

5. GIS duomenų bazių sudarymas

Šios paskaitos tikslas yra parodyti duomenų bazių ir duomenų bazių valdymo sistemų vietą ir reikšmę GIS. Taip pat bus aptariamas geografinių duomenų kaupimas duomenų bazėse.

5.1 Duomenų bazė ir duomenų bazių valdymo sistema

Bet kokia veikla neįmanoma be savalaikės ir tikslios informacijos. Savo ruožtu didelius informacijos kiekius apdoroti rankiniu būdu neįmanoma. Todėl jau šeštame XX amžiaus dešimtmetyje pradėtos kurti ir naudoti kompiuterinės duomenų saugojimo ir apdorojimo technologijos. Pirmiausia jos įsitvirtino verslo pasaulyje. Plačios apimties inžineriniai projektai, pavyzdžiui, kosmoso įsisavinimo programa, reikalavo sunkiai įsivaizduojamos apimties tiekimų apskaitos. Detalių užsakymų bei jų tiekimo kontrolės kompiuterizavimas labai atpigino gamybos kainas. Kitas pavyzdys – *IBM* kompanijos sukurta bilietų rezervavimo sistema *American Airlines*, kur pirmą kartą susidurta su kompiuterinių tinklų panaudojimu, užtikrinant daugelio nutolusių vartotojų darbą su dideliais duomenų kiekiais vienu metu. Besivystant informacinėms technologijoms, 6-7 dešimtmečiuose buvo sukurtos „duomenų bazės“ bei „duomenų bazių valdymo sistemų“ sąvokos.

Kompiuterijos terminų aiškinamajame žodyne **duomenų bazė yra apibrėžiama kaip „duomenų rinkinys, susistemintas ir sutvarkytas taip, kad juo būtų galima patogiai naudotis“.**

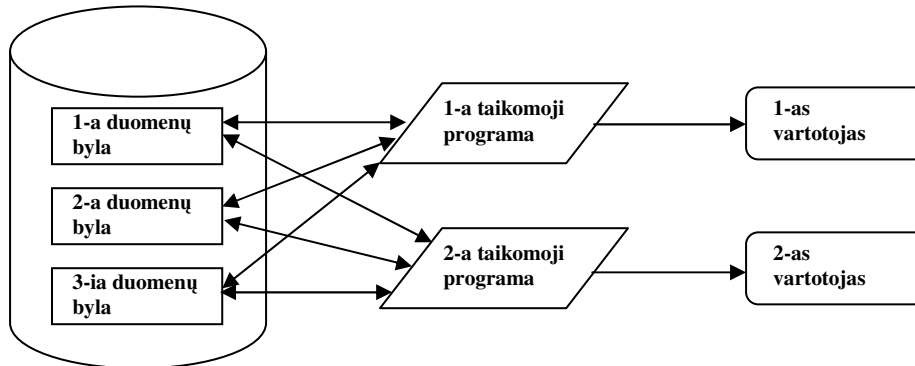
Duomenų bazių valdymo sistema (DBVS) - programinė sąsaja tarp naudotojo ir duomenų bazės, leidžianti vartotojui efektyviai manipuluoti duomenimis: pateikti paieškos užklausas, rikiuoti, įvesti naujus duomenis, pašalinti arba atnaujinti duomenis.

Duomenų bazėje saugoma informacija apie įvairius objektus bei jų tarpusavio ryšius. Pavyzdžiui, duomenų bazėje gali būti sukaupti duomenys apie žmones: jų vardai bei adresai. Vardai gali būti siejami tarpusavyje, naudojant tokias charakteristikas, kaip „klientas“, „draugas“, „šeima“.

Duomenų bazėje taip pat galime kaupti duomenis apie procesus: „dirvos erozija“, „vandens tarša“ ir pan. Pagrindinis tikslas, kodėl duomenys kaupiami duomenų bazėse, yra tarpusavyje susieti anksčiau buvusius nepriklausomus faktus ir situacijas. Paprasčiausiu atveju, tai gali būti faktų iš

duomenų bazės atgaminimas, pavyzdžiui, pateikiami adresai, susieti su žmonių vardais. Tačiau, norint įvertinti įvairiapusių ryšius tarp duomenų, kaip, pavyzdžiui, pateikti žmonių, gyvenančių padidintos rizikos zonoje dėl transporto keliamo triukšmo, sąrašą, gali tekti atlikti sudėtingas duomenų apdorojimo procedūras.

Ankstyvosios duomenų bazių sistemos, kaip ir kita tuometinė programinė įranga, buvo kuriamos siekiant atlikti aiškiai nusakytas funkcijas su tiksliai standartizuotais duomenų rinkiniais. Duomenys kaupiami kaip vienas ar keletas kompiuterinių bylų, kurios prieinamos naudojant specializuotą duomenų bazės programinę įrangą ir būdus, kurie sistemos kūrėjui atrodė tinkamiausi. Šis rinkmenų apdorojimo sprendimas valdant duomenų bazes yra iliustruojamas 5.1 pav.

