

1. Įvadas

1.1. Duomenų bazės, duomenų bazių valdymo sistemos

Duomenų baze (DB) galima vadinti rinkinį tarpusavyje susijusių duomenų, kurie apdorojami programomis. Tokia duomenų bazė, loginiu požiūriu, yra panaši į elektroninę kartoteką. Studentų kartotekos pavyzdys lentelė pateiktas 1.1 pav.

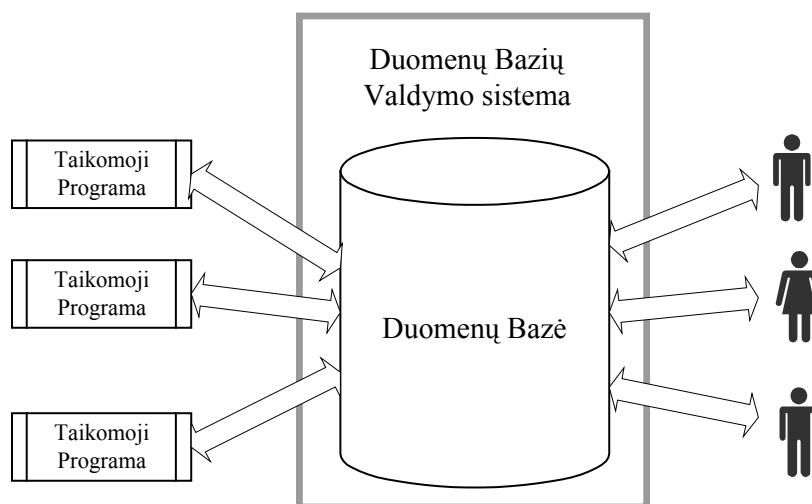
Vardas	Pavardė	Gimimo data	Studijų programa	Pakopa	Kursas
Jonas	Baltakis	1983.03.05	Informatika	Bakalaurantas	3
Ona	Gražulytė	1983.04.15	Informatika	Magistrantas	1

1.1 pav. Studentų duomenų bazė

Panašios duomenų bazės (kartotekos) vartotojai atlieka tokias operacijas:

- naujo įrašo (naujos kortelės) įtraukimas;
- esamo įrašo (kortelės duomenų) keitimas;
- esamo įrašo (kortelės) šalinimas;
- duomenų paieška duomenų bazėje (kartotekoje).

Minėtom ir panašiom operacijoms atlikti, kai duomenys saugomi kompiuterio atmintyje, reikia programinės įrangos. Tokia programinė įranga kartu su duomenų baze vadinama **duomenų bazės sistema** (DBS). Dažnai, kalbant apie kompiuterines sistemas, pastarosioms priskiriami ir jų vartotojai. DB vartotojais laikomi asmenys, betarpiškai operuojantys duomenų baze, bei taikomosios programos, kurios vykdymo metu kreipiasi į DB. Pagrindinė tokios programinės įrangos paskirtis yra suteikti DB vartotojui galimybę dirbti su DB, neatsižvelgiant į technines detales. Programinė įranga, sprendžianti tokį uždavinį vadinama **duomenų bazių valdymo sistema** (DBVS). Supaprastinta DBS pavaizduota 1.2 pav. Kitaip tariant, DBVS leidžia vartotojui žiūrėti į DB kaip į žymiai aukštesnio lygio objektą, negu į įrašų failą. Šiuo požiūriu, DBVS atlieka vartotojo sąsajos su DB vaidmenį. Vartotojas operuoja DB duomenimis formalia kalba, pvz. SQL.



1.2 pav. Supaprastinta duomenų bazės sistema

1.2. Duomenų bazių valdymo sistemų funkcijos

Be jau minėtos pagrindinės DBVS funkcijos, kuri leidžia vartotojui dirbti su DB naudojant logines sąvokas, DBVS atlieka ir kitas svarbias funkcijas. DBVS sprendžia ne tik vartotojo nepriklausomumo nuo fizinių DB realizavimo ypatybių problemą, bet ir kitus uždavinius. Tuo pačiu, galime kalbėti ir apie reikalavimus DB valdymo sistemoms. Aptarsime bene pagrindinius iš jų.

