

Data Mining In Modern Astronomy Sky Surveys:  
*Databases*  
*& Sloan Digital Sky Survey*

Ching-Wa Yip

[cwyip@pha.jhu.edu](mailto:cwyip@pha.jhu.edu); Bloomberg 518

## Estimation of Hubble's Constant

```
> # Read Hubble data file
> data <- read.csv('H:/public_html/teaching/hubbletable1.csv')
> # Show data
> data
```

	ObjectName	Distance_Mpc	RecessionVelocity_kms
1	SmallMag	0.032	170
2	LargeMag	0.034	290
3	NGC6822	0.214	-130
4	NGC598	0.263	-70
5	NGC221	0.275	-185
6	NGC224	0.275	-220
7	NGC5457	0.450	200
8	NGC4736	0.500	290
9	NGC5194	0.500	270
10	NGC4449	0.630	200
11	NGC4214	0.800	300
12	NGC3031	0.900	-30
13	NGC3627	0.900	650
14	NGC4826	0.900	150
15	NGC5236	0.900	500
16	NGC1068	1.000	920
17	NGC5055	1.100	450
18	NGC7331	1.100	500
19	NGC4258	1.400	500
20	NGC4151	1.700	960
21	NGC4382	2.000	500
22	NGC4472	2.000	850
23	NGC4486	2.000	800
24	NGC4649	2.000	1090

```
16   NGC1068      1.000      920
17   NGC5055      1.100      450
18   NGC7331      1.100      500
19   NGC4258      1.400      500
20   NGC4151      1.700      960
21   NGC4382      2.000      500
22   NGC4472      2.000      850
23   NGC4486      2.000      800
24   NGC4649      2.000     1090
```

```
> # Plot Recession Velocity vs. Distance of galaxies
> plot(data$Distance_Mpc, data$RecessionVelocity_kms)
> # Fit linear model
> fit1 <- lm(data$RecessionVelocity_kms ~ data$Distance_Mpc)
> # Add best-fit straight line
> abline(fit1, col = 'red')
> # Show best-fit parameters
> fit1
```

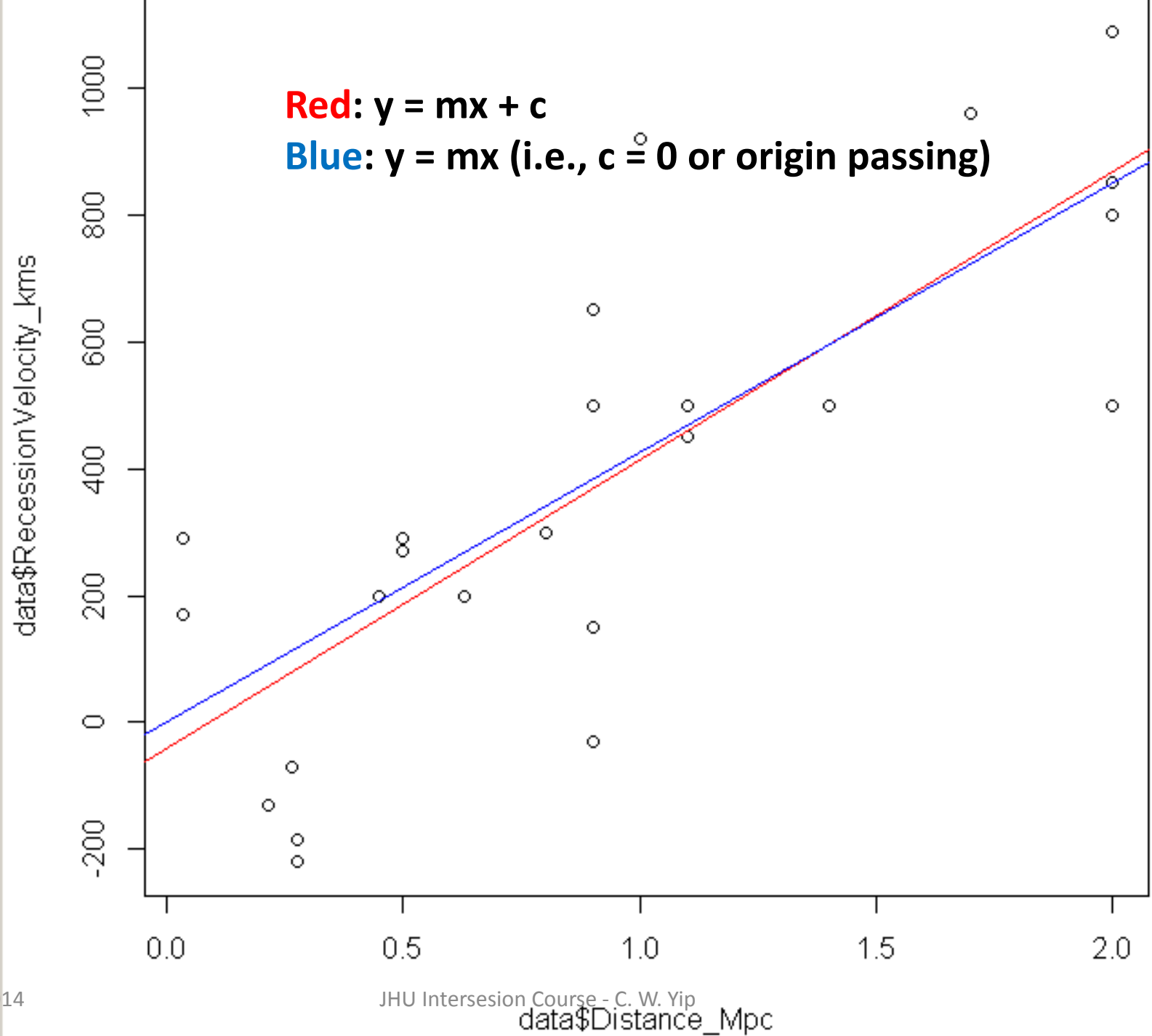
Call:

```
lm(formula = data$RecessionVelocity_kms ~ data$Distance_Mpc)
```

Coefficients:

```
      (Intercept)  data$Distance_Mpc
            -40.78             454.16
```

```
> # Fit origin-passing linear model
> fit2 <- lm(data$RecessionVelocity_kms ~ data$Distance_Mpc + 0)
> # Add best-fit straight line
> abline(fit2, col = 'blue')
> # Show best-fit parameters
> fit2
```



## Discussion HW2

- The calculated values (454 km/s/Mpc) is a factor of a few larger than the WMAP value (71 km/s/Mpc).
- This discrepancy suggests that there could be systematic error in Hubble's measurements of Recession Velocity or/and Distance. The error could be due to the measurement techniques and/or the local galaxy sample.
- Photon count = 100 implies  $\text{SNR} = \sqrt{100} = 10$ .
- By using the simplified Rose Criterion, the minimum number of photons for 100% feature detection is  $5^2 = 25$ .

## Further Readings on Data Mining and Machine Learning

- **Statistical Data Analysis** (Cowan)
  - Practical reference/textbook
- **A Modern Introduction to Probability and Statistics** (Dekking, Kraaikamp, Lopuhaä, Meester)
  - Self-content textbook
  - Freely downloadable online
- **All of Statistics** (Wasserman)
  - Comprehensive; Advanced read
- **Neural Networks for Pattern Recognition** (Bishop)
  - Focus on concepts
  - Freely downloadable online

# From Data to Information

- We don't just want data.
- We want information from the data.

