

მონაცემთა ბაზიდან მონაცემთა  
ამოღების ძირითადი პრინციპები  
**“Structured Query Language/SQL”-ის**  
გამოყენებით



# მონაცემთა ამოღება მონაცემთა ბაზიდან

---

- ❑ ხერხი: მონაცემთა ბაზიდან ზუსტად იმ მონაცემების ამოღება რომელიც გვჭირდება
- ❑ მონაცემთა მოპოვება: საჭირო მონაცემების (ინფორმაციის) ამოღების პროცესი
- ❑ მნიშვნელოვანია:
  - არსებული მონაცემთა ქსელის კარგად ცოდნა კარგად გვესმოდეს რა ტიპის მონაცემთა მიღებაა საჭირო
  - ე.წ. **“Query Language:-**ის გამოყენების კარგად ცოდნა

# ძირითადი და მნიშვნელოვანი ტერმინები

---

- **“Query”**: ფუნქციონალურ აქტივობებს და ბანსაზღვრავს მონაცემთა ბაზიდან ბარკვეული მონაცემების ამოღების მოთხოვნები. ის შეიქმნება ბანსილული იყოს როგორც ე.წ. “high-level” კომპიუტერული პროგრამა
- **“Transaction”**: ფუნქციონალურ აქტივობებს მონაცემთა შეცვლის მოთხოვნების ფორმულირებას. ის შეიქმნება ბანსილული იყოს როგორც ე.წ. “high-level” კომპიუტერული პროგრამა

# Relational DBMS-SQL

---

- ❑ მონაცემები წარმოდგენილია როგორც ცხრილების ერთობლიობა, ახალი ცხრილები შეიძლება შეიქმნას არსებულ სხვა ცხრილებიდან
- ❑ **Select.....From.....Where.....**
- ❑ განვიხილოთ სამი ძირითადი ე.წ. “Relational operations”:
  - Attribute Projection** (სვეტები, Select clause )
  - Join** (ცხრილების დაკავშირება, From clause)
  - Tuple Selection** (რიგები/ჩანაწერები, Where condition)

# SQL პროგრამული ენა

---

- SQL---Structured Query Language
- ეფუძნება “relational algebra, calculus “
- ძირითადი კონსტრუქცია:
  - ცხრილები ( სვეტები, რიგები)
  - ნორმალიზაცია ( კარგი დიზაინის მქონე ცხრილები)
- ISO სტანდარტები 1980 წლიდან

# SQL პროგრამული ენა

---

□ SQL-ს ექრდნობა:

IBM DB2

Microsoft SQL Server

ORACLE

SYBASE

INFORMIX

MS ACCESS

PostgreSQL

MySQL

# SQL statement- ობ ობოპობო

---

- Database Definition Language (**DDL**)
  - CREATE (TABLE, INDEX, VIEW, etc.)
- Database Manipulation Language (**DML**)
  - SELECT
  - UPDATE, DELETE, INSERT

# Tuple selection

- ჩვენ გვინდა შევადგინოთ ისეთი “query”, რომელიც საშუალებას მოგვცემს მოვნიშნოთ არსებული მოანეცემების ბარკევეული ნაწილი

- რომელი მეზღვაურების ასაკი არის 50 წელზე მაღლა?

*Sailor*

<u>sid</u>	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0



<u>sid</u>	sname	rating	age
31	lubber	8	55.5



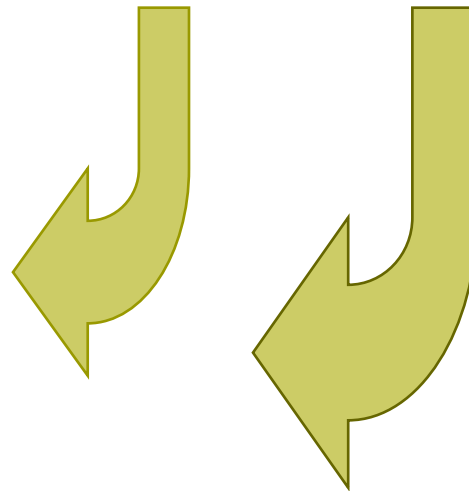
# Tuple selection

---

- SQL-ის გამომყენებით ჩვენ ვამბობთ:

From the Sailors, where Age>50

```
SELECT *  
FROM   Sailor AS s  
WHERE  s.age > 50
```



# Attribute projection

- ჩვენ გვინდა შევადგინოთ ისეთი “query”, რომელიც საშუალებას მოგვცემს მოვნიშნოთ არსებული ატრიბუტების გარკვეული ნაწილი

- რა არის მუდგავართა სახელები და მათი ასაკი?