Naudojant popierinę duomenų saugojimo technologiją, norint tuos pačius duomenis vartoti sprendžiant skirtingus uždavinius, dažnai tenka dubliuoti duomenis. Pvz., studentų skyrius naudoja "savus" studentų sąrašus, o studentų atstovybė ar buhalterija "savus". Panaši situacija yra ir saugant bei apdorojant duomenis, esančius failuose. Dažna taikomoji programa, dirbdama su failais, naudoja savas kopijas ar net savus duomenų formatus. Tokia padėtis nepageidautina. DBVS keliamas uždavinys **minimizuoti duomenų perteklių**. Dažnai nepavyksta visiškai išvengti duomenų dubliavimo, bet minimizuoti jų perteklių yra būtina.

Yra žinoma, kad atidarius duomenų failą rašymui, operacijų sistema kitiems vartotojams neleidžia ne tik į jį rašyti, bet ir skaityti iš jo tol, kol rašymo operacija nebus baigta ir failas nebus uždarytas. DB yra skiriama plačiam vartotojų ratui, todėl DBVS keliamas uždavinys užtikrinti **efektyvų bendrą DB vartojimą**. DBVS pasiekia tai lanksčiai naudodama duomenų blokavimą (izoliavimą). Pagal vieną iš pagrindinių naudojamų strategijų idėjų, vienam vartotojui keičiant DB įrašą, kitiems vartotojams tuo pačiu metu leidžiama peržiūrėti ar net keisti kitus įrašus.

DB vadinama **vientisa** (integralia), jei ji tenkina tam tikrus apribojimus (sąlygas) duomenims ir išsaugo tuos apribojimus modifikuojant (keičiant, šalinant, įterpian) duomenis. Tokio apribojimo pavyzdžiu gali būti "studento bakalauranto kursas negali būti didesnis už 4, o magistranto – už 2". DBVS, kuri užtikrina DB integralumą, atliekant kiekvieną DB modifikavimą, privalo tikrinti, ar duomenų pakeitimas nepažeis nustatytų apribojimų. Jei vartotojas bando modifikuoti DB taip, kad po to kuri nors sąlyga taptų nepatenkinta, tai DBVS privalo neatlikti tokios operacijos ir informuoti apie tai vartotoją.

Kita labai svarbi DB savybė yra **duomenų neprieštaringumas**. Tarkime, kuriam nors studentui studentų bazėje yra du įrašai, kurių viename nurodyta, kad jis studijuoja pirmame kurse, o kitame, kad - antrame. Tokiu atveju, vartotojui užklausus iš bazės duomenų apie to studento kursą, užklaustos rezultatas priklausys nuo to, iš kurio įrašo duomuo bus paimtas. Aišku, tokia situacija yra neleistina. DB valdymo sistemai keliamas uždavinys užtikrinti, kad panašių prieštarų būtų išvengta. Minėta priešvara neatsiras, pavyzdžiui, jei studentų duomenų bazėje kiekvienam studentui visuomet bus skiriama tik viena eilutė.

Nemažiau svarbu apsaugoti duomenis, esančius duomenų bazėje, nuo tyčinio ar netyčinio svarbių duomenų sunaikinimo (pašalinimo) ar pakeitimo. Duomenų bazės **saugumo** sąvoka apima ir apsaugą nuo neleistinos duomenų peržiūros. Šiam tikslui pasiekti, DBVS atsakingajam DB vartotojui (administratoriui) pateikia priemones, kurios leidžia apibrėžti kiekvienam vartotojui ar jų grupei teises duomenų atžvilgiu. DBVS, turėdama teisių sąrašą, aptarnaujant vartotojų užklausas, stebi, kad niekas neviršytų jam priskirtų teisių.

Saugant duomenis failuose, įmanoma sukurti programas, kurios, atsižvelgdamos į duomenų formatą, galės labai efektyviai atlikti paiešką pagal iš anksto numatytus kriterijus net ir labai dideliuose failuose. Tačiau, dažnai negalima iš anksto numatyti visus galimus paieškos kriterijus ir, atsiradus poreikiui, tenka ieškoti duomenų pagal naują kriterijų ar jų kombinaciją. DBVS keliamas uždavinys dideliuose duomenų masyvuose efektyviai atlikti ne tik **planuotas** (pagal iš anksto numatytus kriterijus), bet ir **neplanuotas užklausas**.

