

# მონაცემთა ბაზების ძირითადი კონსტრუქციები

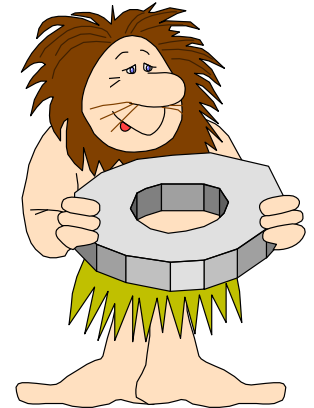


# რატომ მონაცემთა ბაზები?

---

## მონაცემთა ბაზები უზრუნველყოფს:

- ❑ დიდი მოცულობის მონაცემთა შენახვას, მართვას და კონტროლს
- ❑ მონაცემთა ინტეგრაციას
- ❑ მონაცემთა მოპოვებას და ანალიზს გარკვეული მექანიზმის მეშვეობით, “SQL”
- ❑ ე.წ. “Transactional integrity”  
მრავალრიცხოვანი მომხმარებელი  
მრავალრიცხოვანი ედიტირება



# რა არის მონაცემთა ბაზა?

---

- ❑ დიდი მოცულობის, სტრუქტურული, მონაცემებისა ინტეგრირებული კოლექცია
- ❑ ეფუძნება მოდელს, რომლის რელობას გამოსახავენ
- ❑ იყენებს კომპიუტერულ პროგრამას, რომელიც ორგანიზებას უკეთებს მონაცემთა ბაზების შენახას და მართვას  
(მონაცემთა ბაზების მართვის სისტემები/  
database management system-DBMS)



# რატომ ვიყენებთ მონაცემთა ბაზის მართვის სისტემებს (DBMS)?

---

- ❑ მონაცემთა დამოუკიდებლობა და ეფექტურად მოპოვება და გამოყენება
- ❑ ამცირებს მონაცემებთან მუშაობის ღირს
- ❑ მონაცემთა მთლიანობა და უსაფრთხოება
- ❑ მონაცემთა ერთგვაროვანი ადმინისტრირება
- ❑ მონაცემთა ერთდროული მოხმარება და მართვა (სხვადასხვა მომხმარებლის მიერ)



# მონაცემები, მონაცემთა ბაზები და მონაცემთა ბაზების სისტემები

---

- ❑ მონაცემები- წარმოდგენილი სხლასხვა ფორმატით, განთავსებული მონაცემთა ბაზაში
- ❑ ინფორმაცია- აღამიანის მიერ ინტერპრეტირებული მონაცემები
- ❑ ცოდნა/შეცნობა-აღამიანის მიერ მონაცემთა გაანალიზება (სხლასხვა საინფორმაციული წყაროებიდან)

# მონაცემები, ინფორმაცია, ცოდნა/შეცნობა

---



# მონაცემები vs მონაცემთა ბაზების მართვის სისტემები (DBMS)

---

- DBMS გვთავაზობს მეთოდებს მონაცემთა ორგანიზებისთვის, რომლის ძირითადი პრინციპებია:
  - ყველა მონაცემი განთავსებულია შესაბამის ცხრილში
  - ყველა ჩანაწერის იდენტიფიცირება არის შესაძლებელი ცხრილური მონაცემები დაკავსშირებულია (cross reference) სხვა ჩანაწერებთან, სხვა ცხრილებთან
- ეს მეთოდები არ არის სპეციფიკური, ისინი შეიძლება გამოყენებული იქნას ნებისმიერი მონაცემთა ბაზებისთვის. ისინი ერთობლიობაში DBMS-ს მონაცემთა მოდელს ქმნიან

