SQL queries on Mock data:
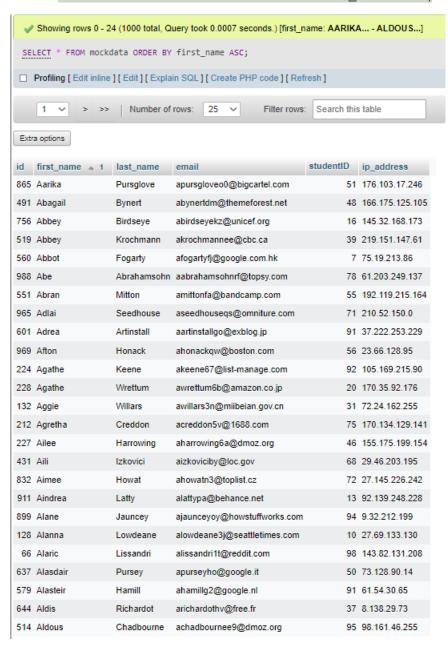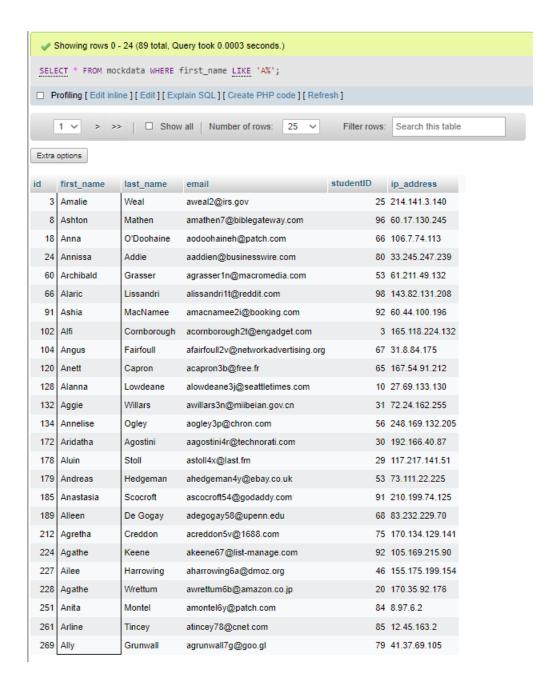
1) Write a query that alphabetizes the first names in ascending order.

SELECT * FROM mockdata ORDER BY first_name ASC;
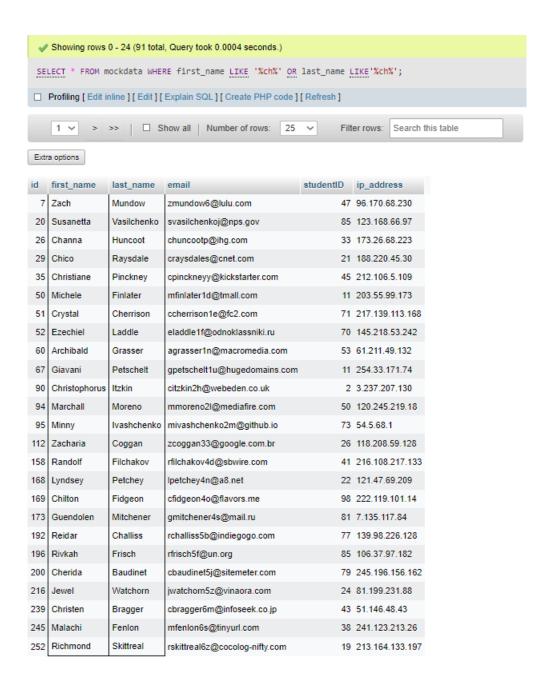
id	first_name ▲ 1	last_name	email	studentID	ip_address
865	Aarika	Pursglove	apursgloveo0@bigcartel.com	51	176.103.17.246
491	Abagail	Bynert	abynertdm@themeforest.net	48	166.175.125.105
756	Abbey	Birdseye	abirdseyekz@unicef.org	16	145.32.168.173
519	Abbey	Krochmann	akrochmannee@cbc.ca	39	219.151.147.61
560	Abbot	Fogarty	afogartyfj@google.com.hk	7	75.19.213.86
988	Abe	Abrahamsohn	aabrahamsohnrf@topsy.com	78	61.203.249.137
551	Abran	Mitton	amittonfa@bandcamp.com	55	192.119.215.164
965	Adlai	Seedhouse	aseedhouseqs@omniture.com	71	210.52.150.0
601	Adrea	Artinstall	aartinstallgo@exblog.jp	91	37.222.253.229
969	Afton	Honack	ahonackqw@boston.com	56	23.66.128.95
224	Agathe	Keene	akeene67@list-manage.com	92	105.169.215.90
228	Agathe	Wrettum	awrettum6b@amazon.co.jp	20	170.35.92.176
132	Aggie	Willars	awillars3n@miibeian.gov.cn	31	72.24.162.255
212	Agretha	Creddon	acreddon5v@1688.com	75	170.134.129.141
227	Ailee	Harrowing	aharrowing6a@dmoz.org	46	155.175.199.154
431	Aili	Izkovici	aizkoviciby@loc.gov	68	29.46.203.195
832	Aimee	Howat	ahowatn3@toplist.cz	72	27.145.226.242
911	Aindrea	Latty	alattypa@behance.net	13	92.139.248.228
899	Alane	Jauncey	ajaunceyoy@howstuffworks.com	94	9.32.212.199
128	Alanna	Lowdeane	alowdeane3j@seattletimes.com	10	27.69.133.130
66	Alaric	Lissandri	alissandri1t@reddit.com	98	143.82.131.208
637	Alasdair	Pursey	apurseyho@google.it	50	73.128.90.14
579	Alasteir	Hamill	ahamillg2@google.nl	91	61.54.30.65
644	Aldis	Richardot	arichardothv@free.fr	37	8.138.29.73
514	Aldous	Chadbourne	achadbournee9@dmoz.org	95	98.161.46.255

