

Zeynəb Arzu qızı BİNNƏDƏLİYEVƏ
Qərbi Kaspi Universiteti, magistrant
E-mail: binnedeliyevazeyneb@gmail.com

İCTİMAİ İAŞƏ OBYEKLƏRİ ÜÇÜN PAYLANMIŞ İNFORMASIYA SİSTEMLƏRİNİN QURULMASI VƏ TƏTBİQİ

Xülasə

Məqalədə informasiya sistemlərindən ictimai obyektlərdə səmərəli istifadəsinin təmin olunmasının necə qurulmasından danışılır. Kafe və restoranlarda müştərilərin sifarişlərini onlayn verilməsi işinin idarə edilməsinin mümkünlüyünü vurğulayır. Müəyyən olunmuş məbləğin elektron formada necə həyata keçirilməsini göstərir. Verilənlər bazasından istifadə edərək lazım olan informasiyanı ən qısa zaman ərzində necə tapmağın yollarını nümayiş etdirir.

Açar sözlər: paylanmış sistemlər, verilənlər bazası, axtarış, proqram, strukturlaşdırma, elektron avadanlıqlar.

UOT: 334

JEL: O00

DOI: <https://doi.org/10.54414/SORT6802>

Giriş

Müasir dünyada müxtəlif informasiya texnologiyaları fəal şəkildə inkişaf etməkdədir. Bu tip texnologiyanın tez-tez istifadə olunduğu mövzu sahələrindən biri də bir növ mühasibat uçotu tələb edən sahələrdir. Coğrafi olaraq paylanmış istehsal müəssisələrində mühasibat uçotunun təşkili problemi xüsusilə çətinləşir ki, bu da müxtəlif məkanda paylanmış informasiya sistemlərinin integrasiyasını tələb edir. Belə bir obyektdə misal olaraq iaşə təşkilatını göstərmək olar. Qəhvə zənciri, alınmış qida inqrediyentlərinin uçotunu və satışın uçotunu tələb edən biznesin idarə edilməsi ilə bağlı fəaliyyətdir. Bu cür mühasibat uçotu çoxlu sayda coğrafi cəhətdən paylanmış obyektlər: müəssisənin özü, təchizatçıları, işçiləri və istifadə etdikləri uçot sistemləri ilə çətinləşir.

Əsas hissə

Paylanmış informasiya sistemləri hal-hazırda geniş yayılmış və ən çox istifadə olunan müasir vasitələrdəndir. Bu sistemlər adətən böyük ölçüdə verilənlərin ötürülməsi, bazada saxlanması, üzərində əməliyyatlar aparılması üçün istifadə olunur. Paylanmış informasiya sistemlərinə qısaca olaraq PİS kimi də müraciət olunur. Bu şəbəkələrdə ixtiyari və hesablama materialları

fərqli coğrafi yerlərdə yerləşən serverlərdə saxlanılır. Müxtəlif ərazilərdə olmaqlarına baxmayaraq istifadəçilərə bütöv sistem kimi çıxış edə bilər.

Bu mövzunun aktuallığı coğrafi olaraq paylanmış istehsal müəssisələrində mühasibat uçotunun təşkilinin mürəkkəbliyindədir ki, bu da məkanda paylanmış müxtəlif informasiya sistemlərinin integrasiyasını tələb edir [1]. Zəncirli restoran və kafələrin mövzu sahəsi bu mövzuda araşdırma üçün nümunə kimi uyğundur.

Tədqiqat obyekti ikipilləli müştəri-server arxitekturasına əsaslanan, vahid məlumat bazasından istifadə etməklə şəbəkəsinin müxtəlif nöqtələrində sifarişlərin qəbulu və uçotunun təşkili problemini həll edən mərkəzləşdirilməmiş informasiya sistemidir.

