest their hands and arms when not keying
- be legible and have a matt surface to minimise glare
- You should also ensure that users:
- adjust the keyboard to a comfortable position
- keep their wrists straight when typing
- use the keys lightly and don't overstretch their fingers
- take frequent breaks

Pointing devices

If an employee uses a mouse, trackball or other pointing device for long periods, they may develop problems in their hands, wrists or upper body.

You can minimise risk by:

- training the user to use the mouse with a relaxed arm and a straight wrist
- informing the user that their fingers should rest lightly on the buttons and not press too hard

- limiting the length of intensive sessions using pointing devices
- ensuring users take frequent short breaks
- ensuring the device is suitable for the task
- training the user in keyboard shortcuts
- training the user to adjust the software settings - these control the reaction time between the mouse and the cursor on the screen

Laptop health and safety

Some of the design features on laptops and other portable computers can make them uncomfortable to use for long periods. Employees shouldn't routinely use laptops where full-sized equipment is available. Alternatively, they should be provided with a laptop docking station so that they can work with a full-sized keyboard and screen.

What to consider when buying laptops

When you're thinking about buying portable computers, they should ideally:

- be as light as possible - 3 kilograms or under
- be fitted with as large and clear a screen as possible - preferably detachable or height-adjustable
- come with a lightweight carrying case with handle and shoulder straps
- have as long a battery life as possible and have extra transformer/cable sets so that the user has a set in each main location where the equipment is used
- have a tilt-adjustable keyboard
- be able to be used with a docking station
- have friction pads underneath to prevent the laptop from sliding across work surfaces
- have enough memory and speed for the software used

Employee laptop training

You must provide training for employees using portable display screen equipment (DSE).

In addition to ordinary DSE training, training for laptop users should include information on:

- comfortable postures for using a laptop
- using the keyboard at the right height
- adjusting the screen to reduce reflection and glare
- the need for regular rest breaks
- how to report any problems that develop
- how to reduce the manual handling risk, eg by reducing the amount of extra equipment and paperwork to be carried
- how to reduce the risk from theft or mugging

Workstation health and safety: desks, chairs and posture

A user's workstation furniture should minimise the risk of injuries. Posture also plays a vital role in preventing upper limb disorders.

Desk health and safety

The work desk or work surface should:

- be big enough to allow the user to arrange the screen, keyboard, documents, etc in a flexible way
- have a matt surface
- be big enough to let the user work comfortably and to change position
- be stable and positioned so that it's comfortable and easy to use where an employee uses a document holder

Office chair health and safety

Workstation chairs should:

- be stable and allow the user to work comfortably
- be adjustable in height
- have a seatback adjustable in height and tilt

Other workstation considerations

Other features of the workstation you should consider include:

- arm supports must not interfere with movement, eg by stopping the user getting the chair under the work surface
- seat width should be wider than the hip width of the person using it
- the back rest should provide support for the back in all sitting positions, particularly the lumbar region.
- a foot rest may be needed if users are unable to rest their feet flat on the floor
- the importance of correct and comfortable seating should be emphasised to employees in a risk assessment

Posture when using a computer

Adopting the right posture when working at a computer is very important in order to prevent health problems. You must ensure that employees are aware of good posture and technique, including the importance of:

- avoiding sitting in the same position for long periods
- keeping their forearms roughly horizontal, and their eyes at the same height as the top of the display screen equipment
- ensuring items that are regularly used are within easy reach
- positioning a mouse or other pointing device close by so they can use it with a relaxed arm and a straight wrist

Working conditions for computer users

The general conditions in the workplace also have an effect on the health and safety of display screen equipment users.

You need to assess:

Noise levels - the equipment shouldn't be so noisy that it distracts the user. If you can't use quieter equipment, consider soundproofing or moving the equipment. You could use sound-insulating partitions between noisy equipment and the rest of the workstation as an alternative.

Lighting - surrounding windows must have curtains or blinds which users can adjust to prevent reflected glare. If needed, provide users with lighting appropriate to their tasks and particular workstation. Users should have control over their lighting to prevent reflected glare.

Temperature - the equipment should not give out so much heat that the user becomes uncomfortable.

Humidity - it's important that you maintain ventilation and humidity at a level which keeps the user comfortable.

Computer health and safety: task design and rest breaks

You must plan the activities of computer users so that they don't work for long uninterrupted periods on display screen equipment (DSE). You can do this through a combination of rest breaks and changes in work activity which allow users to change posture.

You also have a duty to tell employees about the importance of changing activities and taking breaks and to encourage them to do both. See what you need to do about health and safety.

Task design

Good design of the task can be as important as the right choice of furniture and equipment. Whenever possible you should:

- design jobs so that employees have a mix of activities and some say over which tasks they perform and when
- match staffing levels to workload so that individuals are neither overworked nor underworked
- give employees some say in the way work is planned and carried out

Rest breaks

An employee's need for rest breaks will vary depending on the type of work they are doing and how intensely they are working. As a general rule however:

Short, frequent breaks are better than longer, less frequent ones. A five to ten minute break after 50-60 minutes' DSE work is better than a 15-20 minute break after two hours.

