

7-Mavzu.Ma'lumotlarni obyektli bog'lash texnologiyasi

Mijoz - server tizimi uchun ishlangan tatbiqiy komponentlarning SQL - serverlar ko'rinishidagi axborot tizimi yadrosi bilan o'zaro ta'sirini unifikatsiyalash, shaxsiy MBBT boshqaruvidagi tarqoq ma'lumotlar bazasini murakkab detsentralizatsiyalangan getrogen taqsimlangan tizimlarga integratsiyalash uchun ham o'xshash yechimlarni ishlab chiqishga imkon yaratdi. Bunday yondashish *ma'lumotlarni obyektli bog'lash* nomini olgan.

Tor ma'noda, ma'lumotlarni obyektli bog'lash texnologiyasi bir foydalanuvchi tomonidan tashkil qilingan bitta lokal bazaning boshqa foydalanuvchi tomonidan tashkil etilgan va eksploatatsiya qilinuvchi, balkim boshqa hisoblash qurilmasidagi lokal baza ma'lumotlaridan foydalanishni ta'minlash masalasini yechadi.

Ushbu masalaning yechimi "*ma'lumotlardan foydalanish obyektleri*" - dBase texnologiyasini zamonaviy shaxsiy MBBT texnologiyalari (MS Access, MS FoxPro va h.) tomonidan madadlashga asoslangan.

Ma'lumotlardan foydalanish obyekt

Ta'kidlash lozimki, obyekt deganda obyektga mo'ljallangan dasturlash va zamonaviy obyektga moijallangan operatsion muhit asoslanuvchi ma'lumotlar va ularni ishlash usullarining bir butunga (obyektga) integratsiyasi tushuniladi.

Boshqacha aytganda, "*ma'lumotlardan foydalanish obyekt*" texnologiyasini madadlovchi MBBT lokal bazalarga boshqa fayllardagi, balkim boshqa hisoblash qurilmalaridagi va boshqa MBBT boshqaruvidagi ma'lumotlardan foydalanish obyektlarini kiritish va ishlatish imkoniyatiga ega bo'ladi.

Texnik nuqtayi nazaridan, "*ma'lumotlardan foydalanish obyektleri*" texnologiyasi yuqorida aytib o'tilgan *ODBCprotokoliga* asoslangan. ODBC protokoli nafaqat mijoz - server tizimlaridagi SQL serverlari ma'lumotlaridan *foydalanish standarti* sifatida, balki relyatsion MBBT boshqaruvidagi har qanday ma'lumotlardan foydalanish standarti sifatida qabul qilingan.

ODBC drayverlari

ODBC protokoli asosida ma'lumotlardan bevosita foydalanish uchun *ODBC drayverlari* deb ataluvchi ma'lumotlar bo'lgan qurilmalarda initsializatsiyalanuvchi maxsus dasturiy komponentlar yoki boshqaruvida tashqi ma'lumotlar bazasi yaratilgan va eksploatatsiya qilinayotgan MBBTning initsializatsiyalanuvchi yadrolari ishlatiladi. Obyektli bog'lash asosida tashqi ma'lumotlar bazalaridan foydalanish prinsipi va xususiyati sxema ko'rinishida keltirilgan.

Zamonaviy shaxsiy MBBT awalo. "*o'zining*" formatidagi tashqi ma'lumotlar bazasidagi *obyektlardan* (jadvallardan, so'rovlardan, shakllardan) *to'g'ridan-to'g'ri foydalanish* imkoniyatini ta'minlaydi. Boshqacha aytganda, ishlashning joriy seansida tashkil etilgan ma'lumotlar bazasiga foydalanuvchi maxsus havola - obyektlarini kiritishi va boshqa (tashqi, ya'ni ushbu seansda maxsus tashkil etilmagan) ma'lumotlar bazasi bilan ishiash imkoniyatiga ega. *Joriy ma'lumotlar bazasiga kiritilgan tashqi ma'lumotlar bazasi obyektleri bog'langan deb ataladi* va

odatda, ichki obyektlardan farqlash maqsadida maxsus belgiga ega. Ta'kidlash lozimki, ma'umotlarning o'zi joriy ma'lumotlar bazasi fayliga (fayllariga) sig'maydi va "o'zining" ma'lumotlar bazasi fayllarida qoladi. Joriy ma'lumotlar bazasining tizimli katalogiga foydalanish uchun bog'langan obyektlar xususidagi barcha zaruriy ma'lumotlar joylashtiriladi.

Bu ma'lumotlar - bog'langan obyektlarning ichki nomi va tashqi nomi, ya'ni tashqi ma'lumotlar bazasidagi obyektning haqiqiy nomi, tashqi baza fayliga to'liq yo'l va hokazo.

Bog'langan obyektlar

Foydalanuvchi uchun *bog'langan obyektlar* ichki obyektlardan hech nimasi bilan ham farqlanmaydi. Foydalanuvchi tashqi bog'langan bazalarda ma'lumotlar jadvalini tashkil etishni, qidiruvni, ma'lumotlarni o'zgartirishni, yo'q qilishni va qo'shishni, btmday jadvallar bo'yicha so'rovlarni tuzishni va h. bajarishi mumkin. Bog'langan obyektlarni ichki ma'lumotlar bazasiga integratsiyalash, ya'ni ichki va tashqi jadvallar orasida bog'lanishni o'rnatish mumkin.

Texnik nuqtayi nazaridan, tashqi ma'lumotlar bazasining bog'langan obyektlari bilan ishlash joriy ma'umotlar bazasining ma'lumotlar bilan ishlashidan unchalik farq qilmaydi.

Joriy ma'lumotlar bazasining tizimli katalogi bo'yicha bog'langan obyekt ma'lumotlariga murojaat qilinganda MBBT yadrosi tashqi ma'lumotlar bazasining mos faylining (fayllarining) makoni va boshqa parametrlari xususidagi ma'lumotlarni topadi va foydalanuvchiga bildirmasdan ushbu faylni (fayllarni) ochadi. So'ngra ma'lumotlardan bevosita foydalanish va ularni manipulyatsiyalash maqsadida oddiy tartib bo'yicha asosiy xotirada ma'umotlarning tashqi fayli sahifalarini buferlashni tashkil etadi.