Mərkəzləşdirilməmiş informasiya sistemində funksiyaların və resursların çoxlu elementlər arasında bölüşdürüldüyü, vahid idarəetmə mərkəzinin olmaması ilə xarakterizə olunan sistem adlandırılı bilər ki, qovşaqlardan birinin nasazlığı bütün sistemi göstərməsin. İşlək olmayan bu anlayışı dar mənada nəzərdən keçirsək, proqram təminatı baxımından paylanmış informasiya sistemi bir-biri ilə qarşılıqlı əlaqədə olan

proqram komponentlərinin məcmusudur. Bunların hər biri öz növbəsində ayrı bir prosesdə icra olunan proqram modulu kimi nəzərdən keçirilə bilər [4]. Məlumat mübadiləsi mərkəzləşdirilmiş qaydada həyata keçirilir. Buna görə də, mərkəzləşdirilməmiş İS element mövqeləri arasındakı əlaqənin sistemin funksional nöqtəyinə nəzərindən və buna görə də sistemin təhlili və sintezi baxımından əsas rol oynadığı bir sistemdir.

Böyük şirkətlərin inkişafı artıq adi çərçivəyə uyğun gəlmir, hər il daha mürəkkəbləşir. Bu gün yalnız bir baş ofisi olan müasir inkişaf edən bir şirkəti təsəvvür etmək çətindir. Ölkəmizin özəliyi, yanacaq-energetika kompleksləri, banklar, ictimai işə müəssisələri və s. kimi geniş ərazini tutan çoxlu sayda iri müəssisələrin olmasıdır. Hər il coğrafi əhatə dairəsi genişlənir və böyük federal korporasiyaların filialları yeni şəhərlərdə, ölkələrdə və hətta qitələrdə açılır. Bu şirkətlərə tikinti və mədənçıxarma təşkilatları, pərakəndə mağaza ticarət təşkilatları, turizm agentlikləri, bank sənayesi və s. Getdikcə daha çox şirkət təbii olaraq müasir proqram məhsulları əsasında mərkəzləşdirilməmiş informasiya kommunikasiya sistemlərini qəbul etməyə başlayır [2].

Texniki hissədə kafe və restoranların şəbəkəsi, yəni müxtəlif informasiya sistemlərindən istifadə etməklə coğrafi olaraq paylanmış istehsal müəssisələri şəbəkəsi nəzərdən keçiriləcək. Şəbəkəyə məxsus RIS-i həyata keçirmək üçün müştəri-server arxitekturasından istifadə edilməlidir. Bu arxitekturalardan istifadənin əsas səbəbi satış və məhsul balanslarının və tədarük cədvəllərinin vahid qeydlərini saxlamaq üçün mərkəzləşdirmə ehtiyacıdır. RIS aşağıdakı komponentləri olan iki səviyyəli sistem kimi təqdim edilə bilər:

- "Müştəri": "müştəri" (mühasibat proqramı modulu, aşağıda müzakirə olunacaq) vasitəsilə müəyyən bir nöqtədə (kafedə) işləyən İS istifadəçiləri (ofisiantlar, baristalar, menecerlər, mühasiblər, kassirlər və s.)

- "Server məlumat bazası": Əsas ofisdə (məlumat mərkəzində) ayrıca serverdə yerləşən bu, bütün "müştərilər" üçün vahid paylaşımıdır, yəni vahid menyular, təchizatçılar və s.

Nəzərdən keçirilən sistemdə uzaq verilənlər bazalarından istifadə statistik məlumatların aparılması və xammalın alınması və istehlakının uçotu kimi toplama əməliyyatlarının yerinə yetirilməsini sadələşdirmək üçün zəruridir [6].

Strukturları coğrafi cəhətdən səpələnmiş iş yerləri və istehsal obyektləri olan təşkilatlarda qeyri-mərkəzləşdirilmiş informasiya sistemlərinin tətbiqi məlumatların bir informasiya sistemindən digərinə əl ilə ötürülməsi zərurəti ilə əlaqədar çəkilən xərcləri və xərcləri azaldacaq [1]. Mərkəzləşdirilməmiş verilənlər bazası yaratmaq sizə bir anda bütün şəbəkəniz üzrə sorğuları toplamağa və dərhal statistika əldə etməyə imkan verəcək. Bir proqrama keçid onlayn bankçılıq, mühasibat uçotu və vergi hesabatları sistemləri, məlumat bazaları və s. kimi sistemlərdən idxal və ixrac əməliyyatlarının əl ilə yerinə yetirilməsinə ehtiyacı aradan qaldıracaq [3]. Bu, xətlərin sayını və əl əməyi xərclərini azaldacaq.

