

## Duomenų bazių teorija ir praktika

A.Ribikauskas, 2009

## Užsiėmimai

- Paskaitos
- Pratybos
- Laboratoriniai darbai
  - Sybase Power Designer
- Kursinis darbas

## Literatūra

- R.Elmasri, S.Navathe. Fundamentals of Database Systems. 3<sup>rd</sup> ed., Addison-Wesley, 2003.
- C.J. Date. Introduction to Database Systems, 7<sup>th</sup> ed., Addison-Wesley, 2003.
- E.J.Naiburg, R.A.Maksimchuk. UML for Database Design, Addison-Wesley, 2003.

## Literatūra

- R.Baronas. Duomenų bazių valdymo sistemos. TEV, 2005.
- R.Baronas. Duomenų bazių sistemos. TEV, 2002.
- <http://www.mif.vu.lt/~baronas/dbvs/book/index.htm>

## Informacinės sistemos

**Sistema** – vientisas objektas, kurio egzistencija yra tam tikro tikslo pasiekimas.

**Informacinė sistema** – tai tarpusavyje susiję priemonės, metodai ir personalas, naudojami informacijai saugoti, apdoroti bei teikti, siekiant pasirinkto tikslo.

## IS dalys

Pagrindiniai IS komponentai yra:

- techninė įranga (aparatura)
  - kompiuteriai
  - kompiuteriniai tinklai
- programinė įranga
  - sisteminė PĮ
  - taikomoji PĮ
- duomenys
- žmonės (personalas)

## IS raida

Pirmosios IS pasirodė 1950 m. Tada buhalterinei apskaitai jau buvo naudojamos pirmosios elektromechaninės buhalterinės mašinos.

Etapai:

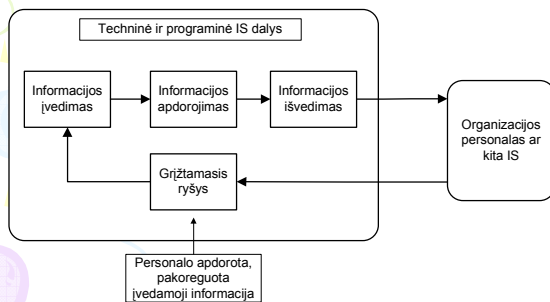
- 1950-60 m. dokumentų paruošimo ir apdorojimo spartinimas
- 1960-70 m. atskaitomybės paruošimo proceso spartinimas
- 1970-80 m. racionaliausio sprendimo priėmimas
- 1980- m. strateginis informacijos šaltinis, leidžiantis firmai gyvuoti ir klestėti.

## Informacinių sistemų naudojimas

Informacinės sistemos, kaip ir technologijos kuriamos, siekiant padidinti darbo našumą ir efektyvumą.

Informacinės sistemos užtikrina informacijos surinkimą, saugojimą, apdorojimą paiešką ir informacijos teikimą įvairiose srityse.

## IS vykstantys procesai



## Pagrindinės IS savybės

- kaip ir bet kokios kitos sistemos, IS gali būti analizuojama ir valdoma (sisteminis sudarymo principas)
- ji yra dinamiška ir besivystanti (ją galima modernizuoti)
- kiekviena IS turi teikti informaciją, kad būtų galima priimti valdymo sprendimus
- tai yra žmogaus – kompiuterio sistema (gali būti nekompiuterizuota, bet dabar tai retenybė)

## IS ir organizacijos struktūra

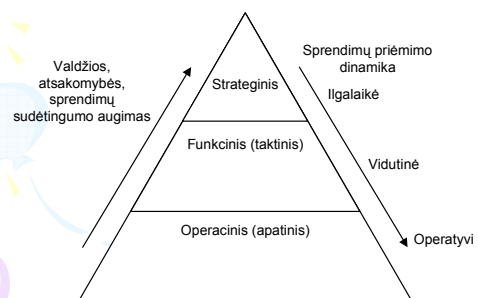
**IS yra organizacijos dalis**

(gali būti sukurta tik atlikus organizacijos valdymo struktūros analizę).

