



---

# SQL is Dead; Long Live SQL: Smart Services for Ad Hoc Databases

Bill Howe

Garret Cole

Alicia Key

Nodira Khoussainova

*Patrick Michaud*

*Kevin Pittman*

*Charlon Palacay*

*Luke Zettlemoyer*

*Yuan Zhou*



Microsoft®

**Research**



# The NoSQL Movement

## Fault-tolerance

by @jrecursive





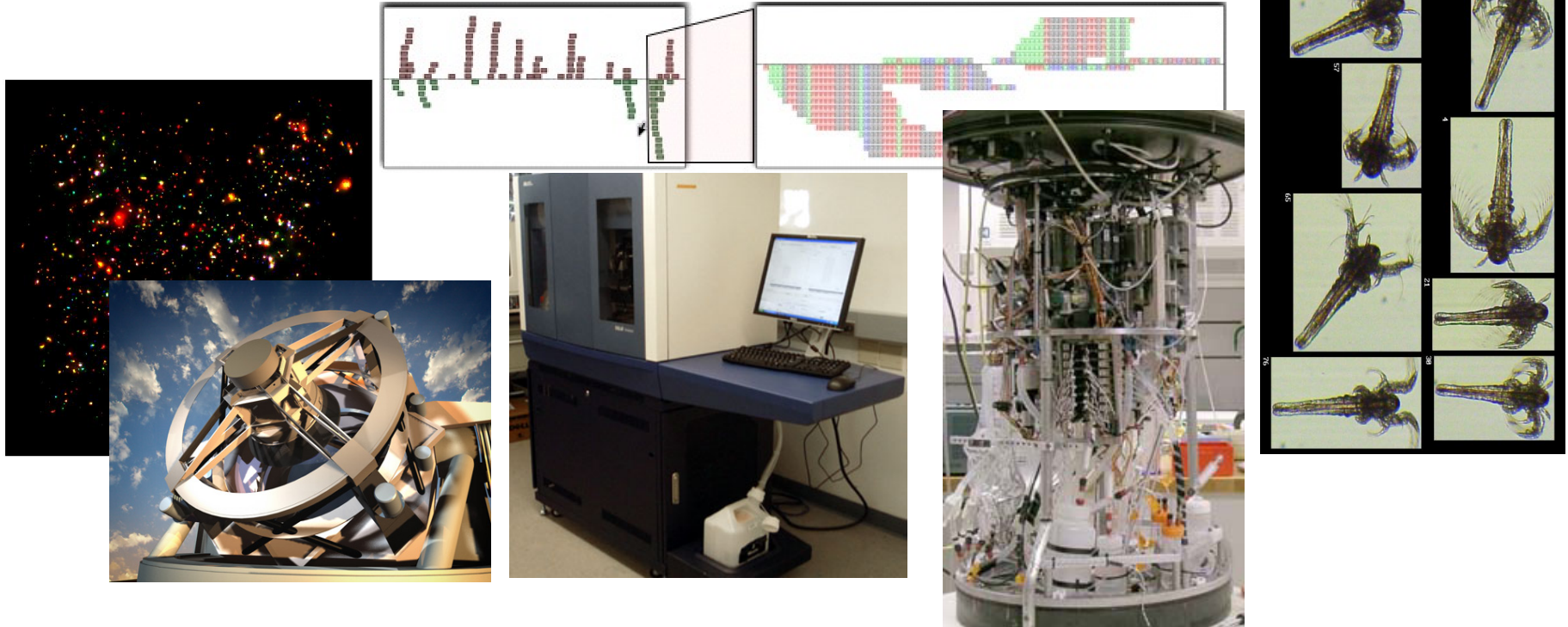
**<http://escience.washington.edu>**

# Science is reducing to querying databases

**Old model: "Query the world" (Data acquisition coupled to a specific hypothesis)**

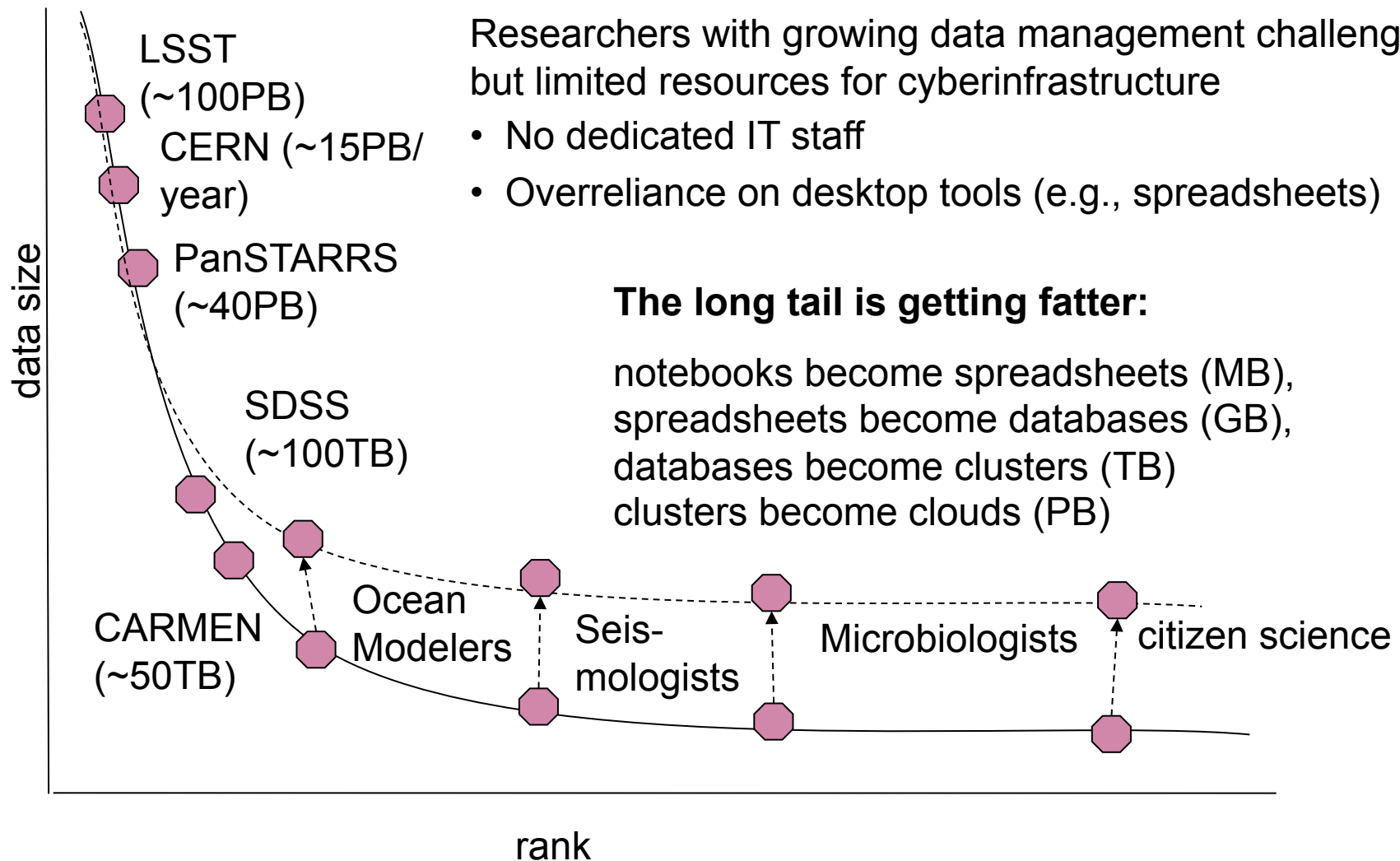
**New model: "Download the world, query the DB" (Data acquired en masse, to support many hypotheses)**

- Astronomy: High-resolution, high-frequency sky surveys (SDSS, LSST, PanSTARRS)
- Oceanography: high-resolution models, cheap sensors, satellites
- Biology: lab automation, high-throughput sequencing,



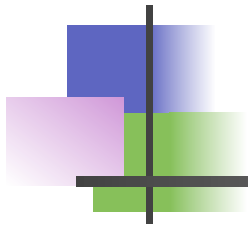
# Context: The Long Tail

[Wired 2004]



*How much time do you spend “handling data” as opposed to “doing science”?*

*Mode answer? 90%*

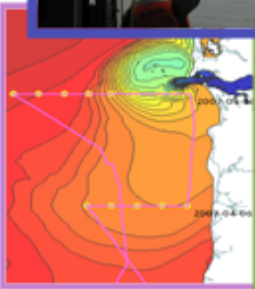


# Example: Environmental Metagenomics





# Environmental Sampling



Questions?

correlate diversity w/environment?

correlate diversity w/nutrients?

find new taxa and their distributions?



find new genes?

compare meta'omes?