Ma'lumotlarning kollektiv ishlanishi

Ta'kidlash lozimki, zamonaviy operatsion tizimlari ma'lumotlari fayllari bilan ko'pchilik foydalanuvchi rejim imkoniyatlari bilan ishlash asosida tashqi ma'lumotlar bazasi bilan (agar u boshqa hisoblash qurilmasida bo'lsa) *shu vaqtning o'zida boshqa foydalanuvchi ishlashi mumkin*. Bu esa umumiy taqsimlangan ma'lumotlarning kollektiv ishlanishini ta'minlaydi.

Bog'langan jadvallarda ma'umotlar hajmining ortishi bilan taqsimlangan tizimlarni qurishning bunday prinsipining tarmoq grafigining jiddiy oshishiga olib kelishini sezish qiyin emas, chunki tarmoq bo'yicha doimo, hatto ma'umotlar nabori emas, balki ma'lumotlar bazasi fayllarining sahifalari uzatiladi.

Bu esa, o'z navbatida, tarmoqning avj yuklanishiga olib keladi. Shu sababli, taqdim etilgan o'zaro bog'langan obyektli lokal ma'lumotlar bazasining sxemalari ishlab chiqarish - texnologik va tashkiliy jarayonlarga bog'lik axborot texnologiyalariga tayangan holda lokal bazalari orasidagi tarmoqdagi ma'lumotlar oqimining jadalligi, yo'nalganligi nuqtayi nazaridan keyingi sinchiklab o'rganib chiqishni talab etadi.

Ma'lumotlar xavfsizligi

Ma'lumotlar xavfsizligini va yaxlitligini ta'minlashning ishonchli mexanizmlarining mavjud emasligi ham jiddiy muammo hisoblanadi. Fayl serveri modelidagidek bir necha foydalanuvchilarning bir xil ma'lumotlar bilan birgalikda ishlashi, faqat operatsion tizimning bir necha ilovalar faylidan bir vaqtda foydalanish bo'yicha funksiyalari yordamida ta'minlanadi. Xuddi shu tarzda keng tarqalgan Fox Pro, dBASE kabi boshqa MBBTning ma'lumotlar bazasidagi ma'lumotlardan hamda jadval ma'lumotlaridan foydalanish ta'minlanadi. Bunda foydalanish bevosita MBBT yadrosi yordamida hamda odatda, MBBT komplekti tarkibiga kiruvchi maxsus qo'shimcha ISAM (Indexed Sequential Access Method) drayverlar yordamida ta'minlanadi.

Bunday yondashish shu tariqa qurilgan taqsimlangan getrogen tizimlarining ko'p protokolligini, ya'ni local ma'lumotlar bazalarini madadlovchi MBBT turlarining "xilma-xilligini" amalga oshiradi. Ammo obyektli bog'lush, amalga oshirilishi va madadlanishi muayyan MBBTning o'ziga xos xususiyatiga bog'liq ma'lumotlar bazasining boshqa obyektlarini (so'rovlar, shakllar, hisobotlar) istisno qilgan holda, faqat bevosita ma'lumotlar jadvali bilan chegaralanadi.

Boshqa MBBT bazalaridan foydalanish masofadagi ma'lumotlar joylashgan hisoblash qurilmalarida o'matiluvchi va bajariluvchi ODBC drayverlarining texnikasi orqali amalga oshiriladi. Bu holda "g'oya" quyidagicha: lokal ma'lumotlar bazasini madadlovchi shaxsiy MBBT tarkibiga ODBC drayveri deb ataluvchi qo'shimcha dasturiy komponentni o'rnatish mumkin.

O'rnatiluvchi ODBC drayveri operatsion tizim katalogining maxsus qism katalogida "ro'yxatga olinadi". Shu tariqa ODBC ma'lumotlari manбайдan to'g'ridan — to'g'ri foydalanishning ishchi zonasi hosil qilinadi.

ODBC protokoli

ODBC ma'lumotlari manбайдan bevosita foydalanish uchun MBBT yadrosi ichki lokal ma'lumotlar bazasining tizimli katalogi bo'yicha manba joylashgan joyni aniqlaydi, ilovalarning o'zaro ta'siri protokoli (API) bo'yicha hisoblash qurilmasida ODBC drayverining masofadagi ma'lumotlar ini chaqiradi ODBC protokoli bo'yicha ma'lumotlardan foydalanish va ishlashi uchun SQL - yo'riqnomalarni yoqlaydi.

Bunday foydalanish rejimi qator parametrlar orqali tartibga solinadi (muolajalarni chaqirish intervali, so'rov ishlanishining maksimal vaqti, so'rovlar bo'yicha shakllantiriluvchi ma'lumotlar naboridan tarmoq bo'yicha bir marta jo'natiluvchi yozuvlar soni, yozuvlarni blokirovka qilish vaqti va h.).

Ushbu parametrlar ODBCning mos drayverini o'rnatishda va ro'yxatga olishda operatsion tizimning maxsus reestriga yoziladi. Bunday yondashishda har bir lokal MBBT, tashqaridan (boshqa hisoblash qurilmalaridan) "uning" ma'lumotlar fayli ma'lumotlaridan foydalanishga murojaat qilinganida o'zining hisoblash qurilmasida SQL - server rolini, ya'ni ma'lumotlar mashinasi rolini bajaradi. Bu holda ma'lumotlarni bevosita ishlashni "o'zining" ma'lumotlar faylining mantiqiy va fizik strukturasi xususiyatlarini biluvchi MBBT tomonidan amalga oshirilishi

odatda, ishlashning yuqori samaradorligini ta'minlaydi, eng asosiysi esa, ma'lumotlar manbaining predmet sohasi mantiqi bo'yicha ma'lumotlar yaxliiligiga cheklashlar bajariladi.

“Raxnalar”

Ma'lumotlarni himoyalash va foydalanishni cheklash tizimlarida “raxnalar”ning paydo bo‘lishi, obyektli bog‘lash texnologiyasining ma'lum muammosi hisoblanadi. Ma'lumotlardan foydalanish muolajalarini bajarish uchun ODBC drayverlarining chaqiriqlari tarkibida, yondan, fayllar va so‘raluvchi obyektlar (jadvallar)dan tashqari (agar mos bazalar himoyalangan bo‘lsa) ochiq holdagi foydalanish parollari boiadi, natijada foydalanish va ma'lumotlarni himoyalash tizimlari tahlillanishi va o`chirilishi mumkin

Dasturni rasmiylashtirish uslubi

Texnologiklik nuqtayi nazaridan dasturning ham muallif, ham uni tekshirishi yoki modifikatsiyalashi ehtimoli bor boshqa dasturchilar tomonidan qabul qilinishini yengillashtiruvchi dasturni rasmiylashtirish uslubi yaxshi hisoblanadi. «Yodda tuting, dasturlarni odamlar o‘qishadi» deya uqtirgandi dasturlash muammolariga bag‘ishlangan mashhur monografiyalardan birining muallifi D.Van Tissel.

