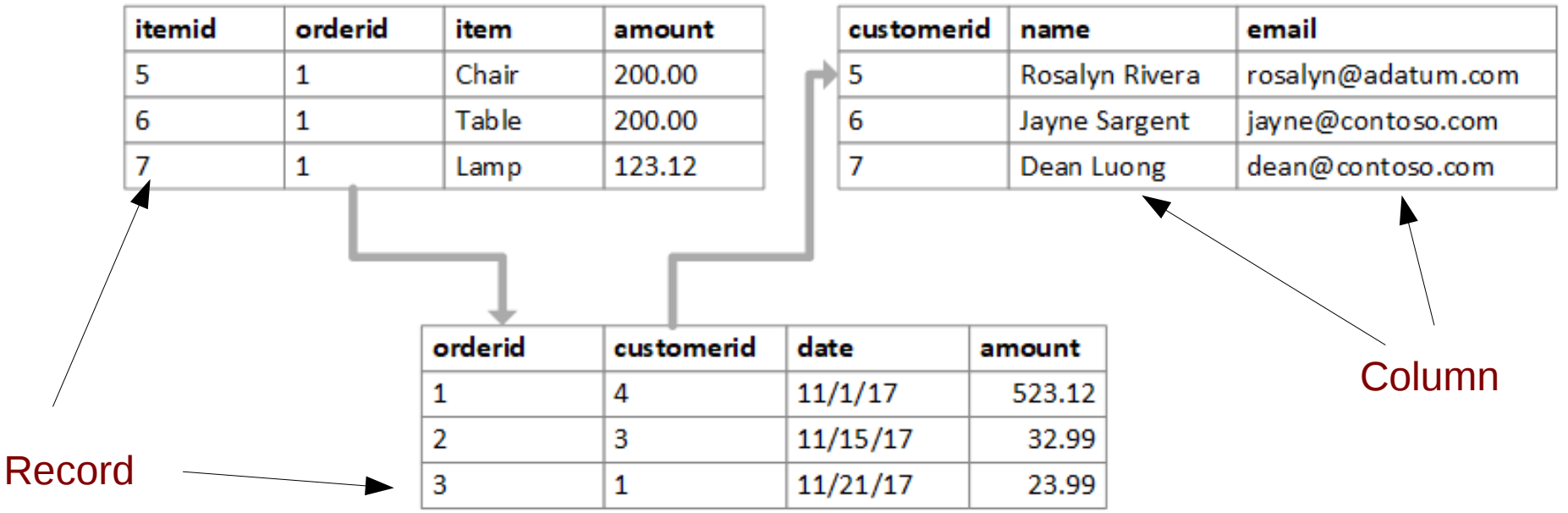
A very brief introduction to SQL

(Whit a large bias to SDSS)

ISYA 41
Socorro
2018

Juan Carlos Muñoz Cuartas
Instituto de Física
Universidad de Antioquia

Data base?



itemid	orderid	item	amount
5	1	Chair	200.00
6	1	Table	200.00
7	1	Lamp	123.12

customerid	name	email
5	Rosalyn Rivera	rosalyn@adatum.com
6	Jayne Sargent	jayne@contoso.com
7	Dean Luong	dean@contoso.com

orderid	customerid	date	amount
1	4	11/1/17	523.12
2	3	11/15/17	32.99
3	1	11/21/17	23.99

Arrangement of tables (many!), entries (many per table)

How do you get “intelligent” use of the data in the sense that you use all data connected through some common datapoints or flags?

How do you use the data “intelligent” if you want to search for as many questions as people in the earth?

Basic instruction set for SQL query

1. **SELECT** clause. Specifies the parameters you wish to retrieve from DB.

(what you want to get!)

2. **FROM** clause. Specifies the database tables from where you want to extract the data.

(from where you want to get it!)

3. **WHERE** clause. Specifies the limitations/constraints you want to impose on the extracted data.