Sequence data

raw data

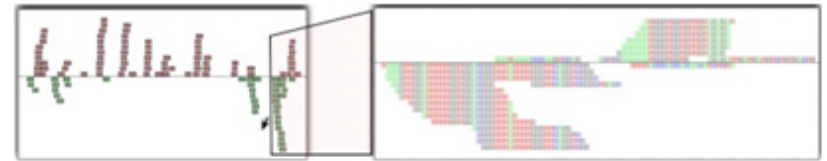
metadata

Plams, TIGPlams, COGs, FICPlams

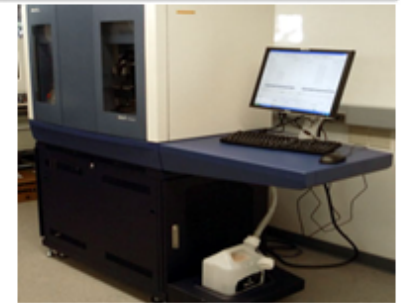
Public annotation DBs



Environmental Sampling



Sequencing

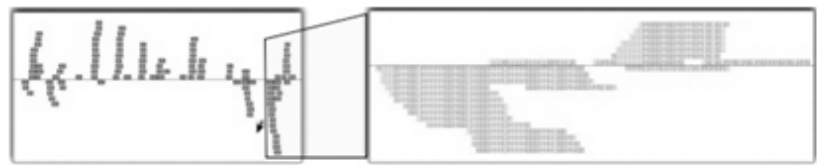


- Questions?
  - correlate diversity w/environment?
  - correlate diversity w/nutrients?
  - find new taxa and their distributions?
  - find new genes?
  - compare meta'omes?
- 





Environmental Sampling



Sequencing



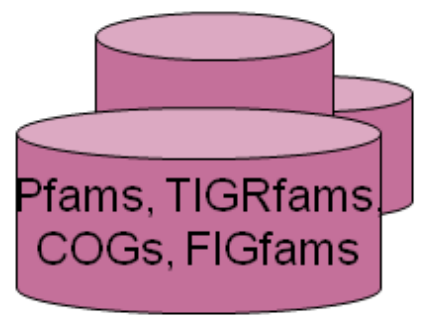
Sequence data



search hits



metadata



Public annotation DBs

Pfams, TIGRfams  
COGs, FIGfams

Questions?

correlate diversity  
w/environment?

correlate diversity  
w/nutrients?

find new taxa and  
their distributions?

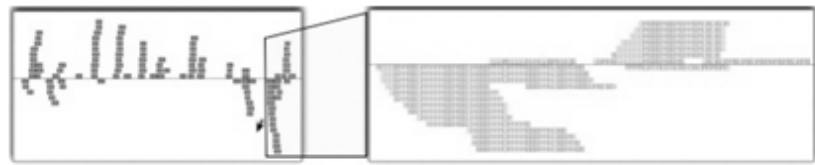
find new genes?

compare meta'omes?

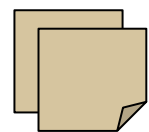




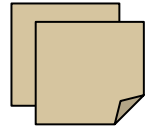
Environmental Sampling



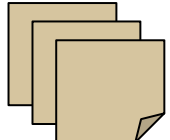
Sequencing



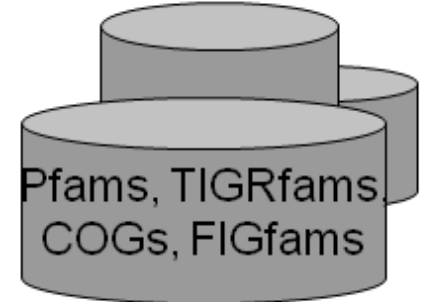
measurements



sequence data



search results



Public annotation DBs

Pfams, TIGRfams, COGs, FIGfams

Questions?

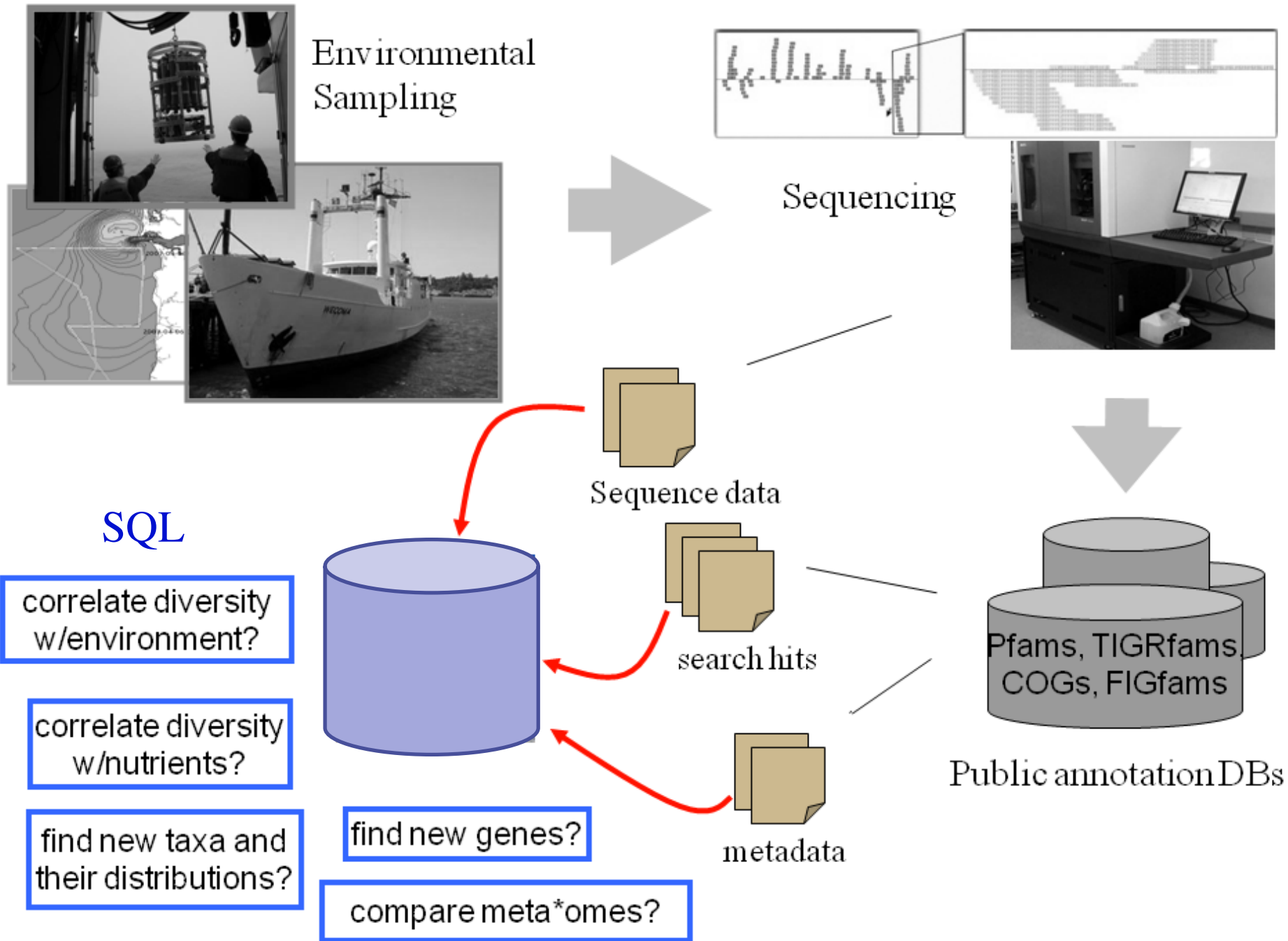
correlate diversity w/environment?

correlate diversity w/nutrients?

find new taxa and their distributions?

find new genes?  
compare meta\*omes?





# Ad Hoc Research Data

## Spreadsheets

### Fasta

```
;LCBO - Prolactin precursor - Bovine  
; a sample sequence in FASTA format  
MDSKGSQKGSRLLLLVVSNLLCQGVVSTPVCNPGNGNCQVSLRDLFDRAMVSHYIHDLS  
EMFNEFDKRYAQKGFITMALNSCHTSSLPTPEDKEQAQQTTHHEVLMSLILGLLSWNDPLYHL  
VTEVRGMKGAPDAILSRATIEEENKRLLEGMEMIFGQVPIGAKETEPYPVWSGLPSLQTKDED  
ARYSAFYNLLHCLRRDSSKIDTYLKLNCRIIYNNNC*
```

```
>MCHU - Calmodulin - Human, rabbit, bovine, rat, and chicken  
ADQLTEEQIAEFKEAFSLFDKDGDTITTKELGTVMRSLGQNPTAEALQDMINEVDADGNGTID  
FPEFLTMMARKMKDITDSEEEIREAFRVFDKDGNGYISAAELRHVMTNLGKELTDEEVDEMIREA  
DIDGDGQVNYEEFVQMMTAK*
```

	A	B	C	D	E	F
1						
2						
3	<u>Date</u>	<u>Start time</u>	<u>End time</u>	<u>Pause</u>	<u>Sum</u>	<u>Comment</u>
4	2007-05-07	9,25	10,25	0		1 Task 1
5	2007-05-07	10,75	12,50	0	1,75	Task 1
6	2007-05-07	18,00	19,00	0		1 Task 2
7	2007-05-08	9,25	10,25	0		1 Task 2
8	2007-05-08	14,50	15,50	0		1 Task 3
9	2007-05-08	8,75	9,25	0	0,5	Task 3
10	2007-05-14	21,75	22,25	0	0,5	Task 3
11	2007-05-14	22,50	23,00	0	0,5	Task 3
12	2007-05-15	11,75	12,75	0		1 Task 3
13						
14						
15						
16						
17						
18						