2) Write a query that outputs every student whose first name starts with an "A".

SELECT * FROM mockdata WHERE first_name LIKE 'A%';

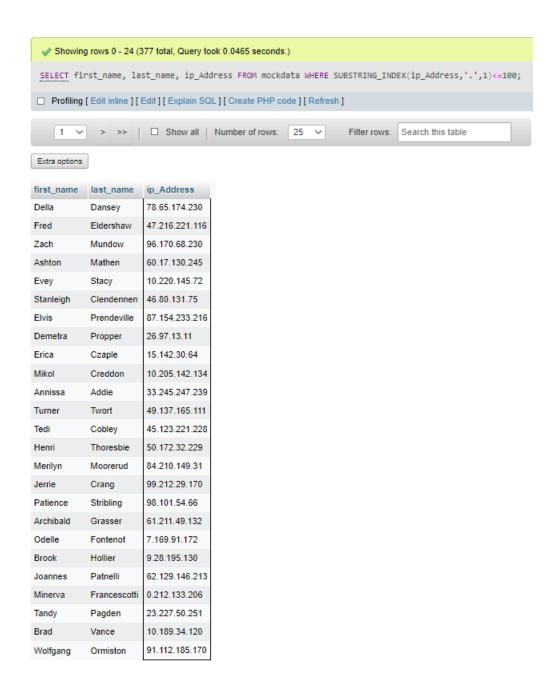| id	first_name	last_name	email	studentID	ip_address
3	Amalie	Weal	aweal2@irs.gov	25	214.141.3.140
8	Ashton	Mathen	amathen7@biblegateway.com	96	60.17.130.245
18	Anna	O'Doohaine	aodoohaineh@patch.com	66	106.7.74.113
24	Annissa	Addie	aaddien@businesswire.com	80	33.245.247.239
60	Archibald	Grasser	agrasser1n@macromedia.com	53	61.211.49.132
66	Alaric	Lissandri	alissandri1t@reddit.com	98	143.82.131.208
91	Ashia	MacNamee	amacnamee2i@booking.com	92	60.44.100.196
102	Alfi	Cornborough	acornborough2t@engadget.com	3	165.118.224.132
104	Angus	Fairfoull	afairfoull2v@networkadvertising.org	67	31.8.84.175
120	Anett	Capron	acapron3b@free.fr	65	167.54.91.212
128	Alanna	Lowdeane	alowdeane3j@seattletimes.com	10	27.69.133.130
132	Aggie	Willars	awillars3n@miibeian.gov.cn	31	72.24.162.255
134	Annelise	Ogley	aogley3p@chron.com	56	248.169.132.205
172	Aridatha	Agostini	aagostini4r@technorati.com	30	192.166.40.87
178	Aluin	Stoll	astoll4x@last.fm	29	117.217.141.51
179	Andreas	Hedgeman	ahedgeman4y@ebay.co.uk	53	73.111.22.225
185	Anastasia	Scocroft	ascocroft54@godaddy.com	91	210.199.74.125
189	Alleen	De Gogay	adegogay58@upenn.edu	68	83.232.229.70
212	Agretha	Creddon	acreddon5v@1688.com	75	170.134.129.141
224	Agathe	Keene	akeene67@list-manage.com	92	105.169.215.90
227	Ailee	Harrowing	aharrowing6a@dmoz.org	46	155.175.199.154
228	Agathe	Wrettum	awrettum6b@amazon.co.jp	20	170.35.92.176
251	Anita	Montel	amontel6y@patch.com	84	8.97.6.2
261	Arline	Tincey	atincey78@cnet.com	85	12.45.163.2
269	Ally	Grunwall	agrunwall7g@goo.gl	79	41.37.69.105