Information



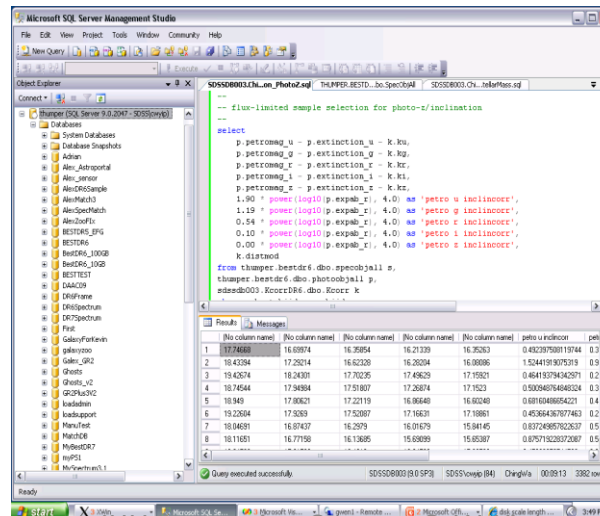
Database



Sensors



Data Analysis  
or  
Data Mining



# Topics

- Database
- Table
- Structured Query Language (SQL)
- Sloan Digital Sky Survey (SDSS) and Web Services
- Example SQL queries in Astronomy:
  - Create binned histograms of galaxies
  - Select targets for follow-up spectroscopy
  - Find extreme galaxies (i.e., outliers)



## Basics of Database

- A database stores a collection of data.
- The data are arranged in database objects such as **tables**.
- Relational Database: a database which uses table(s).
  - The “relation” refers to the relation among different fields within **one table**.
  - The “relation” does not refer to the potential relation among multiple tables.

# Basics of Tables

- Row is called **Record**.
- Column is called **Field**.
- Schema: logical container for database objects that user creates.
- Records are stored in the tables with some order:
  - The records are not necessarily sorted by a particular column.

## Table contains Unique Records: Primary Key

- We want to be able to retrieve each and every record.
- Solution: Each record in a table is unique.
- This unique ID is called **Primary Key**.
- In the SDSS, some Primary Keys are:
  - ObjID (in table PhotoObjAll)
  - SpecObjID (in table SpecObjAll)

First Name	Last Name	Credit Card #
George	Daniels	184715170968
Amy	Lee	207609796702
Brandon	Willis	982767757110
Jennifer	Connolly	486830981903
Andrew	Folks	601571389801

## Table contains Unique Records: Primary Key

- We want to be able to retrieve each and every record.
- Solution: Each record in a table is unique.
- This unique ID is called **Primary Key**.
- In the SDSS, some Primary Keys are:
  - ObjID (in table PhotoObjAll)
  - SpecObjID (in table SpecObjAll)

Primary Key

ID	First Name	Last Name	Credit Card #
1	George	Daniels	184715170968
2	Amy	Lee	207609796702
3	Brandon	Willis	982767757110
4	Jennifer	Connolly	486830981903
5	Andrew	Folks	601571389801

## Foreign Key

- A Foreign Key is a field of a table (*child table*) that uniquely identifies a row in another table (*parent table*).
- A Foreign Key hence ties two tables together.
- In the “Customer and Purchase” tables, CreditCard # is the Foreign Key.

# Un-Normalized Table

- In un-normalized table:
  - Records may grow very quickly.
  - Redundant records may present.
- Solution: Split data into **multiple tables**.
- In Astronomy: Data are fixed once the survey is completed. But tables are long, normalization improves performance.
- In Industry (banking/searching/facebook etc.): Data are growing fast, giving many records for a given user. Normalization is important.

## Un-Normalized Table: Purchase

First Name	Last Name	Credit Card #	Date	Amount
George	Daniels	184715170968	01/05/2013	125.6
Amy	Lee	207609796702	01/07/2013	45.50
George	Daniels	184715170968	01/07/2013	72.35
Brandon	Willis	982767757110	01/09/2013	38.97
Jennifer	Connolly	486830981903	01/08/2013	49.83
George	Daniels	184715170968	01/10/2013	72.35
Andrew	Folks	601571389801	01/12/2013	92.30

- There are redundant data in this table.

## Split Data into 2 Tables: Customer and Purchase

First Name	Last Name	Credit Card #
George	Daniels	184715170968
Amy	Lee	207609796702
Brandon	Willis	982767757110
Jennifer	Connolly	486830981903
Andrew	Folks	601571389801

- No redundant data.
- Two tables grow at different rate!

Credit Card #	Date	Amount
184715170968	01/05/2013	125.6
207609796702	01/07/2013	45.50
184715170968	01/07/2013	72.35
982767757110	01/09/2013	38.97
486830981903	01/08/2013	49.83
184715170968	01/10/2013	72.35
601571389801	01/12/2013	92.30



## Split Data into 2 Tables: Customer and Purchase

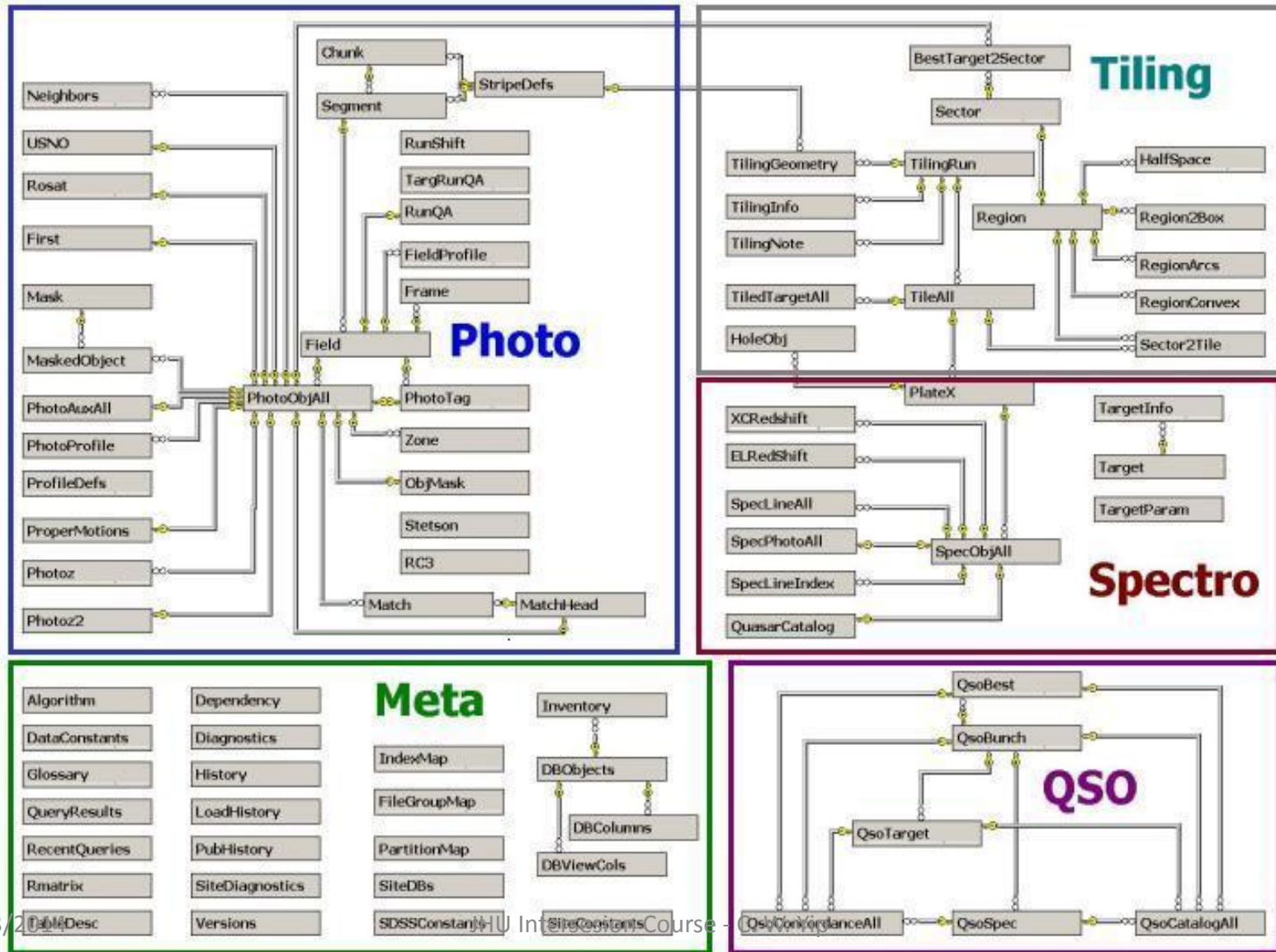
First Name	Last Name	Credit Card #
George	Daniels	184715170968
Amy	Lee	207609796702
Brandon	Willis	982767757110
Jennifer	Connolly	486830981903
Andrew	Felton	601571389801

- No redundant data.
- Two tables grow at different rate!