### ASCII

```
#query GO reference DB reference family e-value description  
lc1|10082_1_CCCI_CCOA_CCOB_CCOC_CFAP_CFAS.6 GO:0006412 TIGRFAM TIGR00001 6e-08 translation  
lc1|10082_1_CCCI_CCOA_CCOB_CCOC_CFAP_CFAS.6 GO:0022625 TIGRFAM TIGR00001 6e-08 cytosolic larg  
lc1|10082_1_CCCI_CCOA_CCOB_CCOC_CFAP_CFAS.6 GO:0000315 TIGRFAM TIGR00001 6e-08 organellar lar  
lc1|10082_1_CCCI_CCOA_CCOB_CCOC_CFAP_CFAS.6 GO:0003735 TIGRFAM TIGR00001 6e-08 structural cor  
lc1|9019_1_CCCI_CCOA_CCOB_CCOC_CFAP_CFAS.1 GO:0005507 TIGRFAM TIGR00003 5.5e-06 copper ion bir  
lc1|9019_1_CCCI_CCOA_CCOB_CCOC_CFAP_CFAS.1 GO:0006825 TIGRFAM TIGR00003 5.5e-06 copper ion trs  
lc1|5439_1_CCCI_CCOA_CCOB_CCOC_CFAP_CFAS.1 GO:0006402 TIGRFAM TIGR00004 5.9e-67 mRNA catabolic  
lc1|5439_1_CCCI_CCOA_CCOB_CCOC_CFAP_CFAS.1 GO:0004521 TIGRFAM TIGR00004 5.9e-67 endoribonucle  
lc1|813_1_CCCI_CCOA_CCOB_CCOC_CFAP_CFAS.2 GO:0009451 TIGRFAM TIGR00005 2.1e-29 RNA modificati  
lc1|813_1_CCCI_CCOA_CCOB_CCOC_CFAP_CFAS.2 GO:0001522 TIGRFAM TIGR00005 2.1e-29 pseudouridine  
lc1|813_1_CCCI_CCOA_CCOB_CCOC_CFAP_CFAS.2 GO:0009982 TIGRFAM TIGR00005 2.1e-29 pseudouridine  
lc1|6708_1_CCCI_CCOA_CCOB_CCOC_CFAP_CFAS.4 GO:0009451 TIGRFAM TIGR00005 1.2e-18 RNA modificati  
lc1|6708_1_CCCI_CCOA_CCOB_CCOC_CFAP_CFAS.4 GO:0001522 TIGRFAM TIGR00005 1.2e-18 pseudouridine  
lc1|6708_1_CCCI_CCOA_CCOB_CCOC_CFAP_CFAS.4 GO:0009982 TIGRFAM TIGR00005 1.2e-18 pseudouridine  
lc1|4_1_CCCI_CCOA_CCOB_CCOC_CFAP_CFAS.1 GO:0009451 TIGRFAM TIGR00005 1.4e-16 RNA modification  
lc1|4_1_CCCI_CCOA_CCOB_CCOC_CFAP_CFAS.1 GO:0001522 TIGRFAM TIGR00005 1.4e-16 pseudouridine synt
```

## ANNOTATIONSUMMARY-COMBINEDORFANNOTATION16\_Phaeo\_genome

##query	length	COG hit #1	e-value #1	identity #1	score #1	hit length #1	description #1
chr_4[480001-580000].287	4500						
chr_4[560001-660000].1	3556						
chr_9[400001-500000].503	4211	COG4547	2.00E-04	19	44.6	620	Cobalamin biosynthesis protein
chr_9[320001-420000].548	2833	COG5406	2.00E-04	38	43.9	1001	Nucleosome binding factor SPN
chr_27[320001-404298].20	3991	COG4547	5.00E-05	18	46.2	620	Cobalamin biosynthesis protein
chr_26[320001-420000].378	3963	COG5099	5.00E-05	17	46.2	777	RNA-binding protein of the Puf
chr_26[400001-441226].196	2949	COG5099	2.00E-04	17	43.9	777	RNA-binding protein of the Puf
chr_24[160001-260000].65	3542						
chr_5[720001-820000].339	3141	COG5099	4.00E-09	20	59.3	777	RNA-binding protein of the Puf
chr_9[160001-260000].243	3002	COG5077	1.00E-25	26	114	1089	Ubiquitin carboxyl-terminal hyd
chr_12[720001-820000].86	2895	COG5032	2.00E-09	30	60.5	2105	Phosphatidylinositol kinase and
chr_12[800001-900000].109	1463	COG5032	1.00E-09	30	60.1	2105	Phosphatidylinositol kinase and
chr_11[1-100000].70	2886						
chr_11[80001-180000].100	1523						

## COGAnnotation\_coastal\_sample.txt

id	query	hit	e_value	identity_	score	query_start	query_end	hit_start	hit_end	hit_length
1	FHJ7DRN01A0TND.1	COG0414	1.00E-08	28	51	1	74	180	257	285
2	FHJ7DRN01A1AD2.2	COG0092	3.00E-20	47	89.9	6	85	41	120	233
3	FHJ7DRN01A2HWZ.4	COG3889	0.0006	26	35.8	9	94	758	845	872
...										
2853	FHJ7DRN02HXTBY.5	COG5077	7.00E-09	37	52.3	3	77	313	388	1089
2854	FHJ7DRN02HZO4J.2	COG0444	2.00E-31	67	127	1	73	135	207	316
...										
3566	FHJ7DRN02FUJW3.1	COG5032	1.00E-09	32	54.7	1	75	1965	2038	2105
...										

**SELECT \* FROM Phaeo P, Coastal C WHERE P.hit = C.hit**

## ANNOTATIONSUMMARY-COMBINEDORFANNOTATION16\_Phaeo\_genome

##query	length	COG hit #1	e-value #1	identity #1	score #1	hit length #1	description #1
chr_4[480001-580000].287	4500						
chr_4[560001-660000].1	3556						
chr_9[400001-500000].503	4211	COG4547	2.00E-04	19	44.6	620	Cobalamin biosynthesis protein
chr_9[320001-420000].548	2833	COG5406	2.00E-04	38	43.9	1001	Nucleosome binding factor SPN
chr_27[320001-404298].20	3991	COG4547	5.00E-05	18	46.2	620	Cobalamin biosynthesis protein
chr_26[320001-420000].378	3963	COG5099	5.00E-05	17	46.2	777	RNA-binding protein of the Puf
chr_26[400001-441226].196	2949	COG5099	2.00E-04	17	43.9	777	RNA-binding protein of the Puf
chr_24[160001-260000].65	3542						
chr_5[720001-820000].339	3141	COG5099	4.00E-09	20	59.3	777	RNA-binding protein of the Puf
chr_9[160001-200000].1	3077	COG077	1.00E-25	26	114	1089	Ubiquitin carboxyl-terminal hydrolase
chr_12[720001-820000].339	3141	COG5099	4.00E-09	20	59.3	777	RNA-binding protein of the Puf

id	query	hit	e_value	query_start	query_end	hit_start	hit_end	hit_length
6409	FHJ7DRN01BYA61.1	TIGR00149	2.20E-21	1	84	43	125	134
6410	FHJ7DRN01BDTEA.1	TIGR00149	3.40E-09	3	42	30	69	134
6411	FHJ7DRN02HEUGQ.1	TIGR00149	1.70E-05	4	46	1	46	134
6412	FHJ7DRN01CA4BO.1	TIGR00149	5.30E-05	4	45	1	45	134
6413	FHJ7DRN01DM2FK.3	TIGR01651	5.70E-64	1	76	511	586	606
6414	FHJ7DRN01B8BPS.1	TIGR01651	1.20E-36	1	52	500	551	606
6415	FHJ7DRN02JM54P.1	TIGR01651	2.20E-24	15	80	301	366	606
6416	FHJ7DRN02FK6C5.2	TIGR00039	2.70E-16	1	45	37	85	153
6417	FHJ7DRN01D019A.1	TIGR00039	8.90E-12	5	65	48	118	153
6418	FHJ7DRN02FYAFO.1	TIGR00039	1.60E-11	1	76	67	153	153

SwissProt

Search in  
Protein

1 result for CC0672 in UniProt

Reduce sequence redundancy

Did you mean cc067?

PIRSF	TIGR01650	GO:0051116	contributes_to
SMART	TIGR01651	GO:0009236	NULL
TIGRFAMs	TIGR01651	GO:0051116	NULL
PROSITE	TIGR01660	GO:0008940	NULL
ProtoNet	TIGR01660	GO:0009061	NULL
	TIGR01660	GO:0009325	NULL
	TIGR01663	GO:0000012	NULL
	TIGR01663	GO:0046403	NULL

family,  
YPR042c  
39.05  
SP55G2.14 SPCC1682.08c

Accession	Entry name	Status	Protein names
Q9AADO	Q9AADO_CAUCR	★	Cobalamin biosynthesis protein



# SQLSHARE

SQLShare is an easier way to store and share your data. Get answers to your research questions right now.

The screenshot shows the SQLShare web application interface. The browser address bar displays 'http://sqlshare.cloudapp.net/'. The page title is 'SQLShare Application'. The main content area is divided into several sections:

- Left Sidebar:** Contains navigation options like 'All queries', 'Favorites', 'Recently Viewed', 'Share with you', 'Upload new data', 'Create new query', and 'Tags' (test, database, blah, sadfsdf, folder, tgrfam).
- Top Bar:** Shows the current database 'TGRFamAnnotation\_sur...', a table 'Table02\_100032', and user information 'Owner: Kevin, Availability: Private'.
- Query Editor:** Contains a SQL query: `SELECT year,month,observeDFlux FROM BioadHepGolarIndexLee`. Below the query is a 'RUN QUERY' button.
- Data Table:** Displays the results of the query in a table with columns: class\_a, class\_b, accession, name, accession, name, accession, name. The table contains two rows of data related to 'Amino acid biosynthesis'.
- Bottom Bar:** Includes buttons for 'DOWNLOAD', 'VISUALIZE', 'COPY', 'QUERY TABLE', and 'DELETE THIS DATASET'.

Log in using your account:

 UNIVERSITY of WASHINGTON



Don't have an account?

Create a [Google Account](#) and start using SQLShare quickly.

## Upload

Upload any tabular data and start analyzing instantly. No need to install, configure, or design a database.

## Modify

Exercise the full power of SQL even with zero programming experience: joins, subqueries, set operations.

## Share

Analyze and compare your data collaboratively. Derive new datasets and share them with your colleagues.