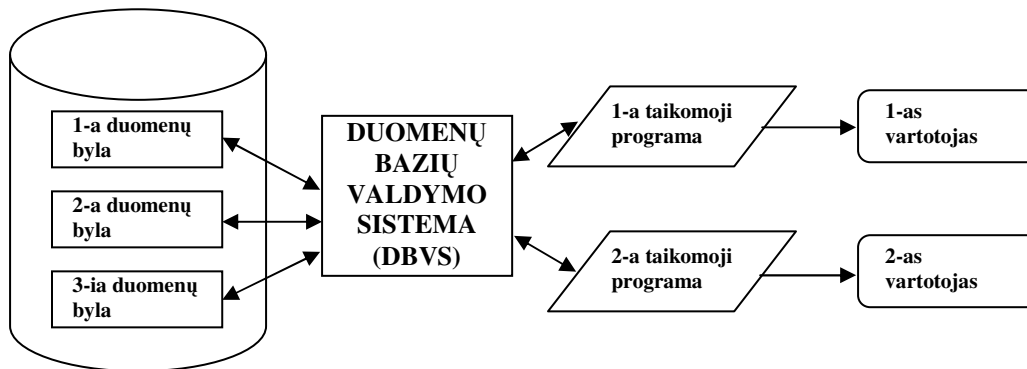


DUOMENŲ BAZĖ

5.1 pav. Duomenų bylų paskirstymas tarp taikomųjų programų, naudojant rinkmenų apdorojimo aplinką

Dirbant su duomenų bazėmis dažnai tenka apdoroti atskiras bylas. Tačiau čia susiduriame su keletu trūkumų. Kadangi taikomoji programa turi tiesiogiai pasiekti kiekvieną naudojamą duomenų bylą, ji privalo „žinoti“, kaip duomenys kiekvienoje byloje yra sukaupti. Tam kiekvienoje taikomojoje programoje būtina numatyti instrukcijas, kaip pasiekti duomenis kiekvienoje byloje. Pakeitus duomenų bylą šios instrukcijos taip pat turi būti atnaujintos. Duomenų bylas gali bendrai naudoti skirtingi vartotojai skirtingose taikomosiuose programose. Šiuo atveju būtina centralizuotai kontroliuoti, kurie vartotojai gali naudotis duomenų bazės duomenimis ir kokie pakeitimai yra leistini. Valdant duomenų bazes, naudojant duomenų rinkmenų apdorojimo sprendimą, tokia kontrolė gana problemiška.

Duomenų bazių valdymo sistemą sudaro kompleksas programų, skirtų manipuluoti duomenimis duomenų bazėje. DBVS sukurtos siekiant sutvarkyti bendrą duomenų naudojimą bei užtikrinti duomenų bazių integralumą. DBVS pažanga prieš rinkmenų apdorojimo aplinką yra akivaizdi, nes DBVS veikia kaip centrinė, kontroliuojanti grandis tarp duomenų bazių ir taikomųjų programų, su kuriomis dirba vartotojas – 5.2 pav.



DUOMENŲ BAZĖ

5.2 pav. Duomenų bylų paskirstymas tarp taikomųjų programų, naudojant duomenų bazių valdymo sistemos aplinką

DBVS užtikrina duomenų nepriklausomumą – tai reiškia, kad taikomajai programai nebereikia „žinoti“, kaip duomenys yra fiziškai sukaupti, nes jie pasiekiami per DBVS. Taikomoji programa pateikia duomenų bazių valdymo sistemai komandą paimti duomenis ir reikiamu pavidalu perduoti juos apdorojimui. Jei atliekami duomenų bylos pakeitimai, DBVS užtikrina, kad duomenys bus tinkami taikomosioms programoms be šių modifikavimo. Dar daugiau, DBVS garantuoja, kad tie patys duomenys tiks įvairioms taikomosioms programoms, reikalaujančioms skirtingų duomenų arba skirtingų duomenų formatų. Taigi, DBVS leidžia iš esmės sumažinti pastangas, reikalingas duomenų bazei bei taikomosioms programoms palaikyti.

DBVS sudaro sąlygas su ta pačia duomenų baze vienu metu dirbti keletui vartotojų nedarant duomenų kopijų. Taip paprastėja duomenų bazių administravimas, nes sudaromos prielaidos centralizuotai duomenų bazių naudojimo kontrolei. Pasinaudodamas vidinėmis duomenų bazių sąsajos priemonėmis, net ir eilinis vartotojas gali atlikti sudėtingus analizės uždavinius, o labiau kvalifikuotas – kurti galingus taikomuosius programinius modulius.