Valdymo funkcijos:

- organizacinė (org. struktūra (skyriai, etatai, pareigybės, pavaldumas) ir normatyvinė dokumentacija)
- planavimas
- apskaita
- analizė
- kontrolė
- stimuliavimas (finansinis, psichologinis) ir motyvacija
- sprendimų priėmimas

## Valdymo lygmenys



## IS panaudojimas

- Firma specializuojasi prekių tiekimu į tam tikras organizacijas, pvz., ligonines.
- Didelės prekių, produktų atsargos gerai, tačiau brangu saugoti, iššaldomos lėšos.
- Užsakovams statomi terminalai, ir jie gali tiesiai užsakyti norimas prekes.
- Užsakymai patenka į užsakymų apskaitos IS.
- IS padeda priimti operatyvinius sprendimus, kaip norimą prekę pristatyti užsakovui per minimalų laiką, atlieka statistinę analizę ir prognozę.
- Taip sutaupomos lėšos prekių atsargoms saugoti. Rezultatas – geresnis biznis, didesnis pelnas.

## IS struktūra

**Struktūra** – tai sudedamųjų sistemos dalių (posistemų) visuma.

**Posistema** – tai sistemos dalis, išskiriama pagal kokį nors požymį.

Galima skirstyti pagal:

- fizinę dislokaciją
- atliekamas funkcijas
- ...

## Duomenys ir informacija

**Duomenys** – tai saugomi, bet nenaudojami požymiai, stebėjimai.

Atsiradus galimybei ir poreikiui, duomenys panaudojami, ir tampa informacija.

**Informacija** – tai naudojami prasminiai ir susieti duomenys.

Bet kokios IS tikslas yra duomenų apie realaus pasaulio objektus apdorojimas.

## Duomenų bazė

**Duomenų bazė** – tai duomenų apie konkrečius objektus tam tikroje dalykinėje srityje visuma.

Dalykinė sritis (pvz., aukštoji mokykla, ...) – mus dominanti sritis, kurios objektai atvaizduojami duomenų pagalba.

Organizaciją domina tik informacija, padedanti siekti organizacijos tikslo. Turi būti saugomi tik tie duomenys, kurie reikalingi ir susiję su organizacijos veikla.

Pirmas žingsnis kuriant DB – duomenų sutvarkymas, t.y. duomenų struktūros sukūrimas.

## Realūs objektai ir juos atspindintys duomenys

Reikalingus saugoti duomenis ir jų tipus būtina nustatyti iki taikomosios programos algoritmo sudarymo.

Prieš atliekant kokias nors operacijas, reikia žinoti objektus, kuriems tos operacijos bus taikomos, bei aiškiai įsivaizduoti tų objektų, kurie bus gauti, struktūrą.

Galima išskirti objektus, turinčius tas pačias savybes, t.y. juos galima aprašyti panašiais duomenimis.

Tokie objektai sugrupuojami, o grupės (klasės) vadinamos informaciniais objektais.

Tolimesniame informacijos apdorojimo algoritmu (taisyklių rinkinių) ir procedūrų kūrime pakanka žinoti informacinio objekto savybes (objektų bendrąsias savybes).

## Duomenų tipai

- paprastieji (baziniai, nestruktūriniai)
- struktūriniai

Baziniai tipai (vienas vardas, viena reikšmė):

- sveikieji skaičiai
- realieji skaičiai
- tekstiniai duomenys (rašmenys)
- loginiai duomenys
- data, laikas
- grafiniai duomenys (paveikslukai)
- ...

## Sveikieji skaičiai

Galimos operacijos: **+**, **-**, **\***, be įprastų algebrinių veiksmų **div** ir **mod** operacijos.

$$10 \text{ div } 7 = 1$$

$$10 \text{ mod } 7 = 3$$

Priklausomai nuo sistemos, kurioje vaizduojami skaičiai, egzistuoja  $M_{+\infty}$  ir  $M_{-\infty}$  (didžiausias teigiamas ir mažiausias neigiamas skaičiai).