Hesablama məsələlərinin həlli üçün kompüter sisteminin resurslarından istifadəni planlaşdırmaq üçün hazırda müxtəlif üsullardan istifadə olunur. Yüksək yüklü sistemlərə fasiləsiz giriş sistemləri deyilir, yəni məlumat tələb edən strukturlar fasiləsiz iş zamanı uzun gecikmələr olmadan məlumatı qəbul etməyə imkan verir. Tipik olaraq, belə iş üçün çox server həlləri istifadə olunur [7].

Serverləri 2 növə bölmək olar:

- Front-end (son istifadəçilərlə qarşılıqlı əlaqədə olan serverlər);
- Backend (məlumatların saxlanması və giriş sistemi).

Sistem yükü yüksəkdirsə, lakin mürəkkəb keşləmə və ya məlumat kodlaşdırma modeli yoxdursa, o zaman ara proqramın rolu buraxıla bilər. Yüksək yüklü sistemlər adətən çoxlu ön, arxa serverlərdən və ya server qruplarından ibarətdir. Coğrafi cəhətdən səpələnmiş layihələr üçün ön, arxa qrupların yerləri resurs istehlakına əsasən müxtəlif məlumat mərkəzləri arasında paylana bilər.

Bulud birləşməsində serverlər prosessor resurslarını paylaşırlar. Bir və ya bir neçə maşının nasazlığı xidmətin uğursuzluğuna səbəb olmayacaq, onun yükü qalan bulud iştirakçıları arasında yenidən bölüşdürüləcək [5]. Çox vaxt proqram

təminatı çox yivli bulud hesablamaları ilə istifadə üçün xüsusi olaraq hazırlanır. Tətbiq modulları arasında qarşılıqlı əlaqə daha dərin ola bilər. Bu ssenari üçün front-end və back-end server proqramları eyni buludda yerləşə bilər.

Mərkəzləşdirilməmiş sistemlər qarşılıqlı əlaqə vasitəsilə ekosistem yalnız proqram səviyyəsində əlaqə saxlayan təcrid olunmuş server kompüterlərinin yaradılması üzərində qurulur [8]. Bu strukturun miqyası bulud hesablamalarından daha sadə və şəffafdır.

Ön, arxa serverlər arasındakı əlaqə çoxdan çox bölünür, lakin hamısı deyil, yəni bir ön server sorğuları məhdud sayda arxa serverlərə paylayır hücumun bütün uğursuzluqları zəruridir və ya uğursuzdur. Sorğu növbəsi müxtəlif geri serverlər arasında bərabər paylanır və ya çəkilir.

Tipik olaraq, yüksək yüklü sistemlər paylanmış hesablama və saxlama sistemləri ilə dizayn edilir. Əl müdaxiləsi olmadan düzgün işləməyi təmin edin. Qüsurlara dözümlülüyü artırmaq üçün biz nasazlıq halında özünü diaqnostika edən və problemləri həll edən sistemlər yaradılır. Bir nasazlıq aşkar edildikdə, hesabat modulu (bug tracker, bug logger) sazlama məlumatını dəstək xidmətinə ötürür, bu məlumatı aldıqdan sonra biz sistemi təkmilləşdiririk və yenilənmiş alqoritmi müştəriyə ötürürük [6]. Yüksək yüklü sistemlərin inkişafı müşayiət olunan sənədlərin və mənbə kodlarının müştəriyə ötürülməsi ilə həyata keçirilir. Proqram vahid test edilmişdir. Burada ən vacib məqam üçüncü tərəflərin müdaxilə etməməsidir, funksionallıq üçüncü şəxslər tərəfindən tamamlandıqda, bizə vaxtında məlumat vermək və sənədləri köçürmək lazımdır. Load Balancing paylanmış -paralel hesablama sistemindən istifadə edərək paylanmış hesablamları təşkil etmək üçün istifadə olunur.

