

PLDB

Alvin Cheung

PL+DB

Programming
Languages

Data
Management

SELECT... FROM ..
WHERE ...

SELECT... FROM ..
WHERE ...

SELECT... FROM ..
WHERE ...



Developing DBMS Applications

End-User Data Discovery



Users want high-level languages and tools to interact with the DBMS

ASP.NET	PHP Fat-Free Framework	CodeIgniter	Drupal	Magento	WordPress
AngularJS	Lift	Drupal	Drupal	Drupal	Drupal
Ruby on Rails	CherryPy	Drupal	Drupal	Drupal	Drupal
ASP.NET MVC	Restlet	Drupal	Drupal	Drupal	Drupal
Django	Lithium	Vaadin	Sails.js	TurboGears	
Laravel	OpenUI5	Yesod	Sinatra	Merb	
Meteor	Tapestry	Compojure	Grails	Ramaze	
Spring	Flight	Revel	Tornado	Ratpack	
Express	CompoundJS	Martini	Phalcon	Aura	
CodeIgniter	ZK	Mithril	Dojo	seaside	
Symfony	Flatiron	beego	Struts	Zotonic	
Ember.js	Noir	Ring	web.py	PureMVC	
Flask	Catalyst	SproutCore	Bottle	Tipfy	
JSF	Nitrogen	Mojolicious	Pyramid	Horde	
CakePHP	Snap	SilverStripe Sapphire	Kohana	Cappuccino	
Flex	Camping	Scalatra	Wicket	Swiz	



Easy integration with app

Impedance mismatch

Maintain single code base

End-to-end consistency

Don't need to learn SQL (!)

Performance issues

Use PL techniques to help users
interact with and build DBMS

End-User Tools



Is there something equivalent to argmax in SQL?

If I'm reading your question correctly, the following query should do it

```
SELECT x.*  
FROM YOUR_TABLE x  
WHERE x.column = (SELECT MAX(y.column)  
FROM YOUR_TABLE y)
```

Of course, if you have more than one row where the column a attains its maximum, then you'll get more than one row back from the query.



Hello computer.
What in the world does
"SELECT * FROM"
mean?



This is a query that returns
the rows with the max value
of column A from table T.

Queries from Constraints

Time	Sensor ID	Reading
Time > 1pm		
	10	

→ All rows returned must have time > 1pm

→ One of the rows have sensor ID = 10



```
SELECT * FROM sensors  
WHERE time > 1pm OR id = 10
```

Time	Sensor ID	Reading
10am	10	20.1
11am	10	4.1
10am	20	30.2
1pm	20	35

→ User wants to group by sensor ID

Sensor ID	Reading

→ Ask user which aggregate to use

PL for Building DBMS

- **Brandon Haynes** (this morning)
PipeGen: Automatic Generation of Data Pipes for Hybrid Analytics
- **Shumo Chu** (later this afternoon)
Formal Verification Framework for Query Optimization

PL for Writing DBMS Apps

- **Cong Yan**
Leveraging Lock Contention to Improve Transaction Application Performance
- **Maaz Ahmad**
Translating Sequential Java Programs to MapReduce Using Verified Lifting