

# DBVS realizavimas

---

- Duomenų saugojimas diske
  - Duomenų saugumas
- Duomenų tipų ir struktūrų saugojimas
  - Fiksuoto ir kintamo ilgio duomenų tipų realizavimas
  - Fiksuoto ir kintamo dydžio įrašai
  - Blokų ir įrašų adresavimas diske: duomenų bazės erdvės adresai.
  - Nuorodų konvertavimas tarp duomenų bazės adreso ir operatyvios atminties adreso (swizzling).
  - Labai didelių binarinių objektų saugojimas DBVS

## Diskų gedimai

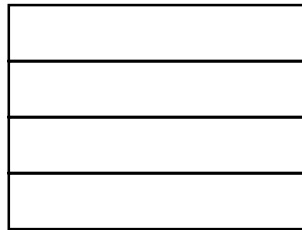
- Nenuolatinės (intermittent) sektoriaus klaidos.
- Nuolatinė (permanent) sektoriaus klaidos.
- Viso disko lūžis (disk crash)

# Apsauga nuo duomenų praradimo: kontrolinės sumos

- Parity check
- Stable storage
- RAID

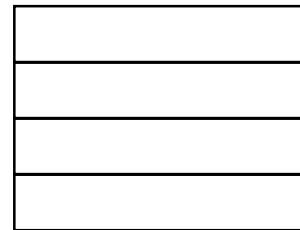
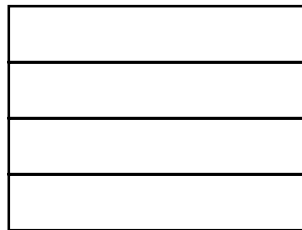
# RAID lygiai

□ 1 lygio RAID (mirroring):

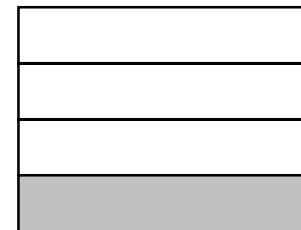
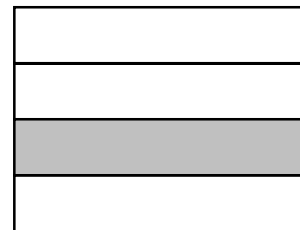
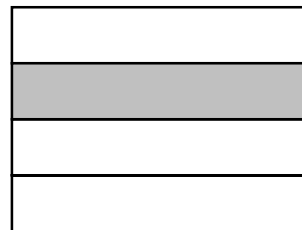
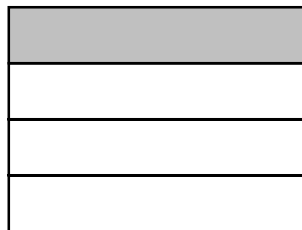


- Duomenys
- Papildoma kontrolė

□ 4 lygio RAID:



□ 5 lygio RAID:



# Terminai

- Tertiary Storage
- Disk/ Secondary Storage
- Blocks
- Sectors
- Disk Controller
- Seek Time
- Rotational Latency
- Transfer Time
- Mirroring
- Elevator Algorithm
- Two-Phase, Multiway Merge-Sort
- Checksums
- Stable Storage
- Parity Check
- Intermittent Failure
- RAID

## SQL duomenų tipai

- Fiksuoto ilgio duomenų tipai
  - CHAR( $n$ )
  - INTEGER
  - DATE
- Kintamo ilgio duomenų tipai
  - VARCHAR( $n$ )
  - DECIMAL( $m, n$ )
  - DATETIME
- Labai dideli binariniai objektai
  - BLOB

