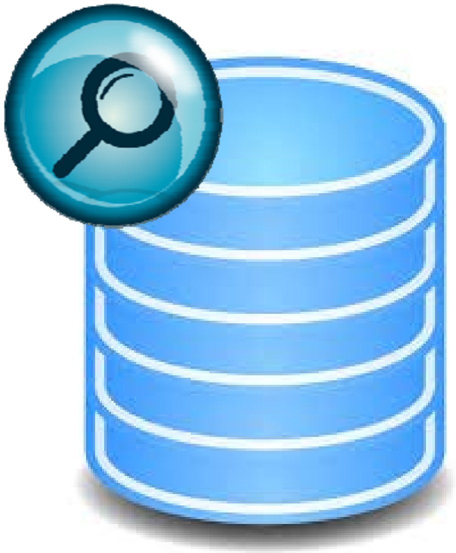
DBSnap

Learning Database Queries By Snapping Blocks

Yasin Silva Jaime Chon
Arizona State University

What is DB*Snap*?

- Web-based application to build database queries (relational algebra) by snapping blocks
- DB*Snap* supports the construction of intuitive database query trees
- Highly dynamic
  - Shows query results as the query is being built
  - User can inspect intermediate results of any query node



DB*Snap* Blocks

- Datasets

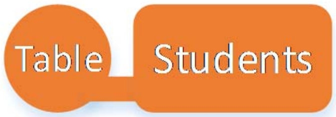


Table Students

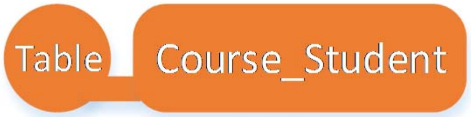
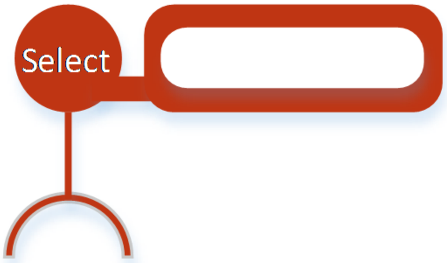
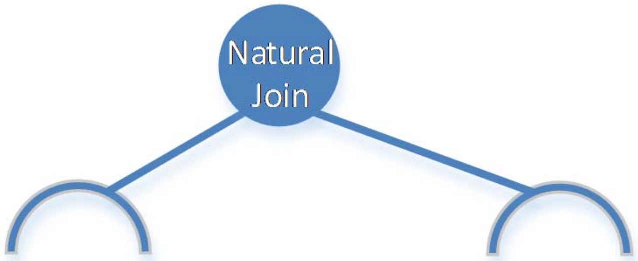


Table Course_Student

- Operators



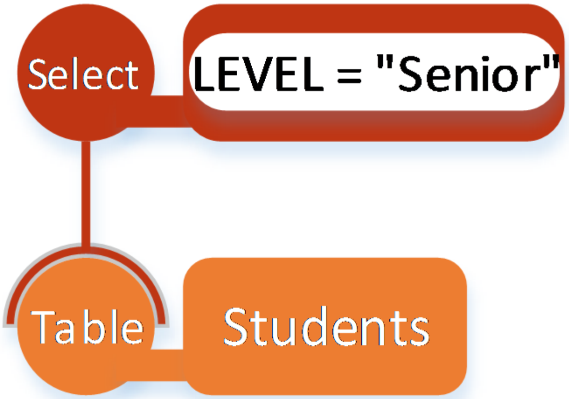
Select



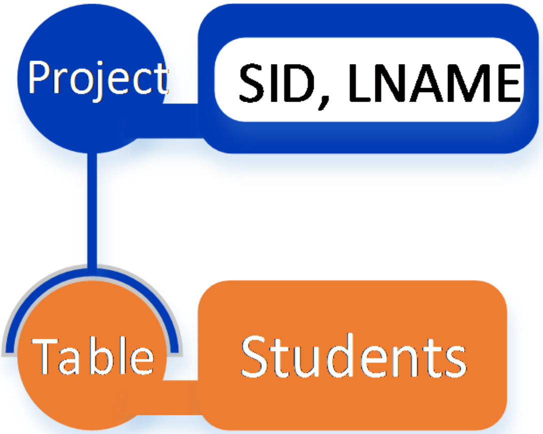
Natural Join

DB*Snap* Operators

- Selection
  - List only the senior students
  - $\sigma_{\text{LEVEL="Senior"}}$(Students)