A single spreadsheet is not the best approach for storing big data!

Credit Card #	Date	Amount
184715170968	01/05/2013	125.6
207609796702	01/07/2013	45.50
184715170968	01/07/2013	72.35
982767757110	01/09/2013	38.97
486830981903	01/08/2013	49.83
184715170968	01/10/2013	72.35
601571389801	01/12/2013	92.30

# 85 Tables in SDSS DR7



# Sloan Digital Sky Survey (2000-)



## • Photometric + Spectroscopic Surveys

- 11,000 square degree footprint (DR7)
- $5.9 \times 10^8$   $u, g, r, i, z$  photometry
- $1.6 \times 10^6$  fiber spectra

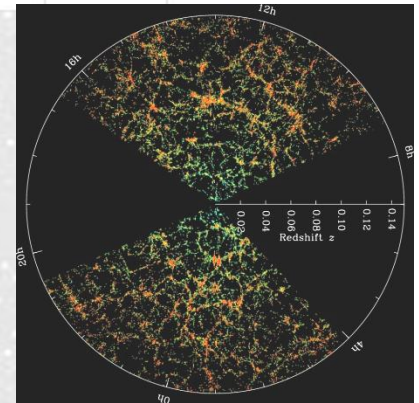
## • Phases

- SDSS I (2000-05)
- SDSS II (2005-08)
- SDSS III (2008-14)
- SDSS 4 (Current)

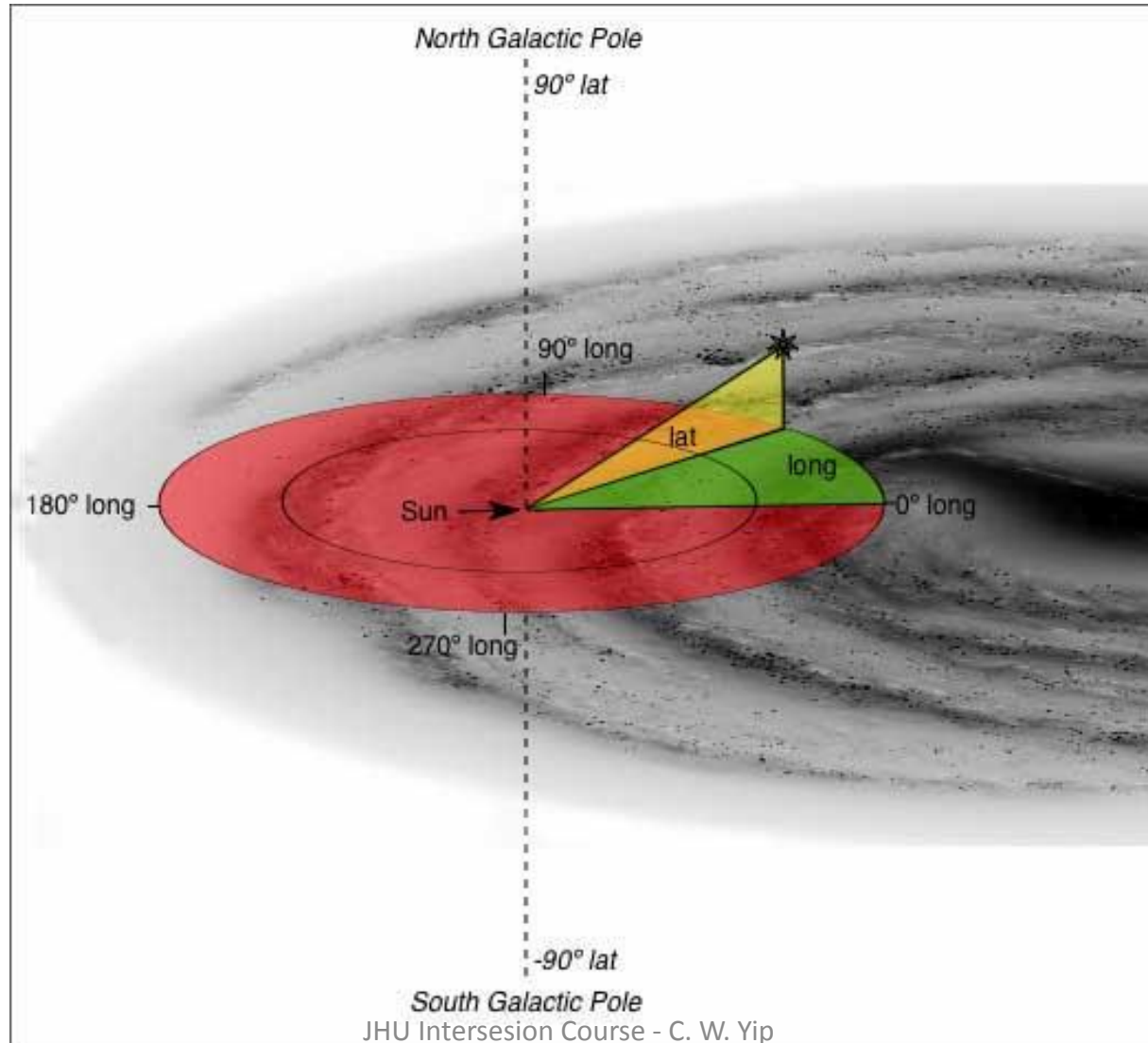
## • Data are public

- Web interfaces for data download & exploration
  - SkyServer, DAS, etc.

(Galaxy Distribution)