## Simbolinių eilučių realizavimas

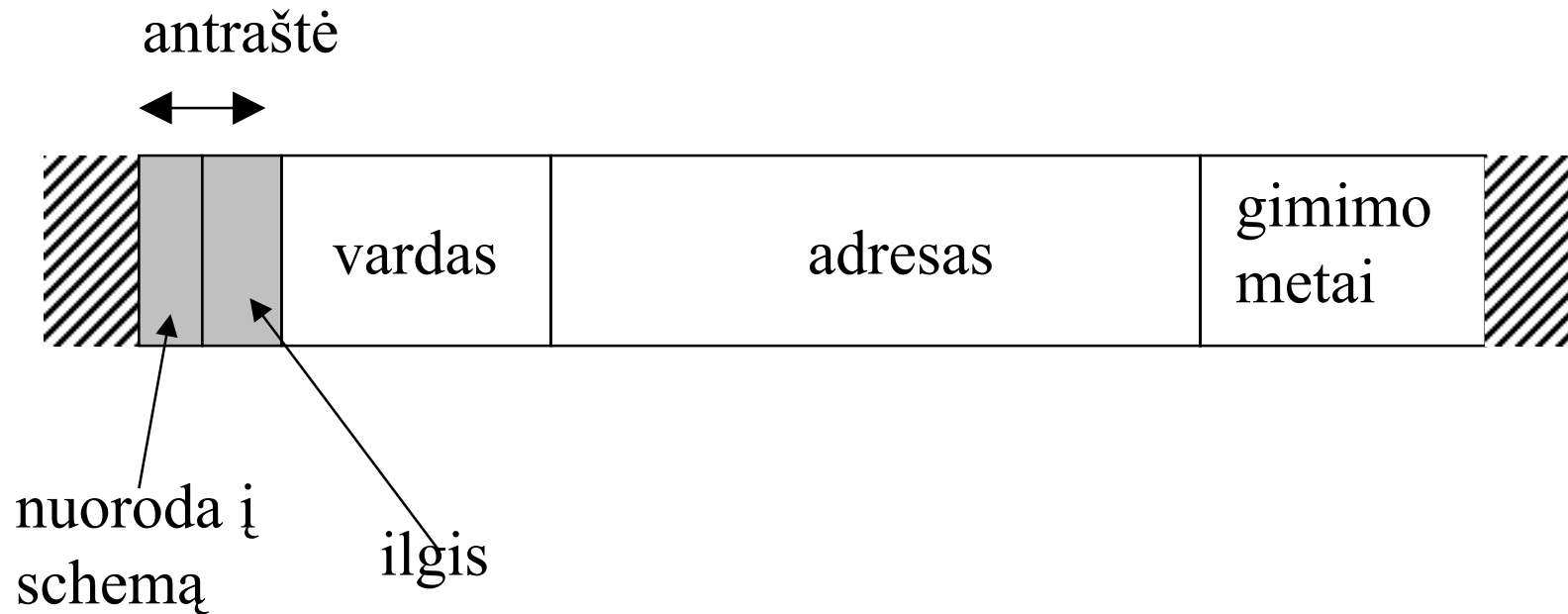
### □ CHAR(*n*):

- Išskiriama *n* baitų. Jei įrašomi duomenys trumpesni negu *n* baitų, prirašoma “tuščių” simbolių

### □ VARCHAR(*n*)

- Pradžioje saugomi 1 arba 2 baitai rodantys sekos ilgį. Paskui juos saugomi sekos simboliai.
- Saugomi sekos simboliai. Sekos pabaigą žymi specialus simbolis *nul*.

# Fiksuoto ilgio įrašai



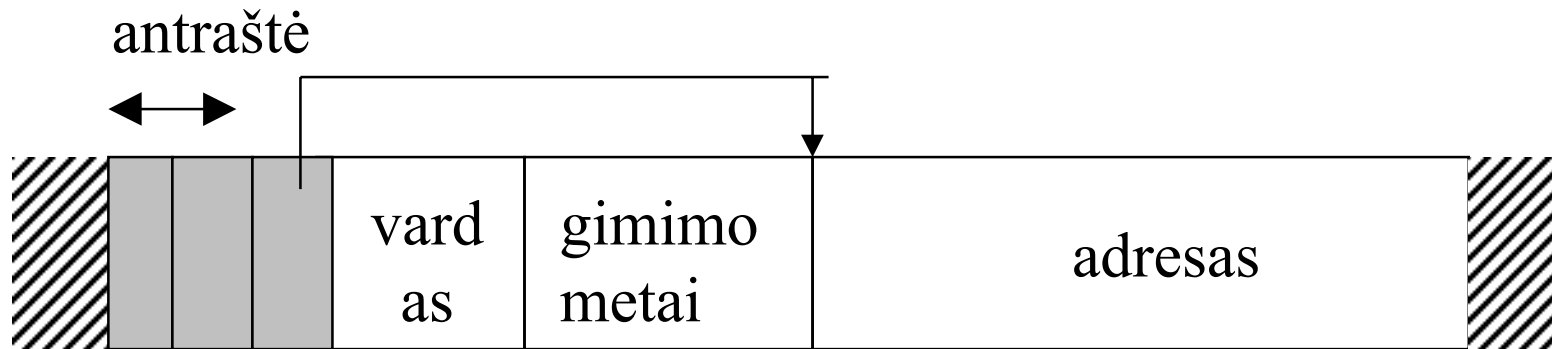
- Laukeliai fiksuoto ilgio, išlygiuoti ties keturių baitų riba
- Įrašo antraštė saugo duomenis bendrus visam įrašui.
- Įrašai pakuojami bloke.
- Blokas gali turėti savo antraštę.