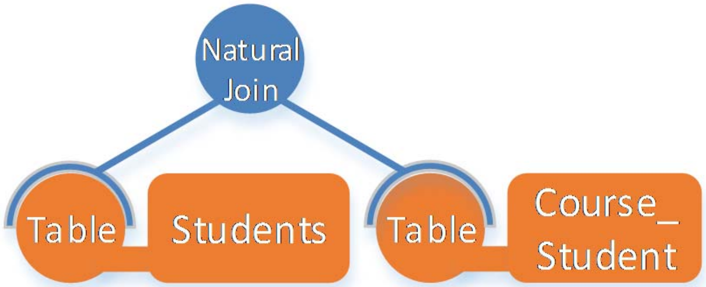


- Projection
  - List the ID and last name of students
  - $\pi_{\text{SID,LNAME}}$(Students)

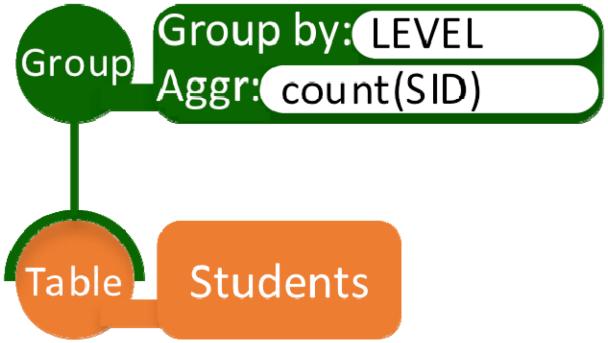


DB*Snap* Operators

- Natural Join
  - List the students and the courses they are taking
  - Students ⋈ Course_Student

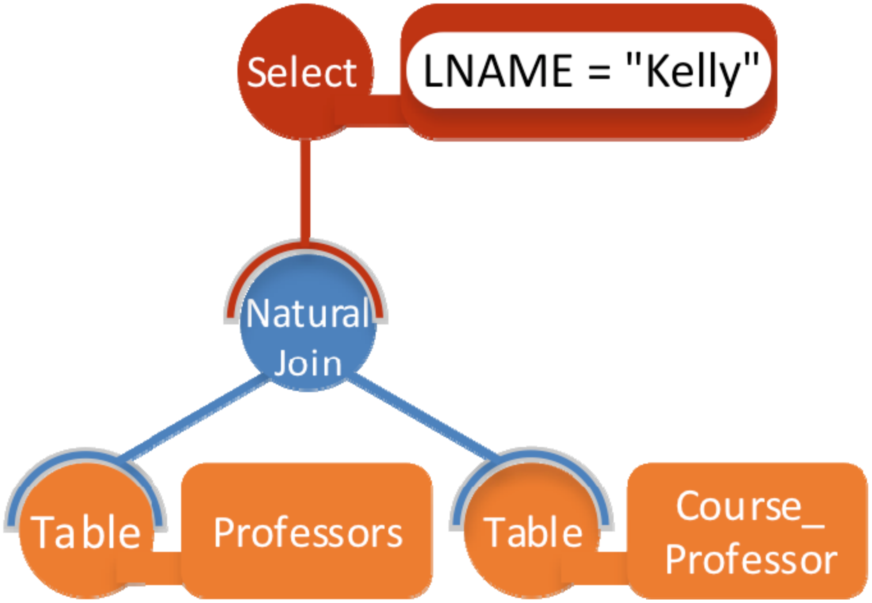


- Aggregation
  - Compute the number of students per level
  - $_{\text{LEVEL}}G_{\text{count(SID)}}$(Students)