## Your datasets

All datasets

Favorites

Recently viewed »

Shared with you...

Upload dataset

New query

## Your Datasets

Name	Sharing / Owner	Created
Amazon: TIGRFam Hit Counts with Sample Metadata, only TE_20174 Hit counts for each TIGRFam protein with	billhowe@washington.edu	Nov 10, 2010 11:56 AM
SDSS 200006-g4-0100 SDSS 200006-g4-0100	billhowe@washington.edu	Nov 2, 2010 7:49 PM
Join Training Data from SDSS logs 39 joins extracted from the SDSS logs, plus 40 "bad" joins.	billhowe@washington.edu	Oct 29, 2010 0:47 PM
SeasonStripColorGeo_bbox add bounding box to SeasonStripColor	billhowe@washington.edu	Oct 28, 2010 8:50 AM
SeasonStripColor_bbox Adding bounding box	billhowe@washington.edu	Oct 27, 2010 10:47 PM
SeasonStripColorGeo testing geo coordinates	billhowe@washington.edu	Oct 27, 2010 11:07 AM
SeasonStripColor Cast all px columns to floats	billhowe@washington.edu	Oct 25, 2010 4:46 PM
chunk tabs	billhowe@washington.edu	Oct 24, 2010 8:39 PM
Stripe 82 sequence file meta data Metadata for all images in the stripe 82 subset of the sloan digital sky survey	billhowe@washington.edu	Oct 24, 2010 8:35 PM
900000_chunk.txt description	billhowe@washington.edu	Oct 23, 2010 4:15 PM
800000_chunk.txt description	billhowe@washington.edu	Oct 23, 2010 4:13 PM
700000_chunk.txt description	billhowe@washington.edu	Oct 23, 2010 4:12 PM
600000_chunk.txt description	billhowe@washington.edu	Oct 23, 2010 4:10 PM
500000_chunk.txt description	billhowe@washington.edu	Oct 23, 2010 4:09 PM
400000_chunk.txt description	billhowe@washington.edu	Oct 23, 2010 4:07 PM
3900000_chunk.txt description	billhowe@washington.edu	Oct 23, 2010 4:05 PM
3800000_chunk.txt description	billhowe@washington.edu	Oct 23, 2010 4:05 PM
3700000_chunk.txt description	billhowe@washington.edu	Oct 23, 2010 4:03 PM
3600000_chunk.txt description	billhowe@washington.edu	Oct 23, 2010 4:01 PM
3500000_chunk.txt description	billhowe@washington.edu	Oct 23, 2010 4:00 PM
3400000_chunk.txt description	billhowe@washington.edu	Oct 23, 2010 3:58 PM

- Your datasets
- All datasets
- Favorites
- Recently viewed »
- Shared with you...

Upload dataset  
New query

**Amazon: TIGRFam Hit Counts with Sample Metadata** <

Last modified: Nov 4, 2010 1:57 PM rkodner@washington.edu

Hit counts for each TIGRFam protein with all sample metadata including day/night information. From Amazon transect samples.

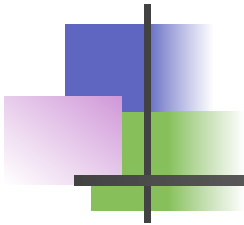
```
SELECT s.TIGRFam, normalized_hit_count, m.*
FROM [rkodner].[Amazon Sample Metadata] m
, [rkodner].[Amazon: TIGRFam Hit Counts by Sample] s
WHERE m.Sample = s.Sample
```

[Copy query](#) [Download](#) [Query dataset](#)

DATASET PREVIEW (Rows 1 - 100 of 22240)

<< first < prev 1 2 3 4 5 next > last >>

TIGRFam	normalized_hit_count	Sample	Station	Latitude	Longitude	SampleTime	Habitat	Depth	Temperature	Salinity	Oxygen	Filter Size	Sample Volume	Com
TIGR00004	1.005687988	TE_20174	SJ0609.003	12.28	-56.12	6/27/2006 8:30:00 AM	West Tropical Atlantic Province; Oligotrophic Open Ocean	5	28.46	31.71	Aerobic	5	110	
TIGR00004	0	TE_20176	SJ0609.003	12.28	-56.12	6/28/2006 10:00:00 PM	West Tropical Atlantic Province; Oligotrophic Open Ocean	5	28.46	31.71	Aerobic	5	40	



share status

name

owner

### AmazonMetadata with Timestamp

Last modified: Nov 3, 2010 10:22 PM

rkodner@washington.edu

description

Convert time and date fields to a single timestamp called SampleTime

SQL

```

SELECT Sample, Station, Latitude, Longitude,
       CAST(CAST(CAST([Local Date] AS date) AS varchar(max))
+ ' ' + CAST(CAST([Local Time] AS time) AS varchar(max)) AS datetime2)
as SampleTime
, Habitat, Depth, Temperature, Salinity, Oxygen, [Filter Size]
, [Sample Volume], [Comment Box]
, [PI], Project, [Chlorophyl A]
FROM [rkodner].[AmazonMetadata.txt]

```

actions

Copy query

Download

Query dataset

preview

DATASET PREVIEW (Rows 1 - 40 of 40)

<< first < prev 1 **2** next > last >>

Sample	Station	Latitude	Longitude	SampleTime	Habitat	Depth	Tem
TA_20174	SJ0609.003	12.28	-56.12	6/27/2006 8:30:00 AM	West Tropical Atlantic Province; Oligotrophic Open Ocean	5	28.4
				6/28/2006	West Tropical Atlantic		

- Your datasets
- All datasets
- Favorites
- Recently viewed »
- Shared with you...
- Upload dataset
- New query

### Amazon: TIGRFam Hit Counts with Sample Metadata

Last modified: Nov 4, 2010 1:57 PM rkodner@washington.edu

Hit counts for each TIGRFam protein with all sample metadata including day/night information.  
From Amazon transect samples.

```
SELECT s.TIGRFam, normalized_hit_count, m.*
FROM [rkodner].[Amazon Sample Metadata] m
, [rkodner].[Amazon: TIGRFam Hit Counts by Sample] s
WHERE m.Sample = s.Sample
```

Copy query Download Query dataset

DATASET PREVIEW (Rows 1 - 100 of 22240)

<< first < prev 1 2 3 4 5 next > last >>

TIGRFam	normalized_hit_count	Sample	Station	Latitude	Longitude
TIGR00004	1.005687988	TE_20174	SJ0609.003	12.28	-56.12
TIGR00004	0	TE_20176	SJ0609.003	12.28	-56.12
TIGR00004	0.985517968	TE_20185	SJ0609.004	12	-54.39
TIGR00004	1.120915221	TE_20186	SJ0609.004	12	-54.39
TIGR00004	0	TE_20187	SJ0609.004	12	-54.39
TIGR00004	2.598848164	TE_20188	SJ0609.004	12	-54.39
TIGR00004	1.299424082	TE_20189	SJ0609.005	11.68	-51.5

### COPYING Amazon: TIGRFam Hit Counts with Sample Metadata

```
SELECT s.TIGRFam, normalized_hit_count, m.*
FROM [rkodner].[Amazon Sample Metadata] m
, [rkodner].[Amazon: TIGRFam Hit Counts by Sample] s
WHERE m.Sample = s.Sample
```

Execute query

DATASET PREVIEW (Rows 1 - 100 of 22240)

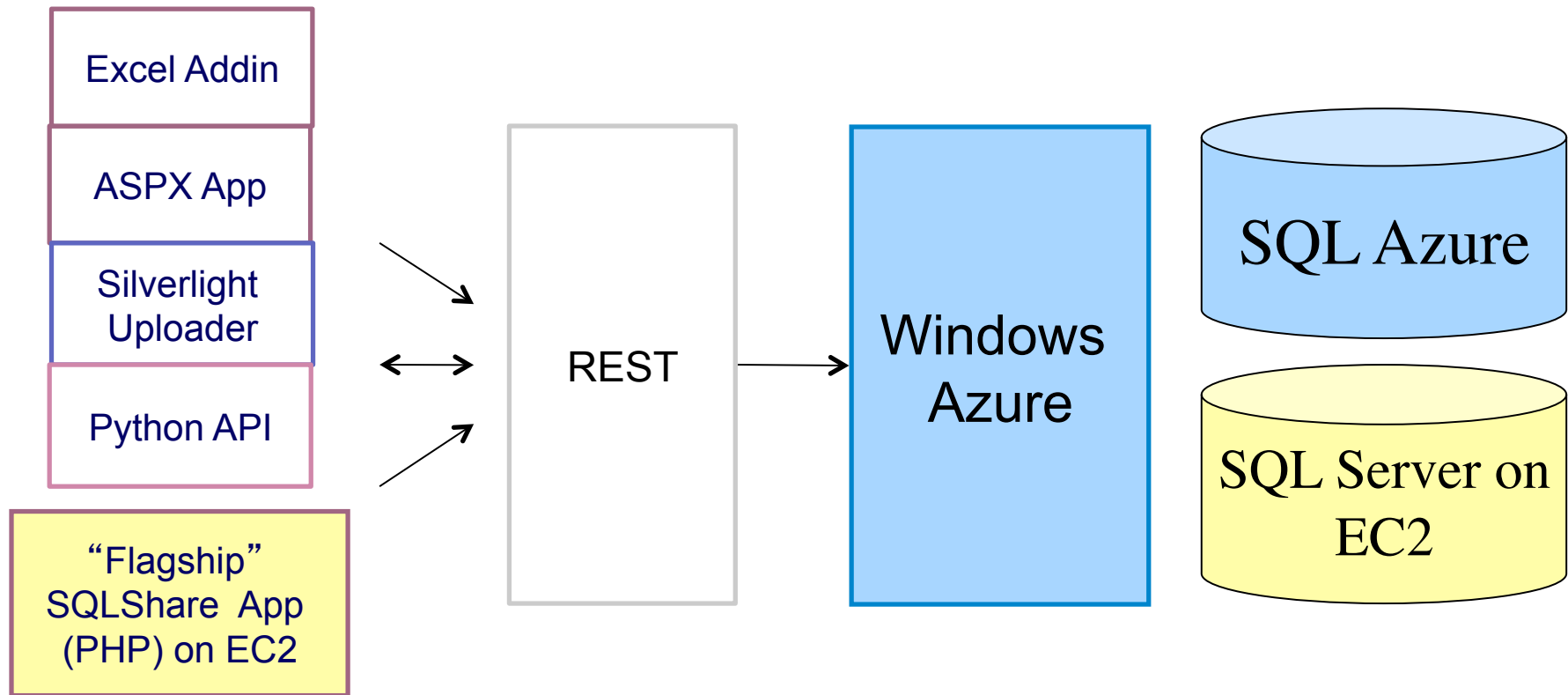
<< first < prev 1 2 3 4 5 next > last >>