Aptarėme ne visas funkcijas. Minėtas ir kitas DBVS funkcijas (reikalavimus, uždavinius) plačiau aptarsime kituose skyreliuose. NAGRINĖSIME ir tai, kokiomis priemonėmis DBVS keliamus jai uždavinius sprendžia.

1.3. Reliacinės ir kitokios DB

Viena iš pagrindinių DBVS funkcijų yra sąsaja tarp vartotojo ir DB. DBVS suteikia vartotojui galimybę kreiptis į duomenų bazę ne fiziniėmis (aparatinėmis ar operacijų sistemos) sąvokomis, bet loginėmis. Sąvokų rinkinys ir jų vartojimo taisyklės sudaro **modelį**. Dauguma DBVS, sukurtų pradedant 70-ųjų pabaigą, vartotojo sąsajai naudoja **reliacinį duomenų modelį**. DB, kurios apibūdinamos naudojant reliacinį modelį, vadinamos **reliacinėmis duomenų bazėmis** (RDB), o atitinkamos jų valdymo sistemos - **reliacinėmis duomenų bazių valdymo sistemomis** (RDBVS).

Reliacinį modelį 1970 m. pasiūlė E.F. Kodas (E.F. Codd), tuomet dirbęs IBM tyrimų laboratorijoje. Pateiksime supaprastintą reliacinio modelio aprašą. Formaliau reliacinį modelį išdėstysime tolimesniuose skyreliuose.

Pagrindinė reliacinio modelio sąvoka yra **lentelė**. Reliacinėse sistemose lentelę sudaro horizontalios **eilutės** (angl. *row*) ir vertikalūs **stulpeliai** (angl. *column*). **Reliacinė sistema** yra vadinama sistema, besiremianti šiais pagrindiniais principais:

- visi duomenys vartotojui pateikiami lentelėmis;
- vartotojui pateikiami operatoriai generuojantys naujas lenteles iš senų. Pavyzdžiui, pateikiamas operatorius, leidžiantis gauti turimos lentelės eilučių poaibį, bei operatorius, įgalinantis gauti lentelės stulpelių poaibį.

Tokios sistemos vadinamos reliacinėmis, nes anglišką žodį *relation* (santykis) yra matematinis lentelės pavadinimas. Daugeliu atžvilgių, terminai “lentelė” ir “santykis” yra sinonimai. Literatūroje vietoje termino “eilutė”, galima sutikti terminus “kortelės” (angl. *tuple*) ir “įrašas” (angl. *record*), o vietoje termino “stulpelis” – “atributas” (angl. *attribute*) ir “laukas” (angl. *field*). Taigi, reliacinė DB - tai duomenų bazė, kurią vartotojai priima kaip lentelių rinkinį. Kitaip tariant, lentelių rinkinys sudaro duomenų bazę.

Kiekviena duomenų bazės lentelė turi savo vardą (pavadinimą). Kiekviena lentelės eilutė aprašo vieną objektą – žmogų, firmą, sandėrį ar pan. Kiekvienas stulpelis aprašo vieną objekto charakteristiką – žmogaus pavardę, vardą, adresą ir pan., firmos telefono numerį, sandėrio numerį, datą ir pan. Duomenų elementas arba reikšmė (angl. *value*), nusakoma lentelės eilutės ir stulpelio susikirtimu. Stulpeliui yra priskiriama galimų reikšmių aibė, vadinama **domenu** (angl. *domain*).

Ne reliacinėse sistemose duomenys vartotojui pateikiami kitokiomis struktūromis nei reliacinės sistemos lentelės. Pavyzdžiui, **hierarchinėse** sistemose duomenys vaizduojami medžio struktūromis, o tarp operacijų yra judėjimo hierarchinėmis struktūromis į viršų ir žemyn operacijos. **Tinklinės sistemos**, didelė dalimi, yra hierarchinių sistemų apibendrinimas. Duomenys tinklinėse sistemose vaizduojamos orientuotais grafais. Tiek hierarchinės tiek ir tinklinės sistemos priskiriamos ikireliacinėms sistemoms. Šių duomenų modelių dominavimą panaikino reliacinių sistemų atsiradimas. Tačiau ir šiandien tinklinės sistemos efektyviai taikomos specifiniams uždaviniams spręsti, pvz. geografijos informacinėms sistemoms realizuoti.