Kita vertus, kalbėdami apie duomenų bazių valdymo sistemą, mintyse turime komercinį produktą, kurį įsigyti bei palaikyti papildomai kainuoja. DBVS taip pat yra sudėtingesnis nei rinkmenų apdorojimu grindžiamas sprendimas, o tai teoriškai sudaro didesnę klaidų ar duomenų praradimo tikimybę.

Pirmosios GIS naudojo rinkmenų apdorojimo aplinką. Šis sprendimas naudojamas ir dabar (pavyzdžiui, IDRISI programinėje įrangoje). Tačiau bendra tendencija daugumoje GIS produktų – atributų duomenims (ArcInfo) ar apskritai visiems geografiniams duomenims (ArcGIS) valdyti įdiegti DBVS.

5.2 Duomenų bazių valdymo sistemų tipai

Duomenų bazių valdymo sistemos skirstomos pagal duomenų kaupimo bei manipuliavimo jais būdus:

- Hierarchinės duomenų bazių sistemos (*Hierarchical Database Systems*).
- Tinklo duomenų bazių sistemos (*Network Database Systems*)
- Reliacinės duomenų bazių valdymo sistemos – RDBVS (*Relational Database Management Systems*).
- Objektinės duomenų bazių valdymo sistemos – ODBVS (*Object Database Management Systems*).
- Objektinės-reliacinės duomenų bazių valdymo sistemos – ORDBVS (*Object-Relational Database Management Systems*).

Šiuo metu GIS dažniausia naudojami paskutiniai trys DBVS tipai.

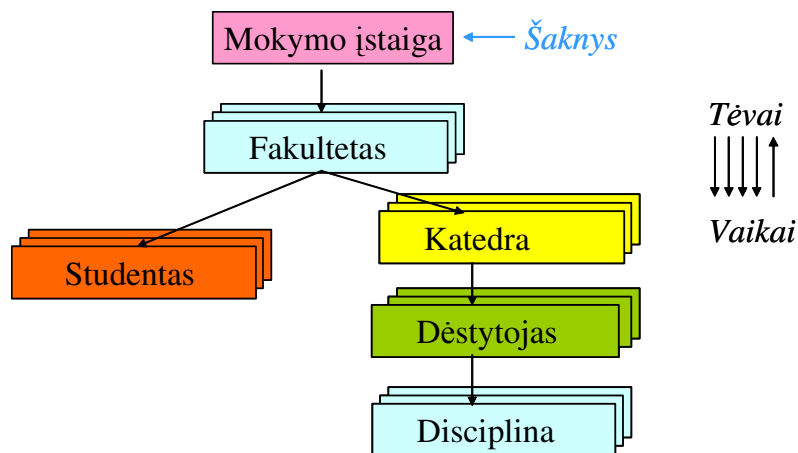
5.2.1 Hierarchinės duomenų bazių sistemos

Hierarchine vadinama DBVS naudoja hierarchinę įrašų struktūrą, organizuotą tarpusavyje sujungtais loginiais lygiais. Atitinkamo lygio įrašas kaupia duomenis, būdingus visiems žemesnio

lygio įrašams. Tarp to paties lygio įrašų ryšių nėra. Kiekviename įrašė yra laukas-raktas, kuris naudojamas hierarchijai organizuoti.

Hierarchinės struktūros viršūnė vadinama „šaknimis“. Kiekvienas duomenų bazės elementas, išskyrus „šaknis“, turi tik vieną su juo susijusį aukštesnio lygio elementą bei vieną ar daugiau jam pavaldžių žemesnio lygio elementų.

Naudojant šį modelį galima atvaizduoti duomenis, kurių organizacija tikrovėje yra hierarchinio tipo. Pavyzdžiui, universiteto organizacinė struktūra yra hierarchinio tipo: mokymo įstaiga – fakultetas – katedra – dėstytojas – dėstoma disciplina (5.3 pav.).