Aynan har qanday dasturni bir necha marta qarab chiqishga to‘g‘ri kelishidan kelib chiqqan holda dasturlarni yozishning yaxshi uslubiga rioya qilish lozim.

Dasturni rasmiylashtirish uslubi quyidagilarni o‘z ichiga oladi:

- dastur obyektlarini (o‘zgaruvchanlarni, funksiyalarni, tiplar, ma'lumotlar va hokazolarni) nomlash qoidalari;
- modullarni rasmiylashtirish qoidalari;
- modullar matnlarini rasmiylashtirish uslubi.

Dastur obyektlarini nomlash qoidalari

Dasturiy obyektlar nomlarini tanlashda quyidagi qoidalarga rioya qilish lozim:

- obyekt nomi uning mazmuniga muvofiq bo'lishi shart, masalan:

Maxitem — maksimal element;

Nextitem — keyingi element;

- agar dasturlash tili imkon bersa bir necha so‘zdan iborat nomlarni vizual ajratish uchun «—« timsolidan foydalanish mumkin, masalan:

Max_item, Next_jtfem

- yozilishi bo‘yicha yaqin nomlardan qochish zarur, masalan: Index va Indec

Modullarni rasmiylashtirish qoidalari

Har bir modul minimum quyidagilarni ichiga olgan sarlavha bilan shaylantirilishi shart:

- modul nomi;
- uning vazifasining qisqacha bayoni;
- o‘lchov birliklari ko‘rsatilgan holda kirish, chiqish parametrlarining qisqacha bayoni;

- foydalaniluvchi (chaqiriluvchi) modullar ro'yxati;
- algoritm (usul) yoki cheklovlar qisqacha bayoni;
- dastur muallifi ism-sharifi;
- identifikatsiyalovchi axborot (versiya raqami yoki so'nggi to'g'rilash sanasi).

Algoritm bayoni

- kesimlar jamg'arish usuli bilan umumlashtiriladi, $n > 0$;
- ma'naviy qaram, ya'ni muayyan mashina tipiga yo'naltirilgan va skalyar kodlar optimallashtirishini mashinaviy buyruqlar darajasida bajariladi, masalan, ortiqcha yuborishlarni olib tanlash, yanada samarali buyruqlardan foydalanish va hokazo;
- mashinaviy-mustaqil vositalar optimallashtirishini kirish tili darajasida bajariladi, masalan sikldan konstant (sikl indeksiga bog'liq bo'lmagan) ifodalarning hisoblanishlarini chiqarish va hokazo.

Tabiiyki, kompilyator ishiga aralashish mumkin emas, biroq buyruqlar darajasida dasturni optimallashtirishning ko'plab imkoniyatlari mavjud.

Xotirani tejash usullari

Xotirani tejash bo'yicha choralar qabul qilish qandaydir holatlarda ushbu xotiradan tejamsiz foydalanilganini nazarda tutadi. Faqat samaradorlik tavsifiga salmoqli ta'sir ko'rsatuvchi ma'lumotlar joylashuvi operatsiyalarning tahlili ma'noga egaligini hisobga olgan holda xotirani tuzilmaviy tiplarning (massivlar, yozuvlar, obyektlar va hokazolarning) ma'lumotlarga ajratishga alohida e'tibor qaratish lozim.

Xotiradan foydalanishga cheklovlar mavjudligida eng avvalo qayta ishlash jarayonida tuzilmaviy tiplarning boshlang'ich ma'lumotlari dubllashtirishni talab qilmaydigan qayta ishlash algoritmlarini tanlash lozim. Yuklatilgan massivda operatsiyani bajaruvchi massivlarni xillash algoritmlari, masalan yaxshi ma'lum <<pufakcha>> usuli bilan xillashtirish misol tariqasida xizmat qilishi mumkin.

Agar dasturda cheklangan vaqtdan foydalanuvchi katta massivlar zarur bo'lsa, bu holda ularni dinamik xotiraga joylashtirish va qayta ishlash yakunlangandan so'ng yo'qotib tashlash mumkin. Yana shuni esda tutish lozimki, tuzilmaviy ma'lumotlarni tagdasturga <<vazifasi bo'yicha>> berishda ushbu ma'lumotlarning nusxalari stekda joylashadi. Agar ma'lumotlarni <<murojaat bo'yicha>> biroq o'zgarmas (const bayonida) berilsa, ba'zan nusxalanishdan saqlanishga erishiladi. Oxirgi holatda stekda faqat ma'lumotlar manzili joylashadi

Bajarish vaqtini kamaytirish usullari

Yuqorida avval eslatib o'tilganidek, bajarish vaqtini kamaytirish uchun birinchi navbatda dasturning takrorlanishlar ko'p miqdoriga ega siklik uchastkalarini tahlil qilish lozim, Ularni yozishda imkoniyatga ko'ra quyidagilar zarur:

- sikllardan konstant, ya'ni sikl parametrlariga bog'liq bo'lmagan ifodalarning hisoblanishini chiqarish;

- qo'shish, ayirish va surishlar bilan almashtirgan holda ko'paytirish hamda bo'lishning «uzun» operatsiyalaridan qochish;
- tiplarning ifodalarda qayta tuzilishini minimallashtirish;
- shartli ifodalar yozuvni optimallashtirish, ortiqcha tekshiruvlarni chiqarib tashlash;
- indekslar bo'yicha massivlar elementlariga ko'p martalik murojaatlarni (ayniqsa, ko'p o'lchamlarini, chunki element manzili hisoblab chiqarilishida indekslar qiymatiga ko'paytirish operatsiyalaridan foydalaniladi) chiqarib tashlash — xotiradan massiv elementini birinchi marta o'qib, uni skalyar o'zgaruvchanda eslab qolish va kerakli joylarda qo'llash lozim;

«Xatolardan himoyali» dasturlash

Dastur kompilyatsiyasi va jamlashtirilishi bosqichlarida aniqlanilmaydigan dasturlash xatolaridan istalgani oxir-oqibatda uch usul bilan namoyon bo'lishi mumkin: xato haqida tizimiy axborot berilishiga, kompyuterning osilib qolishiga va noto'g'ri natijalar olinishiga olib keladi.