The employee should have some choice over when to take breaks.

Employees should be encouraged to do different tasks or activities during their break, ideally away from the workstation.

Train employees to use computers safely

Under health and safety law, you must train your employees in the safe use of display screen equipment (DSE). This training should include:

- the health risks of using DSE incorrectly
- how to position and adjust all equipment and furniture to make it comfortable
- the importance of good posture and the need to change position frequently
- the right technique for using a keyboard, mouse, etc
- how to use the hardware and software
- the need for short, frequent breaks from screen work
- how to recognise the warning signs of health problems from DSE use
- how to report any problems
- entitlement to an eyesight test
- the need to clean and inspect DSE regularly to identify any problems

You must also tell employees about the minimum standards relating to DSE and risk assessment.

Where you upgrade or replace equipment or furniture, you must ensure that you retrain staff accordingly and revise the risk assessment.

Homeworkers

As an employer, you have a duty to ensure that employees working at home or from other locations do so safely. Homeworkers may carry out their own risk assessments using an ergonomic checklist, but you will have to train them to do so.

Employees using workstations at home or at other locations away from the office will need training in the safe use of DSE.

They will also need extra training covering:

- how to conduct a risk assessment of their workstation
- the importance of good posture, taking breaks, etc to make up for the lack of direct supervision
- how to report any problems

Computer workstation risk assessment

By law, employers must carry out risk assessments on computer workstations and the employees who use them. This includes homeworkers. A risk assessment should cover all aspects of the task and workplace.

These aspects will include:

- the physical layout of equipment and furniture
- the job being done
- the user's posture
- any special needs of staff
- the need for rest breaks
- the general office environment

Where you identify a problem, you must take steps to reduce the risk to staff.

It's good practice to arrange a re-assessment if there are any changes affecting the workstation or the user directly. For example, if new equipment or furniture is brought in, a re-assessment will help identify the best ergonomic layout. Similarly, when an employee informs you that she is pregnant, a review should be carried out urgently and repeated regularly as the pregnancy develops. Where an employee reports a repetitive strain injury, a re-assessment should be carried out to help prevent a recurrence of symptoms.

Using a checklist

Employees who are involved in risk assessments are more likely to report any problems as they occur. A good way of doing this is to get staff to fill in an ergonomic best practice checklist covering each aspect of the workstation. You should get completed checklists reviewed by an assessor who has been trained to help identify any problems and find a solution.

Free eye tests for employees who use computers

All employees who regularly use display screen equipment (DSE) have the right to ask you to pay for **eye and eyesight tests**. This will be carried out by an optometrist or doctor and it's your duty to pay the fee. DSE users have the right to regular tests thereafter.

Employers only have to pay for **glasses** if special ones are needed for DSE work and the employee can't use normal glasses. Where special glasses are prescribed, you only have to cover the basic cost, ie the lenses and frames. It's a good idea to set a fixed limit for the basic cost of glasses and use your staff handbook to communicate this. If an employee wants to buy frames or glasses that exceed this limit, you must still pay the basic cost but the employee must make up the difference.

To prevent temporary eye strain and related problems, such as headaches, fatigue and stress, it is important to ensure that your employees take regular rest breaks and that workstations meet the minimum risk standards.

CONCLUSION

As a result of the final work, the database and web-application was designed and developed the information system for Controlling execution of Orders. The developed product meets the goals and is implemented on the basis of the MySQL database management system and PHP programming language. The functions of the organization's office are analyzed and the main type of its activity is revealed. On the basis of which the database of the Information System for the Execution of orders was developed. Each user or any other creator of the program can expand the capabilities of this program, widely use it and recommend it to other developers.

REFERENCES AND SOURCES

1. Piccoli, Gabriele; Pigni, Federico *Information systems for managers: with cases (Edition 4.0 ed.)*. Prospect Press. p. 28.
2. Rainer, R. Kelly Jr and Casey G. Cegielski. *Introduction to Information System: Support and Transforming Business Fourth Edition*. New Jersey: John Wiley and Sons, Inc., 2012.
3. В. Васвани. *MySQL: использование и администрирование.* – «Питер», 2011. – 368 с.
4. Дмитрий Котеров, Алексей Костарев. *PHP*. — СПб.: «БХВ-Петербург», 2005. — С. 1120.
5. Berners-Lee, Tim; Connolly, Daniel (June 1993). *"Hypertext Markup Language (HTML): A Representation of Textual Information and MetaInformation for Retrieval and Interchange"*.
6. Dan Cederholm (2009): *Web Standards Solutions, The Markup and Style Handbook*, Friends of Ed.Meyer, Eric A. (2006).
7. Cascading Style Sheets: *The Definitive Guide*, Third Edition. O'Reilly Media, Inc.
8. C. J. Date with Hugh Darwen: *A Guide to the SQL standard : a users guide to the standard database language SQL, 4th ed*., Addison Wesley, USA 1997.