(Conditions imposed on the data itself)

Example 1

```
SELECT objID,
Field, ra, dec
FROM PhotoObj
WHERE run=1336 and field = 11
```

(Obtains the unique objectID)

(Obtains the field number, and coordinates)

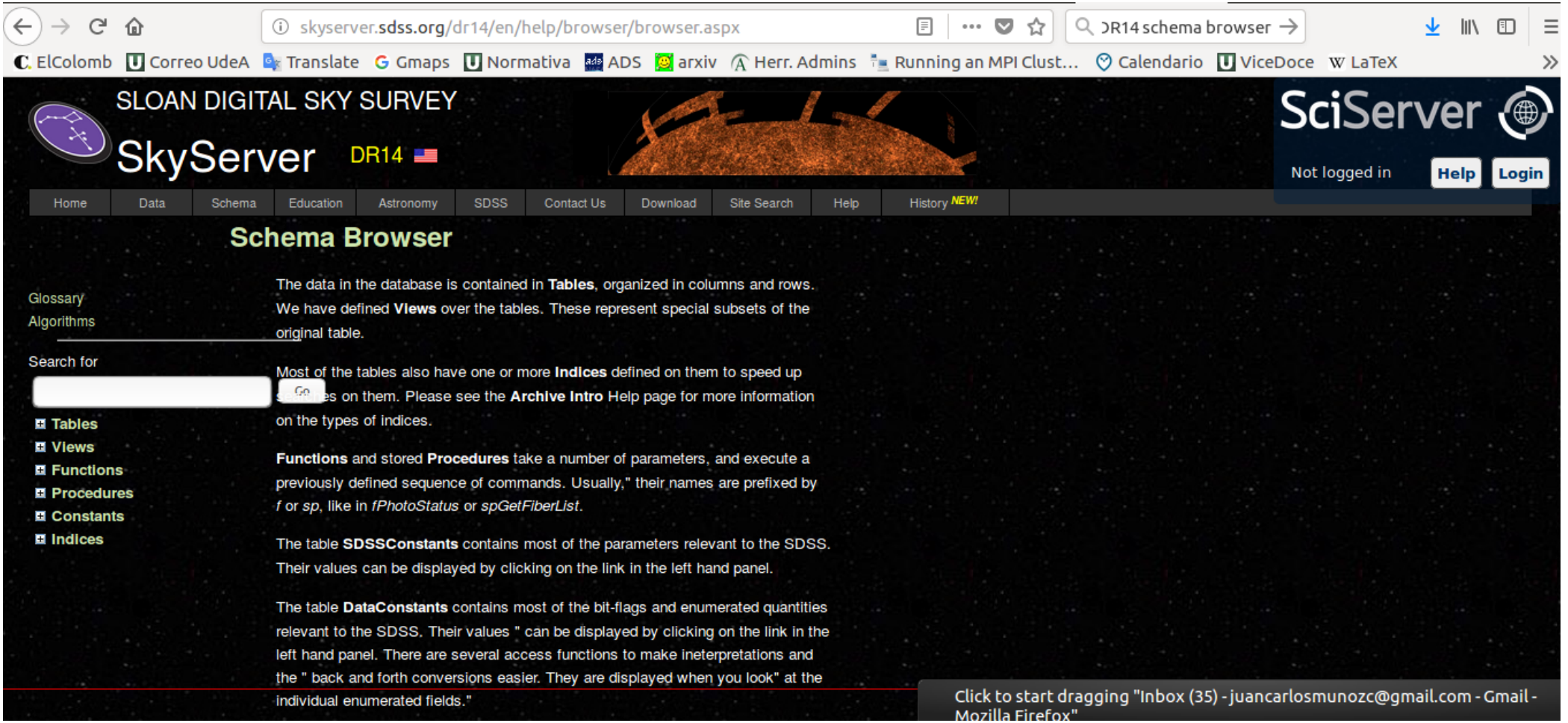
(From the photometric data table)

(Conditions that matches our criteria)

How do you know what are the entries of the database tables?