## Kintamo ilgio įrašai

- Įrašai su kintamo ilgio laukais
- Įrašai su laukais, saugančiais reikšmių sąrašus (repeating fields)
- Kintamo formato įrašai

## Įrašai su kintamo ilgio laukais



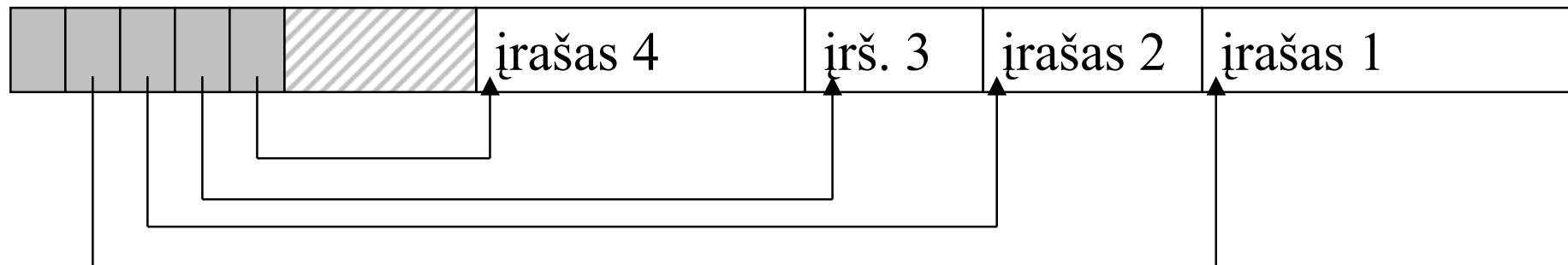
- Fiksuoto ilgio laukai saugomi pirmiausia. Po jų saugomi kintamo ilgio laukai.
- Įrašo antraštėje saugoma:
  - Nuoroda į įrašo schemą
  - Įrašo ilgis
  - Nuorodos į visus kintamo ilgio laukus.
- Galima saugoti fiksuoto ilgio įrašą, o kintamo dydžio laukelius pakeisti nuorodomis į kitame bloke saugomas reikšmes

# Įrašų dėstymas bloke

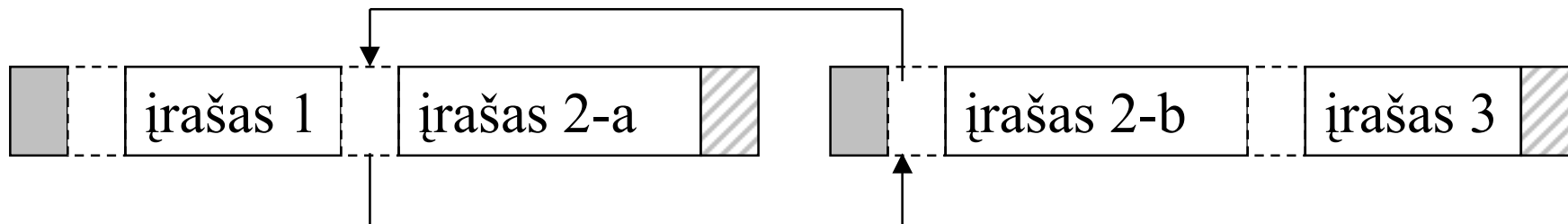
□ Fiksuoto ilgio įrašai, keletas įrašų viename bloke:



□ Kintamo ilgio įrašai, keletas įrašų viename bloke:



□ Vienas įrašas saugomas keliuose blokuose:



## Blokų, įrašų ir laukelių adresai

- Įrašai saugo nuorodas į laukelius,
- blokai saugo nuorodas į indeksus,
- indeksai saugo nuorodas į blokus ir t.t.
  