TIGRFam	normalized_hit_count	Sample	Station	Latitude
TIGR00004	1.005687988	TE_20174	SJ0609.003	12.28
TIGR00004	0	TE_20176	SJ0609.003	12.28
TIGR00004	0.985517968	TE_20185	SJ0609.004	12
TIGR00004	1.120915221	TE_20186	SJ0609.004	12
TIGR00004	0	TE_20187	SJ0609.004	12
TIGR00004	2.598848164	TE_20188	SJ0609.004	12
TIGR00004	1.299424082	TE_20189	SJ0609.005	11.68
TIGR00004	2.37966475	TE_20190	SJ0609.005	11.68

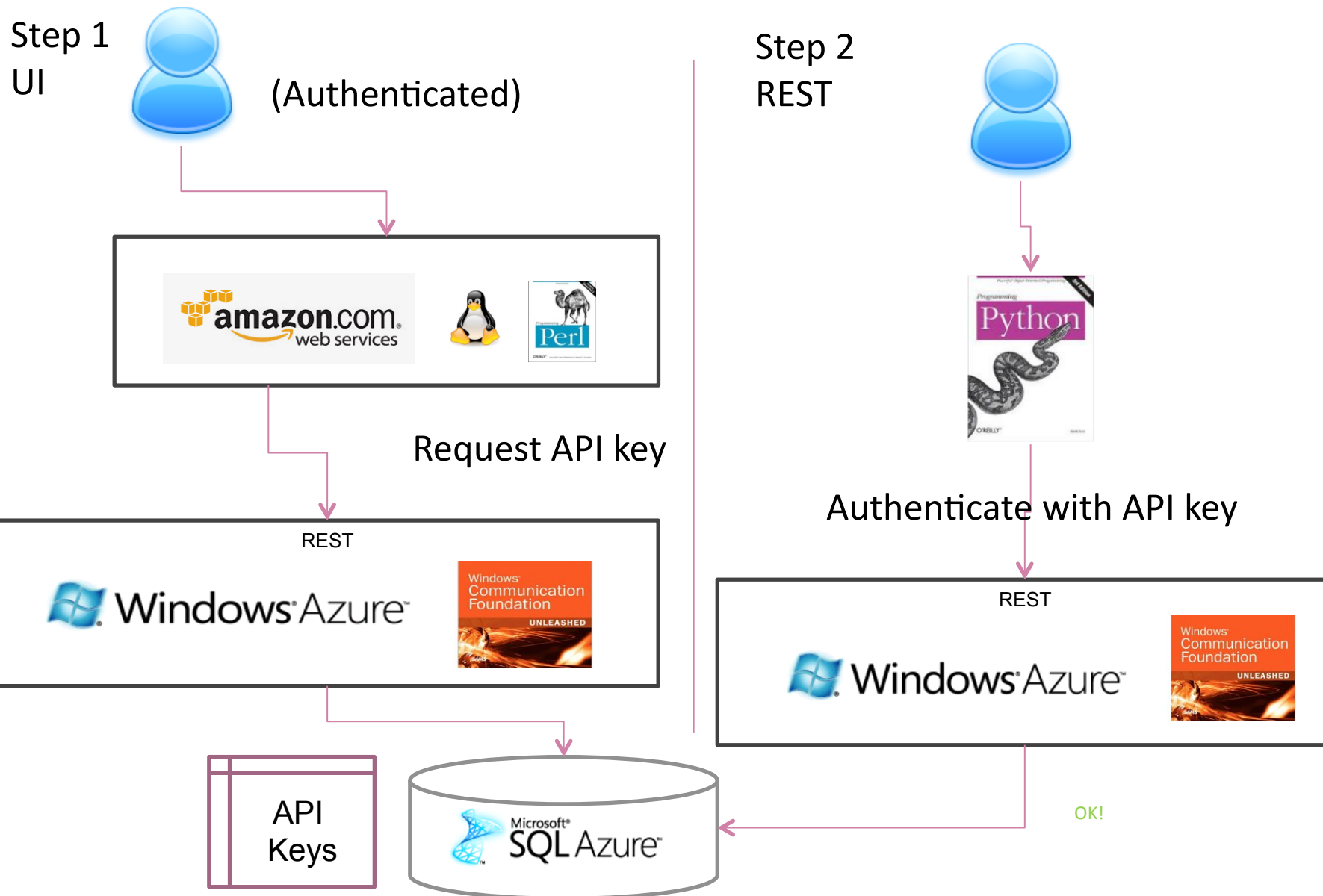
Save as

Cancel

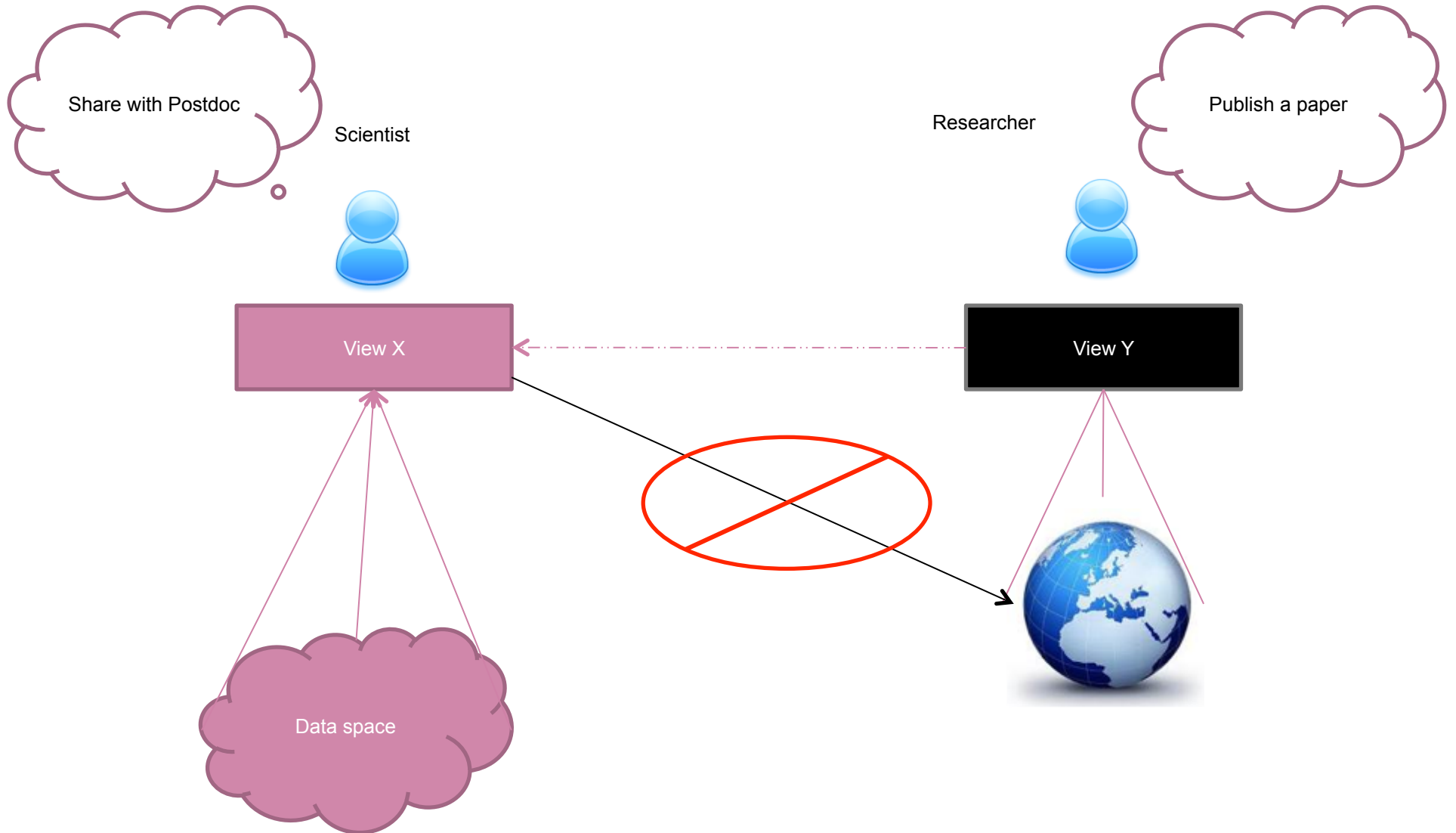
# Architecture



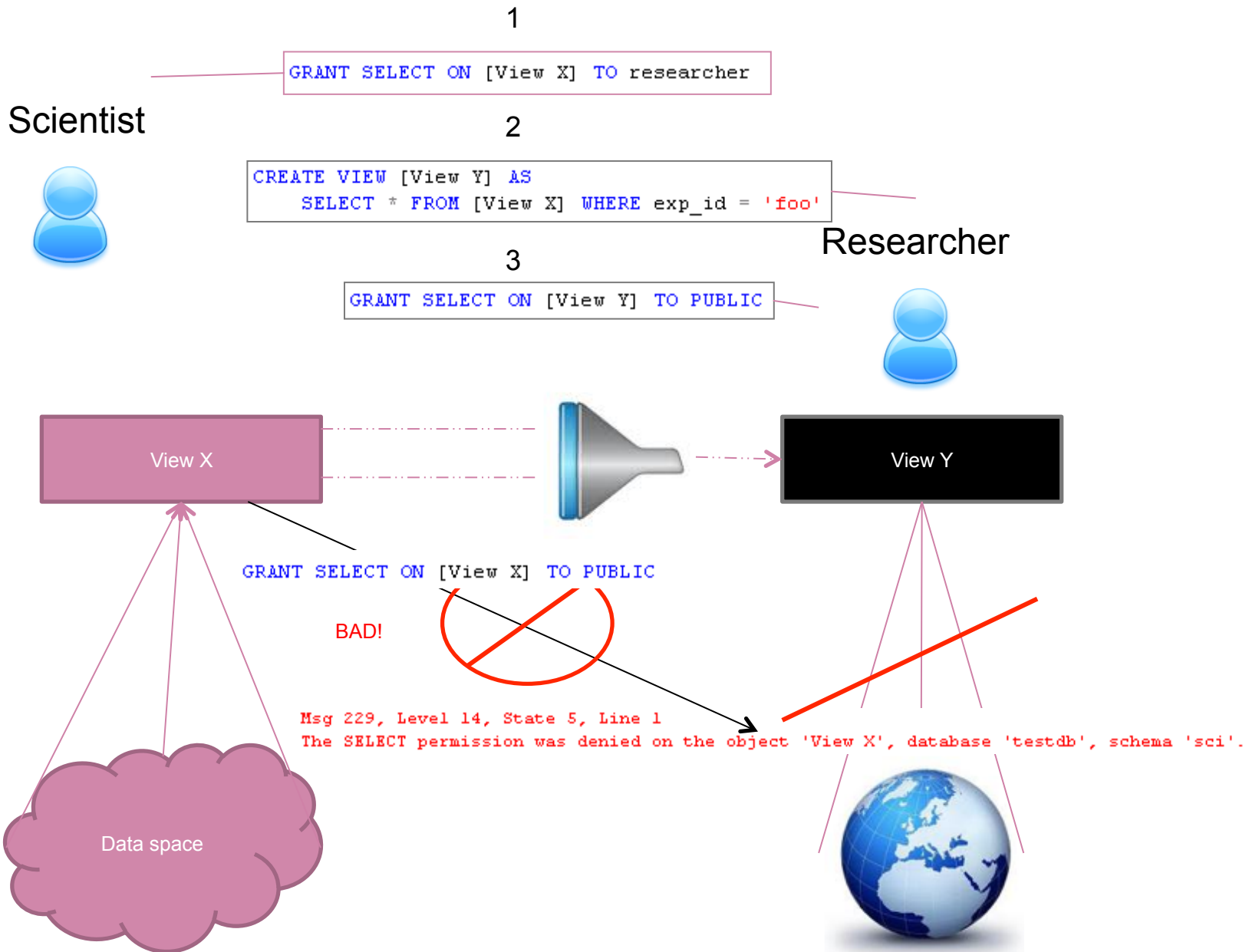
# API Authentication



# Sharing data







# View-oriented workflow

**sdss\_single\_column** 🔒

[Click here to add a description](#)

```
select *,
  case when [count] = distinct_count then 1 else 0
from dbo.single_column_stats
```

**sdss\_column\_composite** ⏪

[Click here to add a description](#)

```
tc.distinct_count as target_distinct_count,
tc.min_value as target_min_value,
tc.max_value as target_max_value,
tc.unique_values as target_unique_values
FROM sdss_join_column_jaccard jc
LEFT JOIN sdss_single_column sc ON jc.source_id = sc.source_id
LEFT JOIN sdss_single_column tc ON jc.target_id = tc.source_id
```

**sdss\_join\_column** 🔒

[Click here to add a description](#)

```
select distinct * from dbo.join_column_stats
```

**sdss\_join\_column\_jaccard** 🔒

[Click here to add a description](#)

```
END as jaccard_projected_value_set,
CASE projected_value_union_bag
WHEN 0 THEN 0
ELSE
(convert(float, projected_value_intersection) /
END as jaccard_projected_value_bag
FROM sdss_join_column order by jaccard_projected_value_set
```

**sdss\_column\_composite\_with\_features** ⏪

[Click here to add a description](#)

```
END AS B,
CASE WHEN (projected_value_union_set > projected_value_intersection)
THEN 1
ELSE 0
END AS C,
* FROM [gbc3].[sdss_column_composite]
where projected_value_union_set is not null
```

**go\_features\_specified** ⏪

[Click here to add a description](#)

```
SELECT * FROM [gbc3].[sdss_column_composite_with_features]
where experiment_id = 'sdss_subsample_request_go'
```



## More examples

---

- Which samples have not been cloned?
- How often does each RNA hit appear inside the annotated surface group?
- How many plasmids were bombarded in July and have a rescue and expression?

```
SELECT *  