Pirmosios reliacinės sistemos atsirado 1970-ųjų pabaigoje – 1980-ųjų pradžioje. Tarp šiuo metu labiausiai paplitusių ir labiausiai išvystytų RDBVS galima paminėti firmos IBM sukurtą DBVS DB2, korporacijos Oracle produktą ORACLE, Sybase Inc. produktą SYBASE, firmos Microsoft DBVS SQL Server ir kt.

Nors reliacinis modelis pasirodė labai patogus vartojimui, tačiau jis nėra idealus. Pastaraisiais dešimtmečiais buvo gana aktyviai ieškoma naujų duomenų modelių. Paminėtina **deduktyvios, ekspertinės ir objektinės** duomenų bazių valdymo sistemos. Ypač aktyviai domimasi pastarosiomis. Objektinės DBVS pastaraisiais metais plinta žymiai sparčiau negu reliacinės. Tačiau reliacinės sistemos ir toliau žymiai plačiau vartojamos negu visos kitos kartu paėmus, tiek Lietuvoje, tiek ir visame pasaulyje. Dėl šios priežasties, tolimesniuose skyreliuose mes nagrinėsime tik reliacines sistemas. Pažyminį reliacinę (-inis) dažnai praleisime.

1.4. SQL

Vartotojo sąsajos organizuoti (užklausoms formuluoti) yra vartojama tam tikra formalizuota kalba. Daugumoje reliacinių sistemų yra vartojama vienas iš kalbos **SQL** (Structured Query Language) dialektų. Ši kalba buvo sukurta firmos IBM tyrimų centre 1970-jų metų pabaigoje. Praktiškai nuo pačios pirmosios firmos IBM sukurtos reliacinės sistemos System R SQL kalba tapo Amerikos nacionaliniu (ANSI) bei tarptautiniu (ISO) standartais.

Didele dalimi, SQL kalba vartojama reliacinėms operacijoms aprašyti. Šioje kalboje išskiriamos trys sakinių grupės: duomenų apibrėžimo sakiniai, kitaip dar vadinami duomenų apibrėžimo kalba (DDL, Data Definition Language); manipuliavimo duomenimis sakiniai (DML, Data Manipulation Language) ir duomenų valdymo sakiniai (DCL, Data Control Language), t.y. $SQL = DDL + DML + DCL$. **DDL** sakiniiais, tarp kitų veiksmų, yra kuriamos duomenų bazės ir lentelės. Kuriant naują DB yra nurodomas jos vardas (pavadinimas), fizinė vieta ir kai kurios kitos savybės. Kuriant (apibrėžiant) lentelę, privaloma nurodyti lentelės vardą, jos stulpelių vardus ir stulpelių tipus. Stulpelio tipu nusakoma, kokios rūšies duomenys galės būti įrašomi stulpelyje. SQL leidžiamos tokios duomenų rūšys: tekstiniai duomenys, skaičiai, dvejetainiai duomenys, datos ir laikai. Kiekviena duomenų rūšis su tam tikromis fizinėmis realizacijos savybėmis (pvz., baitų kiekis, skiriamas reikšmės vaizdavimui) nusakoma **duomenų tipu**. Duomenų tipo sąvoka yra reliacinės teorijos domeno sąvokos atitikmuo. SQL duomenų tipus plačiau aptarsime vėliau.

DML sakiniiais formuluojamos užklauskos (angl. *query*) duomenims surasti, keisti, šalinti ir įterpti. **DCL** sakiniiais koordinuojamas bendras efektyvus vartotojų darbas su DB. SQL - tai kalba, kuria galima bendrauti su reliacinėmis duomenų bazėmis.

Dažnai realiame informacijos valdyje duomenys būna nežinomi arba nepilni: studentas gali nesakyti mums, kaip jis išlaikė egzaminą, kiek per vasaros atostogas užsidirbo pinigų; pildant anketą galima nepildyti amžiaus ir šeimyninės padėties grafų. Įvedant informaciją į kompiuterį, į tai atsižvelgiama. Kartais tam naudojama sąvoka "nulis". Tačiau, tai nevisada padeda, nes nežinojimas nevisada reiškia nulį. Tai, kad nesužinojome, kiek draugas užsidirbo, nereiškia, kad jis užsidirbo nulį litų. Atsižvelgiant į fakto apie duomenų nebuvimą svarbą, SQL kalboje kiekviename duomenų tipe yra speciali reikšmė **NULL**. Šia reikšme žymima, kad duomenys yra nežinomi, šiuo metu dar neįvesti, jau pašalinti ir pan. Tai lyg reikšmė, reiškianti reikšmės nebuvimą.