Biroq dastur ishining natijasi muhim bo'lgunga qadar xatolar odatda ko'p marotaba noto'g'ri oraliq natijalar, noto'g'ri boshqaruvchi o'zgaruvchanlar, noto'g'ri ma'lumotlar tiplari, ma'lumotlar tuzilmalarining indeksleri va hokazolar tarzida namoyon bo'ladi. Bu esa xatolarni ular og'ir oqibatlarga olib kelgunga qadar aniqlanishga va zararlantirishga urinib ko'rish mumkinligini bildiradi. Xatolarni erta aniqlash va zararsizlantirishning maxsus yo'llari qo'llaniladigan dasturlashni himoyali yoki xatolardan himoyali dasturlash deb nomlanadi. Undan foydalanilganda noto'g'ri natijalar olish ehtimoli salmoqli ravishda kamayadi.

Ichki va tashqi xatolar

Xatolarni va ularning mumkin bo'lgan erta ko'rinishlarini batafsil tahlili shuni ko'rsatadiki, quyidagilar tekshirilishi maqsadga muvofiqdir:

- kiritish-chiqarish operatsiyalari bajarilishining to'g'riligi;
- oraliq natijalarning (boshqaruvchi o'zgaruvchanlar ifodalarining, ma'lumotlar indeksleri; tiplari ifodalarining, raqamli argumentlar ifodalarining va hokazolarning) imkonliligi.

Boshlang'ich ma'lumotlarni noto'g'ri belgilashning sabablari ham ichki xatolar — kiritish-chiqarish yoki dasturiy ta'minlash qurilmalarining xatolari, ham tashqi xatolar — foydalanuvchining xatolari bo'lishi mumkin. Bunda quyidagilarni farqlash qabul qilingan:

- *berish xatolari* — apparat vositalar, masalan nosozlik oqibatida ma'lumotlarni buzadi;
- *qayta tuzish xatolari* — dastur boshlang'ich ma'lumotlarni kirish formatidan ichki formatga noto'g'ri qayta tuzadi;
- *qayta yozuv xatolari* — foydalanuvchi ma'lumotlarni kiritishda yanglishadi, masalan ortiqcha yoki boshqa timsolni kiritadi;
- *ma'lumotlar xatolari* — foydalanuvchi noto'g'ri ma'lumotlarni kiritadi.

Uzatish xatolari odatda apparatli nazorat qilinadi

Qayta tizimdagi xatolardan himoyalaniş uchun kiritishdan soʻng maʼlumotlar odatda darhol foydalanuvchiga namoyish qilinadi («aks-sado»). Bunda dastlab ichki formatga qayta tuzish, soʻngra teskarisi bajariladi. Biroq mazkur bosqichda qayta tuzishning barcha xatolarini bartaraf etish odatda oʻta murakkab, shu boisdan dasturning tegishli fragmentlarini ekvivalent parchalash va chegara ifodalari usullaridan foydalanilgan holda batafsil testlashtiriladi.

Notoʻgʻri maʼlumotlarni odatda faqat foydalanuvchi aniqlashi mumkin.

Oraliq natijalarning imkoniligi tekshiruvi

Oraliq natijalarning tekshiruvlari nafaqat maʼlumotlarni notoʻgʻri belgilash xatolarining, balki kodlashtirish va loyihalashdagi ayrim xatolarning kechikkan namoyoni ehtimolini kamaytirishga ham imkon beradi. Bunday tekshiruv mumkin boʻlishi uchun dasturda har qanday kelib chiqishga ega, masalan modellashtirilayotgan jarayonlar mohiyati bilan

bogʻliq cheklovlar mavjud oʻzgaruvchanlardan foydalanilishi zarur.

Biroq shuni nazarda tutish lozimki, dasturda har qanday qoʻshimcha operatsiyalar qoʻshimcha resurslardan (vaqt, xotira va hokazo resurslardan) foydalanishni talab qiladi hamda ularda ham xatolar boʻlishi mumkin. Shu boisdan barcha oraliq natijalarni emas, faqat tekshiruv maqsadga muvofiqlarini, yaʼni xatoni aniqlashga imkon berish ehtimoli bor va murakkab boʻlmaganlarini tekshirish maʼno kasb etadi. Masalan:

- agar massiv elementining indeksi qandaydir tarzda hisoblanilsa, bu indeks imkonli ekanligini tekshirish lozim;
- takrorlanishlari miqdori oʻzgaruvchan ifodasi bilan belgilanuvchi sikl barpo etilayotgan boʻlsa, ushbu oʻzgaruvchan ifodasi salbiy emasligiga ishonch hosil qilish maqsadga muvofiqdir;
- agar qandaydir hodisaning ehtimolligi belgilanayotgan boʻlsa, olingan ifoda 1 dan koʻp emasligini, barcha ehtimolli mustaqil hodisalarning ehtimolligi umumlashmasi 1 ga tengligini va hokazolarni tekshirish maqsadga muvofiq.

Xatoliklar jamgʻarilishini bartaraf etish

Hisoblashlar natijalarining xatolarini kamaytirish uchun quyidagi tavsiyalarga rioya qilish zarur:

- yaqin raqamlarni ayirishdan saqlanish lozim (mashinali nol);
- katta raqamlarni kichiklariga boʻlishdan saqlanish kerak;
- davomiyligi uzun raqamlarni qoʻshishni mutlaq kattalik boʻyicha kichiklaridan boshlash zarur;
- imkon qadar operatsiyalar miqdorini kamaytirishga intilish darkor;
- qusurlar baholanishlari maʼlum usullardan foydalanishi joiz;
- haqiqiy raqamlar tengligi shartidan foydalanmaslik lozim;
- hisoblashlarni ikki yoqlama aniqlik bilan amalga oshirish, natijalarni esa yagona aniqlikda berish shart.

Istisnolarning qayta ishlanishi

Ma'lumotlarni kiritishda va hisoblashlar jarayonida to'liq nazorat qilish, qoidaga ko'ra, imkondan hosil bo'lganligi bois avariya vaziyatlar qayta ishlanishini tutib olish ko'zda tutilishi lozim.

Apparatli va dasturli qayd etiluvchi xatolarni tutish hamda qayta ishlash uchun ayrim dasturlash tillarida, masalan Delphi, Pascal, C++, Java da istisnolar qayta ishlanishi vositalari ko'zda tutilgan. Ushbu vositalardan foydalanish foydalanuvchiga hech narsani anglatmaydigan, dasturning avariya yakunlanishi haqidagi axborotning unga berilishiga yo'l qo'ymaslikka imkon yaratadi. Buning o'rniga dasturchi mazkur xatoni tuzatishga imkon beruvchi amallarni ko'zda tutish imkoniyatini yoki, agar buning iloji yo'q bo'lsa, foydalanuvchiga vaziyatning aniq bayoni bor axborotni berish hamda ishni davom ettirish imkoniyatini oladi.