FROM plasmiddb  
WHERE NOT (ISDATE(cloned)  
OR cloned = 'yes')
```

```
SELECT hit, COUNT(*) as cnt  
FROM tigrfamannotation_surface  
GROUP BY hit  
ORDER BY cnt DESC
```

```
SELECT count(*)  
FROM [bombardment_log]  
WHERE bomb_date BETWEEN  
'7/1/2010' AND '7/31/2010'  
AND rescue clone IS NOT NULL  
AND [expression?] = 'yes'
```



## My favorite example

---

- Find all TIGRFam ids (proteins) that are missing from at least one of three samples (relations)

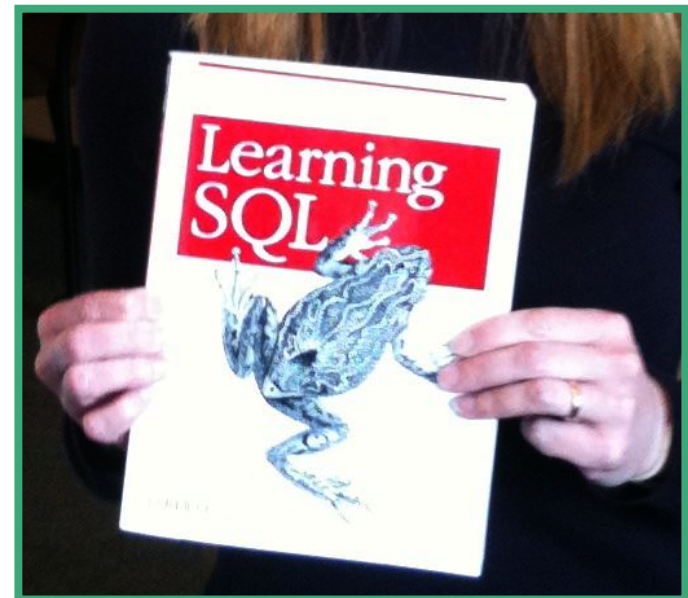
```
SELECT col0 FROM [refseq_hma_fasta_TGIRfam_refs]
UNION
SELECT col0 FROM [est_hma_fasta_TGIRfam_refs]
UNION
SELECT col0 FROM [combo_hma_fasta_TGIRfam_refs]

EXCEPT

SELECT col0 FROM [refseq_hma_fasta_TGIRfam_refs]
INTERSECT
SELECT col0 FROM [est_hma_fasta_TGIRfam_refs]
INTERSECT
SELECT col0 FROM [combo_hma_fasta_TGIRfam_refs]
```

# So what's the point?

- Databases appear underused in (long tail) science
- Conventional wisdom says “Scientists wont write SQL”
  - This is utter horseshit
  - c.f. SDSS, Life Under Your Feet, UW eScience
- Instead, we blame the up-front costs
  - installation
  - configuration
  - **schema design**
  - performance tuning
  - loading
  - app-building



***So we ask:***

***What kind of platform can deliver SQL to scientists?***



## Desiderata for a “SQL Delivery Vector”

---

- Logical data independence is a good idea; let's do more of that
- Loading data is always a pain; let's make that easier
- No updates to science data; let's cache aggressively and support append/replace only
- Scale is small ( $O(100)$  spreadsheets); let's not worry too much about physical tuning
- Reliable schemas are elusive at the frontier of research; let's worry about that later (or not at all)
- SQL is not difficult to learn – *given a set of relevant “starter queries” to build from*



# Status

---

- about 6 months old, no active advertising
- 4 groups associated with have data uploaded
- 50 unique users
- 566 uploaded tables
- 181 views
- 16GB of data
- 3 short/demo papers; “flagship” paper underway
- Interested parties
  - NatureMapping project, Washington Sea Grant, H2O, HIV Global Enterprise, San Juan country MRC, NatureMapping Project, MSR in various contexts
- “quiet” release underway; wide announcement pending scalability planning – we need advice!



