



# Learning Database Queries By Snapping Blocks

Yasin Silva      Jaime Chon  
Arizona State University

# What is DBSnap?

- Web-based application to build **database queries** (relational algebra) by **snapping blocks**
- DBSnap 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



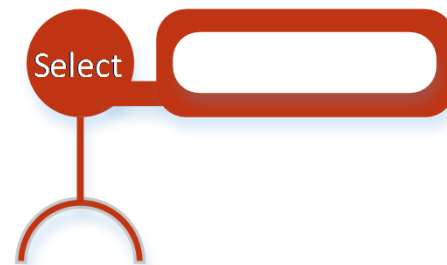
# DBSnap Blocks



- Datasets

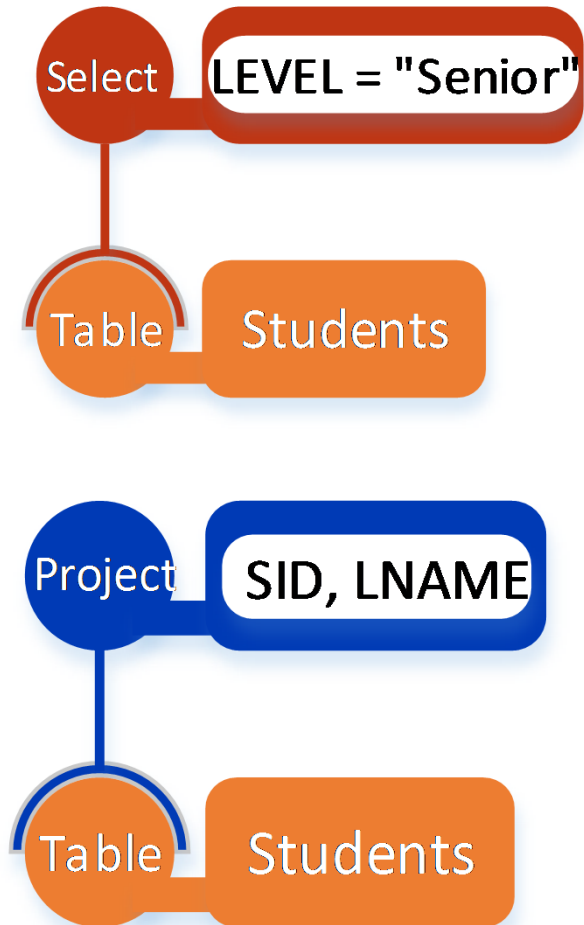


- Operators



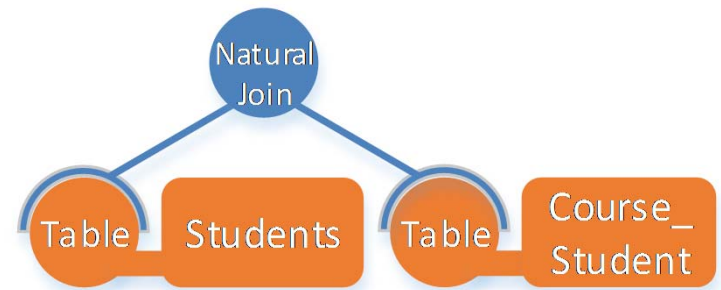
# DBSnap Operators

- Selection
  - List only the senior students
  - $\sigma_{\text{LEVEL}=\text{"Senior"}}(\text{Students})$
- Projection
  - List the ID and last name of students
  - $\pi_{\text{SID,LNAME}}(\text{Students})$

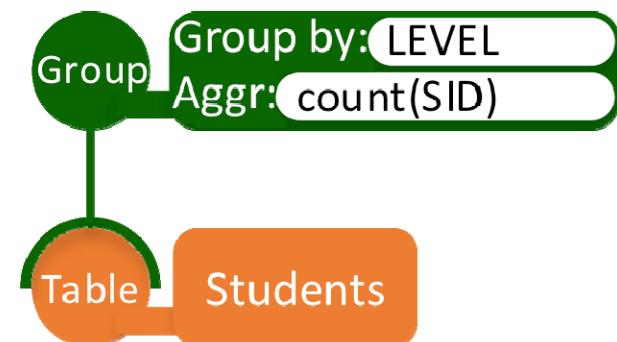


# DBSnap Operators

- Natural Join
  - List the students and the courses they are taking
  - $\text{Students} \bowtie \text{Course\_Student}$