# რა არის მონაცემთა მოდელი?

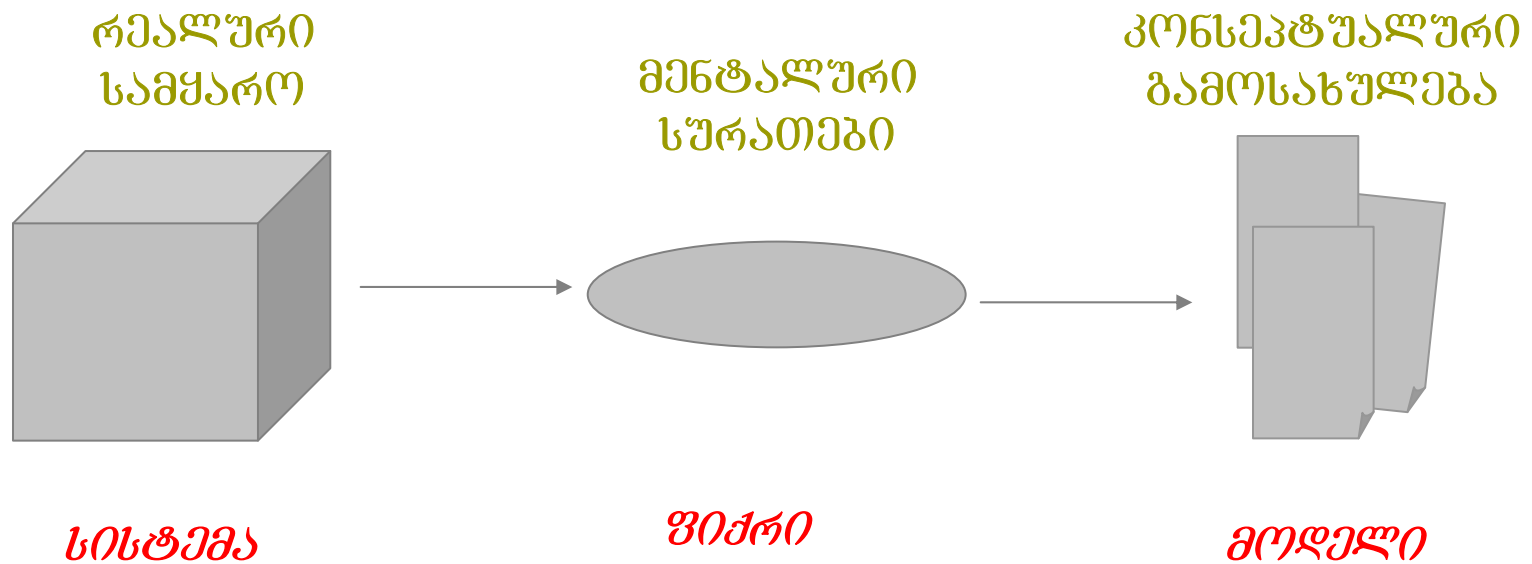
---

- რეალური სამყაროს კომპიუტერულად გამოსახვა
  - მონაცემთა აღწერისათვის საჭირო კონცეფციების ერთობლიობა
  - აღწერს მონაცემთა ტიპებს და მათი დამუშავების პროცესს მონაცემთა ბაზების აპლიკაციებში
- ბანსაზღვრავს სემანტიკას და მონაცემთა სტრუქტურას;
- ბანსაზღვრავს მონაცემების ურთიერთკავშირს
- ბანსაზღვრავს წესებს და შეზღუდვებს(“constraints”)
- მონაცემთა დამუშავების სხვადასხვა ხერხებს