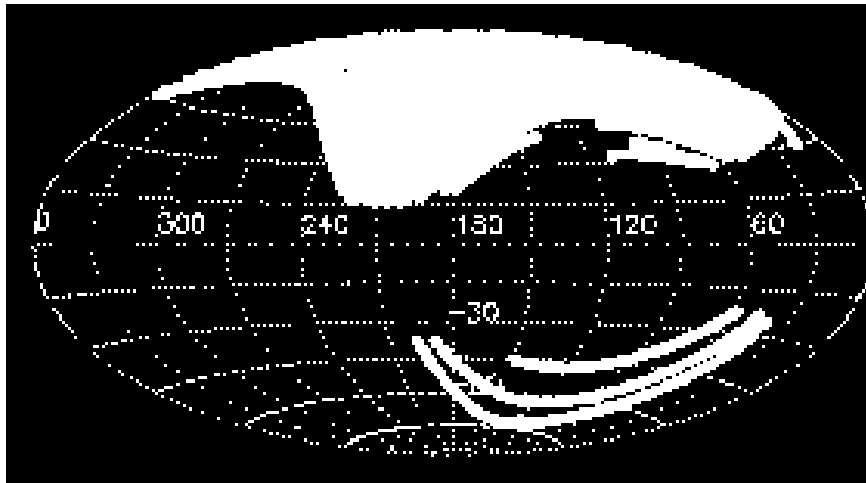


# SDSS Footprints (DR7): in Galactic Coordinate Systems

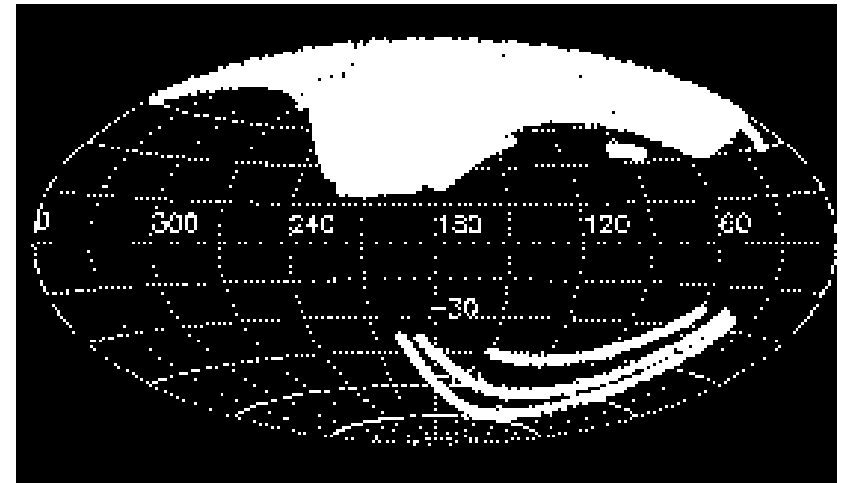


# SDSS Footprints (DR7): in Galactic Coordinate Systems

## Photometry



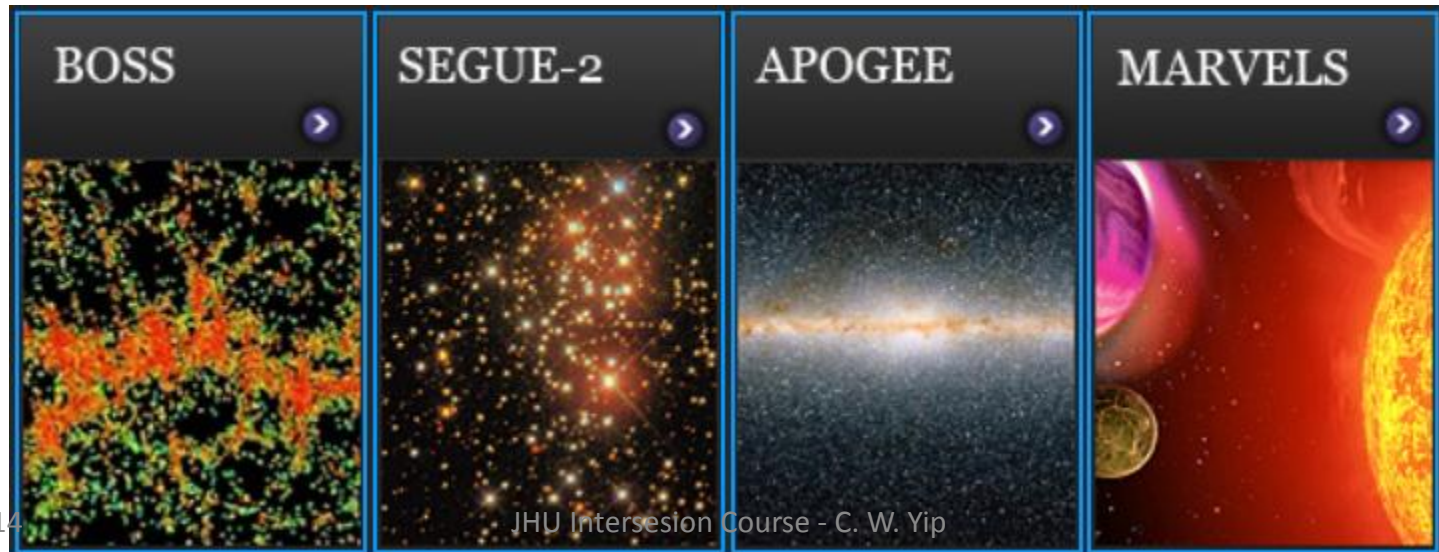
## Spectroscopy



Southern Stripes: Offer repeated scans (time-domain information) of the sky!

# SDSS III

- BOSS
  - Map distribution of galaxies out to redshift of 0.7, which has imprints information about the cosmic microwave background.
- SEGUE-2
  - Map Milky Way structure by measuring optical spectra of 119,000 stars.
- APOGEE
  - Map dust-obscured disk and bulge of Milky Way by measuring Infrared spectra of stars.
- MARVELS
  - Search for exoplanets by monitoring radial velocities of 11,000 stars.



## Statistics of SDSS Databases (Data Release 7, or DR7)

- Number of tables: 85
- Data Volume:
  - Images (16 TB)
  - Tables (18 TB)
  - Data Products (27 TB)
- PhotoObjAll
  - Number of rows: 585,634,220
  - Number of columns: 454
- SpecObjAll
  - Number of rows: 1,640,960
  - Number of columns: 63

## Web Services for SDSS Data

- SkyServer and CasJobs
  - Nolan Li, Alex Szalay, Ani Thakar, Tamas Budavari et al.
- Spectrum Services
  - Dobos et al.
- Open SkyQuery
  - Dobos et al. 2014 in prep.



# SkyServer.org - Team

The team behind the skyserver are multitalented and have various backgrounds. You have seen the names - here are the faces.

Tamas Budavari



William O'Mullane



George Fekete



Adrian Pope



Sam Carliles



Jordan Raddick



Nolan Li



Alex Szalay



Maria Nieto-Santisteban



Ani Thakar



Tanu Malik



Jan van den Berg





# Sloan Digital Sky Survey / SkyServer



- Home
- Tools
- Schema
- Projects
- Astronomy
- SDSS
- Contact Us
- Download
- Site Search
- Help

## Welcome to the DR7 site!!!

This website presents data from the Sloan Digital Sky Survey, a project to make a map of a large part of the universe. We would like to show you the beauty of the universe, and share with you our excitement as we build the largest map in the history of the world.

## News

The site hosts data from **Data Release 7 (DR7)**. **What's new in DR7, what's new on this site, and known problems.** [More...](#)

## For Astronomers

A separate branch of this website for professional astronomers. [\(English\)](#) [More...](#)

SDSS is supported by



Powered by **Microsoft**

## SkyServer Tools

- Famous places
- Get images
- Visual Tools
- Explore
- Search
- Object Cross-ID
- CasJobs

## Science Projects

- Basic
- Advanced
- Challenges
- For Kids
- Games and Contests
- Teachers
- Links to other projects

## Info Links

- About Astronomy
- About the SDSS
- About the SkyServer
- SDSS Data Release 7
- SDSS Project Website
- Open SkyQuery
- Images of RC3 Galaxies

## Help

- Getting Started
- FAQ
- How To
- Glossary
- Schema Browser
- Sample SQL Queries
- Details of SDSS Data

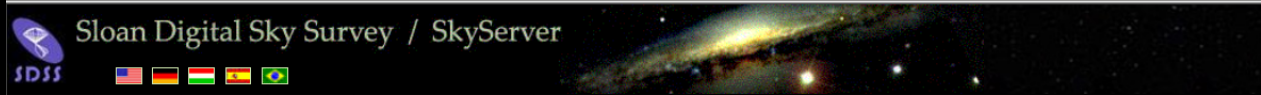


The contents for boundaries of the different regions

Region	Size	Depth	Start	End	Description
SDSS	1	1	2000-01-01	2000-12-31	SDSS DR7
SDSS	1	1	2001-01-01	2001-12-31	SDSS DR7
SDSS	1	1	2002-01-01	2002-12-31	SDSS DR7
SDSS	1	1	2003-01-01	2003-12-31	SDSS DR7
SDSS	1	1	2004-01-01	2004-12-31	SDSS DR7
SDSS	1	1	2005-01-01	2005-12-31	SDSS DR7
SDSS	1	1	2006-01-01	2006-12-31	SDSS DR7
SDSS	1	1	2007-01-01	2007-12-31	SDSS DR7
SDSS	1	1	2008-01-01	2008-12-31	SDSS DR7
SDSS	1	1	2009-01-01	2009-12-31	SDSS DR7
SDSS	1	1	2010-01-01	2010-12-31	SDSS DR7
SDSS	1	1	2011-01-01	2011-12-31	SDSS DR7
SDSS	1	1	2012-01-01	2012-12-31	SDSS DR7
SDSS	1	1	2013-01-01	2013-12-31	SDSS DR7
SDSS	1	1	2014-01-01	2014-12-31	SDSS DR7

Contact Us

Site Traffic  
Privacy Policy



## Schema Browser

Glossary  
 Algorithms

Search for

- Tables
- Views
- Functions
- Procedures
- Constants
- Indices

### TABLE PhotoObjAll

Contains a record describing the attributes of each photometric object

The table has views:

- **PhotoObj**: all primary and secondary objects; essentially this is the view you should use unless you want a specific type of object.
- **PhotoPrimary**: all photo objects that are primary (the best version of the object).
  - **Star**: Primary objects that are classified as stars.
  - **Galaxy**: Primary objects that are classified as galaxies.
  - **Sky**: Primary objects which are sky samples.
  - **Unknown**: Primary objects which are no one of the above
- **PhotoSecondary**: all photo objects that are secondary (secondary detections)
- **PhotoFamily**: all photo objects which are neither primary nor secondary (blended)

The table has indices that cover the popular columns.

name	type	length	unit	ucd	description
objID	bigint	8		ID_MAIN	Unique SDSS identifier composed from [skyVersion, rerun, run, camcol, field, obj].
skyVersion	tinyint	1		CODE_MISC	0 = OPDB target, 1 = OPDB best
run	smallint	2		OBS_RUN	Run number
rerun	smallint	2		CODE_MISC	Rerun number
camcol	tinyint	1		INST_ID	Camera column
field	smallint	2		ID_FIELD	Field number
obj	smallint	2		ID_NUMBER	The object id within a field. Usually changes between reruns of the same field.
mode	tinyint	1		CLASS_OBJECT	1: primary, 2: secondary, 3: family object, 4: outside chunk boundary.
nChild	smallint	2		NUMBER	Number of children if this is a composite object that has been deblended. BRIGHT (in a flags sense) objects also have nchild == 1, the non-BRIGHT sibling.
type	smallint	2		CLASS_OBJECT	Morphological type classification of the object.
clean	int	4		CODE_MISC	Clean photometry flag for point sources (1=clean, 0=unclean).
probPSF	real	4		STAT_PROBABILITY	Probability that the object is a star. Currently 0 if type == 3 (galaxy), 1 if type == 6 (star).
insideMask	tinyint	1		CODE_MISC	Flag to indicate whether object is inside a mask and why
flags	bigint	8		CODE_MISC	Photo Object Attribute Flags

# Using Microsoft SQL Server in Astronomy (Szalay & Gray)

Other choices:

- Oracle
- MySQL

The screenshot shows Microsoft SQL Server Management Studio with a query window open. The query is a SELECT statement that filters astronomical objects based on extinction and inclination. The results window displays a table with 8 rows of data.

```
--  
-- flux-limited sample selection for photo-z/inclination  
--  
select  
  p.petromag_u - p.extinction_u - k.ku,  
  p.petromag_g - p.extinction_g - k.kg,  
  p.petromag_r - p.extinction_r - k.kr,  
  p.petromag_i - p.extinction_i - k.ki,  
  p.petromag_z - p.extinction_z - k.kz,  
  1.90 * power(log10(p.expab_r), 4.0) as 'petro u inclincorr',  
  1.19 * power(log10(p.expab_r), 4.0) as 'petro g inclincorr',  
  0.54 * power(log10(p.expab_r), 4.0) as 'petro r inclincorr',  
  0.10 * power(log10(p.expab_r), 4.0) as 'petro i inclincorr',  
  0.00 * power(log10(p.expab_r), 4.0) as 'petro z inclincorr',  
  k.distmod  
from thumper.bestdr6.dbo.specobjall s,  
  thumper.bestdr6.dbo.photoobjall p,  
  sdsbdb003.KcorrDR6.dbo.Kcorr k
```

	(No column name)	(No column name)	(No column name)	(No column name)	(No column name)	petro u inclincorr	peti
1	17.74668	16.69974	16.35854	16.21339	16.35263	0.492397508119744	0.3
2	18.43394	17.29214	16.62328	16.28204	16.08086	1.52441919075319	0.9
3	19.42674	18.24301	17.70235	17.49629	17.15921	0.464193794342971	0.2
4	18.74544	17.94984	17.51807	17.26874	17.1523	0.500948764848324	0.3
5	18.949	17.80621	17.22119	16.86648	16.60248	0.68160486654221	0.4
6	19.22604	17.9269	17.52087	17.16631	17.18861	0.453664367877463	0.2
7	18.04691	16.87437	16.2979	16.01679	15.84145	0.837249857822637	0.5
8	18.11651	16.77158	16.13685	15.69099	15.65387	0.875719228372087	0.5

Query executed successfully. SDSSDB003 (9.0 SP3) SDSS\cwyp (84) ChingWa 00:09:13 3382 rows

# Practical SQL

- We execute **queries** to manage and retrieve the data.
- The queries are written in Structured Query Language (SQL), which has the form:

```
SELECT column(s)  
FROM table(s)  
WHERE predicate(s) are true
```

- SQL queries can get long and complicated.

# A Simplest Query: Count Rows

New Query | [Icons] | master | Execute [Execute Icon]

Object Explorer

Connect [Icons]

SQLQuery\_Data...SS\cwyip (68)

```
SELECT COUNT (*)
FROM THUMPER.BESTDR7.dbo.PhotoObjAll
```

- [-] BESTDR6
- [-] BESTDR7
  - [-] Database Diagrams
  - [-] Tables
    - [-] System Tables
    - casuser.wflags\_cmp1329pm05DR7
    - dbo.Algorithm
    - dbo.Ap7Mag
    - dbo.BestTarget2Sector
    - dbo.Chunk
    - dbo.DataConstants
    - dbo.DBColumns
    - dbo.DBObjects
    - dbo.DBViewCols
    - dbo.Dependency
    - dbo.Diagnostics
    - dbo.DR3QuasarCatalog
    - dbo.DR5QuasarCatalog
    - dbo.ELRedShift
    - dbo.Field
    - dbo.FieldProfile
    - dbo.FieldQA
    - dbo.FileGroupMap
    - dbo.First
    - dbo.Frame
    - dbo.Glossary
    - dbo.HalfSpace
    - dbo.History
    - dbo.HoleObj
    - dbo.IndexMap
    - dbo.Inventory
    - dbo.LoadHistory
    - dbo.Mask
    - dbo.MaskedObject
    - dbo.Match
    - dbo.MatchHead
    - dbo.Neighbors
    - dbo.ObjMask
    - dbo.OrigField
    - dbo.OrigPhotoObjAll
    - dbo.PartitionMap
    - dbo.PhotoObjAll
    - dbo.PhotoProfile

Results | Messages

	(No column name)
1	585634220

1/23/2014

# Show Top Records

- dbo.PhotoObjAll
  - Columns
    - objID (PK, bigint, not null)
    - skyVersion (tinyint, not null)
    - run (smallint, not null)
    - rerun (smallint, not null)
    - camcol (tinyint, not null)
    - field (smallint, not null)
    - obj (smallint, not null)
    - mode (tinyint, not null)
    - nChild (smallint, not null)
    - type (smallint, not null)
    - clean (int, not null)
    - probPSF (real, not null)
    - insideMask (tinyint, not null)
    - flags (bigint, not null)
    - rowc (real, not null)
    - rowcErr (real, not null)
    - colc (real, not null)
    - colcErr (real, not null)
    - rowv (real, not null)
    - rowvErr (real, not null)
    - colv (real, not null)
    - colvErr (real, not null)
    - rowc\_u (real, not null)
    - rowc\_g (real, not null)
    - rowc\_r (real, not null)
    - rowc\_j (real, not null)
    - rowc\_z (real, not null)
    - rowcErr\_u (real, not null)
    - rowcErr\_g (real, not null)
    - rowcErr\_r (real, not null)
    - rowcErr\_j (real, not null)
    - rowcErr\_z (real, not null)
    - colc\_u (real, not null)
    - colc\_g (real, not null)
    - colc\_r (real, not null)
    - colc\_j (real, not null)
    - colc\_z (real, not null)
    - colcErr\_u (real, not null)
    - colcErr\_g (real, not null)
    - colcErr\_r (real, not null)
    - colcErr\_j (real, not null)
    - colcErr\_z (real, not null)
    - sky\_u (real, not null)
    - sky\_g (real, not null)
    - sky\_r (real, not null)
    - sky\_j (real, not null)
    - sky\_z (real, not null)
    - skyErr\_u (real, not null)
    - skyErr\_g (real, not null)
    - skyErr\_r (real, not null)
    - skyErr\_j (real, not null)
    - skyErr\_z (real, not null)