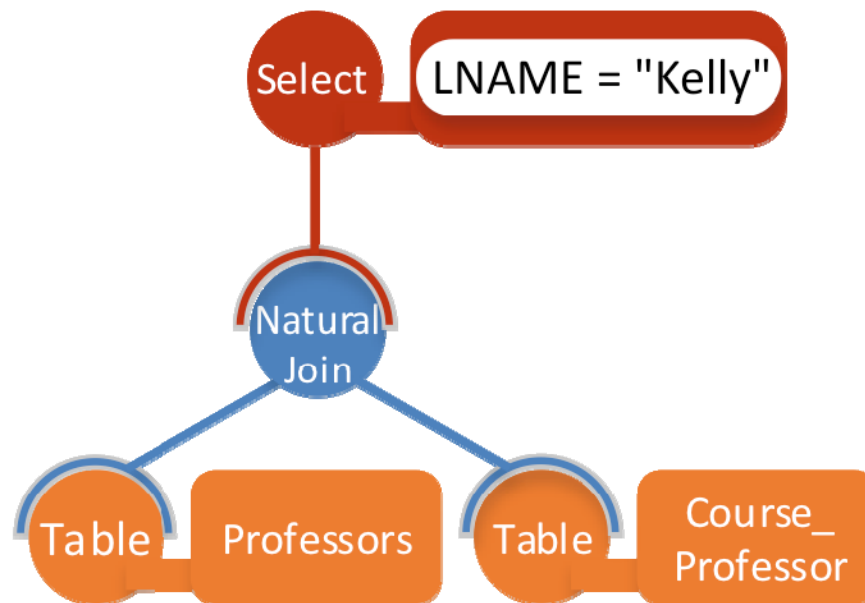


- Aggregation
  - Compute the number of students per level
  - $\text{LEVEL} \overset{\text{G}}{\text{count(SID)}}(\text{Students})$



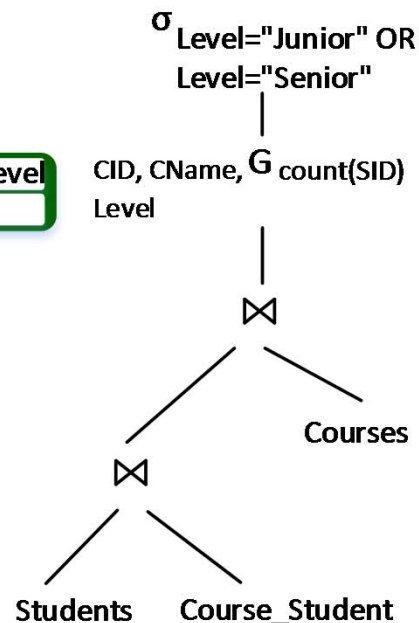
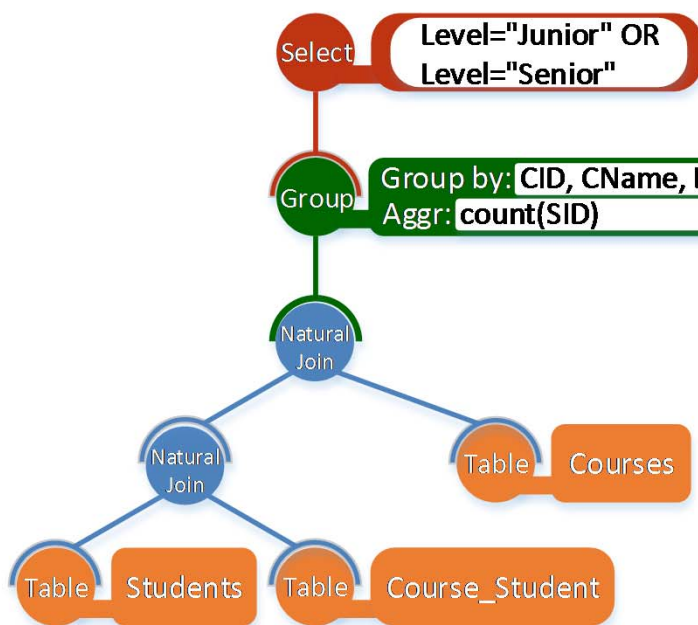
# DBSnap Query Example 1

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



# DBSnap Query Example 2

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



$\sigma_{Level="Junior" \text{ OR } Level="Senior"}$   
 $(\rho_{CID, CName, Level} \mathcal{G}_{count(SID)})$   
 $((Students \bowtie Course\_Student)$   
 $\bowtie Courses))$

The screenshot shows the DBSnap web application interface. The browser address bar displays `http://www.public.asu.edu/~ynsilva/dbsnap/app.html`. The interface is divided into several sections:

- Operator palette (1):** Located on the left, it contains various operators such as Project, Group, Rename, and Natural Join, each with associated attributes and functions.
- Dataset palette (2):** Located below the operator palette, it lists available tables: Students, Courses, and Professors.
- Query area (3):** The central workspace where a query is built using a relational algebra diagram. The diagram shows a Natural Join operator connecting two table nodes (Students and Course\_Student), followed by a Select operator with the condition `LEVEL="Senior"`.
- Relational algebra panel (4):** Located at the bottom of the query area, it displays the SQL equivalent of the diagram: `SELECT LEVEL="Senior" ((Students) NJOIN (Course_Student))`.
- Query result panel (5):** Located on the right, it displays the results of the query. The table has columns: SID, FNAME, LNAME, LEVEL, and AGE. The results are:
 