Yük balans, paylanmış sistemin bir hissəsi olan müəyyən hesablama qovşaqlarına bərabər yüklə balanslaşdırıcıların təmin edilməsini nəzərdə tutur. Sorğuları emal etmək üçün tapşırıqlar gəldikdə, balanslaşdırıcı bu tapşırıqla bağlı hesablamaları hansı aşağı axın hesablama qovşağının yerinə yetirməli olduğuna qərar verməlidir. Bundan əlavə, balanslaşdırma hesablamaların bir hissəsinin ən çox yüklənmiş hesablama qovşaqlarından daha az yüklənmiş qovşaqlara dərhal ötürülməsini nəzərdə tutur.

Böyük hesablama gücü tələb edən vəzifələr arasında kifayət qədər müxtəlif sahələri ayırmaq olar. Müasir informasiya dünyası getdikcə daha çox əməliyyat sistemi brauzerinin müştəri kimi çıxış etdiyi internet proqramlarına keçir. Böyük hesablama gücü tələb edən tapşırığın parlaq nümunəsi sosial şəbəkə portallarıdır, məsələn, "VKontakte". Bu şəbəkənin gündəlik trafiki 90 milyondan çox insana çatır. Heç bir fayl server tipli sistem belə sayda aktiv verilənlər bazası istifadəçisini idarə edə bilməyəcək. Buna görə də, bu cür problemləri həll etmək üçün yeganə düzgün həll bu və ya digər konfigurasiyada müştəri-server platforması qurmaq üçün vasitələrdən kömək istəmək olardı [7].

Digər misal YouTube video sosial şəbəkəsidir. Gündəlik trafik yüz milyonlarla istifadəçi təşkil edir və durmadan artır. Hər dəqiqə portala minlərlə saatlıq video yüklənir. Belə portalın düzgün işləməsini təmin etmək üçün müştəri-server arxitekturasında işləyən yaxşı qurulmuş proqramla birlikdə çox böyük hesablama gücü tələb olunur.

Paylanmış informasiya sistemləri və coğrafi olaraq paylanmış strukturlar anlayışlarını birləşdirərək, ümumi informasiya sahəsi Mexanizm kompleksi vasitəsilə vahid sistemdə virtual və məntiqi birləşmənin əsasını təşkil edən coğrafi paylanmış informasiya kompleksinin tərifinə gələ bilərik. O, şirkətin faktiki biznes prosesləri və məlumat axınları nəzərə alınmaqla qurulur [3].

İş asanlaşdırmaq və mümkün olduqda insan faktorları ilə bağlı səhvləri aradan qaldırmaq üçün bütün mexanizm nəzərdən keçirilməlidir.

Bu məqsədlə mühasibat uçotu və hesabatla bağlı kompüter proqramları nəzərdə tutulmuşdur. Onlar bir baxışda bir ay, həftə, gün üçün bütün təchizatçılar, məhsullar, bütün satılan məhsullar və təchizatçılar haqqında məlumat verməyə kömək edir.

Restoranlar, alınmış qida inqrediyentlərinin uçotunu və satışın uçotunu tələb edən biznesin idarə edilməsi ilə bağlı fəaliyyətdir. Bu cür mühasibat uçotu çoxlu sayda coğrafi cəhətdən paylanmış obyektlər: müəssisənin özü, təchizatçıları və işçiləri və istifadə etdikləri uçot sistemləri ilə çətinləşir [2]. Buna görə də bu problemi həll

etmək üçün müştəri-server arxitekturasına əsaslanan paylanmış informasiya sisteminin qurulması tövsiyə olunur.

Əvvəlcə proqramdan istifadə istifadəçi və təşkilat haqqında məlumatların qurulması ilə başlayır. Bu məlumat yalnız sistem administratorları tərəfindən redaktə edilə bilər. İşdə növbəti addım alınacaq və son məhsulun istehsalı üçün istifadə olunacaq məhsulların kataloqunun ilkin doldurulmasıdır [2]. Administrator hesabı olan istifadəçilər də bu funksiyadan istifadə edə bilərlər. Əsas biznes prosesini modelləşdirək.