Neigiami skaičiai – kairysis bitas naudojamas ženklui.

Neigiamų skaičių vaizdavimui naudojamas **papildomas kodas** (vietoje skaičiaus **-x** vaizduojamas skaičius **2N - x**, kur **N** - skiriamų skaičių pavaizduoti bitų kiekis).

## Sveikieji neigiami skaičiai

Papildomas kodas gaunamas tokiu būdu:

1. Pavaizduojamas atitinkamas teigiamas skaičius.
2. Visi bitai invertuojami, t.y. 0 keičiamas 1 ir atvirkščiai.
3. Pridedamas 1.

```
32 766 01111111 11111110
32 767 01111111 11111111
-32 768 10000000 00000000
-32 767 10000000 00000001
... ..
-2 11111111 11111110
-1 11111111 11111111
```

Galioja:

- $M_{+\infty} + 1 = M_{-\infty}$
- $M_{-\infty} - 1 = M_{+\infty}$

## Kiti baziniai tipai

- **Realieji skaičiai.** Galimos operacijos: **+**, **-**, **\***, **/**. Skaičiavimai vykdomi tam tikru tikslumu, priklausomai nuo jų vaizdavimui skiriamos atminties kiekio.
- **Tekstiniai duomenys.** Pagrindinė jų savybė – galimybė palyginti. Lyginamas simbolio kodas.
- **Eilutės tipo duomenys.** Atskirais atvejais gali būti priskiriami prie bazinių ar prie struktūrinių duomenų tipų. Taip pat galima, palyginti, sudėti ar išskirti fragmentą.
- **Loginiai duomenys.** Tipas, galintis įgyti tik dvi reikšmes: **False** ir **True**. Atliekamos trys pagrindinės operacijos: **AND** (konjunkcija), **OR** (disjunkcija), **NOT** (inversija).

## SQL Server 2000 duomenų tipai

Tipas	Aprašas
bigint	Sveiki skaičiai nuo $-2^{63}$ iki $2^{63}-1$
int	Sveiki skaičiai nuo $-2^{31}$ iki $2^{31}-1$
smallint	Sveiki skaičiai nuo $-2^{15}$ iki $2^{15}-1$
tinyint	Sveiki skaičiai nuo 0 iki 255
bit	0 arba 1
decimal	Fiksuoto kablelio skaičiai nuo $-10^{38} + 1$ iki $10^{38} - 1$
numeric	Fiksuoto kablelio skaičiai nuo $-10^{38} + 1$ iki $10^{38} - 1$
money	Piniginės reikšmės nuo $-2^{63}$ iki $2^{63}-1$
smallmoney	Piniginės reikšmės nuo -214 748,3648 iki +214 748,3647
float	Slankaus kablelio skaičiai nuo $-1.79E+308$ iki $1.79E+308$
real	Slankaus kablelio skaičiai nuo $-3.40E+38$ iki $3.40E+38$
datetime	Data ir laikas nuo 1753.01.01 iki 9999.12.31 3.33 ms tikslumu
smalldatetime	Data ir laikas nuo 1900.01.01 iki 2079.06.06 1 min tikslumu

## Struktūriniai duomenų tipai

- tiesinės struktūros (eilutės ir pan.)
  - masyvai
  - sąrašai (įrašai)
  - stekai
  - eilės
- grafai
  - hierarchinės struktūros (medžiai)

## Masyvas

**Masyvas** – tai vieno tipo duomenų rinkinys.

Duomenys, vadinami masyvo elementais arba komponentais, yra sujungti bendru vardu (identifikatoriumi) ir adresuojami išskaičiuojamu indeksu. T.y. visi masyvo elementai turi tą patį vardą ir skiriasi tik savo indeksu.

Daugelio matavimų masyvas – jei indeksų ne vienas.