Schema database

http://skyserver.sdss.org/dr14/en/help/browser/browser.aspx



Tables → Views → Functions and procedures → Constants

Imposing no conditions to the query

```
SELECT ra, dec
FROM Galaxy
```

Take all RA and DEC coordinates for all galaxies in the Galaxy VIEW

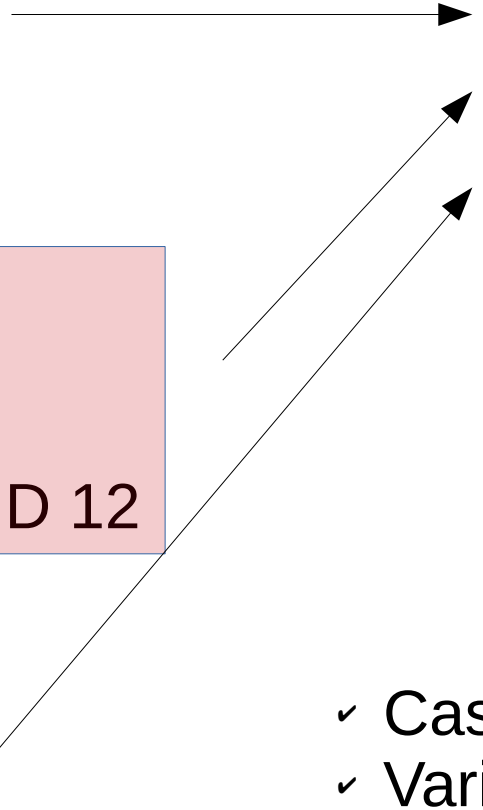
Rich syntax for SQL

(sometimes source of mistakes)

```
SELECT TOP 100 u,g,r,i,z
FROM Galaxy
WHERE r<12 and r>0
```

```
SELECT TOP 100 u,g,r,i,z
FROM Galaxy
WHERE r BETWEEN 0 AND 12
```

```
select top 100 u,g,r,i,z
From Galaxy
WHEre r<12 and r>0
```



They all do the same query!

- Case insensitive
- Various forms for execution of comparison

Considering invalid flags-values

```
SELECT ra, dec, u, err_u
FROM PhotoObj
WHERE ra BETWEEN 180 AND 181
AND dec BETWEEN -0.5 AND 0.5
AND u BETWEEN -9999 AND 20.0
AND err_u BETWEEN -1000 AND 0.1
```

What is the meaning of this query?

Using data from two tables

```
SELECT s.psfMag_g
FROM Star s, Field f
WHERE s.fieldID = f.fieldID
and  s.psfMag_g < 20
and  f.pspStatus = 2
```

Note that now we have an “alias” for each table

And so we use it

What is the meaning of this query?

```
SELECT p.objID, f.field, g.run
FROM PhotoObj p, Field f, Segment g
WHERE f.fieldid = p.fieldid
and  f.segmentid = g.segmentid
```

Data is taken from three different tables

Conditions are used to join them together

Or three tables!

```
Select  plateX.plate, plateX.mjd,  specObj.fiberID,
PhotoObj.modelMag_u, PhotoObj.modelMag_g,
PhotoObj.modelMag_r, PhotoObj.modelMag_i, PhotoObj.modelMag_z,
PhotoObj.ra, PhotoObj.dec, specObj.z, PhotoObj.ObjID
from
    photoObj, specObj, plateX
where
    specObj.bestObjid = PhotoObj.ObjID
    and plateX.plateID = specObj.plateID
    and class = 'qso'
    and specObj.zWarning = 0
    and specobj.z between 0.3 and 0.4
```

Outer joins

What if you are not rigorous about the data from the query?

Non inclusive query → use outer joins

Select P.objID, P.ra, P.dec, S.SpecObjId,S.ra, S.dec

From PhotoObj as P left outer join SpecObjAll as S on P.objID = s.BestObjID

All data from P is taken, missing data from s is filled with NULL

Calling functions from the DB

select top 100 z, dered_u, dbo.fCosmoAbsMag(dered_u, z, DEFAULT,DEFAULT,DEFAULT,DEFAULT,DEFAULT)

from SpecPhoto

where z > 0 and zWarning=0

Your SQL command was:

```
select top 100 z, dered_u, dbo.fCosmoAbsMag(dered_u, z, DEFAULT,DEFAULT,DEFAULT,DEFAULT,DEFAULT)
from SpecPhoto
where z > 0 and zwarning = 0
```

Your query output (max 500,000 rows):

z	dered_u	Column1
0.5379699	23.57166	-18.8945730913955
0.4058664	17.83873	-23.893199709389

FUNCTION **fCosmoAbsMag**

Returns the absolute magnitude of a galaxy at a particular redshift.