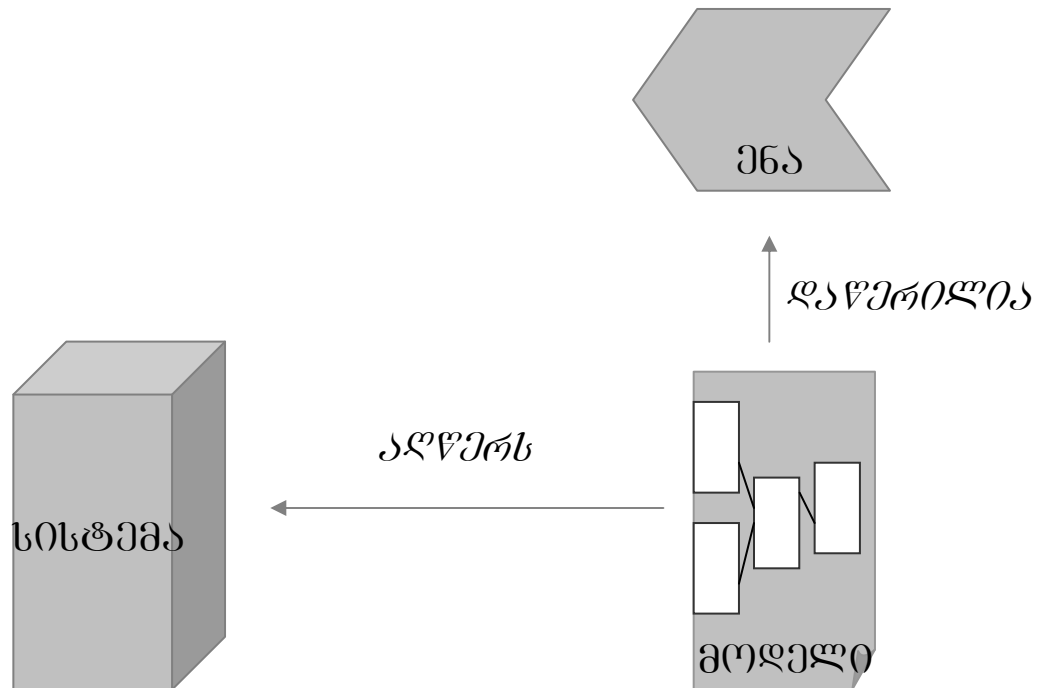
# მოდელეობი

---



# სისტემები და მოდელები

მოდელი არის სისტემების შესახებ ცოდნის აბსტრაქტული გამოსახულება, რომელიც იყენებს კარგად განსაზღვრულ ენას



# მონაცემთა მოდელები

---

- ❑ მონაცემთა მოდელი არის ინტეგრირებული კოლექცია:
  - მონაცემთა სტრუქტურირების ძირითადი ცნებების
  - მონაცემთა მართვის მექანიზმების შესახებ
- ❑ მონაცემთა ბაზის სქემა კონკრეტული მონაცემების ერთობლიობის აღწერა , რომელიც იყენებს მოცემულ მონაცემთა მოდელს
- ❑ მონაცემთა მოდელი შეიძლება იყოს:
  - Relational
  - Object-oriented
  - Object-relational

# მონაცემთა მოდელების ისტორია

---

- 1950 ---- File Systems
- 1960 ---- Hierarchical model
- 1970 ---- Network model
- 1980 ---- Relational mode
- 1990 ---- Object-Oriented model
- 1990 ---- Object-Relational

# მონაცემთა მოდელი

---

- Relational Data model (RM) ---მისი ძირითადი პრინციპია რომ მონაცემები თავმოყრილია ცხრილებში, რომლებიც ჩანაწერების ერთობლიობაა

<i>sid</i>	<i>name</i>	<i>login</i>	<i>age</i>	<i>gpa</i>
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@cs	18	3.2
53650	Smith	smith@math	19	3.8

- Relation schema:

Student (*sid*:char(5), *name*; varchar(20), *login*: varchar(15), *age*: integer, *gpi*: real)

# Relational Data model (RM)

Diagram illustrating the Relational Data Model (RM) structure. A table represents a relation with four attributes: PlateNumber, Make, MadeIn, and Colour. The table contains five tuples. Arrows indicate the relationship between the labels and the table components: 'Attributes' points to the header row, 'Tuple' points to the first row, and 'Value' points to the data cells.

PlateNumber	Make	MadeIn	Colour
AA961	Ford	1997	Green
ZD021	Jaguar	1995	Red
XR336	Opel	1976	Blue
XX321	Fiat	1989	White
AB112	Mercedes	1992	Yellow

**Relation:** ცხრილი ჩანაწერები (tuples) და სვეტები (attributes) რომლებიც განსაზღვრულია ე.წ. “relation schema”-ის მიერ

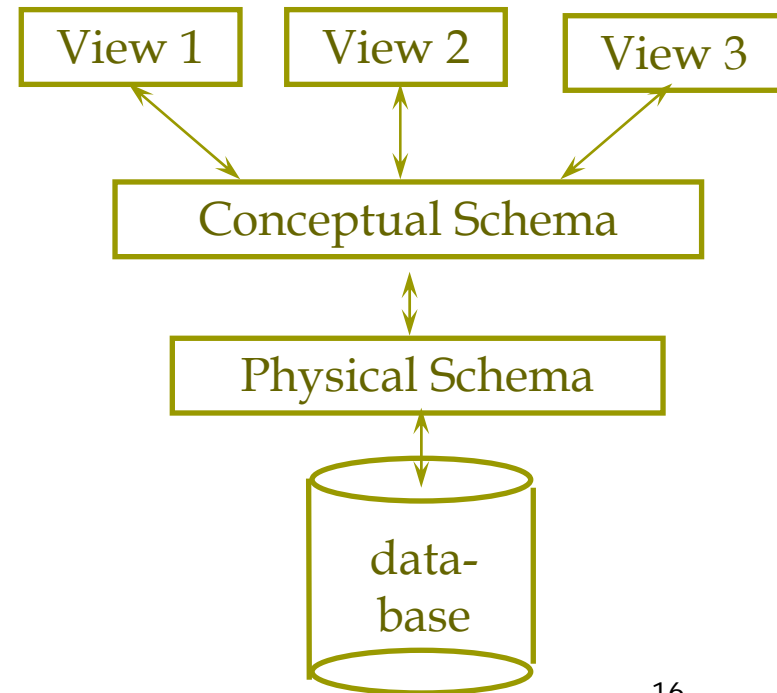
# მონაცემთა ბაზების დიზაინი

---

- მონაცემთა ბაზების დიზაინისათვის მნიშვნელოვანია განსაზღვრული იყოს:
  - სტრუქტურა/Storage structures;
  - დიზაინის წესები / Design of rules;
  - დიზაინისთვის საჭირო ოპერაციები / Design operations