İnformasiya sistemlərinin qurulması kifayət qədər mürəkkəb prosesdir. İqtisadi mövzu sahəsində fəaliyyət göstərən, biznes proseslərini idarə edən və ya qeydləri aparan sistemlər verilənlər bazası texnologiyasına əsaslanır.

NoSQL texnologiyası tədricən inkişaf etsə də, relational verilənlər bazası yaratmaq texnologiyası hələ də görkəmli mövqe tutur.

Əlaqəli verilənlər bazası qurulması dünyasında bir çox məlumat formatı var. Məlumatlar sözdə DBMS tərəfindən işlənir. Buna görə də, bütün dünyada DBMS-ni yerli DBMS və sənaye DBMS-ə bölmək adətdir [1].

Müəssisə daxilində nisbətən kiçik tapşırıqları yerinə yetirmək üçün istifadə edilən yerli müəssisə DBMS. Məsələn, bu əmək haqqı, anbarda malların uçotu ola bilər. Ümumiyyətlə, mühasibat uçotu tələb edən istənilən vəzifə yerli DBMS texnologiyasından istifadə etməklə həll edilə bilər. Yerli DBMS-lərin iki əhəmiyyətli məhdudiyyəti var. Birinci hədd cədvəldə icazə verilən maksimum sahələrin sayı, cədvəldəki qeydlərin sayı, cədvəlin ölçüsü və s. Məsələn, Visual FoxPro-da cədvəldə icazə verilən qeydlərin maksimum sayı 1 milyard qeyddən çox ola bilməz və cədvəlin maksimum fayl ölçüsü 2 GB-dan çox ola bilməz. Buna görə də, yerli məlumat bazası üçün müəyyən məhdudiyyətlər var. Kiçik və orta müəssisələr üçün bir cədvəldə 1 milyard rekorda nail olmaq çox çətin, ona görə də belə məhdudiyyətlərin əhəmiyyəti yoxdur [3]. Bundan əlavə, bəzi köhnə qeydləri ayrıca verilənlər bazasında arxiv olaraq üzərinə yazmaqda arxiv nüsxələrini yarada bilərsiniz.

DBMS-nin eyni vaxtda istifadəçilərinin sayı açıq şəkildə məhduddur. Ümumiyyətlə, onların sayı bir və ya iki ondan çox ola bilməz. Şəbəkə yaxşı işlədikcə istifadəçilərin sayı onlarla artdı.

Lakin istifadəçi çox olduqda sistem yavaş işləməyə başlayır. Səbəb yerli DBMS-dən istifadə edən fayl server sistemi texnologiyasındadır.

İnkişafın növbəti mərhələsi verilənlər bazası üzərində işləyən sistemlərin qurulması üçün müştəri-server texnologiyasıdır.

Buna görə də verilənlər bazası qurulmasının DBMS arxitekturası kimi tanınan iki əsas modeli mövcuddur: kiçik sistemlərin layihələndirilməsi üçün uyğun olan fayl server modeli və müəssisə səviyyəli sistemlər üçün uyğun olan müştəri server modeli.

Fayl server sistemi aşağıdakı kimi işləyir. Proqram verilənlər bazasına sorğu göndərdikdə, bütün məlumatlar proqrama ötürülür və verilənlərin işlənməsi müştəri proqramı tərəfindən həyata keçirilir, məsələn, bir dəyərin digəri ilə toplu şəkildə dəyişdirilməsi [2]. Bu, şəbəkəyə çox yük qoyacaq. Çox sayda kompüter eyni verilənlər bazasına daxil olduqda, şəbəkə nasaz olmağa başlayır.

Müştəri-server yanaşması üçün vəziyyət fərqlidir. Müştəri proqramı sorğu şəklində məlumatları serverə göndərir. Server müştərinin sorğusunu idarə edir və ona bütün

məlumatları deyil, yalnız tələb olunan nəticələri göndərir. Beləliklə, şəbəkə trafiki yüklənməmiş qalır.