*Sailor*

<u>sid</u>	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0



sname	age
dustin	45.0
lubber	55.0
rusty	35.0

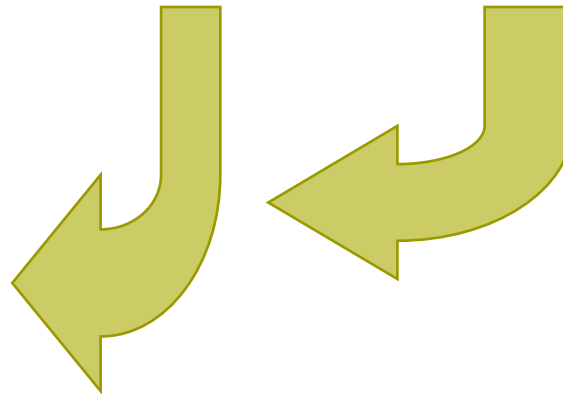
# Attribute projection

---

- SQL-ის გამომყენებით ჩვენ ვამბობთ:

From Sailors, give names and ages

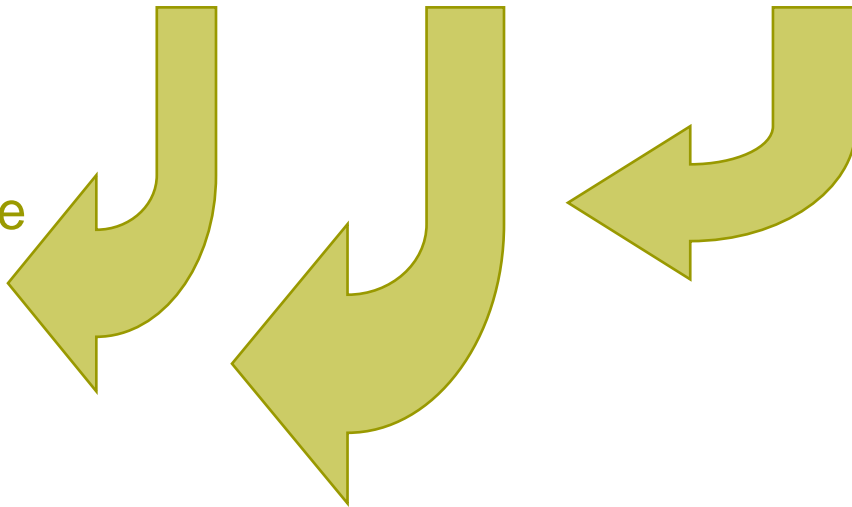
```
SELECT  s.sname, s.age  
FROM    Sailor AS s
```



# “Tuple selection” and “Attribute projection”

From Sailors, Age above 50, give names and ages

```
SELECT    s.sname, s.age
FROM      Sailor AS s
WHERE     s.age > 50
```



*Sailor*

<u>sid</u>	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0



sname	age
lubber	55.5

# JOINS

---

## JOINS Operations:

- EQUI
- INNER
- OUTER
- LEFT
- RIGHT

# JOINS

---

```
❑ SELECT DISTINCT s.sname
FROM Sailor AS s, Reserves AS r
WHERE s.age > 30 AND s.sid = r.sid
```

ასევე შესაძლებელია ესე დაიწეროს:

```
SELECT DISTINCT s.sname
FROM Sailor AS s
INNER JOIN
Reserves AS r
ON s.sid = r.sid
WHERE s.age > 30
```

## Sailor

sid	sname	rating	age
22	Dustin	7	45
31	Lubber	8	55.5
34	Marvin	9	39
36	Gail	9	35
58	Rusty	10	35
64	Johnny	6	43.5
93	Rachel	8.5	45

s.sid	sname	rating	age	r.sid	bid	day
31	Lubber	8	55.5	31	103	11/11/1999
31	Lubber	8	55.5	31	108	11/12/1999
31	Lubber	8	55.5	31	112	11/13/1999
31	Lubber	8	55.5	31	125	5/12/1999
34	Marvin	9	39	34	108	1/5/1998
58	Rusty	10	35	58	103	1/1/1998
58	Rusty	10	35	58	112	5/5/1998
64	Johnny	6	43.5	64	103	1/2/1999
93	Rachel	8.5	45	93	112	4/12/1998