## Sąrašas

**Sąrašo** duomenų elementai (DE) atskiriami vienas nuo kito nustatant jiems pastovų ilgį, t.y. tas pats DE užima atmintyje vienodą baitų kiekį.

DE, kurį sudaro skirtingų tipų duomenys, vadinamas **įrašu (record)**.

Sudarant sąrašo struktūrą, nurodomi jo įrašų sudarančių domenų baziniai tipai ir ilgiai.

Pvz., sąrašo apie detales struktūra gali būti aprašyta taip:

Detalė	Nr.	Ilgis
CHAR*20	INT*2	REAL*4
Varžtas	16	35,6

## Sąrašų tipai

- **Vientisųjų sąrašų** įrašai saugomi bendroje vientisoje fizinių adresų erdvėje ir yra panašūs į vienmačius masyvus (įrašas ekvivalentus masyvo elementui).

Vientisą sąrašą privalumas – greitas reikalingo įrašo radimas. Pvz., įrašo apie detalę ilgis  $L=26$  baitai. Jei reikia  $n$ -to įrašo, tai jis prasideda nuo  $(n-1)*L+1$  baito, t.y. jį rasti nesunku.

- **Sutvarkytieji sąrašai** – tai sąrašai, kuriuose įrašai išdėstyti eilės tvarka pagal išrūšiuotas kurio nors įrašų duomens (požymio) reikšmes.

Sąrašo privalumas – paprasta rasti įrašą pagal išrūšiuoto požymio reikšmę.

## Rodyklės

Naudojant sutvarkytus sąrašus ir pridėdant DE ar jį trinant, reikalingas nuoseklus sąrašo perrašymas, kuris yra ypač neefektyvus išorinėje duomenų laikmenoje.

Išėitis – rodyklės (**pointer**) naudojimas. Kadangi fizinis įrašo adresas yra skaičius, galima įrašyti jį į kitą atminties ląstelę ir ją naudoti kaip nuorodą į minėtą įrašą.

**Rodyklė** – atminties ląstelė, sauganti kitos ląstelės adresą.

Rodyklę galime traktuoti kaip būdą nurodyti kitos ląstelės buvimą. Jų pagalba galima kurti sudėtingas duomenų struktūras ir jas valdyti.

## Rodyklių panaudojimas

Pvz., bibliotekoje yra knygų katalogas pagal jų pavadinimus. Jame lengvai rasime ieškomo pavadinimo knygą, tačiau sunku bus rasti visas vieno autoriaus knygas.

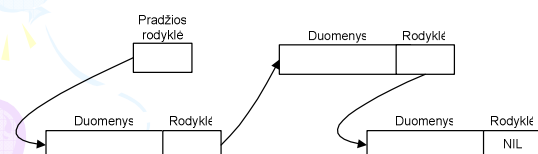
Kiekvienos knygos kataloge aprašyme įdėjus rodyklę į sekančią to paties autoriaus knygą kataloge – problemos nebėra.

...			
30	Atsisveikinimas su ginklais	E. Hemingvėjus	105
...			
105	For whom the bell tolls	E. Hemingvėjus	509
...			
509	Senis ir jūra	E. Hemingvėjus	820
...			

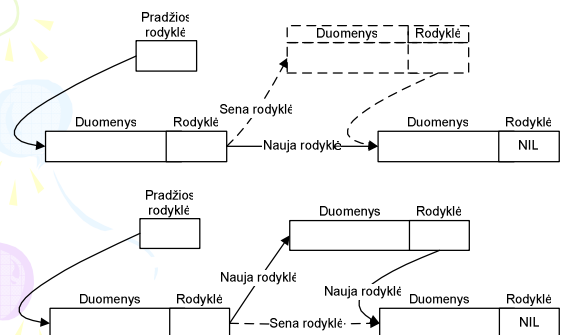
## Nuorodinis sąrašas

**Nuorodiniai sąrašai**, kuriami naudojant rodykles, yra žymiai dinamiškesni.

Nuorodinis sąrašas gaunamas, vientisą sąrašą kiekvieną DE papildant rodykle.