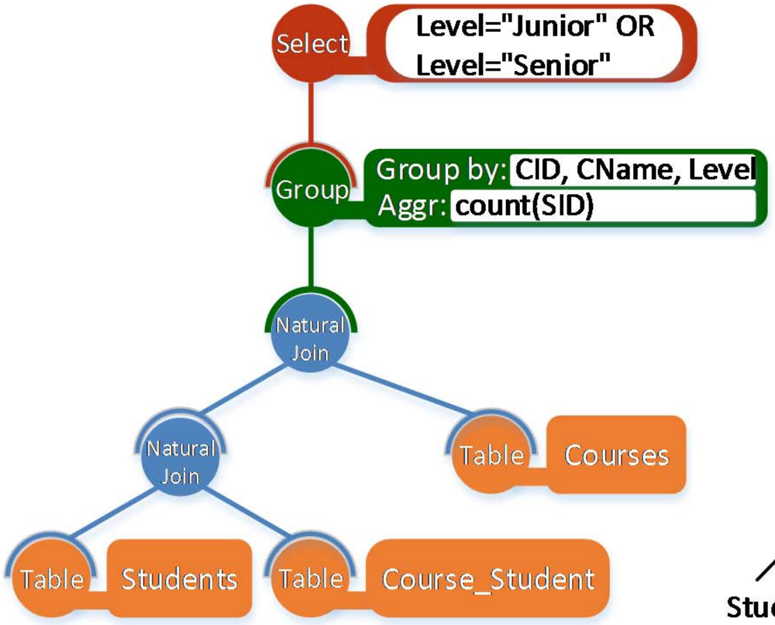
DB*Snap* Query Example 1

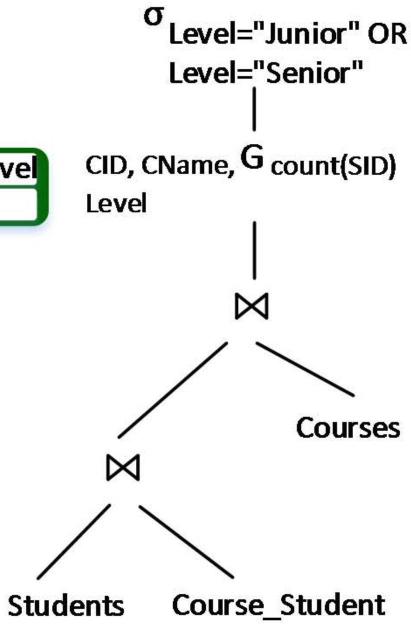
- List the professors with last name "Kelly" and the courses (CID) they are teaching
- $\sigma_{\text{LNAME="Kelly"}}$ (Professors ⋈ Course_Professors)