Join

Projection

## Reserve

sid	bid	day
31	103	11/11/1999
31	108	11/12/1999
31	112	11/13/1999
31	125	5/12/1999
34	108	1/5/1998
58	103	1/1/1998
58	112	5/5/1998
64	103	1/2/1999
93	112	4/12/1998

```
SELECT s.sname, r.day
FROM Sailor AS s, Reserves AS r
WHERE s.age>30 AND s.sid=r.sid;
```

sname	day
Johnny	1/2/1999
Lubber	5/12/1999
Lubber	11/11/1999
Lubber	11/12/1999
Lubber	11/13/1999
Marvin	1/5/1998
Rachel	4/12/1998
Rusty	1/1/1998
Rusty	5/5/1998

# JOINS

Person

Car

PID	CID
1	A
2	B
3	C
4	null

CID	type
A	Ford
B	Volvo
C	Honda
D	Kia

Left Outer Join

PID	CID	CID	type
1	A	A	Ford
2	B	B	Volvo
3	C	C	Honda
4	null	null	null

Inner Join

PID	CID	CID	type
1	A	A	Ford
2	B	B	Volvo
3	C	C	Honda

Right Outer Join

PID	CID	CID	type
1	A	A	Ford
2	B	B	Volvo
3	C	C	Honda
null	null	D	Kia

Full Outer Join

PID	CID	CID	type
1	A	A	Ford
2	B	B	Volvo
3	C	C	Honda
4	null	null	null
null	null	D	Kia



# SQL - Relational Operations

---

## □ Attribute Projection

- აჩვენებს სვეტების მონაცემებს (attributes)
- '\*' მონიშნავს ყველა ატრიბუტს ცრილიდან
- მაგ. `SELECT * FROM STUDENT;`
- მაგ. `SELECT * FROM ENROLLMENT;`

# SQL, Comparison Operators

---

=, <, >, <>, =>, etc.; BETWEEN

```
SELECT *
FROM EMP
WHERE WORKDEPT <> 'D11';
```

LIKE comparison operator:

```
SELECT *
FROM EMP
WHERE LASTNAME LIKE 'A*';
```

```
SELECT *
FROM EMP
WHERE LASTNAME LIKE '*R';
```

# SQL -- LOGICAL OPERATORS

---

## LOGICAL OPERATORS:

- AND, OR , NOT

- მაგალითი:

```
SELECT *  
FROM EMP  
WHERE WORKDEPT = 'D11' OR WORKDEPT =  
      'A11';
```

# SELECT Statement

---

- General Form:

SELECT- list of columns or expressions

FROM -list of tables or views

WHERE- list of row conditions

GROUP BY- column

ORDER BY - list of columns [ASC/DESC];

# UPDATE Statement

---

□ The UPDATE statement:

- UPDATE JAnn.EMP SET JOB = 'PROGRAMR'
- WHERE WORKDEPT = 'D11' AND JOB = 'DESIGNER';

# DELETE Statement

---

- შებვიძლია მხოლოდ მთელი ჩანაწერის (tuples) წაშლა

მაბ: delete from r where P;

ჩანაწერი(ები) (tuples) r -ში რომლისთვისაც P არის მართებული წაიშლება

- თუ “where clause”-ს არ მივუთითებთ, ცხრილის ყველა ჩანაწერი წაიშლება

# DELETE example

---

მაგალითები:

წავშალოთ ყველა ჩანაწერი რომელიც Smith- ს ეკუთვნის

```
DELETE FROM depositor  
WHERE cname='Smith'
```

წავშალოთ სესხის ყველა თანხა 1300 –სა და 1500-ს შორის

```
DELETE FROM loan  
WHERE loan BETWEEN 1300 and 1500
```

# INSERT Statement

---

INSERT:

```
INSERT INTO  
DEPT (DEPTNO, DEPTNAME, MGRNO)  
VALUES ('F01', 'BCIS DEPT', '000200');
```