Tipik olaraq, müştəri-server texnologiyası ilə əlaqəli verilənlər bazası şəbəkədə veb server adlanan ayrıca kompüterdə yerləşir. Verilənlər bazasını emal edən proqram SQL dilindən istifadə edir, ona görə də klient-server texnologiyası çərçivəsində bu proqram çox vaxt SQL server adlanır.

Müştəri proqramı əvvəlcə SQL serverinə, DBMS-ə daxil olur, sonra isə DBMS verilənlər bazasını emal edir. Beləliklə, məlumatların təhlükəsizliyinin idarə edilməsinin mərkəzləşdirilməsinə nail olunur. Məsələn, SQL-də yazılmış və verilənlər bazasında saxlanılan cədvəllər və saxlanılan prosedurlar bütün müştərilərdə işləyəcək [1]. Müştəri-server texnologiyasından istifadə edərkən müştərilərin sayı faktiki olaraq qeyri-məhdud olur. Buna misal olaraq hər hansı bir əsas veb resursunu göstərmək olar. Məsələn, ok.ru portalına giriş sistemində eyni vaxtda işləyən minlərlə istifadəçini əhatə edə bilər. Onların sayı

yalnız DBMS-nin yerləşdiyi server klasterinin fiziki imkanları ilə məhdudlaşır.

Access kimi yerli verilənlər bazası da ayrıca şəbəkə kompüterində yerləşə bilər və müştərilər şəbəkədəki müxtəlif kompüterlərdə olacaq, lakin bu, sistemi müştəri-server sisteminə çevirmir [1]. Fayl server şəbəkəsi arxitekturasına malik bir sistem olaraq qalmağa davam edəcək.

Müştəri server sistemləri yalnız sənaye DBMS-dən ibarətdir.

Tətbiq arxitekturasının seçimini və verilənlər bazası ilə qarşılıqlı əlaqə metodunu tamamlamadan əvvəl onların müsbət və mənfi tərəflərini, mümkün riskləri və təhdidləri nəzərə almalısınız. Məsələn, verilənlər bazasına fiziki giriş itirilsə restoranlar şəbəkəsi necə işləyəcək?

Bu hadisə sifarişləri qəbul edə bilməmək, sifarişləri qeyd etmək və fəaliyyətin dayandırılması ilə nəticələnəcək. Bu riski aradan qaldırmaq üçün iki variant nəzərdə tutula bilər: müvəqqəti “kağız” uçotun tətbiqi və ya verilənlər bazasının lokal sürətinin saxlanması və onun paylaşılan serverlə sinxronlaşdırılması. Bununla belə, zəncirin mərkəzləşdirilmiş anbarında balansları nəzərə almaq zərurəti ilə müştəri-server həllərindən tamamilə imtina etmək və avtonom həllərə keçid mümkün deyil. Buna görə də, “server nasazlığı” hadisəsi baş verdikdə, məlumat bazasının yerli nüsxəsini müvəqqəti olaraq gətirməklə və sonradan uzaq verilənlər bazası ilə sinxronizasiya etməklə fəaliyyətlər davam etdirilə bilər [3]. Müştəri modulunun uğursuzluğu ehtimalı azdır, çünki hər bir sayt birdən çox fərdi kompüterlə təchiz edilmişdir və bununla da dublikasiya yaranır. Elektrik enerjisinin kəsilməsi işıqlandırmanın itirilməsi, sobaların dayanması səbəbindən bütün obyektin tam dayanmasına səbəb olacaq ki, bu da istehsalı mümkün edəcək və bu barədə ətraflı düşünməyin mənası yoxdur.

Müştəri-server qarşılıqlı əlaqə modeli və arxitekturasının üstünlüyü ondan ibarətdir ki, müştəri tətbiqinin mənbə kodu və server tətbiqi ayrılır. Üstəlik, bu model müştəri maşınında daha aşağı aparat resurs tələblərinə malikdir, çünki əksər hesablama əməliyyatları serverdə yerinə yetiriləcək və müştəri-server arxitekturası çox çevikdir.