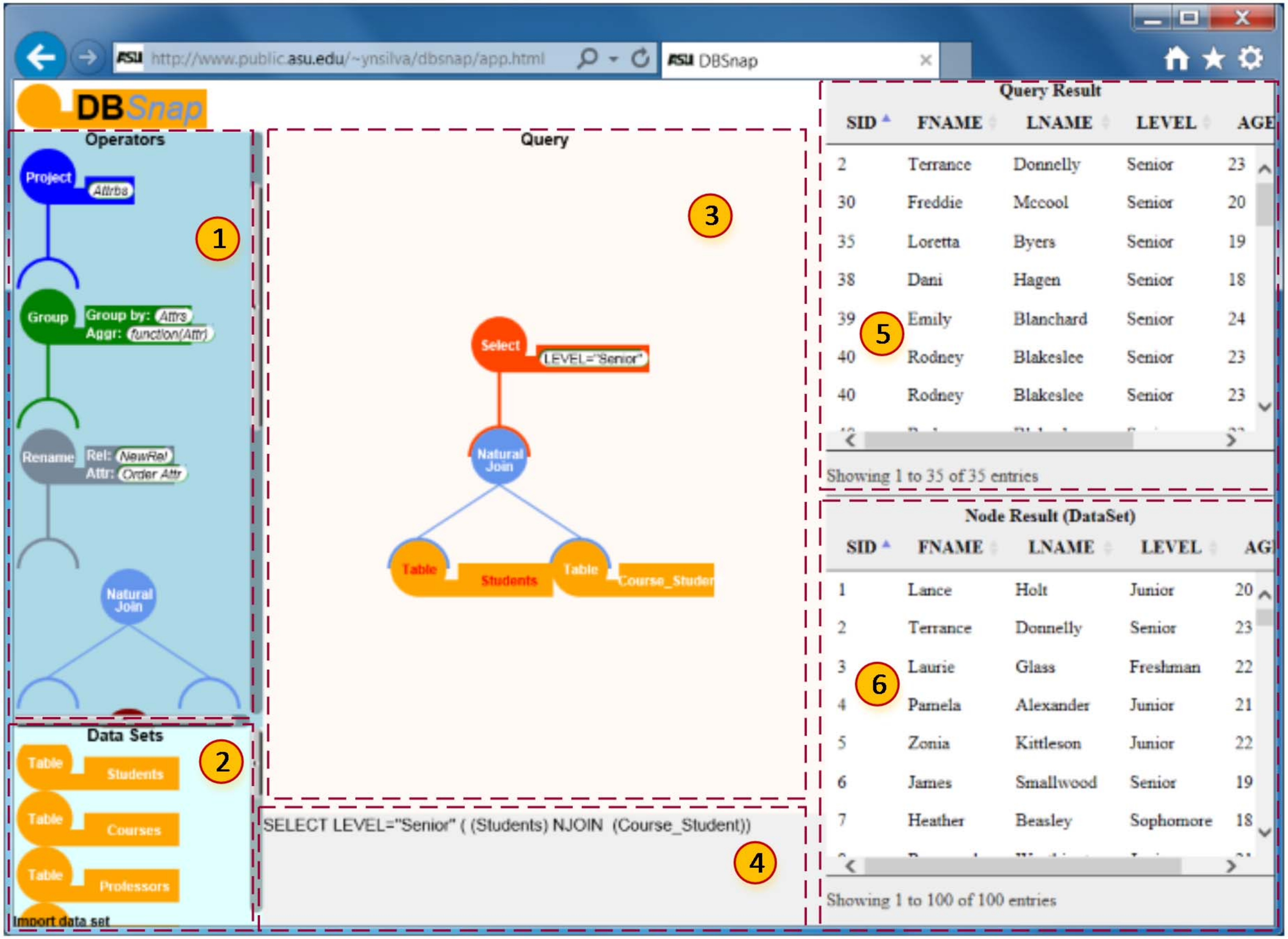
DB*Snap* Query Example 2

- Compute the number of students of each level registered in each course considering only Junior and Senior students




σ Level="Junior" OR Level="Senior"
CID, CName, Level G count(SID)
⋈
Courses
⋈
Students
Course_Student

$$\sigma_{Level=\text{``}Junior\text{''}\ \mathrm{OR}\ Level=\text{``}Senior\text{''}}\\ ({}_{CID,CName,Level}G_{count(SID)}\\ ((Students \bowtie Course\_Student)\\ \bowtie Courses))$$



SELECT LEVEL="Senior" ((Students) NJOIN (Course_Student))

Query Result

SID	FNAME	LNAME	LEVEL	AGE
2	Terrance	Donnelly	Senior	23
30	Freddie	Mccool	Senior	20
35	Loretta	Byers	Senior	19
38	Dani	Hagen	Senior	18
39	Emily	Blanchard	Senior	24
40	Rodney	Blakeslee	Senior	23
40	Rodney	Blakeslee	Senior	23

