

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.

One method:

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

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)	W1, W2, W3, W4
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

User interface for SQL, Sequel Pro

Structure tab

Query tab

Search facility

MySQL 5.1.73 localhost/Nsample/systems_tab

Select Database Structure Content Relations Triggers Table Info Query

Table History Users Console

Search: id

id	systemid	comp_structure	hier_level	Type	Sep	PA	vmag1	vmag2	kmag1	kmag2	per	per_flag	M1	MCODE1	M2	MCODE2	CMT	Source
1	ABD_21	A, B, *	L1	Cpm	9.33497	133.668	8.25	0	6.368	7.592	6.2196	U	1	k	0.6	k	NULL	2M
2	ABD_72	A, B, *	L1	v	22.231	97.9696	0	8.99	7.651	7.828	7.475	U	1.3	k	1.3	k	NULL	2M
3	ABD_90	A, B, *	L1	Cpm	245.663	25.0888	8.61	11.28	6.965	8.823	8.8303	U	1.1	k	0.8	k	NULL	2M
4	ABD_98	AB, C, *	L1	v	581.214	166.82	7.48	0	5.943	6.267	8.79	U	1	k	1	k	NULL	2M
5	ARG_2	A, B, AB	L1	Cpm	99.5152	116.219	10.77	11.89	6.267	7.455	7.4223	U	0.6	k	0.3	k	NULL	2M
6	ARG_2	AB, C, *	L1	X	2760.4	134.975	10.77	0	6.267	10.73	9.6644	U	0.6	k	0.03	k	NULL	2M
7	ARG_10	A, B, *	L1	v	80.4661	279.968	0	9.855	7.585	8.058	8.127	U	1.2	k	1	k	NULL	2M
8	BPC_4	Aa, Ab, A	L1	v	14.2619	217.147	0	7.74	5.787	6.262	6.527	U	1.1	k	1	k	NULL	WDS
9	BPC_4	A, B, *	L1	Cpm	1258.39	302.018	0	0	5.787	11.609	9.5781	U	1.1	k	0.04	k	NULL	EL16
10	BPC_6	A, B, *	L1	Cpm	22.3744	311.987	0	12.55	7.08	7.921	6.9714	U	0.8	k	0.5	k	NULL	AF15
428	THA_55	Aa, Ab, A	L1	s	0.01	0	11.82	0	NULL	NULL	2.0303	U	0.6	0.4	0.4	SB1	Sig. RV var	
427	THA_30	Aa1, Aa2, Aa	L11	s	0.0051	0	0	0	7.62	NULL	1.4511	U	0.9	0.9	0.9	SB	Malkov+2	
13	BPC_37	A, B, *	L1	v	71.2911	78.428	4.779	9.54	4.298	6.096	7.345	U	0.5	k	0.7	k	NULL	WDS
14	BPC_39	A, B, *	L1	v	551.154	279.764	0	12.22	6.136	7.854	9.1033	U	1.3	k	0.7	k	NULL	AF15
15	BPC_44	A, B, AB	L12	X	275.672	227.746	5.02	7.04	5.008	5.91	8.6221	U	1.8	k	1.4	k	NULL	EL16
16	BPC_44	A, B, *	L1	Cpm	416.3	171	5.02	7.09	5.008	10.734	8.8906	U	1.4	k	0.09	k	NULL	WDS
17	BPC_52	A, B, *	L1	Cpm	4681.34	37.0746	8.93	11.36	NULL	NULL	9.6108	U	0.5	k	0.2	k	NULL	EL16
18	BPC_52	Ba, Bb, B	L12	Cpm	2.88	180	11.36	11.49	NULL	NULL	4.9158	U	0.2	k	0.2	k	NULL	WDS
19	BPC_57	A, B, *	L1	Cpm	323.321	137.432	0	0	5.811	7.039	8.6455	U	1.3	k	0.9	k	NULL	WDS
20	CAR_2	A, B, *	L1	v	31.495	313.386	0	0	10.332	11.378	7.8798	U	0.5	k	0.3	k	NULL	2M
21	COL_12	Aa, Ab, A	L111	Cpm	85.5037	219.289	6.392	10.014	6.189	8.14	8.2552	U	2	k	1.2	k	NULL	2M
22	COL_12	A, B, AB	L11	Cpm	140.456	160.472	6.392	7.307	6.189	6.552	8.5412	U	2	k	1.8	k	NULL	2M
23	COL_12	AB, C, *	L1	Cpm	363.299	187.153	6.392	0	6.189	10.619	9.2512	U	2	k	0.5	k	NULL	2M
24	COL_14	A, BC, *	L1	v	72.9141	56.3961	0	0	7.235	7.444	8.0248	U	1.3	k	1.2	k	NULL	2M
25	COL_3	A, B, *	L1	Cpm	409.901	105.968	10.87	0	8.405	12.429	9.2372	U	1.7	k	0.09	k	NULL	2M
26	COL_29	A, B, AB	L11	v	18.7295	202.782	9.762	13.03	8.16	8.888	7.1326	U	0.8	k	0.7	k	NULL	2M
27	COL_29	AB, C, *	L1	Cpm	383.263	357.148	9.762	0	8.16	11.659	9.21	U	0.8	k	0.1	k	NULL	2M
28	COL_38	A, B, *	L1	Cpm	11.4578	257.905	8.2	8.95	7.207	7.76	6.7506	U	1.2	k	1.1	k	NULL	2M
29	COL_46	Aa, Ab, A	L11	Cpm	33.9484	31.2971	10.96	11.01	8.253	8.232	7.5321	U	0.8	k	0.8	k	NULL	2M
30	COL_46	A, B, *	L1	Cpm	141.805	21.5962	10.96	0	8.253	9.523	8.5085	U	0.8	k	0.5	k	NULL	2M
31	ECH_7	A, B, *	L1	v	231.901	133.877	0	12.142	6.862	8.488	8.9435	U	2.5	k	1	k	NULL	2M
423	ECH_8	A, B, *	L1	v	0.4	0	0	0	9.594	9.694	4.9219	U	1.1	1.1	1.1	ARCH		
422	ECH_7	Ba, Bb, B	L11	v	0.14	0	0	0	NULL	NULL	4.2987	U	1	0.5	0.5	ARCH		
34	OCT_25	A, B, *	L1	v	20.7301	282.771	11.37	11.74	8.728	9.523	7.6278	U	1.2	k	0.9	k	NULL	2M
35	OCT_33	A, B, *	L1	v	6.77805	92.8624	0	12.4	5.326	6.262	5.1913	U	0.3	k	0.1	k	NULL	2M
281	THA_6	A, B, AB	L11	v	26.8391	335.479	4.514	12.17	4.108	4.481	6.8641	U	3	k	0.7	k	NULL	2M
37	THA_6	AB, C, *	L1	Cpm	1394.52	322.285	4.514	0	4.108	7.549	9.4376	U	3	k	0.7	k	NULL	2M
38	THA_18	A, B, *	L1	v	52.4255	74.8022	0	10.36	6.204	7.537	7.4707	U	1.2	k	0.7	k	NULL	2M
39	THA_32	Aa, Ab, A	L11	v	8.09247	223.363	0	4.26	4.824	5.309	6.1465	U	2.7	k	2	k	NULL	2M
40	THA_32	A, B, *	L1	Cpm	85.9497	134.542	0	0	4.824	8.701	7.7693	U	2.7	k	0.5	k	NULL	2M

