SQL databases with python

What is an SQL database?

"Indexed" data – when a query is performed
Only a section of the database is queried

Much faster data access (only important for large datasets)

What are the advantages?

Centralised data access → fewer mistakes and lost files and irritating filenames i.e. “data-final-2-superfinal_1.txt”

‘Dynamic results’ – results are easily updated from user changes

Your database gets more comprehensive the more you use it, building something worthwhile → important in large projects

What are the advantages?

Collection of data from various sources in one place makes huge cross-matching very simple.

Code	Reference	Parameters
NO04	Nordström et al. (2004)	RV
BA12	Bailey et al. (2012)	RV
EL14	Elliott et al. (2014)	RV
MA14	Malo et al. (2014)	RV, H_α, Li, L_x
RAVE	Kordopatis et al. (2013)	RV
DE15	Desidera et al. (2015)	RV, Li
RO13	Rodriguez et al. (2013)	RV, H_α, Li, NaI
RI06	Riaz et al. (2006)	RV,H_α,L_x/L_{bol},CaH1,2,3
BE15	Bell et al. (2015)	*V* mag.
IPHAS	Barentsen & et al. (2014)	H_α
GALEX	Bianchi et al. (2011)	NUV, FUV mag.
URAT1	Zacharias et al. (2015)	*R*,*B*,*V* mag.
CATA	Drake et al. (2009)	*V* mag. (+ variability)
ASAS	Pojmanski (2002)	*V* mag. (+ variability)
RO13	Rodriguez et al. (2013)	RV
SH12	Shkolnik et al. (2012)	RV
SC12	Schlieder et al. (2012b)	RV, SpT
SPM4	Girard et al. (2011)	*B*,*V* mag.
RE09	Reiners & Basri (2009)	RV, Li presence
RI15	Riviere-Marichalar et al. (2015)	RV, H_α
M013	Moór et al. (2013)	RV, Li, H_α,L_x/L_{bol}
GO06	Gontcharov (2006)	RV, *V* mag.
DE12	de Bruijne & Eilers (2012)	RV
FR13	Frith et al. (2013)	RV
MO08	Morin et al. (2008)	RV
LO10	Looper et al. (2010)	RV
KH07	Kharchenko et al. (2007)	RV
HO91	Hoffleit & Jaschek (1991)	RV
LE13	Lépine et al. (2013)	H_α, CaH2,3
MO01a	Montes et al. (2001)	RV
LO06	López-Santiago et al. (2006)	RV, Li
MA10	Maldonado et al. (2010)	RV
WISE	Cutri et al. (2003)	*W*1, *W*2, *W*3, *W*4
RI14	Riedel et al. (2014)	*V* mag.
VI56	Vyssotsky (1956)	*V* mag.
ME06	Mermilliod (2006)	*V* mag.
LA01	Lawson et al. (2001)	*V* mag.
KH09	Kharchenko et al. (2009)	*V* mag.
NOMAD	Zacharias et al. (2005)	*B*,*V*, *R* mag.
WDS	Mason et al. (2001)	Multiplicity
MA13	Malo et al. (2013)	RV,H_α,Li
GSC	Lasker et al. (2008)	*B*,*U*,*V* mag.
TYC	Høg et al. (2000)	*B*,*V* mag.
3XMM	Xmm-Newton Survey Science Centre (2013)	X-ray: flux
CHAN	Evans et al. (2010)	X-ray: flux
BSC	Voges et al. (1999)	X-ray: counts, HR
FSC	Voges et al. (2000)	X-ray: counts, HR

One method:

Cross-matches all this information
And feeds it into my 'go-to' table

User interface for SQL, Sequel Pro

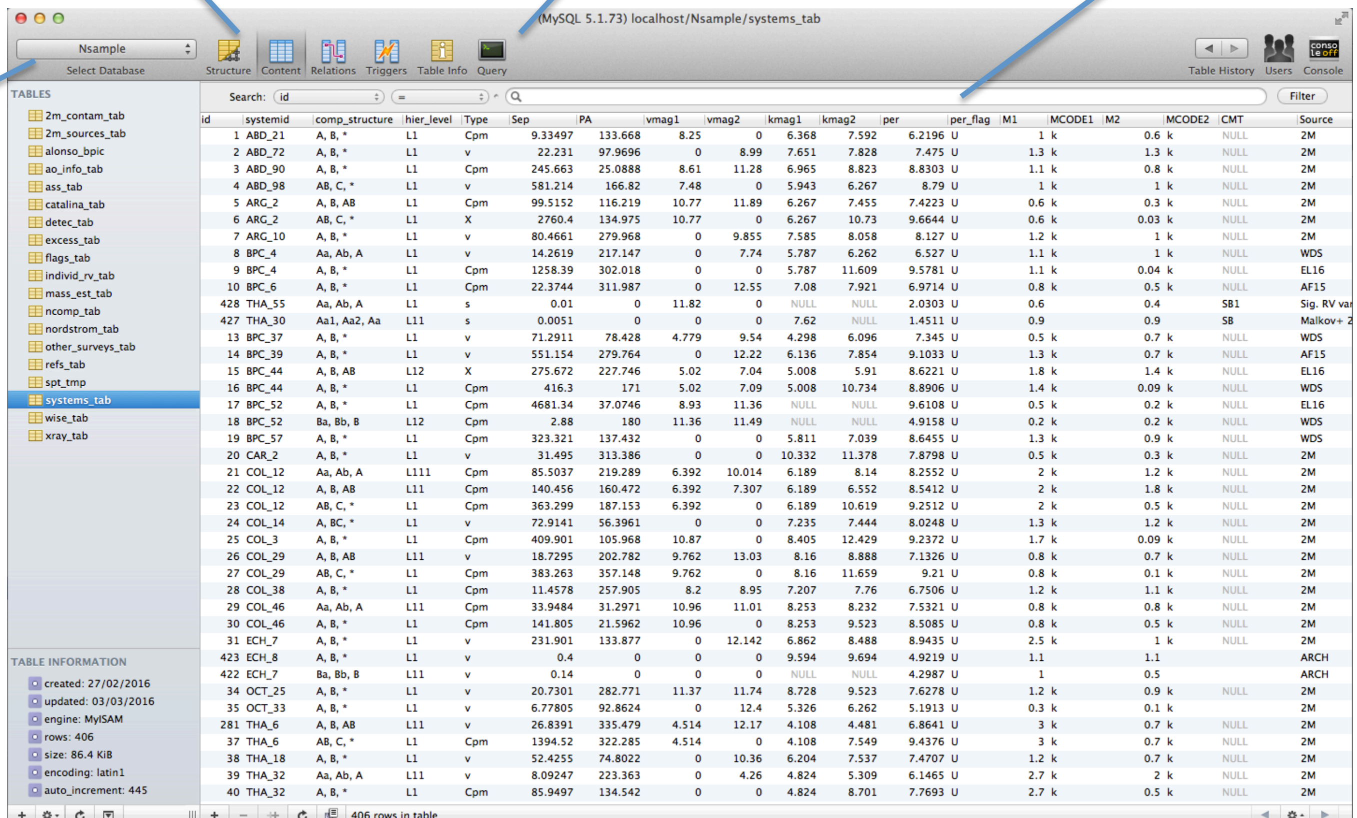
Structure tab

Query tab

Search facility

Database name

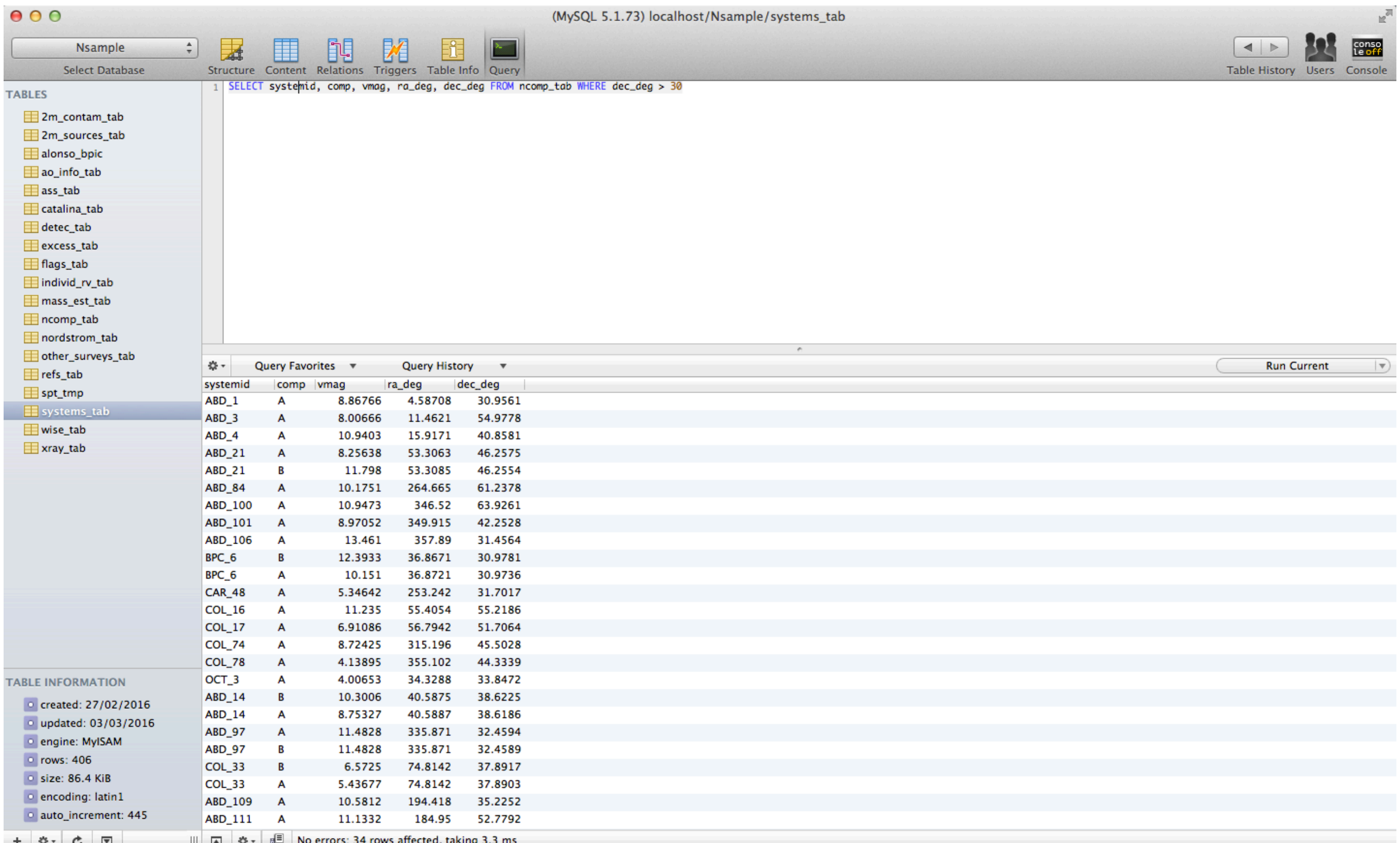
List of tables