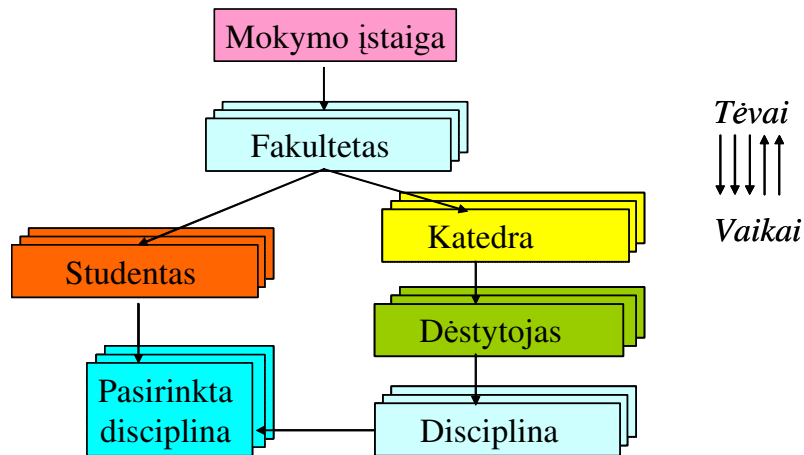


5.3 pav. Duomenų bazės organizacija naudojant hierarchinį duomenų modelį

Hierarchinės duomenų bazių sistemos yra lengvai suprantamos ir duomenys jose nesunkiai atnaujinami. Jos užtikrina aukštą priėjimo prie didelių duomenų kiekių greitį, nedideles bylų apimtis. Pagrindiniai hierarchinio duomenų modelio trūkumai yra sunkiai modifikuojami ryšiai tarp duomenų bei sudėtingas užklausų formavimas. Efektyviausia čia analizuoti duomenis, kai visos užklausos numatytos ir suprogramuotos iš anksto. Tačiau hierarchinio duomenų modelio reikšmė GIS yra gana ribota – jis kartais naudojamas kaupiant skaitmeninius žemėlapius.

5.2.2 Tinklo duomenų bazių sistemos

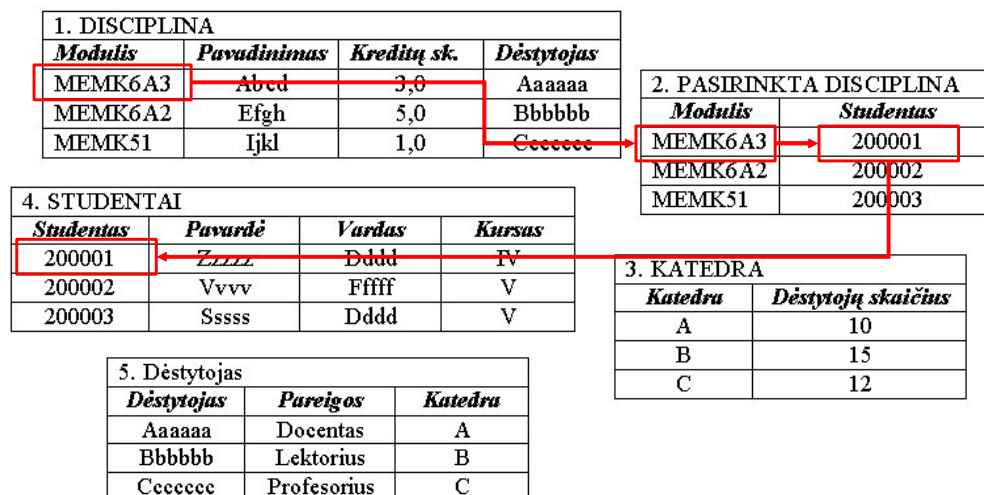
Tinklo duomenų modelyje žemesnio lygio elementas gali būti pavaldus daugiau nei vienam aukštesnio lygio elementui (5.4 pav.). Tai supaprastina sudėtingų realiame pasaulyje egzistuojančių ryšių tarp geografinių objektų atvaizdavimą. Tinklo duomenų modelis glaudesnis nei hierarchinis, duomenys, kaupiami skirtinguose kompiuterinės laikmenos sektoriuose, prieinami ir atgaminami sparčiau. Tačiau tinklo duomenų bazių sistemos sudėtingesnės nei hierarchinės.



5.4 pav. Duomenų bazės organizacija naudojant tinklo duomenų modelį

5.2.3 Reliacinės duomenų bazių valdymo sistemos

Reliaciniame duomenų modelyje atskirame įrašė nėra duomenų laukų hierarchijos – kiekvienas laukas gali būti naudojamas kaip raktas. Visa informacija yra kaupiama dvimatėse lentelėse, kurios, panaudojant identiškus atributų laukus, gali būti sujungiamos tarpusavyje (5.5 pav.). Naudojant reliacinį duomenų modelį šiandien yra kaupiama didžioji dalis geografinių duomenų.



5.5 pav. Duomenų bazės organizacija naudojant reliacinį duomenų modelį

Projektuojant reliacines duomenų bazes siekiama produktyvaus bei efektyvaus sprendimo. Teoriškai visą informaciją galime kaupti vienoje lentelėje. Tokiu atveju duomenų paieška taptų spartesnė, tačiau duomenų bazės nebūtų tokios kompaktiškos. 1970 metais Ted Codd iš IBM firmos suformulavo reliacinių duomenų bazių sudarymo taisykles:

- Langelyje, kur susikerta lentelės stulpelis ir eilutė, turi būti tik viena reikšmė.
- Vieno stulpelio reikšmės turi būti apie tą patį dalyką.
- Kiekviena eilutė – unikali (nėra pasikartojančių įrašų).
- Stulpelių išdėstymas nėra svarbus.
- Eilučių išdėstymas taip pat nėra svarbus.