SQLQuery\_Data... (cwyp (68))\*

```
SELECT TOP 10 *
FROM THUMPER.BESTDR7.dbo.PhotoObjAll
```

Results Messages

	objID	skyVersion	run	rerun	camcol	field	obj	mode	nChild	type	clean	probPSF	insideMask	flags	rowc	rowcErr	colc	colcErr
1	587722951693303809	1	745	40	1	518	1	3	1	6	0	1	0	34628571982	67.31985	0.008972449	49.60184	0
2	587722951693303810	1	745	40	1	518	2	2	0	6	0	1	0	310109256160076	67.25411	0.002876111	49.69498	0
3	587722951693303811	1	745	40	1	518	3	3	0	0	0	-9999	0	281619127075148	-6.699471	0.1010417	1098.149	0
4	587722951693303812	1	745	40	1	518	4	3	1	6	0	1	0	268435970	37.28694	0.01529532	1793.392	0
5	587722951693303813	1	745	40	1	518	5	3	0	6	0	1	0	268435968	37.10624	0.007697065	1793.54	0
6	587722951693303814	1	745	40	1	518	6	3	1	6	0	1	0	17592454611530	43.78992	0.01509366	1875.807	0
7	587722951693303815	1	745	40	1	518	7	3	0	6	0	1	0	17599434461772	43.60673	0.005338188	1875.988	0
8	587722951693303816	1	745	40	1	518	8	3	1	6	0	1	0	448635372704330	60.17922	0.007902309	496.6865	0
9	587722951693303817	1	745	40	1	518	9	3	11	6	0	1	0	1046988272308748	48.23613	0.1025441	567.655	0
10	587722951693303818	1	745	40	1	518	10	3	0	6	0	1	0	483787533062164	48.23613	0.1025441	567.655	0

# Predicates (or Conditions)

- These inequalities can be used in predicates:

=

>

<

>=

<=

<>

(“not equal”)

```
SELECT COUNT(*)  
FROM PhotoObjAll  
WHERE PetroMag_r < 17.7
```

It means:

Count the number of rows in PhotoObjAll where the r-band measured magnitude is brighter than 17.7.



## ORDER BY

- We use “ORDER BY” to sort the output into increasing order.

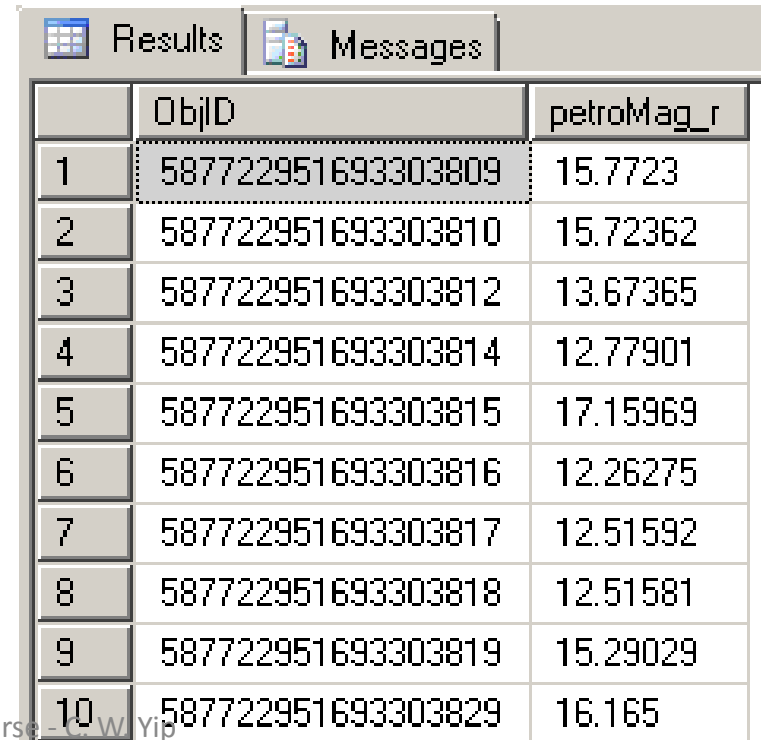
```
SELECT Top 10 ObjID, petroMag_r  
FROM PhotoObjAll  
WHERE petroMag_r < 17.7  
ORDER BY ObjID
```

	ObjID	petroMag_r
1	587722951693303809	15.7723
2	587722951693303810	15.72362
3	587722951693303811	-9999
4	587722951693303812	13.67365
5	587722951693303814	12.77901
6	587722951693303815	17.15969
7	587722951693303816	12.26275
8	587722951693303817	12.51592
9	587722951693303818	12.51581
10	587722951693303819	15.29029

## Nullable?

- A field that is allowed to have no values is called “nullable”.
- Determined when creating the database.
- In SDSS, many unavailable fields have values “-9999”.

```
SELECT Top 10 ObjID, petroMag_r
FROM PhotoObjAll
WHERE petroMag_r < 17.7
AND petroMag_r <> -9999
ORDER BY ObjID
```



The screenshot shows a database query results window with two tabs: "Results" and "Messages". The "Results" tab is active, displaying a table with 10 rows. The table has two columns: "ObjID" and "petroMag\_r". The data is as follows:

	ObjID	petroMag_r
1	587722951693303809	15.7723
2	587722951693303810	15.72362
3	587722951693303812	13.67365
4	587722951693303814	12.77901
5	587722951693303815	17.15969
6	587722951693303816	12.26275
7	587722951693303817	12.51592
8	587722951693303818	12.51581
9	587722951693303819	15.29029
10	587722951693303829	16.165

# Aggregate (“Bag”) Functions: Group a field from multiple rows together

- Commonly used aggregate functions include:

COUNT()

MIN()

MAX()

AVG()

STDEV()

- For big tables, aggregate functions may take a long time to finish.

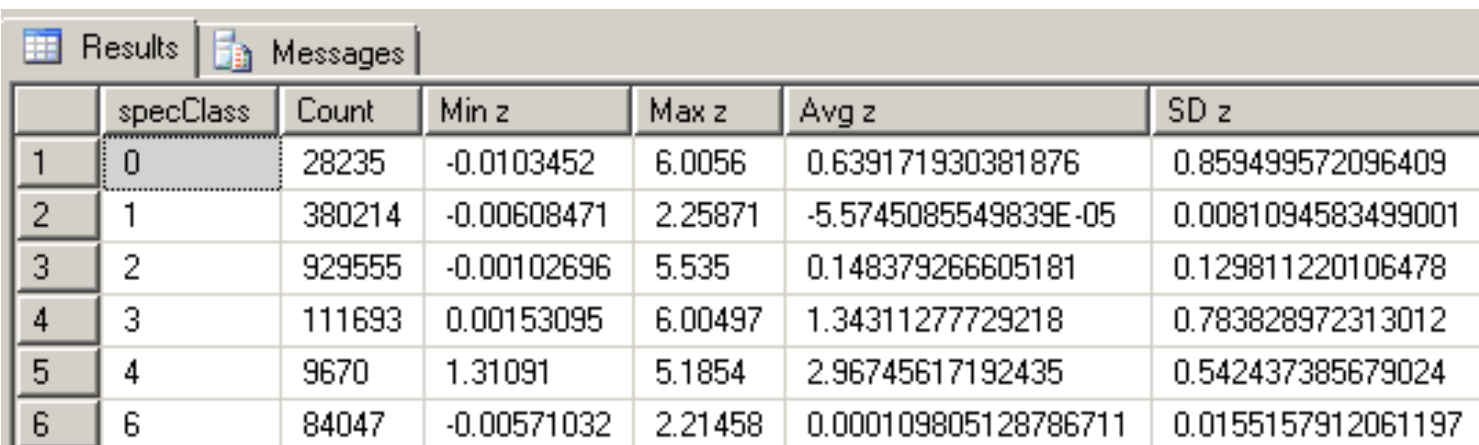
```
SELECT COUNT(*) as 'Count', MIN(z) as 'Min z', MAX(z) as 'Max z', AVG(z) as 'Avg z', STDEV(z) as 'SD z'  
FROM SpecObjAll  
WHERE specClass = 2
```