Showing 1 to 35 of 35 entries

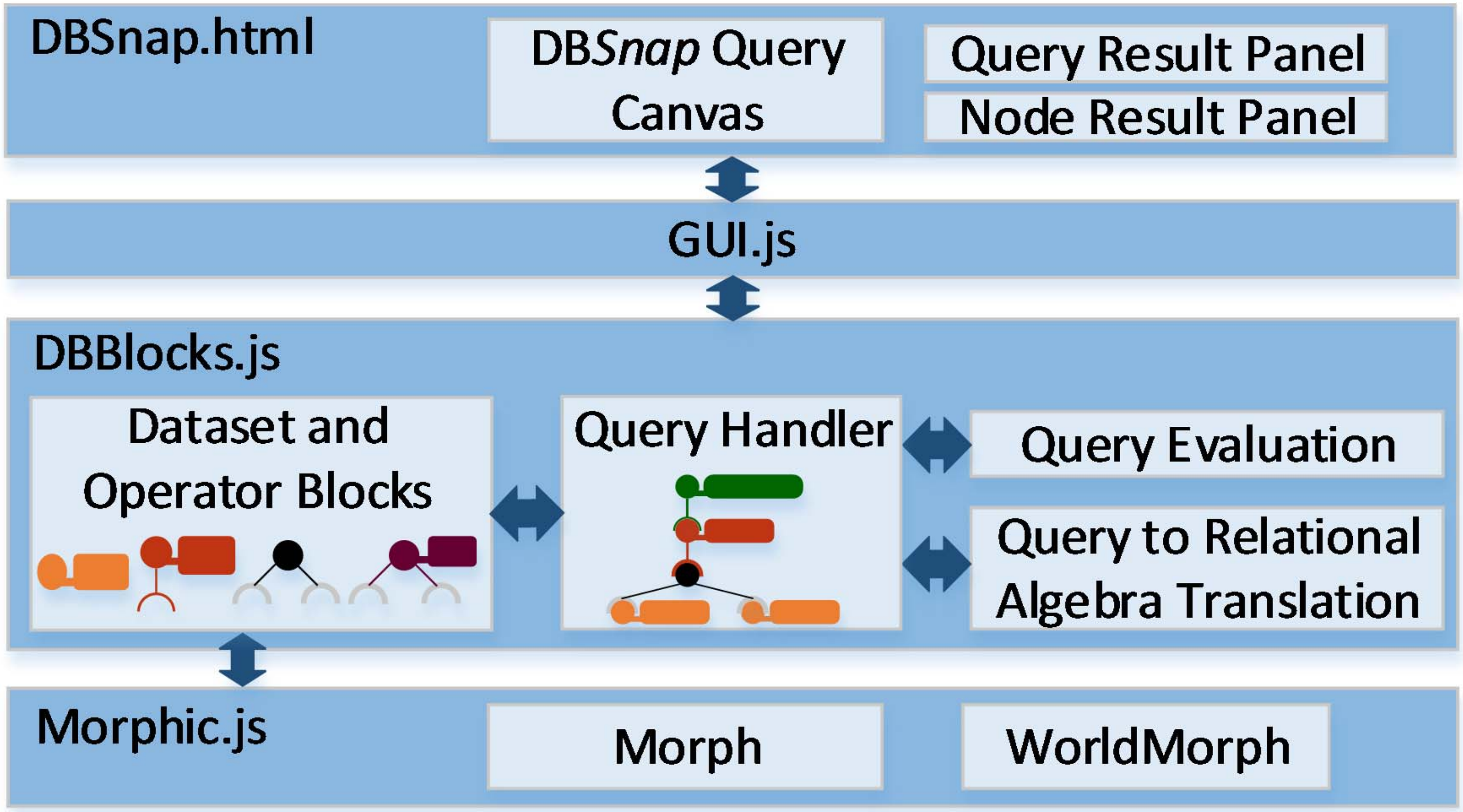
Node Result (DataSet)

SID	FNAME	LNAME	LEVEL	AG
1	Lance	Holt	Junior	20
2	Terrance	Donnelly	Senior	23
3	Laurie	Glass	Freshman	22
4	Pamela	Alexander	Junior	21
5	Zonia	Kittleson	Junior	22
6	James	Smallwood	Senior	19
7	Heather	Beasley	Sophomore	18

Showing 1 to 100 of 100 entries

1. Operator palette
2. Dataset palette
3. Query area
4. Relational algebra panel
5. Query result panel
6. Node result panel

DB*Snap*'s Architecture



Comparison of Educational Tools

Feature	DBSnap	WinRDBI	iDFQL	Relational	RALT	Query Visualiser	Bags
Implementation technologies	HTML5, JavaScript	Java	Borland C++	Python	Java Swing	.NET, Mono	HTML5, JavaScript
Publicly available	●	●	●	●	●	●	●
Open source code	●			●		●	●
Block-based query editor	●		●		●		●
Shows RA expression	●	●		●		●	
Web application	●						●
Uses tree-based representation to build queries	●		●		●		
Automatically updates query result	●				●		
Intermediate results	●		●		●	●	●
Side-by-side queries	●	●	●		●	●	●
Can work without DB connection	●	●		●		●	●
Build-in datasets	●	●		●		●	●
Allows importing or connecting to custom data	●	●	●	●	●	●	
Views				●		●	

DBSnap Demo

DB*Snap*

Thank you!

Try DB*Snap* now!
http://www.public.asu.edu/~ynsilva/dbsnap/