## Įrašo trynimas ir įterpimas



## Sąrašai ir masyvai

Naujas DE išsaugomas bet kurioje laisvoje atminties ląstelėje – sąrašas gali būti ne vientisas.

Į kiekvieną įrašo lauką galima kreiptis įrašo bei lauko vardu. Pvz. DE įrašas vardu **B**. Tada galima kreiptis **B.Detale**, **B.Nr.**, **B.Ilgis**.

Ir masyve ir įrašė galima tiesiogiai kreiptis konkrečių elementu.

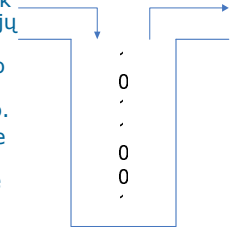
Skirtumai:

- įrašė gali būti skirtingi duomenų tipai
- masyvo elementų indeksacija lankstesnė, nes galima apskaičiuoti reikalingą indeksą

## Stekas

Apribojant vientisą sąrašą galimybe pridėti ar šalinti tik galinius sąrašo elementus, jų panaudojimas yra patrauklesnis, nes nesudaro "skylių" sąrašė, reikalaujančių jo perrašymo.

**Stekas** – tai sąrašas, kuriame visi įterpimo ir šalinimo veiksmai atliekami viename gale, t.y. paskutinis į jį patalpintas elementas paimamas pirmuoju (**LIFO – Last In First Out**).



## Eilė

**Eilė** – tai toks sąrašas, kur duomenų pridėjimas vyksta viename sąrašo gale, o paėmimas – kitame (**FIFO – First In First Out**).

Eilės ilgis iš anksto neapibrėžtas ir gali būti begalinis.



## Grafai

**Grafus** nagrinėja grafų teorija.

Grafą **G** sudaro dvi elementų aibės:

1. viršūnių aibė **V**
2. briaunų aibė **B**

Galima užrašyti  $G=(V,B)$ .

Briauna paprastai turi kryptį iš vienos viršūnės į kitą.

Tam tikros grafų rūšys turi savo pavadinimus.

## Medis

**Medis** – tai grafas, kuriame nėra ciklų.

- Yra viena viršūnė, į kurią neateina jokia briauna (yra tik išeinančios). Tai **šaknis**.
- Iš šaknies į kiekvieną viršūnę veda vienintelis kelias.

Praktiškai įdomesni dvejetainiai (**binary**) medžiai, kur kiekvienam mazgė atsišakojama dviem kryptim.



## Hierarchinės struktūros

Nereguliarūs duomenys dažnai gali būti pateikiami kaip **hierarchinės struktūros**. Pvz., pašto, interneto adresų sistemos.

Hierarchinėje struktūroje kiekvieno elemento adresas apibrėžiamas kreipties keliu (maršrutu), vedančiu nuo struktūros viršūnės iki ieškomo elemento. Kelio užrašymo būdai gali būti įvairūs.

## DB raida. Duomenų apdorojimo sistemos

Pirmosios IS, skirtos paspartinti rankinių duomenų apdorojimą, dar vadinamos **duomenų apdorojimo sistemomis**.

Duomenų apdorojimo sistema – tai automatizuota organizacijos dokumentuose esančių duomenų apdorojimo sistema.

Mėgdžiojant rankines operacijas, panašiai dirbama ir kompiuteryje. Vieno dokumento ar dokumentų aplanko informacija talpinama failu. Keitimasis duomenimis – keitimasis failais.

## Failinės duomenų sistemos

Iki ~1960 m. prieiga prie failų duomenų tik nuosekli. (Informacijos nešėjai – magnetinės juostos, perfojuostos ir pan.).

Darbas vyksta paketiniu režimu, dažniausiai naktimis apdorojami per dieną sukaupti duomenys.

Norint gauti surūšiuotas ataskaitas – rūšiavimui vykdoma duomenis apdorojanti transformacija.

Vėliau pradėti naudoti tiesioginės kreipties failai. Bet tai neišsprendė daugelio problemų.