Parameters:

Go

@m float: aparent magnitude of the object

- @z float: redshift
- @OmegaM float: matter density. If set as DEFAULT, then 0.27891507
- @OmegaL float: dark energy density. If set as DEFAULT, then 0.721
- @OmegaR float: radiation density. If set as DEFAULT, then 8.493e-5
- @omega0 float: dark energy state equation. If set as DEFAULT, then -1
- @h_0 float: (hubble constant[Km/s/MPc])/(100[Km/s/MPc]). If set as DEFAULT, then 0.701

  Reference: http://lambda.gsfc.nasa.gov/product/map/dr3/pub_papers/fiveyear/cosmology/wmap_5yr_cosmo.pdf
- returns AbsMag: absolute magnitude

```
select dbo.fCosmoAbsMag(17.5,0.1,0.27891507,0.721,8.493e-5,-1,0.701);select dbo.fCosmoAbsMag(17.5,0.1,DEFAULT,DEFAULT,DEFAULT,DEFAULT,DEFAULT)
```

return -20.8155123821697 and -20.8155123821697

see also fCosmoDistanceModulus

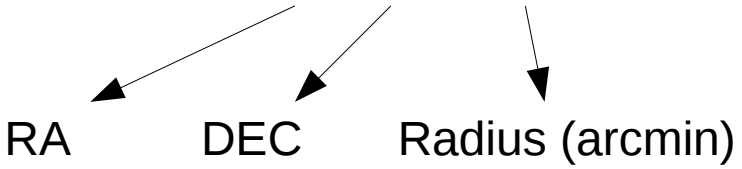
Input and output parameters

name	type	length	inout	pnum
@m	float	8	input	1
@z	float	8	input	2
@OmegaM	float	8	input	3
@OmegaL	float	8	input	4
@OmegaR	float	8	input	5
@omega0	float	8	input	6
@h_0	float	8	input	7
	float	8	output	0