Oralama tuzilmaviy nazorat

Oralama tuzilmaviy nazorat ishlab chiqish jarayonida xatolarni imkon qadar yanada erta aniqlashni ta'minlashga imkon beruvchi nazorat texnologik operatsiyalarining jamlanmasini ifoda etadi. Nomdagi «oralama» atamasi ishlab chiqishning barcha bosqichlarida nazorat bajarilishini aks ettiradi. «Tuzilmaviy» atamasi har bir bosqichda nazorat qiluvchi operatsiyalar bajarilishi bo'yicha aniq tavsiyalar mavjudligini anglatadi. Oralama tuzilmaviy nazorat maxsus nazorat sessiyalarida bajarilishi shart, zero ularda ishlab chiquvchilardan tashqari maxsus taklif etilgan ekspertlar qatnashishlari mumkin. Sessiyalar o'rtasidagi vaqt sessiyaga chiqariladigan material hajmini belgilaydi: xususiy sessiyalarda materialni o'rtacha porsiyalarda, kamyob sessiyalarda esa salmoqli fragmentlarda qarab chiqiladi. Navbatdagi sessiya uchun materiallar qatnashchilarga ularni oldindan o'ylab chiqishlari uchun avvalroq berilishi shart. Dastlabki sessiyalardan biri ixtisoslashtirishlar belgilanishi bosqichida tashkil etilishi darkor. Ushbu sessiyada ixtisoslashtirishlarning to'liqligi va aniqligi tekshiriladi, bunda dasturiy ta'minot ixtisoslashtirishlari qanchalik to'g'ri va to'liq belgilanganligining aniqlashga qodir buyurtmachi yoki predmetli soha bo'yicha mutaxassis qatnashishi maqsadga muvofiq.

Konseptual qarorlarni tahlil qilish

Qismlar bo'yicha qo'lda loyihalash bosqichida ishlab chiqilayotgan dasturiy ta'minot algoritmlari ma'lumotlarning muayyan majmualarida tekshiriladi va olingan natijalar tegishli ixtisoslashtirishlar bilan solishtiriladi. Asosiy vazifa — ixtisoslashtirishlar tushunilishi to'g'riligiga ishonch hosil qilish hamda loyihaga asos qilinayotgan konseptual qarorlarning afzalliklari va kamchiliklarini tahlil qilish.

Amalga oshirish bosqichida modullarning ijobatlanish, testlarning terilish rejasi (davomiyligi), shuningdek alohida modular matnlari tekshiriladi. Barcha bosqichlar uchun adabiyot manbalari bo'yicha va avvalgi ishlab chiqishlar tajribasidan kelib chiqqan holda shakllantirilgan eng ko'p uchrovchi xatolar

ro'yxatiga ega bo'lish maqsadga muvofiq. Bunday ro'yxatlar hammasini birvarakayiga ketma-ket tekshirishga emas, muayyan holatlarda kuch-g'ayratga markazlashtirishga imkon beradi. Bunda barcha topilgan xatolar maxsus hujjatda qayd etiladi, biroq ular tuzatilmaydi.

Oralama tuzilmaviy nazorat xatolarni erta aniqlash bilan birga loyiha bo'yicha sifatli hujjatlar o'z vaqtida tayyorlanishini ham ta'minlaydi.

Dasturiy ta'minotga va uni loyihalash uchun boshlang'ich ma'lumotlarga talablarni belgilash

Bazi predmeting (masalaning) qo'yilish bosqichi dasturiy mahsulot yaratishning eng mas'uliyatli bosqichlaridan biridir. Ushbu bosqichda ishlab chiqilayotgan dasturiy ta'minlashga asosiy talablar shakllantiriladi. Funktsiyalar va tasarrufiy talablar qanchalik to'liq belgilanganiga hamda loyihalash jarayonini belgilovchi prinsipial qarorlar qanchalik to'g'ri qabul qilinganiga ishlab chiqish qiymati va uning sifati ko'p jihatdan bog'liq.

Dasturiy mahsulotlarning funksional alomat bo'yicha tasnifi

Har bir dasturiy mahsulot muayyan funksiyalarni bajarish uchun mo'ljallangan. Barcha dasturiy mahsulotlarni ahamiyatiga ko'ra uch guruhga: tizimli, amaliy va gibrid guruhlariga ajratish mumkin.

Tizimli guruhga odatda hisoblash tizimlarining (ham alohida kompyuterlarning, ham tarmoqlarning) funksiyalar bajarilishini ta'minlovchi dasturiy mahsulotlar mansub bo'ladi. Bular operatsion tizimlar, qobiqlar va boshqa dasturlar (utilitlar).

Operatsion tizimlar, qoidaga ko'ra, resurslarni (protssessor va xotira bilan), jarayonlarni (vazifalar va oqimlar bilan) va qurilmalarni boshqaradi. Operatsion tizimlarni tashkillashtirishning

murakkabligi avtomatlashtirish darajasi va boshqaruv jarayonlarining erishuvchi samaradorligi bilan shartlaniladi. Jumladan, multidasturiy operatsion tizimlar yakkadasturiy tizimlardan ancha murakkabroq, bu MS DOS hamda WINDOWS misolida yaxshi ko'rinadi.

Qobiqlar

Qobiqlar (masalan, NORTON COMANDER) O'z vaqtida tizimli MS DOS fayllaridan foydalanuvchining yanada qulay interfeysini tashkillashtirish uchun paydo bo'lgandi. FAR kabi zamonaviy qobiqlar faylli tizim bilan ishlashda foydalanuvchiga odatiy muhitni ta'minlash uchun qo'llaniladi.

Muayyan funksiyalar, jumladan fayllarni arxivlashtirish, kompyuterni viruslar yuqishidan tekshirish, axborotga uzoqlashgan daxlni amalga oshirish va boshqa funksiyalar bajarilishini ta'minlovchi dasturlar hamda tizimlarni utilitlarga kiritish qabulqilingan.

Amaliy dasturlar va tizimlar muayyan foydalanish vazifalarini hal etishga yo'naltirilgan.

Foydalanuvchilarni ikki xilga ajratiladi:

- dastur ishlab chiquvchilar;
- kompyuter tizimlardan o‘z maqsadlariga erishish uchun foydalanuvchi nodasturchilar.