## Failinių sistemų trūkumai

- Duomenų perteklumai
- Silpna duomenų kontrolė (duomenų elementai – sinonimai bei omonimai (tas pats žodis – skirtinga prasmė))
- Nepakankamos duomenų valdymo galimybės (nėra ryšių tarp atskirų duomenų grupių, pvz., pirkimo ir realizacijos prekyboje)
- Didelės sąnaudos programuojant (kiekvienam uždaviniui reikalingos savitos duomenų struktūros, didelės programavimo sąnaudos duomenims pasiekti į darbą su failais orientuotose aplinkose)
- Failų formatų nesuderinamumas

## Duomenų bazių valdymo sistemos

**Duomenų bazių valdymo sistema (DBVS, Database Management System)** – tai programinių priemonių kompleksas, leidžiantis:

- Aprašyti (sukurti) duomenų struktūras
- Įvesti duomenis (aktualaus būvio palaikymas)
- Duomenų paieška (svarbiausia funkcija)
- Įvairių ataskaitų sudarymas

Papildomos funkcijos:

- Duomenų vientisumo palaikymas
- Duomenų neprieštaringumo kontrolė
- Duomenų apsauga
- Kelių nepriklausomai dirbančių vartotojų vienašališkas darbas su duomenimis
- Duomenų bazės būsenos atstatymas po avarijų

## Duomenų bazė Duomenų bazių sistema

Kadangi dažniausiai kalbama apie kompiuterizuotą duomenų apdorojimą, tai susiaurėja ir duomenų bazės supratimas:

**Duomenų bazė (DB)** – tai rinkinys logiškai tarpusavyje susijusių duomenų vienetų, kurie gali būti apdorojami viena ar keliomis taikomosiomis programomis (ar programų sistemomis), siekiant patenkinti organizacijos informacinius poreikius.

Duomenų bazė kartu su DBVS vadinama **duomenų bazės sistema**. Jai priskiriami ir vartotojai. Vartotojai šiuo atveju – tiesiogiai DB naudojantys asmenys (pvz., DB administratorius) bei DB naudojančios taikomiosios programos.

## Taikomųjų programų ir duomenų nepriklausomumas

Duomenų ir taikomųjų programų nepriklausomumas pasiekiamas, kai DB turi savo **sisteminiame kataloge** metaduomenis (**meta-data**, duomenys apie duomenis), sudarančius duomenų žodyną (**data dictionary**).

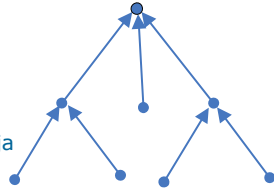
Išorės vartotojai mato pirmiausia šį žodyną, ir tik tada naudojami duomenimis, todėl pakeitus DB struktūrą, vartotojas atskirais atvejais to gali nepastebėti (pvz., pailginus lauką 1 baitu).

## Hierarchinis duomenų modelis

Pirmiausia pritaikyti hierarchinės struktūros DB modeliai.

Struktūrose naudojami rodyklės – tai fiziniai įrašų adresai.

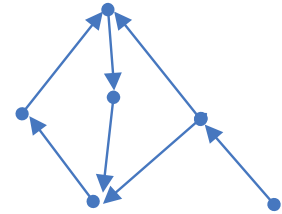
Hierarchinė struktūra – nepakankamai lanksti, ja negalima sumodeliuoti daugelio dalykų.



## Tinklinis duomenų modelis

Veikiai buvo pradėtas naudoti tinklinis DB modelis.

Modelyje taip pat naudojamos rodyklės.



## Reliacinės DB

Populiariausias šiuo metu – **reliacinis (sąryšinis) DB modelis**.

1970 m. E. F. Codd straipsnis apie reliacinį modelį. Svarbiausi informacijos apdorojime yra ryšiai tarp duomenų. Iki tol atskirų failų skirtingų įrašų ryšių palaikymui buvo įterpiamos tik fizinės rodyklės (fiziniai disko adresai).