Nəzərə alınan arxitekturanın çatışmazlıqlarına server avadanlığının dəyərinin müştəri avadanlığının qiymətindən əhəmiyyətli dərəcədə yüksək olması daxildir. Serverlər xüsusi təlim keçmiş və təlim keçmiş işçilər tərəfindən saxlanılmalıdır. Şəbəkədəki server sıradan çıxsa, müştərilər onunla əlaqə qura bilməyəcək və bu, fəaliyyətlərin yerinə yetirilməsini çətinləşdirəcək. Hal-hazırda, bir çox növ proqram inkişaf etdirmə alətləri var, onların bir çoxu çox oxşar funksiyaları yerinə yetirir. Xüsusi proqramlaşdırma dili və ya inkişaf mühiti müəyyən bir problemi həll etmək üçün nəzərdə tutulmuşdur. Hazırda belə nəticəyə gəlmək olar ki, inkişaf mühitinin seçimi artıq həll olunacaq problemdən asılı deyil, proqramçının konkret dilə üstünlük verməsindən getdikcə daha çox asılıdır [2]. Bu, inkişaf sistemlərinin artan universallığı və inkişafı ilə bağlıdır.

MySQL RAM-da saxlanacaq buferin ölçüsünü təyin etmək imkanına malikdir. Əgər RAM ölçüsü kitabxananın ölçüsündən böyükdürsə, bütün kitabxana RAM-a sığacaq və daha sürətli miqyasda işləyəcək.

Üstəlik, belə bir DBMS-nin verilənlər bazasında qeydlərin miqdarı artarsa, problem köhnə ilk qeydləri xronoloji ardıcılıqla arxivləşdirmək və verilənlər bazasını təmizləmək, sonra isə istifadəçilərin sayı və istifadəçilərin sayı kimi həll edilə bilər [3]. DBMS-də Təmizləmə yükü artıqca bu məsələ aktual olaraq qalır. Sənaye DBMS-ləri yüklə bağlı problemləri həll etmək üçün nəzərdə tutulmuşdur.

Masaüstü verilənlər bazası sistemləri arasında təqdimata layiq olanlar dBase və Visual dBase, Microsoft Access, Microsoft FoxPro və Visual FoxPro və Paradoxdur.

MySQL verilənlər bazasını idarə etməyin klassik metodu, o cümlədən verilənlər bazasının özünün yaradılması, verilənlər cədvəllərinin və onlar arasında əlaqənin yaradılması, cədvəllərin doldurulması və digər əməliyyatların hamısı adi Windows-a bənzər bir əmr pəncərəsi şəklində müştəri proqramı vasitəsilə həyata keçirilir.

Hal-hazırda, Access-də olduğu kimi MySQL verilənlər bazalarını qrafik şəkildə idarə edən ki-fayət qədər çox sayda müştəri proqramı var. Təbii ki, burada görmənin formalaşma mexanizmi tamamilə fərqlidir.

Hər bir ümumi təyinatlı dilin özünəməxsus üstünlükləri və mənfi cəhətləri, həmçinin xüsusi tətbiq sahələri var. Ancaq hazırda bütün dillər kifayət qədər inkişaf etdirilib və standartlaşdırılıb ki, bu və ya digər dilin seçimi müəyyən bir proqramçının üstünlük və ya bacarıqlarından daha çox asılıdır, nəinki təyin olunmuş tapşırıqdan. Birgə inkişaf həm də dil seçiminə haqq qazandırır. Bir proqramçı komandanın proqram təminatı üzərində işləyən konkret bir iş tapdıqda, o, komandanın qalan hissəsinin işlədiyi dildə kod yazmağa məcbur olur [2].