5.2.4 Objektinės duomenų bazių valdymo sistemos

Objektinės duomenų bazių valdymo sistemos buvo sukurtos siekiant panaikinti kai kuriuos reliacinių sistemų trūkumus, kurių pagrindinis – negebėjimas tiesiogiai duomenų bazėse kaupti informacijos apie objektų būklę bei elgseną. RDBVS skirtos panaudoti sprendžiant įvairius verslo uždavinius – bankininkystėje, personalo valdyme, akcijų prekyboje ir pan., tačiau jų kūrėjai nemanė, kad teks spręsti klausimus, susijusius su geografinių objektų informacijos, garso, vaizdo kaupimu. RDBVS nėra optimalios vykdant daugybę geografinių užklausų. ODBVS gali būti naudojamos informacijai apie objektus kaupti bei pateikia daug priemonių objektinėms užklausoms. Objektinių duomenų bazių valdymo sistemų pavyzdžiai yra GemStone/S Object

Server, sukurta GemStone Systems Inc.; Objectivity/DB, sukurta Objectivity Inc.; ObjectStore (Progress Software); Versant (Versant Object Technology Corp.).

5.2.5 Objektinės-reliacinės duomenų bazių valdymo sistemos

Nors ODBVS techniškai pranašesnės už RDBVS, pirmosios nesulaukė tikėtos komercinės sėkmės dėl to, kad RDBVS neginčijamai vyravo rinkoje ir buvo labiau praktiškai taikomos, o jų kūrėjai operatyviai prisitaikė prie pakitusių vartotojų poreikių, įdiegdami pagrindines ODBVS galimybes į standartinius RDBVS sprendimus. Taip buvo sukurtos hibridinės objektinės-reliacinės duomenų bazių valdymo sistemos, kur standartinė RDBVS geba dirbti su objektais. Šiandien objektinėmis-reliacinėmis DBVS gali būti laikomi tokie produktai, kaip IBM DB2 ir Informix Dynamic Server, Microsoft SQL Server, Oracle. ORDBVS leidžia dirbti su įvairiais duomenų tipais (pavyzdžiui, MS Access: *Text, Memo, Number, Date/Time, Currency, AutoNumber, Yes/No, OLE Object, Hyperlink, Lookup Wizard*), duomenys gali būti perkelti iš bylų, duomenų bazių bei kitų taikomųjų programų, siekiant greitesnio priėjimo prie duomenų, vykdomas duomenų bazės indeksavimas.

5.2.6 Duomenų bazių valdymo sistemos ir geografiniai duomenys

Dažniausiai negalime nusakyti griežtų ribų tarp geografinių informacinių sistemų ir duomenų bazių valdymo sistemų, tačiau tam tikras funkcijų pasidalijimas yra. Pirmosios GIS ir automatizuoto kartografavimo sistemos duomenų bylas naudodavo tiesiogiai, be DBVS. Vėliau atsirado hibridinės sistemos, kuriose atributai buvo kaupiami standartinėse DBVS, paprastai reliacinėse. Objektų padėties informacijai valdyti bei apdoroti kurtos specialios programos, pasitelkiančios RDBVS ryšiui tarp objekto geometrijos bei atributų užtikrinti. Pavyzdžiui, per pastarąjį dešimtmetį populiariausia Lietuvoje programine GIS įranga tapusi ArcInfo iš principo sudaryta iš dviejų komponentų. Info yra komercinė reliacinė duomenų bazių valdymo sistema, skirta atributų duomenims valdyti. Vėliau Info funkcijas šiame produkte perėmė DBVS Tables, taip pat galima naudoti, pavyzdžiui, Oracle. PC ArcInfo versijoje darbui su atributais buvo naudojami komerciniai DBVS produktai: FoxPro, dBase. Geografinių objektų geometrijai kaupti

ir ja manipuluoti naudojama sistema Arc – čia atskiri moduliai atlieka geografinių objektų padėties įvedimo, redagavimo ir analizės funkcijas.

Dviejų pagrindinių komercinių DBVS gamintojai prie jų standartinių ORDBVS produktų neseniai pasiūlė papildomus modulius, palaikančius geografinių duomenų tipus ir funkcijas:

- IBM - DB2 Spatial Extender bei Informix Spatial Datablade.
- Oracle - Oracle Spatial.

Nors tarp šių sistemų yra esminių technologinių, paskirties bei galimybių skirtumų, jos leidžia kaupti, valdyti ir vykdyti nesudėtingą geografinę analizę.

Kitas sprendimas – įdiegti specialią GIS programinės įrangos gamintojų kuriamą įrangą, palengvinančią erdvinį duomenų valdymą standartinėse duomenų bazių valdymo sistemose:

- ESRI Inc. produktas ArcSDE, naudojamas Lietuvos georeferencinių ir daugelio kitų GIS duomenų bazių valdyme.
- Intergraph produktas GeoMedia Transaction Server.
- MapInfo produktas SpatialWare.