E. F. Codd pademonstravo, kad tada stipriai susiaurinamos duomenų manipuliavimo galimybės. Tokios DB labai jautrios fizinės aplinkos pokyčiams. Pvz. keičiant diską, reikia atlikti ir duomenų restruktūrizavimą.

## Reliacinės DB

Reliacinis modelis pagrįstas loginiu duomenų ryšiu, todėl problemų lieka nebedaug.

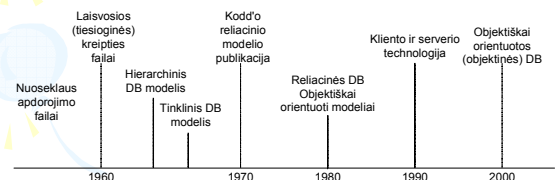
Reliacinė DB valdymo sistema (reliacinė sistema) – tai sistema, kurioje išpildomos dvi pagrindinės sąlygos:

1. Duomenys (vartotojo požiūriu) yra talpinami į lenteles, vad. **sąryšiais** (santykiais, reliacijomis, **relations**).
2. Vartotojui darbui su DB yra siūlomi operatoriai, kurių pagalba iš esamų lentelių gali būti generuojamos naujos (reliacinė algebra).

## DB modelių savybės

DB modelis	Savybės
Nuoseklosios kreipties failai	Visi įrašai apdorojami (skaitomi ir rašomi) nuosekliai.
Laisvosios (tiesioginės) kreipties failai	Tiesioginė kreiptis į įrašą. Sudėtinga rasti daug įrašų, susietų su vienu.
Hierarchinė DB	Palaiko kreiptį į keletą įrašų, susietų su vienu. Ryšiai tik hierarchiniai. Naudoja fizines iš anksto apibrėžtas nuorodas.
Tinklinė duomenų bazė	Palaiko tinklinius ryšius. Naudoja fizines iš anksto apibrėžtas nuorodas.
Reliacinė DB	Palaiko loginius ryšius tarp duomenų. Loginė kreiptis, nepriklauso nuo fizinės realizacijos.

## DB vystymasis



## DB kūrimo ideologija

DB sistemos atsiradimas – tai informacinės sistemos kūrimo paradigmos pakeitimas:

- Kol nebuvo kuriamos duomenų bazės, o naudojamos failinės sistemos – pirmiausia buvo kuriamos taikomosios programos reikalingų atskiriems organizacijos padaliniais ataskaitų generavimui, o po to – generuojamos kažkokios ataskaitos.
- Atsiradus DBVS, pirmiausia apmąstoma, kokie duomenys bus reikalingi, kuriama duomenų struktūra, ir tik po to kuriamos taikomosios programos, pateikiančios reikalingą informaciją. Bloga duomenų struktūra – kančia visiems vėliau programuojantiems, sprendžiantiems konkrečius uždavinius.

## DBVS privalumai

- + Perteklinių duomenų kontrolė
- + Duomenų neprieštaringumas
- + Duomenų vientisumas (reikšmių kontrolė įvedant)
- + Lakoniškumas (ta pati informacija – mažas duomenų tūris)
- + Bendras duomenų naudojimas
- + Duomenų apsauga
- + Standartizacija
- + Augant sistemai, auga lėšų panaudojimo efektyvumas (pakanka vienos terpės duomenims)
- + Didesnis našumas kuriant sistemas
- + Geresnis duomenų prieinamumas (aptarnaujant galutinį vartotoją)
- + Geresnis lygiagrečių užklausų valdymas
- + Būklės atstatymas po trykio ar avarijos

## DBVS trūkumai

- Sudėtingesnės
- Dideli DBVS poreikiai aparatiniams ištekliams
- Didelė DBVS kaina
- Didelė diegimo kaina
- Menkas atskirų uždavinių sprendimo našumas (sprendžiama daug užduočių vienu metu)
- Išėjus sistemai iš rikiuotės, nebegali dirbti nė vienas (centralizacija)