- Fyzinis DB adresas:
  - Baitų seka vienareikšmiškai nurodanti duomenų vietą DBVS diske
  - Mašinos vardas, kontrolerio numeris, disko numeris, bloko numeris, poslinkis bloke
- Loginis adresas:
  - Kiekvienam blokui priskiriamas unikalus numeris. Konvertavimas tarp fizinio ir loginio adreso vykdomas lentelės pagalba.

## Swizzling

- Atmintyje esantys blokai gali būti adresuojamas *DB* adresu arba *atminties adresu*.
- Diske esantys blokai adresuojami tik *DB* adresu.
- Pakraunant blokus į atmintį pildoma speciali lentelė, siejanti *DB* adresą su *atminties adresu*.
- *DB* nuorodas esančias bloko viduje reikia taip pat konvertuoti į atminties adresus (swizzling).
- Konevertavimas gali būti atliktas tada, kai blokas į kurį yra nuoroda taip pat bus pakrautas į atmintį. Iki to laiko jis turi būti adresuojamas *DB* adresu.
- Bloko adresas atmintyje:
  - *DB adresas / atminties adresas + vėliavėle nurodanti adresotipą*

## Swizzling (tęsinys)

- Pakraunant bloką į atmintį, jo viduje randamos visos nuorodos (pagal įrašų schemą) ir įvedame į *transliavimo lentelę* (jei jų ten dar nėra). Taip pat lentelėje saugomi bloko įrašų adresai ir pačio bloko adresas.
- Išankstinė strategija: visų nuorodų swizzling'as atliekamas iš anksto, pirmo bloko pakrovimo metu
- Atidėliojimo strategija: nuorodų swizzlingas atliekamas pirmo pareikalavimo metu.
- Galima iš viso nenaudoti swizzling'o
- Gražinant bloką į diską, vyksta atgalinis visų į jį rodančių nuorodų transliavimas į DB adresą.
- Blokai, kurių adresas *swizzlintas*, pažymimi speciali atžyma (pinned block)

## Įrašų modifikavimas - įterpimas

- Jei įrašai rūšiuojami (pvz., pagal pirminį raktą), įrašai perstumdomi bloko viduje.
- Nesunku perstumdyti įrašus, jei įrašai bloko viduje adresuojami poslinkių lentele.
- Jei naujai įterpiamam įrašui nepakanka vietos bloko viduje:
  - Rask vietos kaimyniniame bloke.
  - Naudok specialų “perpildymo” (overflow) bloką. Bloko antraštėje paliekama speciali vieta “nuoroda į perpildymo bloką”.

## Įrašų modifikavimas - pašalinimas

- Naikinant įrašus gali tekti:
- Koreguoti bloko antraštėje esančią įrašų poslinkių lentelę.
- Jei sunaikinamas paskutinis įrašas “perpildymo” bloke, sunaikinti patį perpildymo bloką.
- Pakeisti bloko antraštėje esančią nuorodą į įrašą specialia nuoroda – antkapiu (tombstone) ☺



## Įrašų modifikavimas - pakeitimas

- Fiksuoto ilgio įrašams joki pokyčio.
- Kintamo ilgio įrašams pasekmės panašios į įterpimą/pašalinimą – gali tekti žaisti su bloko antraštėje esančia poslinkių lentele ir perpildymo blokais..

## BLOB

- BLOB – binary large object: piešinukas, Excel lentelė, AVI arba MPEG filmas ir kt.
- Saugoma daugelyje blokų.
- Gali būti saugomi ištisais cilindrais, kad pagreitinti priėjimo laiką ir sumažinti fragmentaciją.
- Dažnai dirbama apeinant tranzakcijų žurnalus.
- Dažniausiai palaikomos operacijos: padėti. nuskaityti, sunaikinti. Kartais leidžiama skaityti “iš vidurio”.
- Gali būti naudojami netradiciniai indeksai, pavyzdžiui, filmams, bloko paieška pagal sekundžių skaičių nuo filmo pradžios.

## Terminai

- Fields
- Records: fixed and variable length
- Spanned Records
- BLOB
- Offset Table
- Overflow Block
- Database Address
- Pointer Swizzling
- Tombstone
- Pinned Blocks