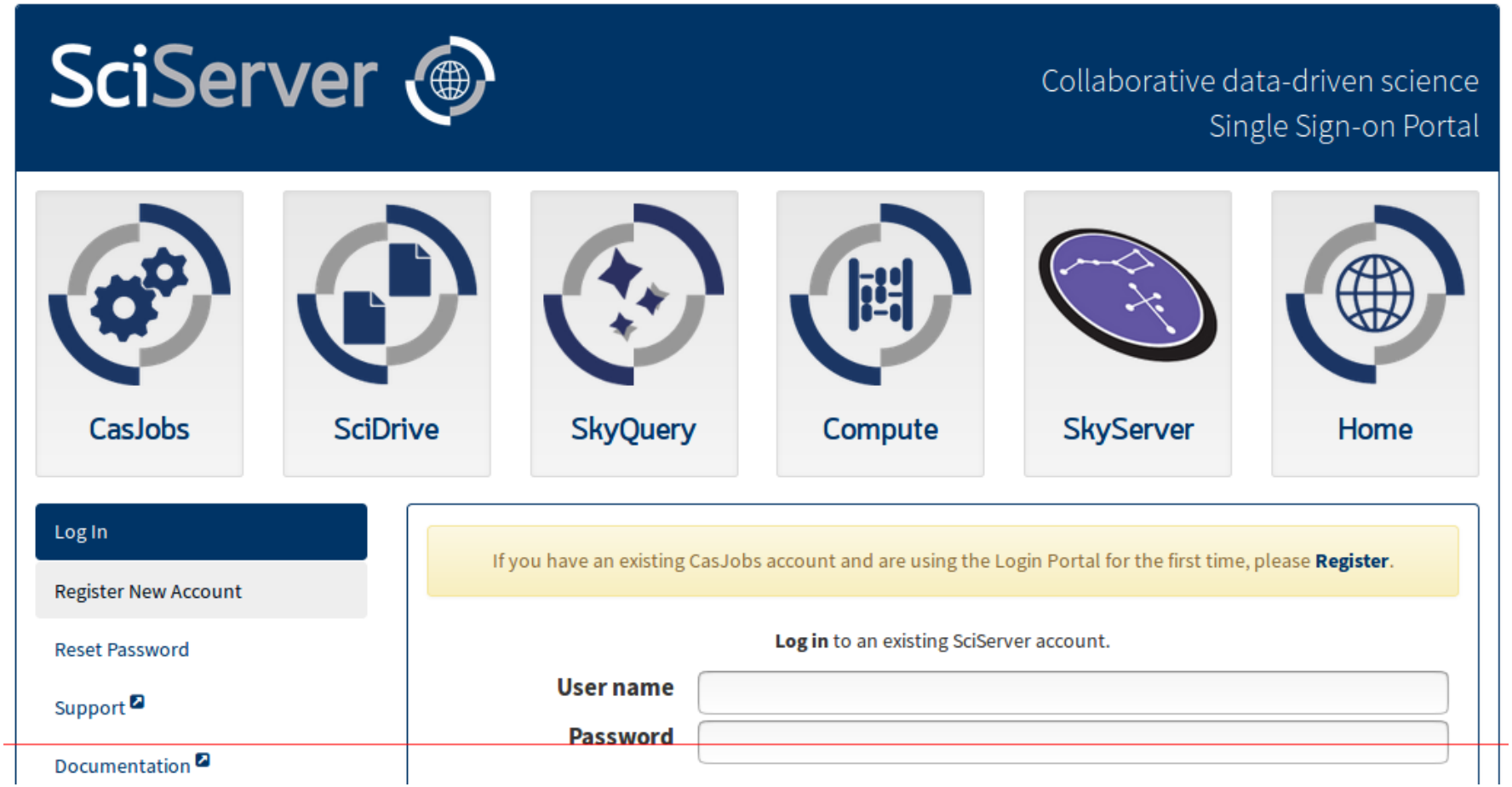
Lets get the objects from the database at a given distance from a point with coordinates RA and DEC

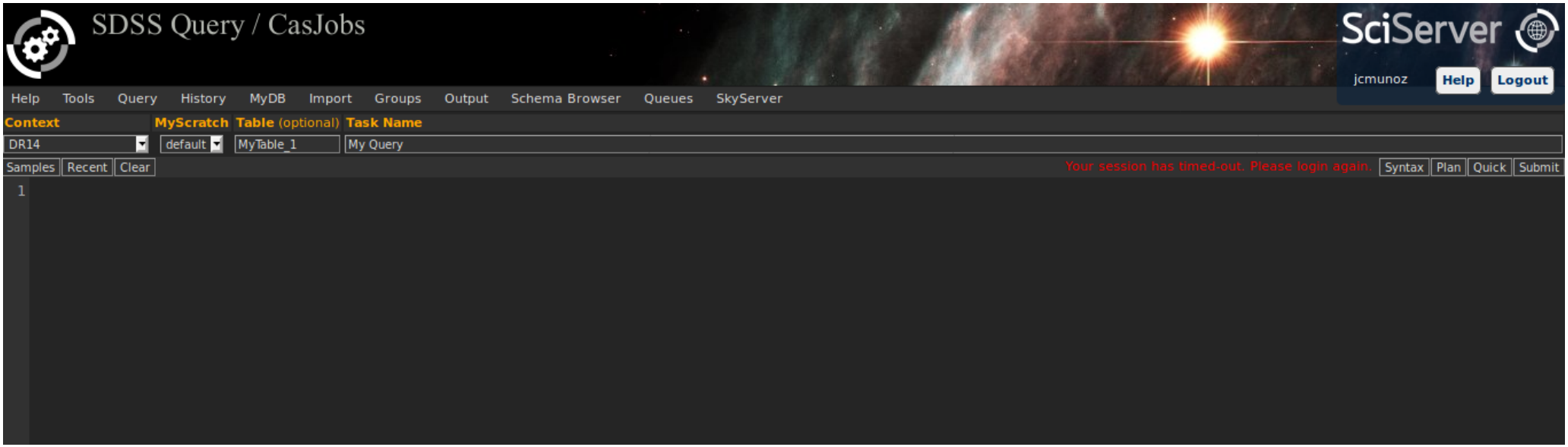
```
Select p.ObjID, p.ra, p.dec, p.u, p.g, p.r, p.i, p.z
From photoObj p, dbo.fGetNearbyObjEq(140,20,5) n
Where p.objID = n.objID  AND p.type = 3
```