TABLE INFORMATION

- created: 27/02/2016
- updated: 03/03/2016
- engine: MyISAM
- rows: 406
- size: 86.4 KiB
- encoding: latin1
- auto_increment: 445

406 rows in table

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”

The screenshot shows the Sequel Pro application window titled "(MySQL 5.1.73) localhost/Nsample/systems_tab". The interface includes a sidebar with a "TABLES" list on the left and a "TABLE INFORMATION" section at the bottom left. The main area displays a SQL query: "SELECT systemid, comp, vmag, ra_deg, dec_deg FROM ncomp_tab WHERE dec_deg > 30". Below the query, the results are shown in a table with the following columns: systemid, comp, vmag, ra_deg, and dec_deg. The table contains 34 rows of data. At the bottom, a status bar indicates "No errors; 34 rows affected, taking 3.3 ms".

TABLES

- Zm_contam_tab
- Zm_sources_tab
- alonso_bplic
- ao_info_tab
- ass_tab
- catalina_tab
- detec_tab
- excess_tab
- flags_tab
- individ_rv_tab
- mass_est_tab
- ncomp_tab
- nordstrom_tab
- other_surveys_tab
- refs_tab
- spt_tmp
- systems_tab
- wise_tab
- xray_tab

TABLE INFORMATION

- created: 27/02/2016
- updated: 03/03/2016
- engine: MyISAM
- rows: 406
- size: 86.4 KIB
- encoding: latin1
- auto_increment: 445

Query: SELECT systemid, comp, vmag, ra_deg, dec_deg FROM ncomp_tab WHERE dec_deg > 30

systemid	comp	vmag	ra_deg	dec_deg
ABD_1	A	8.86766	4.58708	30.9561
ABD_3	A	8.00666	11.4621	54.9778
ABD_4	A	10.9403	15.9171	40.8581
ABD_21	A	8.25638	53.3063	46.2575
ABD_21	B	11.798	53.3085	46.2554
ABD_84	A	10.1751	264.665	61.2378
ABD_100	A	10.9473	346.52	63.9261
ABD_101	A	8.97052	349.915	42.2528
ABD_106	A	13.461	357.89	31.4564
BPC_6	B	12.3933	36.8671	30.9781
BPC_6	A	10.151	36.8721	30.9736
CAR_48	A	5.34642	253.242	31.7017
COL_16	A	11.235	55.4054	55.2186
COL_17	A	6.91086	56.7942	51.7064
COL_74	A	8.72425	315.196	45.5028
COL_78	A	4.13895	355.102	44.3339
OCT_3	A	4.00653	34.3288	33.8472
ABD_14	B	10.3006	40.5875	38.6225
ABD_14	A	8.75327	40.5887	38.6186
ABD_97	A	11.4828	335.871	32.4594
ABD_97	B	11.4828	335.871	32.4589
COL_33	B	6.5725	74.8142	37.8917
COL_33	A	5.43677	74.8142	37.8903
ABD_109	A	10.5812	194.418	35.2252
ABD_111	A	11.1332	184.95	52.7792

No errors; 34 rows affected, taking 3.3 ms

Database structure

Detection limits
(1+ entries per object)

Photometric monitoring
(1+ entries per object)

'Go to' table
one entry per observed object
Use unique identifier!

Multiple systems
(1+ entries per object)

Radial velocities
(1+ entries per object)

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