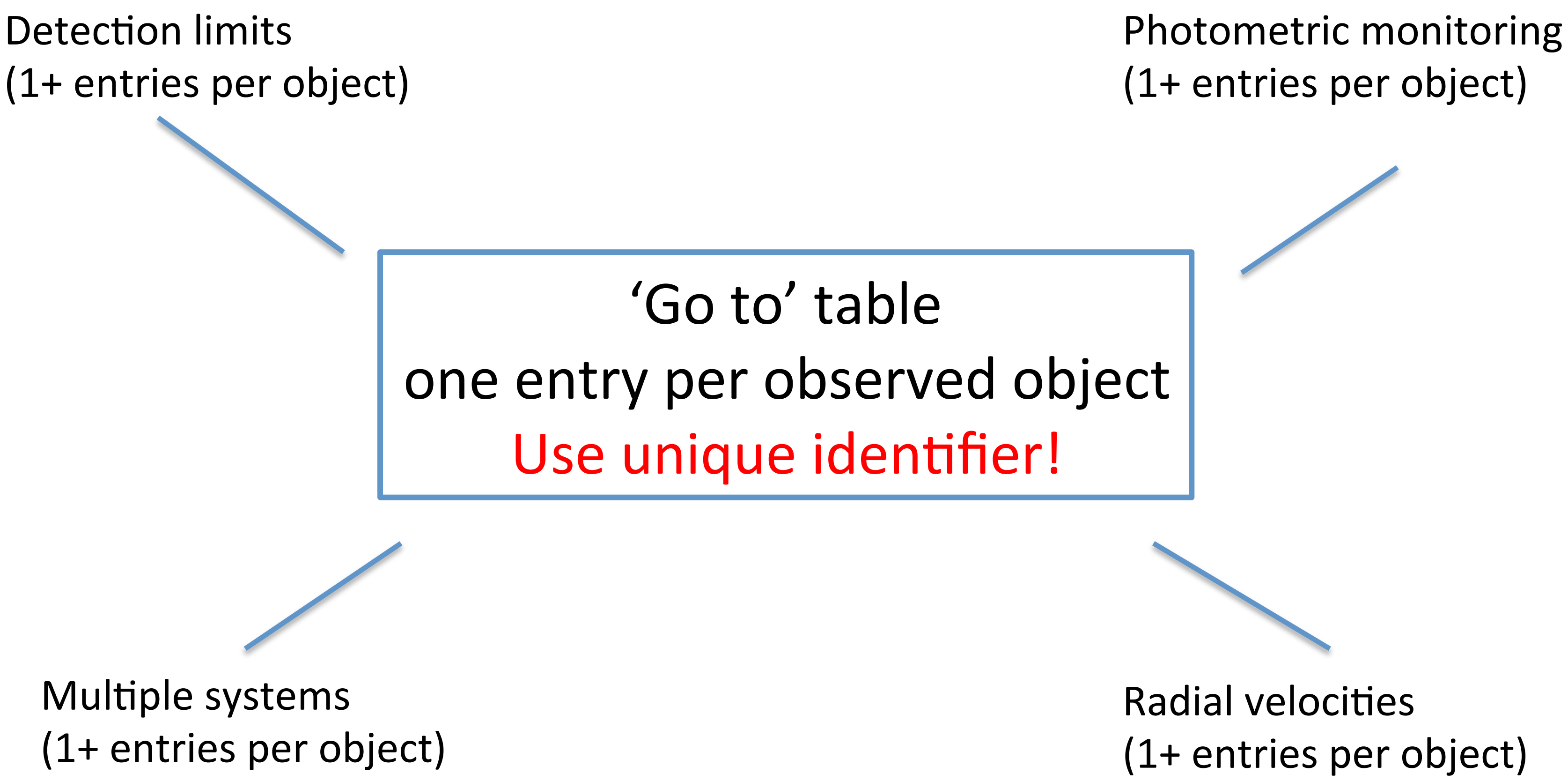
User interface for SQL, Sequel Pro

Example query

“SELECT systemid, comp, vmag, ra_deg, dec_deg FROM ncomp_tab WHERE dec_deg > 30”



Database structure



Creating methods to manipulate your SQL data

Create a class for your object (in my case a "multiple system")

Create a bunch of methods / functions that will act on that object
i.e. do something useful with all that data in the SQL database

Perfect for often-repeated analysis

See additional notes for a few examples of methods