5.3 Duomenų bazių valdymo sistemose naudojamos užklausų kalbos

Duomenų paieškos tipai (arba užklausos) hierarchinėse arba tinklo modeliuose numatomi jau juos kuriant. Norėdamas surasti norimus duomenis tokiose duomenų bazėse, vartotojas privalo iš anksto žinoti jų struktūrą. Užklausų kalbos, kurios reikalauja duomenų bazės hierarchijos žinojimo, yra vadinamos procedūrinėmis užklausų kalbomis (*Procedural Query Language*).

Reliacinėse duomenų bazėse bei jų patobulintuose variantuose atributų hierarchijos nėra. Bet kuris atributas gali būti naudojamas kaip raktas informacijai atkurti, o duomenys skirtingose reliacinėse lentelėse gali būti apjungiami pagal jų bendrus atributų laukus. Šiuo atveju, norint pateikti užklausą, nėra būtina iš anksto žinoti duomenų bazės struktūros. Užklausų kalba, kuri nepriklauso nuo duomenų bazės struktūros, yra vadinama neprocedūrine užklausų kalba (*Non-procedural Query Language*). Labiausiai paplitęs tokios kalbos pavyzdys – 1970-aisiais metais

IBM sukurta SQL (*Structured* arba *Standard Query Language*). Šiuo metu dirbant su reliacinėmis duomenų bazėmis SQL yra *de facto* ir *de jure* standartas (ISO/IEC 9075).

5.4 GIS duomenų bazių projektavimas

Duomenų bazės projektuojamos trimis etapais:

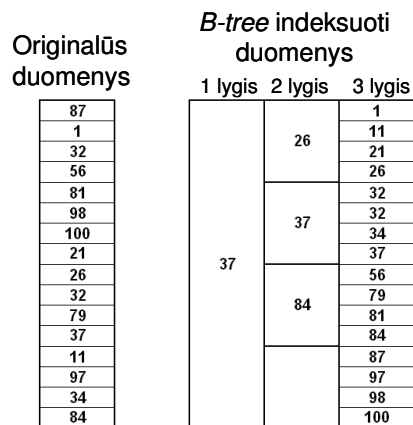
- Konceptinio modelio parengimas:
 - Vartotojo funkcijų nusakymas (pavyzdžiui, miškų valstybės kadastro vedimas, nepertraukiamos miškotvarkos vykdymas ir pan.); duomenų, reikalingų šioms funkcijoms užtikrinti, nusakymas;
 - Kai vartotojo funkcijos nusakytos, yra apibrėžiamos objektų klasės ir funkcijos, ryšiai tarp objektų klasių;
 - Pasirenkami geografinių objektų atvaizdavimo tipai (taškas, linija, poligonas).
- Loginio modelio parengimas:
 - Nusprendžiame, kaip susiesime objektų klases su specifiniais duomenų tipais, palaikomais GIS;
 - Nusakome reikiamus topologinius ryšius, taisykles, koordinačių sistemas ir pan.
- Fizinio modelio parengimas:
 - Sukuriama fizinė duomenų bazės schema (elementų duomenų rinkiniai, elementų klasės, atributų parametrai ir pan.).

5.5 Geografinės informacijos kaupimo optimizavimas

Kai geografiniai duomenys yra surinkti ir patalpinti GIS duomenų bazėje, galime optimizuoti jų struktūrą, taip palengvindami užklausų atlikimą, analizę bei kartografavimą. Yra du geografinių duomenų bazių optimizavimo sprendimai: topologijos sukūrimas bei indeksavimas. Topologijos sukūrimas bei jos teikiami privalumai aptarti 3 skyriuje „GIS naudojami geografinių duomenų modeliai“.

Geografinių duomenų bazės paprastai būna didžiulės apimties, todėl geografinė užklausa – kartais ilgai trunkanti procedūra. Atliekant duomenų paiešką indeksuotoje duomenų bazėje

išvengiame visų lentelės įrašų skaitymo, kadangi sukuriama speciali nuoroda, kuria atitinkantys duomenys kaupiami atskirame stulpelyje. Standartiniai DBVS naudojami indeksai (pavyzdžiui, *B-tree*¹) yra vienmačiai ir ribotai tinka geografiniams duomenims indeksuoti. 5.6 pav. pristatomas daugumoje komercinių DBVS naudojamas *B-tree* (*Balanced Tree*) indeksavimo metodas. Naudojant neindeksuotą duomenų bazę (originalius duomenis), bet kokios reikšmės paiešką sudarytų 16 testų, kurių metu būtų skaitomas kiekvienas duomenų bazės įrašas. *B-tree* indeksas išrikiuoja reikšmes didėjimo tvarka ir sudalija duomenų bazės įrašus į nustatyto dydžio grupes. Sakykime, norime surasti įrašą, kurio reikšmė lygi 81 *B-tree* indeksuotoje duomenų bazėje. Paiešką sudarytų daugiausiai šeši žingsniai: vienas pirmame lygyje (mažiau kaip ar daugiau kaip 37), vienas antrame lygyje (mažiau kaip ar daugiau kaip 84) ir visų keturių įrašų skaitymas trečiame lygyje.