	Count	Min z	Max z	Avg z	SD z
1	929555	-0.00102696	5.535	0.148379266605181	0.129811220106478

# GROUP BY

- We use “GROUP BY” to group output by column(s).
- Often used together with aggregate functions.

```
SELECT specClass, COUNT(*) as 'Count', MIN(z) as 'Min z', MAX(z) as 'Max z', AVG(z) as 'Avg z',  
       STDEV(z) as 'SD z'  
FROM SpecObjAll  
WHERE z <> -9999  
GROUP BY specClass
```



	specClass	Count	Min z	Max z	Avg z	SD z
1	0	28235	-0.0103452	6.0056	0.639171930381876	0.859499572096409
2	1	380214	-0.00608471	2.25871	-5.5745085549839E-05	0.0081094583499001
3	2	929555	-0.00102696	5.535	0.148379266605181	0.129811220106478
4	3	111693	0.00153095	6.00497	1.34311277729218	0.783828972313012
5	4	9670	1.31091	5.1854	2.96745617192435	0.542437385679024
6	6	84047	-0.00571032	2.21458	0.000109805128786711	0.0155157912061197

## SpecClass Data values

name	value	description
UNKNOWN	0	Spectrum not classifiable ( $z_{\text{Conf}} < 0.25$ ).
STAR	1	Spectrum of a star.
GALAXY	2	Spectrum of a galaxy.
QSO	3	Spectrum of a quasi-stellar object.
HIZ_QSO	4	Spectrum of a high-redshift quasar ( $z > 2.3$ ), whose redshift is confirmed by a Ly-alpha estimator (see "Spectroscopic Redshift and Type Determination" section in Algorithms).
SKY	5	Spectrum of blank sky.
STAR_LATE	6	Star dominated bt molecular bands M or later.
GAL_EM	7	Emission line galaxy (placeholder).

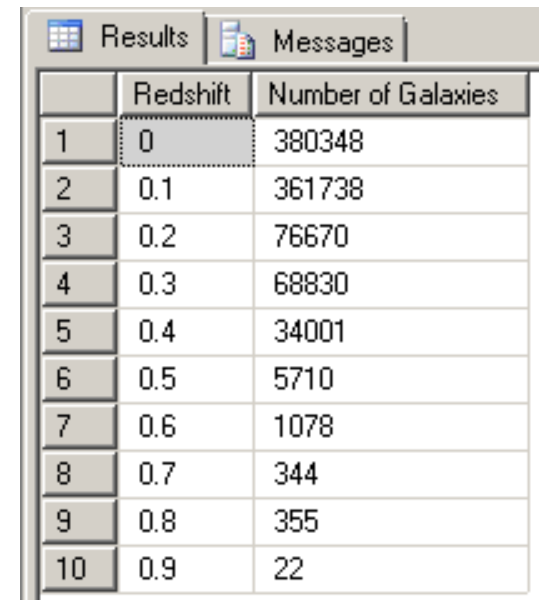
Results		Messages				
	specClass	Count	Min z	Max z	Avg z	SD z
1	0	28235	-0.0103452	6.0056	0.639171930381876	0.859499572096409
2	1	380214	-0.00608471	2.25871	-5.5745085549839E-05	0.0081094583499001
3	2	929555	-0.00102696	5.535	0.148379266605181	0.129811220106478
4	3	111693	0.00153095	6.00497	1.34311277729218	0.783828972313012
5	4	9670	1.31091	5.1854	2.96745617192435	0.542437385679024
6	6	84047	-0.00571032	2.21458	0.000109805128786711	0.0155157912061197

## SQL Example: Create Binned Redshift Histogram of Galaxies

- Suppose we want to know the redshift distribution of galaxies from the SDSS spectroscopic data.

```
DECLARE @binsize FLOAT
SET @binsize = 0.1

SELECT FLOOR(z / @binsize) * @binsize as 'Redshift',
       COUNT(*) as 'Number of Galaxies'
FROM THUMPER.BESTDR7.dbo.SpecObjAll
WHERE specClass = 2
AND z BETWEEN 0 and 1
GROUP BY FLOOR(z / @binsize) * @binsize
ORDER BY FLOOR(z / @binsize) * @binsize
```



The screenshot shows a SQL query results window with two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with three columns: 'Redshift', 'Number of Galaxies', and an unlabeled index column. The data is as follows:

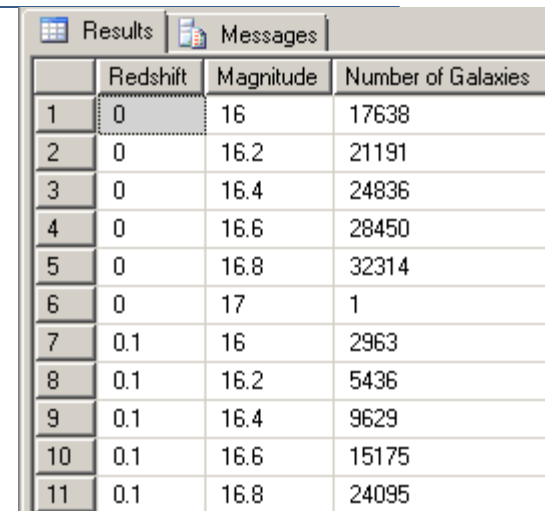
	Redshift	Number of Galaxies
1	0	380348
2	0.1	361738
3	0.2	76670
4	0.3	68830
5	0.4	34001
6	0.5	5710
7	0.6	1078
8	0.7	344
9	0.8	355
10	0.9	22

# SQL Example: Create Binned 2D (Redshift, Magnitude) Histogram of Galaxies

- We can select fields from multiple tables.
- We can also use the clause “JOIN” explicitly for this example.

```
DECLARE @binsize_z FLOAT
DECLARE @binsize_m FLOAT
SET @binsize_z = 0.1
SET @binsize_m = 0.2

SELECT FLOOR(s.z / @binsize_z) * @binsize_z as 'Redshift',
       FLOOR(petroMag_r / @binsize_m) * @binsize_m as 'Magnitude',
       COUNT(*) as 'Number of Galaxies'
FROM THUMPER.BESTDR7.dbo.SpecObjAll s, THUMPER.BESTDR7.dbo.PhotoObjAll p
WHERE specClass = 2
AND s.bestObjID = p.objID
AND s.z BETWEEN 0 and 0.2
AND p.petroMag_r BETWEEN 16.7 and 17.7
GROUP BY FLOOR(s.z / @binsize_z) * @binsize_z, FLOOR(petroMag_r / @binsize_m) * @binsize_m
ORDER BY FLOOR(s.z / @binsize_z) * @binsize_z, FLOOR(petroMag_r / @binsize_m) * @binsize_m
```



The screenshot shows a SQL query results window with a table containing 11 rows of data. The columns are Redshift, Magnitude, and Number of Galaxies. The data is as follows:

	Redshift	Magnitude	Number of Galaxies
1	0	16	17638
2	0	16.2	21191
3	0	16.4	24836
4	0	16.6	28450
5	0	16.8	32314
6	0	17	1
7	0.1	16	2963
8	0.1	16.2	5436
9	0.1	16.4	9629
10	0.1	16.6	15175
11	0.1	16.8	24095

# Data Analysis using Database

- Automated data analysis:

Select data from DB using C# routines with SQL scripts embedded

Perform computations

Output results to DB, if necessary

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays a tree view of databases, including 'thumper (SQL Server 9.0.2047 - SDSS\cwyp)'. The main window displays a SQL query in the 'Query Editor' pane, titled 'SDSSDB003.Chi\_on\_Photoz.sql'. The query is a SELECT statement that filters data from the 'thumper.bestdr6.dbo.photoobjall' table based on various parameters and calculates logarithmic values for extinction and inclination. The 'Results' pane at the bottom shows the output of the query, which is a table with 8 rows and 7 columns. The columns are labeled with '(No column name)' and 'petro u inclincorr'. The status bar at the bottom indicates 'Query executed successfully' and '3382 rows'.

```
--  
-- flux-limited sample selection for photo-z/inclination  
--  
select  
  p.petromag_u - p.extinction_u - k.ku,  
  p.petromag_g - p.extinction_g - k.kg,  
  p.petromag_r - p.extinction_r - k.kr,  
  p.petromag_i - p.extinction_i - k.ki,  
  p.petromag_z - p.extinction_z - k.kz,  
  1.90 * power(log10(p.expab_r), 4.0) as 'petro u inclincorr',  
  1.19 * power(log10(p.expab_r), 4.0) as 'petro g inclincorr',  
  0.54 * power(log10(p.expab_r), 4.0) as 'petro r inclincorr',  
  0.10 * power(log10(p.expab_r), 4.0) as 'petro i inclincorr',  
  0.00 * power(log10(p.expab_r), 4.0) as 'petro z inclincorr',  
  k.distmod  
from thumper.bestdr6.dbo.specobjall s,  
thumper.bestdr6.dbo.photoobjall p,  
sdssdb003.KcorrDR6.dbo.Kcorr k
```

(No column name)	(No column name)	(No column name)	(No column name)	(No column name)	(No column name)	petro u inclincorr	peti
1	17.74668	16.69974	16.35954	16.21339	16.35263	0.492397508119744	0.3
2	18.43394	17.29214	16.62328	16.28204	16.08086	1.52441919075319	0.9
3	19.42674	18.24301	17.70235	17.49629	17.15921	0.464193794342971	0.2
4	18.74544	17.94984	17.51807	17.26874	17.1523	0.500948764848324	0.3
5	18.949	17.80621	17.22119	16.86648	16.60248	0.68160486654221	0.4
6	19.22604	17.9269	17.52087	17.16631	17.18861	0.453664367877463	0.2
7	18.04691	16.87437	16.2979	16.01679	15.84145	0.837249857822637	0.5
8	18.11651	16.77158	16.13685	15.69099	15.65387	0.875719228372087	0.5

(MS SQL Server. Source: Alex Szalay)



# CasJobs

- Available for public.
- Users can register and search the public SDSS data.
- All SDSS data will become public some time after the survey completes.

SDSS Query / CasJobs

Help Tools Create Account Login Not Logged in

User ID

Password

Login

If you do not have a login please **create an account**.

Contact  
\$Name: v3\_5\_16 \$, \$Revision: 1.11 \$, Last modified: Tuesday, April 24, 2007 at 6:52:23 PM

1/23/2014 JHU Intersesion Course - C. W. Yip

**Exercise: Create an account in CasJobs.**

# Open SkyQuery

- An ambitious platform for storing and cross-matching Catalogs from many Astronomy surveys.
- Under big overhaul and new development (2014) by L. Dobos and collaborators.

The screenshot shows a Windows Internet Explorer browser window displaying the Open SkyQuery website. The browser's address bar shows the URL <http://openskyquery.net/Sky/skysite/>. The search bar contains the text "open skyquery". The website header features the NVO logo (National Virtual Observatory) on the left, the "Open SkyQuery" title in the center, and the "Hosted By" logo for Johns Hopkins University on the right. Below the header is a navigation menu with links for "Advanced Query", "Import Data", "Help", and "Contact Us". The main content area displays a large blue heading "Welcome to Open SkyQuery!" followed by a paragraph: "OpenSkyQuery allows you to cross-match astronomical catalogs and select subsets of catalogs with a general and powerful query language. You can also [import a personal catalog](#) of objects and cross-match it against selected databases." Below this paragraph, there is a line of text: "To get started, go to the [Simple Query page](#) to run a simple form query, or go the the [Advanced Query page](#) and look at some of the samples." The date "1/23/2014" is visible in the bottom left corner of the browser window.

## Hooking Up Database using R

- Here we use Microsoft Windows Operating System.
- Two main steps:
  - Set up user's Data Source Name (DNS) in Windows.
  - Install R library for Open Database Connectivity (RODBC).
- See class demonstration.
- The R script can be downloaded from the Course Website.

# Big Astronomy Databases

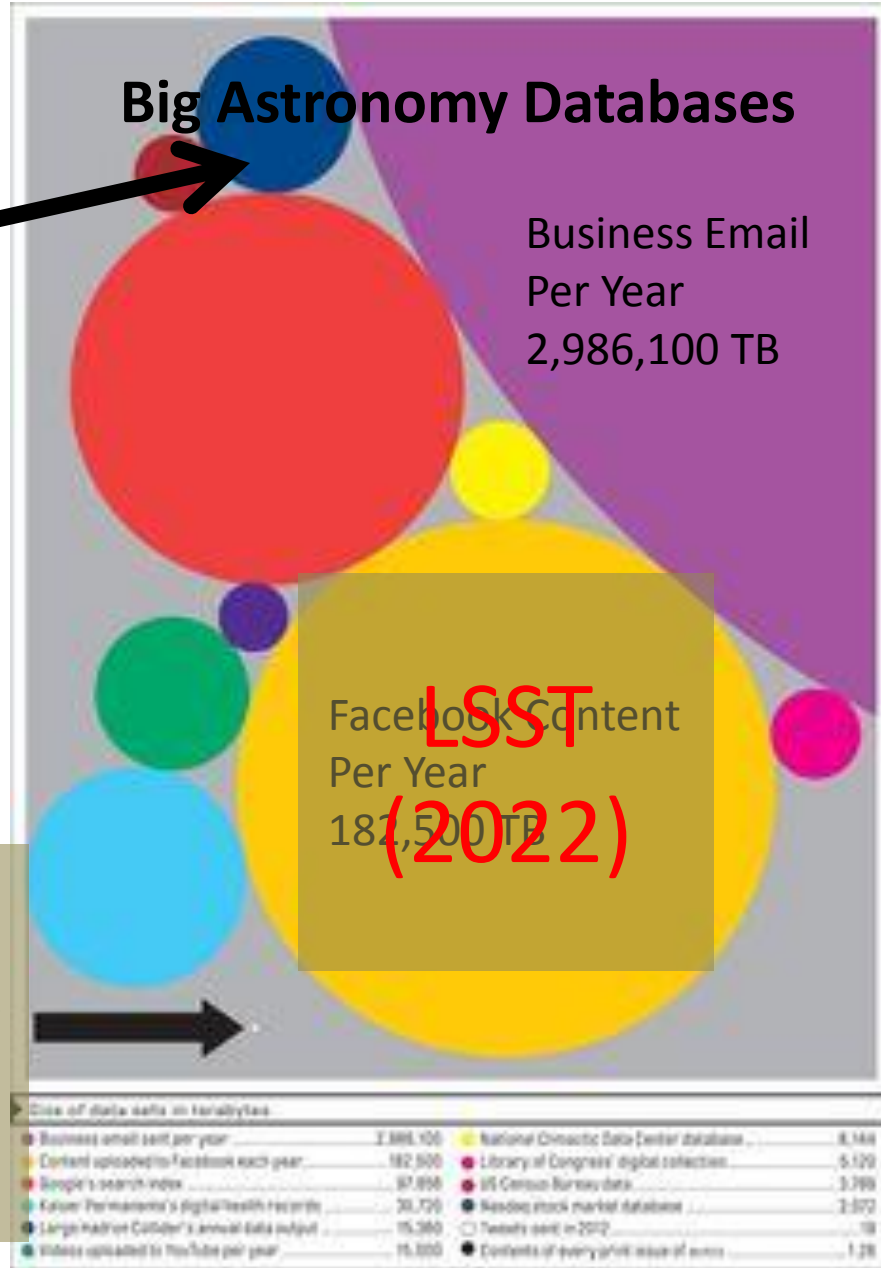
Large Hadron Collider  
15,360 TB

Business Email  
Per Year  
2,986,100 TB

1Mega = 1,000,000 =  $10^6$   
 1Giga =  $10^9$   
 1Tera =  $10^{12}$   
 1Peta =  $10^{15}$   
 1Exa =  $10^{18}$   
 1Zetta =  $10^{21}$

Facebook Content  
Per Year  
182,500 TB  
**LSST  
(2022)**

**SDSS**  
Tweets in 2012  
19 TB  
**(now)**



(WIRED, May 2013)