Utilitlar

Utilitlar deganda to‘g‘ridan to‘g‘ri operatsiyalar tizim tarkibiga kirmaydigan, lekin fayllarni arxivlash, kompyuterni virus bilan zararlanmaganligini tekshirish, axborotlarga masofadan turib ruxsatli foydalanish kabi boshqa ishlarni bajarish funksiyalarini ta’minlovchi dasturlar va tizimlar tushuniladi. Amaliy dasturlar va tizimlar muayyan foydalanuvchining aniq masalalarini hal qilishga mo‘ljallangan bo‘ladi. Dasturlar ishlab chiquvchilar keyingi paytda, odatda dasturlash tizimlariga va ishlab chiqish muhitlariga integratsiya qilinayotgan kompilyatorlar, jamlovchilar, sozlovchilar kabi maxsus instrumental vositalaridan foydalanishadi. Zamonaviy dasturlash, muhitlari, masalan Delphi, Visual C++ dasturiy mahsulotlar ishlab chiqishning vizual texnologiyasini ishlatadi va dasturchilar uchun o‘z ishlanmalariga kiritishlari mumkin bo‘lgan komponentlarning boy kutubxonalarini taqdim etishga imkon beradi.

CASE dasturiy ta’minotni ishlab chiqish

Assess, FoxPro, Oracle kabi ma’lumotlar bazalarini yaratishning instrumental komplekslari, intellektual tizimlar, masalan ekspert, ta’limiy tizimlar, bilimlar nazorati tizimlari va shu kabilarni yaratish vositalari ham ushbu guruhga mansubdir.

Ushbu yo‘nalishdagi so‘nggi yutuq — CASE dasturiy ta’minotni ishlab chiqishning ERwin, BPwin, Paradigm Plus, Rational Rose kabi vositalaridir. Foydalanuvchi-nodasturchilar zamonaviy talablarga muvofiq dasturiy mahsulotlar yaratish va ularning operatsion tizim bilan o‘zaro harakatlanishi spetsifikasi muammolarida professionallar bo‘lmasligi shart. Ular uchun aniq predmet sohaga mo‘ljallangan maxsus dasturiy mahsulotlar ishlab chiqiladi. Bunday mahsulotlarni shartli ravishda umumiy ahamiyatli mahsulotlarga, professional muhitlar yoki paketlarga, o‘rgatuvchi tizimlarga, ko‘ngilochar dasturlarga va hokazolarga ajratish mumkin.

Umumiy ahamiyatli dasturlardan foydalanuvchilarning turli guruhlari foydalanadi. Matnli muharrirlar, masalan WinWord, shuningdek, Excel tipidagi elektron jadvallar, grafik muharrirlar, umumiy ahamiyatli axborot tizimlari ularga mansubdir.

Professional mahsulotlar

Professional mahsulotlar turli sohalardagi mutaxassislar uchun mo‘ljallangan va ularga quyidagilar mansub:

- turli texnik sohalarga yo‘naltirilgan loyihalashni avtomatlashtirish tizimlari;

- trenajer (mashq) — tizimlar, masalan avariya vaziyatda uchuvchilar harakatlari ishlanishi uchun trenajer;
- hisoblagich tizimlari, masalan 1C;
- noshirlik tizimlari, masalan PageMaker, QuarkXpress;
- professional grafik tizimlar, masalan Adobe Illustrator, PhotoShop, CorelDraw va hokazo;
- ekspert tizimlar va boshqalar.

Ishlab chiqarish jarayonlarini avtomatlashtirish tizimlari

Ishlab chiqarish jarayonlarini avtomatlashtirish tizimlari professional tizimlardan shunisi bilan farqlanadiki, ular yagona ishlab chiqarish jarayoni bilan bog‘liq turli kasbdagi foydalanuvchilarga yo‘naltirilgan.

Ta‘limiy dasturlar va tizimlar o‘z nomiga muvofiq ravishda ta‘lim, masalan xorijiy tillar, yo‘l harakati qoidalari va hokazolar ta‘limi uchun mo‘ljallangan.

Ko‘ngilochar tizimlarga o‘yinlar dasturlari, musiqiy dasturlar, axborot, biror ko‘ngilochar mazmundagi testlari bor tizimlar, masalan munajjimlar bashorati va hokazolar mansubdir.

Gibrid tizimlar

Gibrid tizimlar tizimli va amaliy dasturiy ta‘minot alomatlarini o‘zida birlashtiradi. Qoidaga ko‘ra, bular real vaqt tartibida turlicha tiplardagi texnologik jarayonlarni boshqarish uchun mo‘ljallangan katta, lekin tor ixtisosli tizimlardir.

Ishonchlilikni oshirish va ishlanish vaqtini kamaytirish uchun bunday tizimlarga, odatda operatsion tizimlar funksiyalari bajarilishini ta‘minlovchi dasturlar kiritiladi.

Yuqorida sanalgan dasturiy ta‘minot tiplarining har biriga ishlab chiqishdagi funksional talablar bilan birga, odatda muayyan tasarrufiy talablar ham qo‘yiladi.

Dasturiy mahsulotlarga asosiy tasarrufiy talablar

Tasarrufiy talablar ishlab chiqilayotgan dasturiy ta‘minotning bajarish jarayonida namoyon bo‘luvchi ayrim tavsiflarni belgilaydi. Bunday tavsiflarga quyidagilar mansub:

- to‘g‘ri ishlashlilik — texnik topshiriq bilan muvofiqlikda ishlab turishi;
- universallik — har qanday imkonli ma‘lumotlarda to‘g‘ri ishlashni va noto‘g‘ri ma‘lumotlardan himoyani ta‘minlash;
- ishonchlilik (to‘siqlardan himoyalanganlik) — natijalarning to‘liq takrorlanuvchanligini ta‘minlash, ya‘ni turli xildagi izdan chiqishlar mavjudligida ularning to‘g‘riligini ta‘minlash;
- tekshiriluvchanlik — olingan natijalarni tekshirish imkoniyati;
- natijalar aniqligi — berilgandan yuqori bo‘lmagan natijalar xatoligini ta‘minlash;
- himoyalanganlik — axborot konfidentsialligini ta‘minlash;
- dasturiy moslik — boshqa dasturiy ta‘minot bilan birgalikda ishlash imkoniyati;

- samaradorlik — texnik vositalar resurslari, masalan mikroprotssessor vaqti yoki operativ xotira hajmidan mumkin bo'lgan darajadagi minimal miqdordan foydalanish;
- moslashuvchanlik — funksiyalar bajarishning o'zgarayotgan shart-sharoitlariga moslashish maqsadida jadal modifikatsiyalanish imkoniyati;
- takroriy kirishlilik — diskdan qayta yuklatishsiz takroriy bajarish imkoniyati;
- rentabellik (reyentabellik) — bir necha jarayonlardan «parallel» foydalanish imkoniyati.