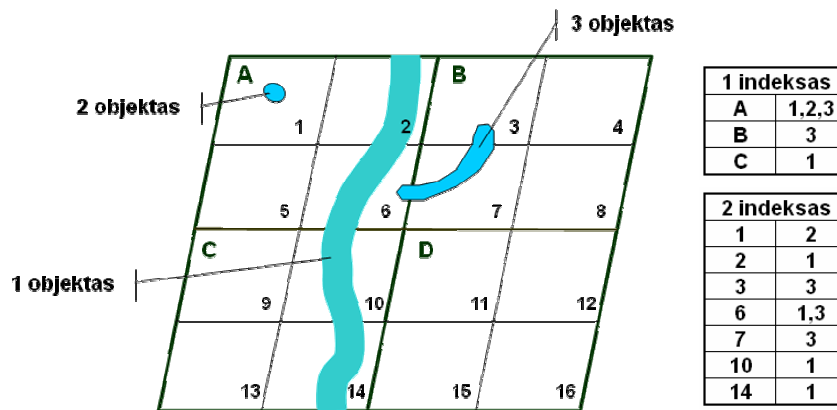
5.6 pav. *B-tree* indekso pavyzdys

Indeksų kūrimas ir atnaujinimas yra imli laikui procedūra. Be to, indeksai užima papildomą disko vietą. Tačiau be duomenų bazių indeksavimo kai kurių operacijų atliktis neįmanoma (pavyzdžiui, vienos vidutinio dydžio miškų urėdijos neindeksuotų taksacinių sklypų poligonų atributų bei sklypų taksacinių charakteristikų lentelių sujungimas, naudojant ArcInfo Workstation programinę įrangą bei standartinį kompiuterį, trunka keletą valandų. Atlikus indeksavimą, operacija trunka sekundes).

GIS dažniausiai taikomi 3 indeksavimo metodai: *GRID*, *Quadtree* ir *R-tree*.

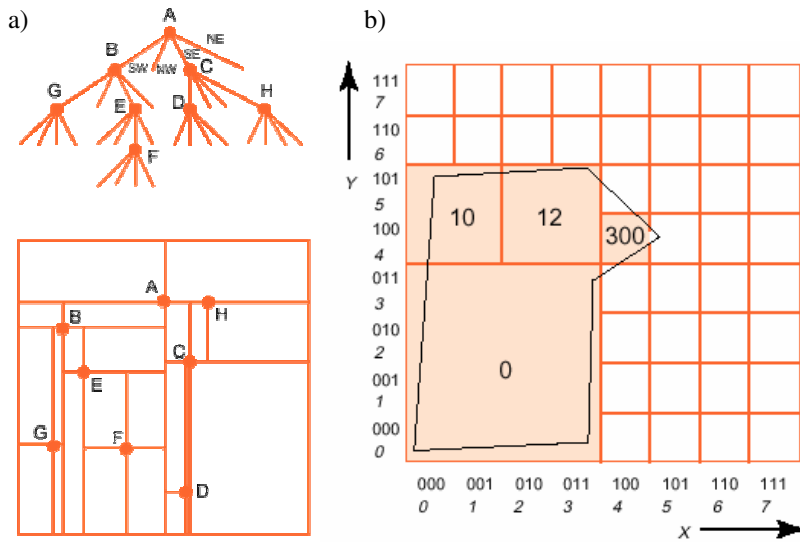
¹ Indeksavimo metodai čia ir toliau įvardijami vartojant jų originalius pavadinimus. Pažodinis vertimas dažnai iškreipia pavadinimo esmę

GRID indeksu suprantamas taisyklingas tinklas, kuris „užklojamas“ ant geografinių objektų. 5.7 pav. pateikti trys objektai, kurie indeksuojami naudojant du tinklo lygius. Pirmo lygio tinklas (1 indeksas) visą mus dominantį plotą suskaido į 4 vienodo dydžio gardeles. Visų trijų objektų dalys patenka į gardelę A, gardelėje B yra tik trečio objekto dalis, o ketvirtadalyje viso ploto – gardelėje D – nėra vieno iš objektų. Tokia pat procedūra pakartojama panaudojant antro lygio tinklą (2 indeksas), kol palyginus tiksliai nusakome viso ploto fragmentus, kuriuose yra dominantys geografiniai objektai.