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
- Node result panel (6):** Located below the query result panel, it displays the results of a specific node in the query. The table has columns: SID, FNAME, LNAME, LEVEL, and AGE. The results are:
 

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

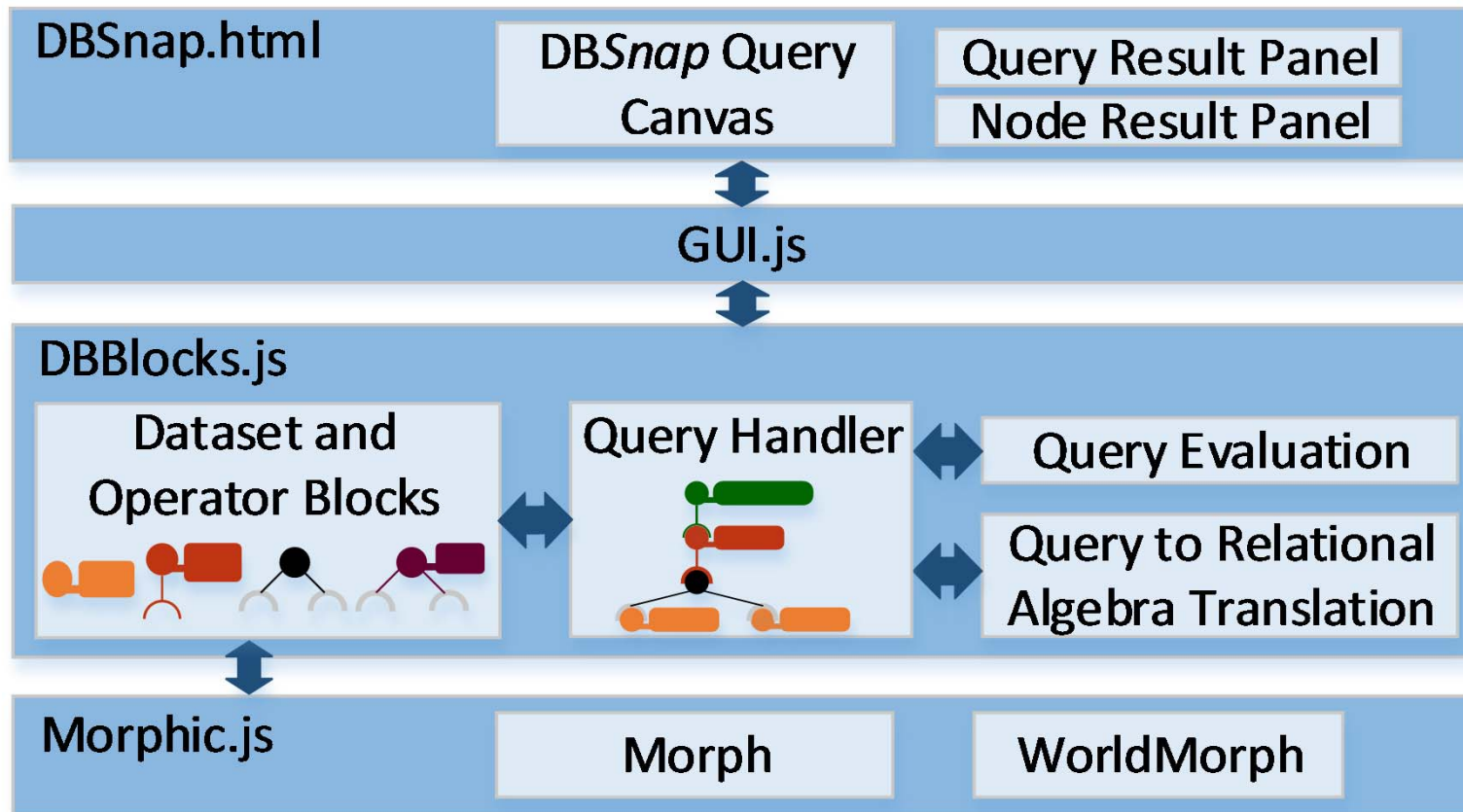
- 1. Operator palette
- 2. Dataset palette

- 3. Query area
- 4. Relational algebra panel

- 5. Query result panel
- 6. Node result panel



# DBSnap'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				•		•	





Thank you!

Try *DBSnap* now!

<http://www.public.asu.edu/~ynsilva/dbsnap/>