“To‘g‘rilik”

To‘g‘rilik har qanday dasturiy ta'minlash uchun majburiy talab hisoblanadi: texnik topshiriqda ko'rsatilgan hamma narsa muqarrar ijobatlanishi shart. Biroq testlash ham, verifikatsiyalash ham yaratilgan dasturiy mahsulot to'g'riligini isbotlay olmasligini tushunish lozim. Shu munosabat bilan odatda muayyan xatolar mavjudligi ehtimoli to'g'risida gapiriladi.

Tabiiyki, kompyuter tizimiga qancha ko'p mas'uliyat yuklatilsa ham dasturiy, ham apparatli izdan chiqishlar ehtimoli shuncha kam bo'lishi darkor.

Universallik

Universallik talablari ham odatda majburiy talablar guruhiga kiradi. Ishlab chiqilgan tizim nuqsonli (nokorrekt) ma'lumotlar uchun natija chiqarishining yoki ma'lumotlarning ayrim termasida o'z ishini buzilishli (avariyali) yakunlanishining hech qanday yaxshi tomoni yo'q. Biroq yuqorida eslatib o'tilganidek, nisbatan murakkab dasturning universalligini ham xuddi uning to'g'riligi kabi isbotlash mumkin emas, shu boisdan dasturning universalligi darajasi haqida so'z yuritish mohiyatan ro'g'ri bo'ladi.

Amalda dasturiy ta'minotining to'g'riligi va universalligiga talablar qancha yuqori bo'lsa, uning ishonchliligiga talablar ham shuncha yuqori bo'ladi. Hisoblash jarayonining barcha qatnashchilari: texnik vositalar, dasturiy vositalar va odamlar to'siqlarning manbalari bo'lishlari mumkin. Texnik vositalar, masalan elektr kuchlanishining keskin sakrashlari yoki tarmoqlar bo'yicha axborotni berishdan to'siqlar tufayli izdan chiqishlarga moyil. Dasturiy ta'minotda xatolar bo'lishi mumkin. Odamlar esa boshlang'ich ma'lumotlarni kiritishda yangilanishlari mumkin.

«Nazorat nuqtalarini yaratish»

Zamonaviy hisoblash qurilmalari hozirda ishonchli. Texnik vositalarning izdan chiqishlari, qoidaga ko'ra, apparatlarda qayd etiladi va ushbu holatda tegishli ravishda hisoblash natijalari tiklanadi. Uzoq muddatli hisoblarda, qoidaga ko'ra, oraliq natijalar saqlanadi (bu yo'l «nazorat nuqtalarini yaratish» nomini olgan), bu esa izdan chiqish sodir bo'lganda hisoblashni oxirgi nazorat nuqtasida yozilgan ma'lumotlar bilan davom ettirishga imkon beradi.

Tarmoqlar bo'yicha axborotni berish ham apparatlar orqali nazorat qilinadi, bundan tashqari ma'lumotlarni berishdagi xatolarni topish va tuzatishga imkon

beruvchi maxsus to‘siqdan himoyalovchi kodlashtirish ham odatda qo‘llaniladi. Biroq texnik vositalarning xatolarini butunlay istisno qilish mumkin emas, shu boisdan ishonchlilikka talablar yuqori bo‘lgan holatlarda, odatda tizimlar dubllashtirishdan foydalaniladiki, bunda ikki tizim vaqti-vaqti bilan olingan natijalarni solishtirgan holda aynan bitta vazifa predmeti bajaradi.

«Ishonchsiz elementi»

Ko‘pincha zamonaviy tizimlarning eng «ishonchsiz elementi» odamlar bo‘ladi. Hozirda yaxshi ma‘lumki, hisoblash moslamasi pultdagi monoton ish sharoitlarida operatorlar juda ko‘p xatolarga yo‘l qo‘yishadi. Muayyan vaziyatlarda xatolar miqdorini kamaytirishga imkon beruvchi vositalar ham ma‘lum. Jumladan, imkoni bo‘lgan joyda ziyoda axborot kiritishdan foydalaniladi, bu kiritiluvchi ma‘lumotlarning to‘g‘riligini tekshirish imkonini beradi.

Bundan tashqari, axborotni kiritish emas, ayrim ro‘yxatdan tanlash zarur bo‘lganda turli xildagi ko‘maklashuvlardan keng foydalaniladi va hokazo.

Hisoblashlar texnologik jarayonlar bilan parallel bajariladigan real vaqt tartibida funksiyalarni bajaruvchi boshqaruv tizimlari ishlab chiqilishida ishonchlilikka yuqori talablar qo‘yiladi. Mazkur talab ilmiy-texnik tizimlar va ma‘lumotlar bazalari uchun ham muhimdir.

Tekshiriluvchanlikni ta‘minlash uchun olingan natijalarga ta‘sir qiluvchi boshlang‘ich ma‘lumotlarni, o‘rnatilgan tartiblar va boshqa axborotni hujjatli qayd etish lozim. Bu, ayniqsa, ma‘lumotlar bevosita datchiklardan kelib tushadigan holatlarda g‘oyatda ahamiyatlidir. Agar bunday ma‘lumotlar natijalar bilan birga chiqarilmasa, natijalarni tekshirish mumkin bo‘lmay qoladi.

Aniqlik yoki natijalar xatoligi kattaligi boshlang‘ich ma‘lumotlar aniqligiga, foydalanilayotgan model adekvatligi darajasi va tanlangan usul aniqligiga, shuningdek, kompyuterda operatsiyalar bajarishning xatoligiga bog‘liq. Odatda natijalar aniqligiga talablar texnologik (masalan, kimyoviy) jarayonlarni boshqaruv tizimlari hamda navigatsiya tizimlari (masalan, kosmik apparatlarning ulashuvini boshqaruv tizimi) uchun eng qattiq bo‘ladi.