5.7 pav. Geografinių duomenų bazių indeksavimas *GRID* metodu

Quadtree vadinama indeksų grupė, kuri grindžiama laipsnišku ploto dalijimu į kvadratus. Iš dalies tai tam tikras *GRID* indeksavimo variantas, kuris skiriasi tik tuo, kad *Quadtree* atveju erdvė nuosekliai dalijama į kvadratus pagal nuo duomenų tankumą. *Quadtree* naudojamas ir geografinėms duomenų bazėms suspausti. *Quadtree* algoritmai skirstomi pagal indeksuojamų geografinių objektų geometriją (taškai, linijos, poligonai, paviršiai ar geografinės matricos), pagal principus, kuriais vadovaujantis plotas yra dalijamas į kvadratus, ir pagal tai, naudojama fiksuota ar kintanti dalijimo skiriamoji geba. 5.8 pateikiami keli *Quadtree* indeksavimo pavyzdžiai.

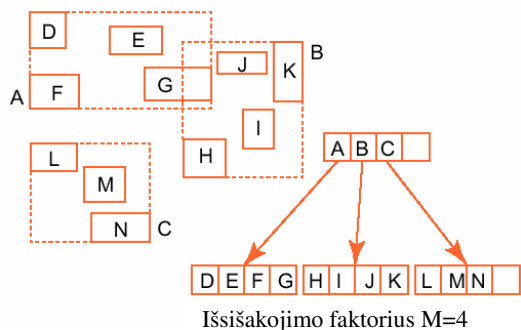


5.8 pav. Indeksavimo *Quadtree* metodu pavyzdžiai: a) taškų indeksavimas, b) regionų indeksavimas

Naudojant taškinį *Quadtree*, erdvė palaipsniui yra padalijama į keturis stačiakampius, priklausomai nuo taškų padėties. Pagal pirmo taško x ir y koordinates, plotas dalijamas į keturis nevienodo dydžio stačiakampius. Tai kartojama tol, kol dalijimui panaudojami visi taškai ir kiekvienam jų sukuriamas indeksas.

Regioninis *Quadtree* naudojamas linijoms, poligonams ir rastriniams duomenims indeksuoti. Indeksas kuriamas duomenų užimamą erdvę dalijant į keturis kvadratus, šiuos – į keturis smulkesnius kvadratus ir t.t. Jei tokio kvadrato gardelės objektas visiškai neužpildo, jis dalijamas toliau tol, kol objekto padėtį galime nusakyti jį formuojančių kvadratų numeriais (šie gali būti indeksuoti naudojant *B-tree* metodą).

Indeksuojant *R-tree* metodu, objektai grupuojami apibendrinant jų padėtį pagal patekimą į mažiausią ribojantį stačiakampį (5.9 pav.).



5.9 pav. Indeksavimas *R-tree* metodu

5.6 Redagavimas ir duomenų palaikymas

Redagavimu yra suprantami geografinių duomenų bazėje daromi pakeitimai, kurių metu pridami nauji objektai ar keičiami esami. Kaip duomenų bazės atnaujinimas yra suprantamas bet koks vieno ar daugiau objektų geometrijos ir/ar atributų pakeitimas arba bet koks duomenų bazės schemas pakeitimas. Sprendžiami tokie klausimai: geometrijos bei atributų redagavimas, duomenų bazės palaikymas (pavyzdžiui, sistemos administravimas), indeksavimas ir topologijos kūrimas bei redagavimas, duomenų importas bei eksportas, orientavimas geografinėje erdvėje ir pan. Paprastai GIS yra aprūpinta priemonėmis objektų geometrijai ir atributams redaguoti. Įvedami duomenys nuolatos yra išsaugomi duomenų bazėje. Siekiant užtikrinti duomenų saugumą ir kokybę, prieinamumas prie duomenų paprastai būna griežtai administruojamas.

Jau ne vienerius metus viena aktualiausių GIS duomenų valdymo problemų yra galimybių vienu metu su ta pačia duomenų baze dirbti keliems vartotojams užtikrinimas. Geografiniai objektai nėra vieni nuo kitų izoliuoti, paprastai jie glaudžiai susiję su kitais juos supančiais objektais. Todėl galima susidurti su problema, kai viena duomenų bazė tuo pačiu metu yra redaguojama kelių vartotojų. Kita problema iškils tokiu atveju, jei skirtingi vartotojai pateiks nevienodas to paties geografinio objekto geometrijos bei atributų versijas.

Bet kokią duomenų bazės redagavimo veiksmų grupę (pavyzdžiui, naujai atsiradusio kontūro įvedimą ar netikslių atributų koregavimą) pavadinkime operacija. Daugelis profesionalių GIS ir kitokių duomenų bazių leidžia vienu metu su ta pačia informacija dirbti ir operacijas atlikti daugiau nei vienam vartotojui. Vienam vartotojui pradėjus operaciją, duomenų bazė yra laikinai,

kol ši operacija pabaigiama, „užrakinama“. Bet koks kito vartotojo bandymas ką nors pakeisti yra automatiškai atmetamas.