# მონაცემთა ბაზა: დიზაინის დონეები

- ❑ Views ---- აღწერს მომხმარებელი რობორ ხელავს მონაცემებს
- ❑ Conceptual schema ---- განსაზღვრავს მონაცემთა ინტეგრირებულ, ლობიკურ სტრუქტურას
- ❑ Physical schema ---- აღწერს გამოყენებულ ფაილებს და ინდექსებს





# მონაცემთა დამოუკიდებლობა

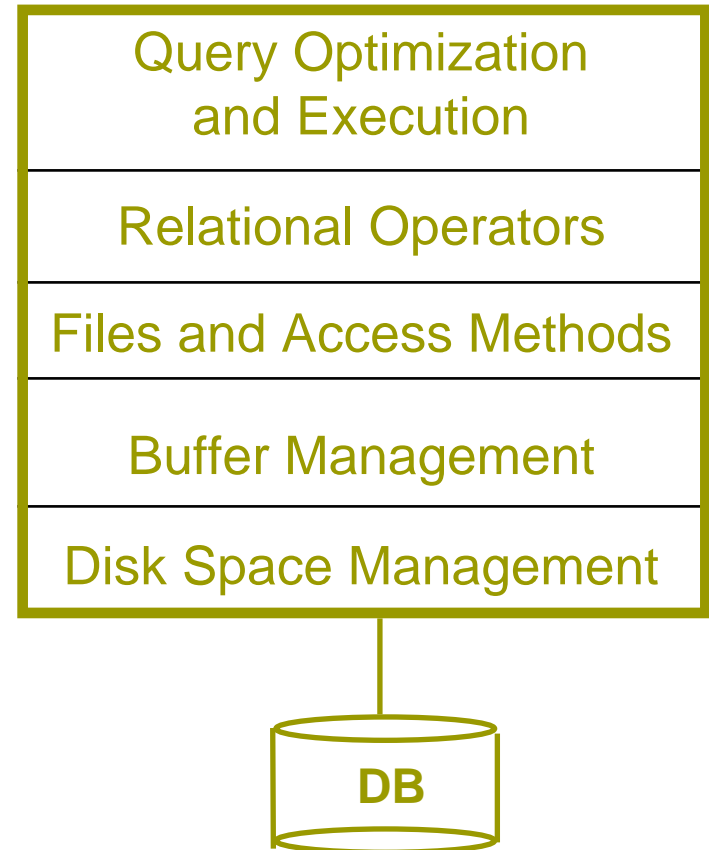
---

- აპლიკაციების ფუნქციონალურობა მეტწილად დამოკიდებულია მონაცემთა სტრუქტურაზე და განთავსებაზე
- მონაცემთა *ლოგიკური დამოუკიდებლობა* იცავს მონაცემთა ლოგიკური სტრუქტურის შეცვლას
- მონაცემთა *ფიზიკური დამოუკიდებლობა* იცავს მონაცემთა ფიზიკური სტრუქტურის შეცვლას

# DBMS- ების არქიტექტურა

---

- ❑ ზოგადად DBMS- ებს აქვთ  
ფუნქციური არქიტექტურა
- ❑ ეს არის მრავალიდან  
ერთ-ერთი შესაძლებელი  
არქიტექტურა



# დასკვნა

---

- ❑ DMBS- ები გამოიყენება დიდი ზომის მონაცემთა შენარჩუნებისთვის და მათი მართვისთვის
- ❑ აღსანიშნავია ალგორითმების უნარი სისტემური დაზიანების შემთხვევაში; კონკურენტული გამოყენება, აპლიკაციის სწრაფი განვითარება, ინტეგრაცია, უსაფრთხოება
- ❑ ე.წ. “level of abstraction” იძლევა მონაცემთა დამოუკიდებლობის შესაძლებლობას
- ❑ DMBS- ებს აქვთ ფენობრივი არქიტექტურა
- ❑ აქვს საკანონმდებლო და მაღალ ანაზღაურებასი სამუშაო ადგილები