Axborotning himoyalanganligini ta‘minlash

Loyihalananayotgan tizim foydalaniladigan axborotning himoyalanganligini (konfidensialligini) ta‘minlash alohida bo‘lib, tarmoqlar mavjudligi sharoitlarida yetarlicha murakkabdir. Himoyalanganlikni ta‘minlash uchun axborotni kodlashtirish va foydalanuvchi identifikatsiyalash kabi sof dasturiy himoya vositalari bilan birga maxsus tashkiliy usullardan ham foydalaniladi.

Davlat va tijorat sirlari bilan bog‘liq axborot saqlanadigan tizimlarga eng qattiq talablar qo‘yiladi. Dasturiy mutanosiblikning talablari ko‘rsatilgan dasturiy ta‘minotni birgalikda o‘rnatish imkoniyatidan tortib, to u bilan o‘zaro harakatini ta‘minlashgacha bo‘lgan o‘zgarishni tushunish mumkin, masalan ma‘lumotlar almashinuvi va h.k. Hammasidan ko‘ra ko‘proq dasturiy ta‘minotning ishlashi mazkur operatsion tizim boshqaruvi ostida bo‘lishini ta‘minlashga to‘g‘ri keladi.

Biroq qandaydir dasturdan ma'lumotlar olishni yoki unga ayrim ma'lumotlarning berilishini ko'zda tutish talab qilinishi mumkin. Ushbu holatda berilayotgan ma'lumotlar formatlarini aniq ta'kidlab ko'rsatish zarur.

Apparatli mutanosiblik

Apparatli mutanosiblik talablari asosan dasturiy ta'minot ishlaydigan uskunaning minimal konfiguratsiyasi tarzida shakllantiriladi. Agar nostandart uskunadan foydalanish nazarda tutilayotgan bo'lsa, uning uchun axborot bilan almashinuv interfeyslari yoki protokollari ko'rsatilishi shart. Bu borada Windows sinfidagi operatsion tizimlar uchun qurilmaning operatsion tizim bilan o'zaro harakatlanishini ta'minlovchi drayver-dasturlar mavjud bo'lmagan qurilmalar nostandart hisoblanadi.

Tizim samaradorligi hisoblash

Tizim samaradorligi hisoblash moslamasining har bir resursi bo'yicha alohida baholanadi. Ko'pincha quyidagi mezonlardan foydalaniladi:

- tizim javobi vaqti (foydalanilayotgan uskuna tez hakarakatlanishiga mansub bo'luvchi) — foydalanuvchi bilan interaktiv tartibda o'zaro ishlovchi tizimlar va real vaqt tizimlari uchun;
- operativ xotira hajmi — operativ xotiraning hajmi cheklangan tizimlarda ishlovchi mahsulotlar uchun masalan MS DOS;
- tashqi xotira hajmi — tashqi xotiradan, masalan ma'lumotlar bazalaridan intensiv foydalanadigan mahsulotlar uchun;
- xizmat ko'rsatiluvchi tashqi qurilmalar miqdori — tashqi qurilmalar, masalan datchiklar bilan intensiv o'zaro harakatlanishni amalga oshiruvchi mahsulotlar uchun.

Samaradorlik talablari bir-biriga qarama-qarshi bo'lishi mumkin.

Masalan, dasturning ayrim fragmenti bajarilish vaqtini kamaytirish uchun operativ xotiraning qo'shimcha hajmi talab qilinishi mumkin.

Moslashuvchanlik, mohiyatga ko'ra, dasturiy ta'minotning texnologik sifatini baholaydi, shu boisdan ushbu tavsifni miqdoriy baholash amalda mumkin emas.

Mahsulotni yaratishda uning modellashtirilishini yengillashtiruvchi texnologiyalardan va maxsus usullardan foydalanilganini qayd etish mumkin.

Takroriy kirishlilik talablari operativ xotiraga rezidentli yuklatilgan dasturiy ta'minotga, masalan drayverlarga qocoyiladi. Mazkur talabni ta'minlash uchun dasturni shunday tashkillashtirish kerakki, uning boshlang'ich ma'lumotlari bajarish jarayonida o'chirilib ketmasligi yoki har bir chaqiruv boshlanishida yoxud tugallanishida tiklanishi kerak.

Rentabellik talablari

Rentabellik talablari takroriy kirishlik talablariga ko‘ra yanada qattiqroq, zero ushbu holatda bajarish jarayonida dastur o'zgartiruvchi barcha ma'lumotlar maxsus blokka ajratilishi shart va uning nusxasi dasturni chaqirishidagi har bir jarayon uchun yaratiladi. Ko‘plab dasturiy tizimlarning murakkabligi ularga bo‘lgan aniq talablarni darhol shakllantirishga imkon bermaydi. Odatda qandaydir dasturiy ta’minotni yaratish g‘oyasidan texnik topshiriqqa kiritilishi mumkin bo‘lgan talablarning aniq shakllantirilishga o‘tishi uchun ishlab chiqish sohasida loyihadan oldingi tadqiqotlarni bajarish zarur bo‘ladi.

Foydalanilgan adabiyotlar

1. Ramez Elmasri, Shamkant B.Navathe. Fundamentals of Database Systems (7th Edition). Pearson. USA, 2015.
2. M.T.Azimjanova, Muradova, M.Pazilova. Informatika va axborot texnologiyalari. O'quv qo'llanma. T.: “O'zbekiston faylasuflari milliy jamiyati”, 2013 y.
3. Holmatov T.X., N.I.Tayloqov. Amaliy matematika, dasturlash va kompyuteming dasturiy ta’minoti. O‘quv qo‘llanma. T.: “Mehnat”, 2000 y.
4. S.S.Qosimov Axborot texnologiyalar O‘quv qo‘llanma. T.:“Aloqachi”, 2006 y.
5. M Aripov, B.Begalov va boshqalar. Axborot texnologiyalari. O‘quv qo‘llanma. T.:, “Noshir”, 2009 y.
6. A.Sattorov. Ma'lumotlar bazasini boshqarish sistemasi Access (Windows 9x/2006) O‘quv qo‘llanma. T.: “Fan va texnologiya”, 2006 y.
7. Axborot xavfsizligi asoslari, I.M.Karimov Tashkent, 2019
8. Информационная безопасность телекоммуникационных систем (технические аспекты), Кулаков М.В., Гарашин А.В. Учеб. Пособие для вузов. М.: Радио и связь, 2014
9. Информационная безопасность государственных организаций и коммерческих фирм. Справочное пособие. М.: 2017.
- 10.Зегжда Д.П. Основы безопасности информационных систем: Учеб. пособие для студ. вузов. – М.: Горячая линия – Телеком, 2016.