Nəticə

MySQL DBMS, JAVA 12 proqramlaşdırma dili və hazır kitabxanalar dəsti (məs. verilənlər bazalarına qoşulmaq üçün JDBC, istifadəçi interfeyslərini yaratmaq üçün SWING). Hazırlanmış sistem müştəri-server arxitekturasına malikdir və biznes proseslərinin məhdud siyahısı və gələcəkdə gözlənilən dəyişiklikləri olmayan coğrafi cəhətdən paylanmış obyektlərdə mühasibat uçotunun təşkili üçün nəzərdə tutulub. Bu halda, konkret həlli inteqrasiya etməkdənsə, konkret müəssisənin ehtiyaclarını hədəfləyən proqram təminatının hazırlanması iqtisadi mənə daşıyır. Nəzərdən keçirilən sistemdə uzaq verilənlər bazalarından istifadə statistikanın aparılması, xammalın alınması və istehlakının uçotu, sonda məlumatlar mərkəzi məlumat bazasına ötürüldükdə statistik məlumatların yaradılması və satışın aparılması kimi ümumiləşdirmə

əməliyyatlarının yerinə yetirilməsini sadələşdirmək üçün zəruridir. İnkişafdan əvvəl mövzu sahəsi tədqiq edilmiş, biznes proseslərinin elektron çertyojları hazırlanmış, informasiya modelləri və məntiqi verilənlər bazası modelləri hazırlanmışdır.

ƏDƏBİYYAT SİYAHISI:

1. Golitsyna, O.L. Alqoritmlərin və Proqramlaşdırmanın Əsasları: Dərslik/O.L.Golitsina, I.I. - M.: Forum, 2-ci nəşr, 2015.
2. Korobov, N.A. Ticarətdə İnformasiya Texnologiyaları/N.A.Korobov, A.Yu. - M.: Akademiya, 2016.
3. Selko, Peşəkarlar üçün Joe SQL. Proqramlaşdırma / Joe Selko. - M.: LORI, 2015.
4. Ben, Forta SQL / Forta Ben 10 Dəqiqədə. - M.: Dialektika/Williams, 2015.
5. Bewley, Alan Learning SQL / Alan Bewley. - M.: Simvol Plus, 2014.
6. Van, Tassel D. Proqramlaşdırma üslubu, inkişafı, səmərəliliyi, sazlanması və sınaqdan keçirilməsi / Van Tassel D. - M.: Mir, 2017. - 332 səh.
7. Vdovenko, L. A. Müəssisə İnformasiya Sistemləri: Uh. Kənd/L.A.Vdovenko-2-ci nəşr, tərcümə. və əlavə - M.: Vuzovskiy uch. / La Verdo Liberal İncəsənət. - Moskva: Maşınqayırma, 2016. -143 s.

Зейнеб Арзу БИННАДАЛИЕВА

Магистрант Западно-Каспийского университета

E-mail: binnedeliyevazeyneb@gmail.com

ИНФОРМАЦИЯ, РАСПРОСТРАНЯЕМАЯ ДЛЯ ОБЪЕКТОВ ОБЩЕСТВЕННОГО РАЗМЕЩЕНИЯ НАСТРОЙКА СИСТЕМЫ

Резюме

В статье говорится о том, как обеспечить эффективное использование информационных систем в общественных учреждениях. Подчеркивается возможность управления онлайн-заказами клиентов в кафе и ресторанах. Там показано, как реализовать указанную сумму в электронном виде. Показано, как с помощью базы данных найти необходимую информацию в кратчайшие сроки.

Ключевые слова: распределенные системы, база данных, поиск, программное обеспечение, структурирование, электронное оборудование.



Zeyneb Arzu BINNEDALIYEVA
Master's student at Western Caspian University
E-mail: binnedaliyevazeyneb@gmail.com

ESTABLISHMENT OF DISTRIBUTED INFORMATION SYSTEMS FOR PUBLIC CATERING FACILITIES

Summary

The article discusses how to ensure the efficient use of information systems in public facilities. It emphasizes the possibility of managing online ordering of customers in cafes and restaurants. It shows how to implement the specified amount in electronic form. It demonstrates how to find the necessary information in the shortest possible time using a database.

Keywords: distributed systems, database, search, software, structuring, electronic equipment.

Daxil olub: 20.06.2025