Formuluojant užklauskas SQL kalba, reikia išreikšti tai, kas norima gauti, nesileidžiant į detales, kaip rezultatą gauti. DBVS uždavinys – nustatyti kaip, koku būdu, kokius algoritmus panaudoti rezultatui gauti. Todėl, SQL gali būti priskirta neprocedūrinėms (deklaratyvioms) programavimo kalboms. Kita ypatybė, kuri daro SQL kalbą netradicine programavimo kalba, yra SQL sakinių vartojimas programuojant programavimo kalbomis. SQL sakiniai gali būti įterpiami į programas, sudaromas bazinė programavimo kalba. Bazinėmis programavimo kalbomis SQL kalbai gali būti daugelis plačiai vartojamų programavimo kalbų: C/C++, PL1, COBOL, FORTRAN ir pan. Kita vertus, SQL sakiniai gali būti vykdomi ir interaktyviai.

Pirmasis tarptautinis SQL kalbos standartas buvo priimtas 1989 m., kuris vadinamas SQL-89 arba SQL1. Dauguma šiuolaikinių RDBVS šį SQL standartą pilnai atitinka. Besivystančios informacinės technologijos įtakojo ir SQL vystymąsi. 1992 m. buvo priimtas kitas standartas, kuriam prigijo SQL-92 ir SQL2 pavadinimai. Šiuo metu, tai – reikalavimai, kurių prisilaiko dauguma komercinių RDBVS. 1999 m. pasirodė naujas standartas SQL3. SQL2 nuo SQL1 skiriasi, daugiausia, kiekybiniais parametrais. SQL3 standarto kalba išsiskiria gana dideliais kokybiniais pakitimais. Į SQL3 įvesti nauji duomenų tipai, numatyta galimybė apibrėžti naujus struktūrinius duomenų tipus. Naujų galimybių atsiradimą labai įtakojo objektinių technologijų paplitimas. Šiandien RDBVS vystosi, siekdamos įdiegti SQL3 standarto galimybes.

Pastebėsime, kad SQL sudaro gana daug sakinių. Kai kurių sakinių sintaksės aprašai užima ne vieną puslapį. 1992 m. išleisto SQL-92 (International Standard Database Language SQL, 1992) aprašas užima virš 600 puslapių. Todėl, nėra galimybės nedidelės apimties knygoje išdėstyti ne tik pagrindines reliacinių DB sąvokas, bet ir SQL sintaksę. Šioje mokomojoje knygoje mes apžvelgsime žymią dalį SQL2 ir tik nedidelę dalį SQL3 sakinių.

Nagrinėjamų SQL sakinių detalios sintaksės nepateiksime. Juo labiau, kad, neretai, SQL sakinių sintaksė konkrečiose DBVS skiriasi tiek nuo standarto, tiek ir tarpusavyje. Praktiškai, SQL dialektų skaičius yra lygus komercinių RDBVS skaičiui. Toliau SQL sakinius pateiksime RDBVS IBM DB2 SQL dialekte, kuris yra artimas SQL2 standartui. Pateikdami sakinių sintaksę prisilaikysime Backus-Naur notacijos. Kita vertus, šioje knygelėje, nesistengsime pateikti visų nagrinėjamų sakinių galimybių. Tikslios sakinių sintaksės siūlome ieškoti komercinio produkto dokumentacijoje. Mūsų tikslas yra išmokyti skaitytoją operuoti pagrindinėmis RDBVS ir SQL sąvokomis. Dėstoma medžiaga skirta suformuoti pagrindą, kuris leistų savarankiškai naudotis komercinio produkto dokumentacija, siekiant patikslinti sakinių sintaksę, galimybes ir pan.

Ši mokomoji priemonė sudaryta taip, kad skaitytojas galėtų palaipsniui susipažinti su pagrindinėmis reliacinių DBVS sąvokomis ir išmokyti jomis operuoti naudojant SQL kalbą. Priemonė nėra orientuota į sudėtingų taikomųjų programų kūrimą - tai kitų informatikos sričių (informacinių sistemų, programų inžinerijos ir kt.) objektas.