# Use cases we are seeing

---

- **Poor man' s LIMS**

- Enter data via spreadsheets, upload to SQLShare for holistic analysis

- **Pilot Projects**

- Before investing in a conventional database design project, throw your data in SQLShare to understand what you're working with

- **Collaborative Query Management System\***

- YouTube for SQL Queries

- **Data “Instrument”\*\***

- Put your scattered data under the “SQLScope”

- **Citizen Science**

- Stage 1: Democratization of Data Collection
- Stage 2: Democratization of Data Analysis

\*Khoussinova, CIDR 2009

\*\*credit: Alex Szalay





# Eating our own dogfood

---

## timings recorded into sqlshare Inbox | X



**Garret Cole** to me

[show details](#) 2:17 PM (9 minutes ago)

[Reply](#)



I have shared a dataset that has the timings recorded as milliseconds:

[https://sqlshare.escience.washington.edu/sqlshare#s=query/gbc3/join\\_column\\_stats\\_timings](https://sqlshare.escience.washington.edu/sqlshare#s=query/gbc3/join_column_stats_timings)

Also I have created a the distinct ERRR WOOPS, i just realized i forgot to include the distinct keyword in my select INTO statement. i'll have to rerun that. But amazingly it goes pretty damn quick, it only took 25 minutes to create select INTO for every single column in the skyserver db

Garret

[Reply](#)

[Forward](#)

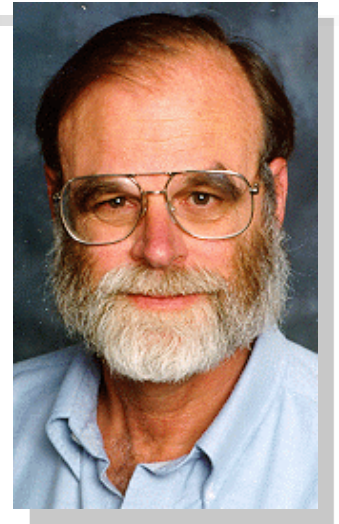


# Bootstrapping new users

---

*A mini version of Jim Gray's 20 questions methodology*

1. Give us your data
2. Give us 20 questions in English
3. Our job
  - upload the data
  - translate the queries
  - share them in SQLShare



*This process has been demonstrably successful, but doesn't scale*



# Automating the Process

---

- **Automatic Starter Queries**
  - Garret Cole (Research Programmer)
  - Nodira Khoussinova (CSE, Phd student)
  - Leilani Battle (CSE, Ugrad)
  - with Phil Bernstein (MSR)
- **Automatic Web Visualization**
  - Alicia Key (Research Programmer)
- **SQL Autocomplete**
  - Nodira Khoussinova (CSE, Phd student)
  - Magda Balazinska (CSE faculty)
- **Automatic English to SQL**
  - Luke Zettlemyer (CSE faculty) and
  - Shaminoo Kapoor (Applied Math, masters student)
- **Personalized Query Recommendation**
  - Yuan Zhou (Applied Math, masters student)



# Automatic Starter Queries

---

- We find that Starter queries are sufficient for users to “self-serve”
- But DB experts must provide these, and this doesn’t scale
- *Hypothesis: We can automatically derive a set of “good” starter queries directly from the data*
- Challenge: With an ad hoc database, we cannot assume a schema, query logs, or prior user input



# Potential Goals for Starter Queries

- SQL training

- Database profiling

- Ex: `SELECT a, COUNT(*) FROM R HAVING COUNT(*) > 1`

- Ex: `SELECT * FROM R INTERSECTION SELECT * FROM S`

- Bootstrapping logical design

- Ex: Reconstruct partitioned tables

```
SELECT * FROM [East Sound]
UNION
SELECT * FROM [Presidents Channel]
...
```

```
SELECT *
FROM [Men's 12M - Alcohol/Drug] a,
[Men's 12M - Demographics] d,
...
WHERE a.patient = d.patient
...
```



# Desiderata for a set of starter queries

---

- Coverage of SQL idioms
- Coverage of data
- Coverage of query complexity



# Approach for Joins

---

- Derive heuristics on what a “good query” means
- Examples of join heuristics:
  - *Two columns exhibiting a foreign key relationship*
  - *Two columns with high Jaccard similarity*
  - *Two columns with similar active domains, where one has higher cardinality, indicates a 1:N join*
  - *many more*
- Relative influence of these heuristics unclear
- So: Extract features from the data covering all cases, and learn a model from existing “starter query” examples



# Features Extracted

---

Feature	Expression
max/min cardinality	$\max/\min( x ,  y )$
cardinality difference	$\text{abs}( x  -  y )$
intersection cardinality	$ x \cap y $
union cardinality	$ x \cup y $
Jaccard similarity	$\frac{ x \cap y }{ x \cup y }$

Compute all of these for both set and bag semantics





# Experimental Design

---

1. SDSS DB: Learn a decision tree on these features, using the joins present in the query logs as ground truth
2. Test the decision tree on the sample queries provided on the SDSS website
3. Gene Ontology DB: Extract features on the Gene Ontology database
4. Test the *same decision tree* on the sample queries from the GO website



# Preliminary Results

---

- SDSS Database:
  - recall and precision both around 91%
  
- GO Database:
  - Recall 93%: 28/30 joins in sample queries classified correctly
  - Precision 96%: 11/12 “bad” joins classified correctly



# Decision Tree

---

- | (1)jaccard\_projected\_value\_set < 0.002: 1.016
  - | | (2)source\_unique\_values < 0.5: 1.246
    - | | (2)source\_unique\_values >= 0.5: -0.623
      - | | | (5)projected\_value\_union\_set < 592.5: 0.441
        - | | | (5)projected\_value\_union\_set >= 592.5: -0.512
          - | | | | (7)target\_count < 100541: -0.406
            - | | | | (7)target\_count >= 100541: 0.23
        - | | | (4)source\_distinct\_count < 195851: 0.985
          - | | | (6)projected\_value\_union\_bag < 1030.5: -0.086
            - | | | (6)projected\_value\_union\_bag >= 1030.5: 0.707
          - | | | (4)source\_distinct\_count >= 195851: -1.137
      - | (1)jaccard\_projected\_value\_set >= 0.002: -1.253
        - | | (3)source\_count < 55: 2.076
          - | | (3)source\_count >= 55: -2.363
    - Tree size (total number of nodes): 22
    - Leaves (number of predictor nodes): 15



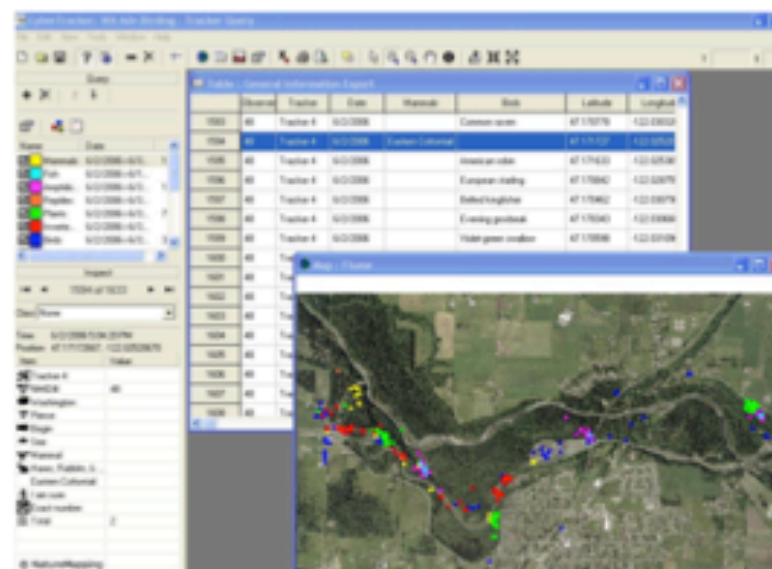
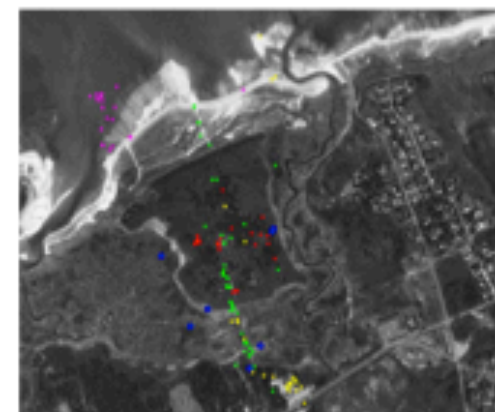
# Ongoing work

---

- Collaborative Features
  - Favorites, Likes
  - “People who ran this query also ran...”
  - Annotations
- Mining clicklogs
  - with Hazeline Asuncion at UW Bothell
- Visualization
- New collaborations
  - Citizen science
  - Global Enterprise HIV Vaccine

# What is NatureTracker?

- Menu-driven program for recording field data
- includes all criteria requested for *NatureMapping* data
- runs on CyberTracker software (“greenware”)
- Used for *NatureMapping* bioblitzes