dbo.fGetNearbyObjEq(140, 20, 5)



This function returns a table!
You can query it inside your query


SciServer
Collaborative data-driven science
Single Sign-on Portal
CasJobs
SciDrive
SkyQuery
Compute
SkyServer
Home
Log In
Register New Account
Reset Password
Support
Documentation
If you have an existing CasJobs account and are using the Login Portal for the first time, please Register.
Log in to an existing SciServer account.
User name
Password


SDSS Query / CasJobs
SciServer
jcmunoz
Help
Logout
Help Tools Query History MyDB Import Groups Output Schema Browser Queues SkyServer
Context
MyScratch
Table (optional)
Task Name
DR14
default
MyTable_1
My Query
Samples Recent Clear
Your session has timed-out. Please login again.
Syntax Plan Quick Submit
1

http://skyserver.sdss.org/dr9/en/help/howto/search/default.asp

SLOAN DIGITAL SKY SURVEY III

SkyServer DR9

Home | Data | Schema | Education | Astronomy | SDSS | Contact Us | Download | Site Search | Help

SQL Tutorial

Back to Help

1. Introduction

2. A Simple Query

Practice

3. Common

SQL Tutorial

NOTE: This is a step-by-step tutorial for those new to SQL. If you already know some SQL and want a more in-depth introduction to how SQL applies to SkyServer, see **Using SQL with SkyServer**.

Another great way to learn SQL is to modify and submit queries that other people have already written. To jump right in to modifying queries written by scientists, see the **Sample SQL Queries.**

Searching for Data: A Tutorial

Did you know that you can search through SkyServer's database for only the objects you are interested in? For example, you might want to find all the bright blue galaxies for which we have obtained spectra. Looking through all 14 million objects with the Navigation tool could take years. But using **the right search**, you could find all the bright blue galaxies with spectra in seconds!

Virgo - Millennium Database

Documentation

CREDITS/Acknowledgments

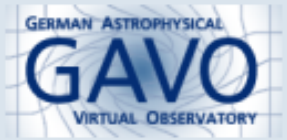
Registration

News

FAQ

Databases

millimil (context)



Streaming queries return unlimited number of rows in CSV format and are cancelled after 30 seconds.
Browser queries return maximum of 1000 rows in HTML format and are cancelled after 30 seconds.

There is a partial mirror of this database in Durham at http://galaxy-catalogue.dur.ac.uk:8080/Millennium/ .
The Durham database does not contain all the latest L-Galaxies models but does contain more recent GALFORM models.

Query (stream)

Query (browser)

Help

Maximum number of rows to return to the query form: 10

Demo queries: click a button and the query will show in the query window.
Holding the mouse over the button will give a short explanation of the goal of the query. These queries are described in some more d

```
SELECT TOP 1000
g.run, f.field, p.objID
FROM photoObj p, fieldf, segment g
WHERE f.fieldid= p.fieldid
 and  f.segmentid = g.segmentid
and  f.psfWidth_r > 1.2
and  p.colc > 400.0
```

When using table valued functions, you must do the joinexplicitly (rather than using "="). To do this, we use the syntax

```
SELECT quantities

    FROM table1

    JOIN table2 on table1.quantity =table2.quantity

    WHERE constraints
```