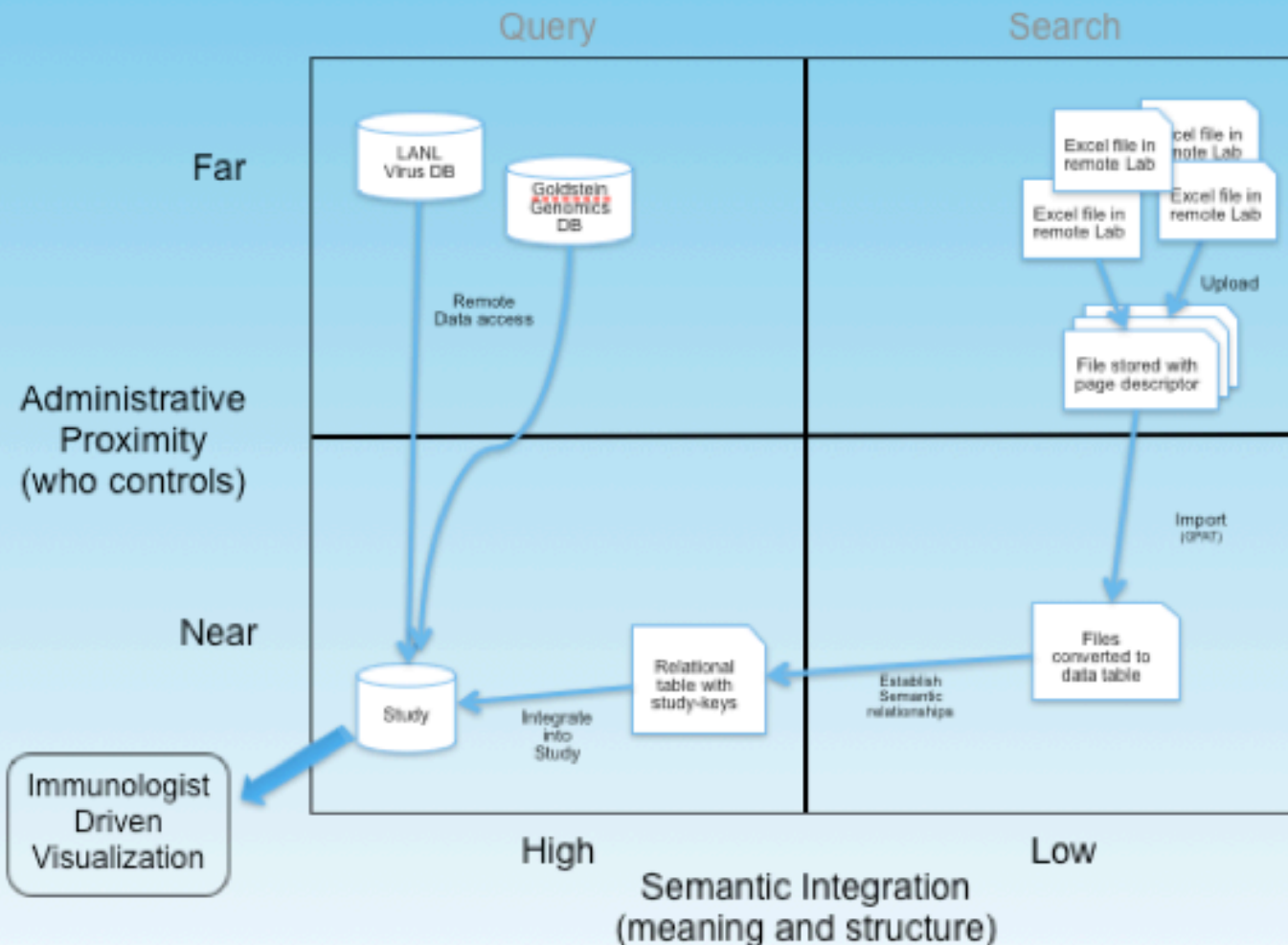


# 2010 Pilot- Outreach and Education-based sampling: Schooner Adventuress

The image is a composite graphic. At the top is a map of the San Juan Islands with labels for 'Harbour', 'Canada United States', 'Orcas Island Airport', 'YMCA Camp Orkila', 'Orcas Village', and 'Eastsound'. Below the map is a blue banner with the text 'Sound Experience' in a stylized font, where 'Sound' is in blue script and 'Experience' is in orange. To the right of the banner is the tagline 'educate inspire empower' in white lowercase letters, with a small image of a sailboat. Below the banner are four photographs: 1) Two young people, a girl in a blue jacket and a boy in a red 'ESAR' hoodie, looking out at sea. 2) A woman in sunglasses and a life vest working on the deck of a sailboat. 3) A woman smiling while holding a clear plastic container, with a boy in a blue shirt looking on. 4) A large white sailboat with the number '5' on its hull sailing on the water. At the bottom is another map showing 'San Juan Island National Historical Park', 'Lopez Island Airport', 'Anacortes Airport', 'Smak Bay', and 'Memorial Park'. Three yellow stars are marked on the map with numbers 6, 7, and 8.

# CHAVI-CAVD Dataspace Concept

Towards an HIV Enterprise Dataspace



**<http://sqlshare.escience.washington.edu>**

(Ask me about the “NoSQL” movement)



# 23<sup>rd</sup> International Conference on Scientific and Statistical Database Management (SSDBM 2011)

<http://www.ssdbm2011.ssdbm.org>

Portland, Oregon, USA

July 20 – 22, 2011

Abstracts due: January 31, 2011

Papers due: February 7, 2011





# Backup slides

---



# Desiderata

---

- Schema-Later
  - Pay-as-you-go by creating and sharing views
- Dataset-level CRUD ops
  - Append and Replace, not update
- Easy ingest
  - Parse “obvious” file formats automatically
  - Excel Add In
  - LearnPADS [Fisher 2009-2010]
- “Starter Queries” for bootstrapping analysis
- Social/Collaborative/Participatory
- Easy Visualization



# Digression: Relational Database History

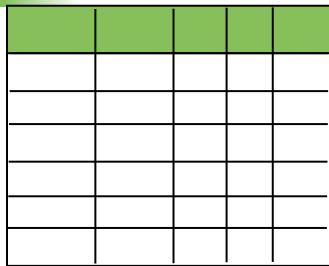
---

Pre-Relational: if your data changed, your application broke.

*“Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed.” -- Codd 1979*

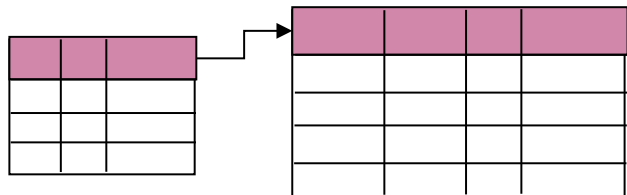
**Key Ideas:** Programs that manipulate tabular data exhibit an algebraic structure allowing reasoning and manipulation independently of physical data representation

# Key Idea: Data Independence



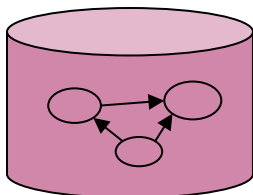
views

*logical data independence*



relations

*physical data independence*



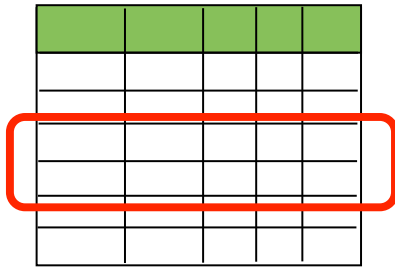
files and pointers

```
SELECT *  
FROM my_sequences
```

```
SELECT seq  
FROM ncbi_sequences  
WHERE seq =  
'GATTACGATATTA';
```

```
f = fopen('table_file');  
fseek(10030440);  
while (True) {  
    fread(&buf, 1, 8192, f);  
    if (buf == GATTACGATATTA) {  
        . . .
```

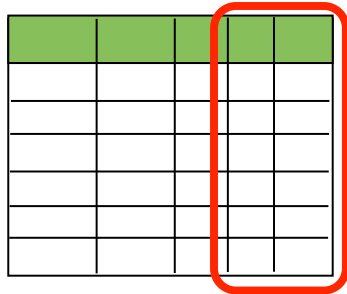
# Key Idea: An *Algebra of Tables*







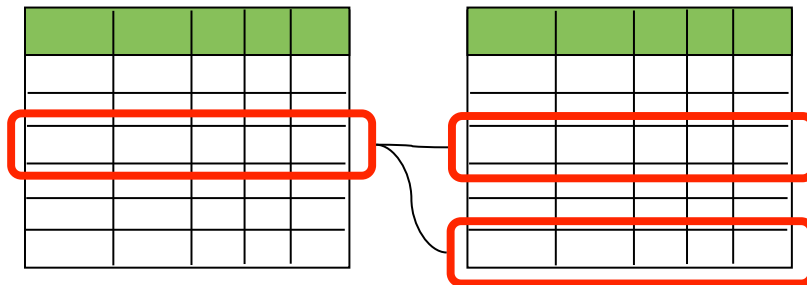
select







project











join

*Other operators: aggregate, union, difference, cross product*



# Key Idea: Algebraic Optimization

---

$$N = ((z*2)+((z*3)+0))/1$$

Algebraic Laws:

1. (+) identity:  $x+0 = x$
2. (/) identity:  $x/1 = x$
3. (\*) distributes:  $(n*x+n*y) = n*(x+y)$
4. (\*) commutes:  $x*y = y*x$

Apply rules **1, 3, 4, 2**:

$$N = (2+3)*z$$

two operations instead of five, no division operator

*Same idea works with the Relational Algebra!*



# RDBMS: Summary

---

- Intuitive data model
  - “just tables”
- Data cleaning, filtering, joins, aggregation, user-defined functions
- Physical and logical data Independence
  - Views are a good idea; let’s use more of those
- Declarative query language + algebraic optimization
  - Describe what you want, not how to get it
- Scalability
  - “SQL is the most successful parallel language in the world”
- Proven results
  - \$15B industry
  - Nearly every (non-search engine) website backed by a RDBMS
  - One of the all-time best examples of CS research impact

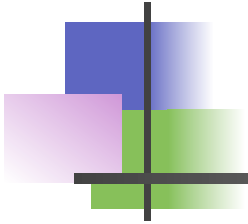




# Usage

---

- about 6 months old, but not yet advertised
- 4 labs around UW campus
- 50 unique users
- 566 uploaded tables
- 181 saved queries (i.e., views)
- 16GB of data



*How do we repeat the success of SDSS in the long tail?*

*How do we build the next 100 SDSS-like systems?*