3) Write a query that outputs every student who has a 'ch' anywhere in their name (first or last).

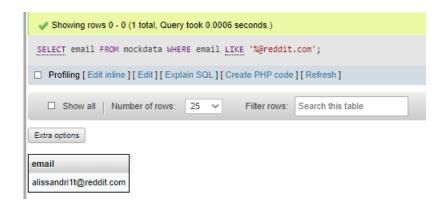SELECT * FROM mockdata WHERE first_name LIKE '%ch%' OR last_name LIKE'%ch%';

id	first_name	last_name	email	studentID	ip_address
7	Zach	Mundow	zmundow6@lulu.com	47	96.170.68.230
20	Susanetta	Vasilchenko	svasilchenkoj@nps.gov	85	123.168.66.97
26	Channa	Huncoot	chuncootp@ihg.com	33	173.26.68.223
29	Chico	Raysdale	craysdales@cnet.com	21	188.220.45.30
35	Christiane	Pinckney	cpinckneyy@kickstarter.com	45	212.106.5.109
50	Michele	Finlater	mfinlater1d@tmall.com	11	203.55.99.173
51	Crystal	Cherrison	ccherrison1e@fc2.com	71	217.139.113.168
52	Ezechiel	Laddle	eladdle1f@odnoklassniki.ru	70	145.218.53.242
60	Archibald	Grasser	agrasser1n@macromedia.com	53	61.211.49.132
67	Giavani	Petschelt	gpetschelt1u@hugedomains.com	11	254.33.171.74
90	Christophorus	Itzkin	citzkin2h@webeden.co.uk	2	3.237.207.130
94	Marchall	Moreno	mmoreno2l@mediafire.com	50	120.245.219.18
95	Minny	Ivashchenko	mivashchenko2m@github.io	73	54.5.68.1
112	Zacharia	Coggan	zcoggan33@google.com.br	26	118.208.59.128
158	Randolf	Filchakov	rfilchakov4d@sbwire.com	41	216.108.217.133
168	Lyndsey	Petchey	lpetchey4n@a8.net	22	121.47.69.209
169	Chilton	Fidgeon	cfidgeon4o@flavors.me	98	222.119.101.14
173	Guendolen	Mitchener	gmitchener4s@mail.ru	81	7.135.117.84
192	Reidar	Challiss	rchalliss5b@indiegogo.com	77	139.98.226.128
196	Rivkah	Frisch	rfrisch5f@un.org	85	106.37.97.182
200	Cherida	Baudinet	cbaudinet5j@sitemeter.com	79	245.196.156.162
216	Jewel	Watchorn	jwatchorn5z@vinaora.com	24	81.199.231.88
239	Christen	Bragger	cbragger6m@infoseek.co.jp	43	51.146.48.43
245	Malachi	Fenlon	mfenlon6s@tinyurl.com	38	241.123.213.26
252	Richmond	Skittreal	rskittreal6z@cocolog-nifty.com	19	213.164.133.197

4) Write a query that selects the first name, last name, and IP address, for every IP address that starts with '100' or less.

SELECT first_name, last_name, ip_Address FROM mockdata WHERE SUBSTRING_INDEX(ip_Address,'.',1)<=100;

| 1 ⌄ | > | >> | | ☐ Show all | Number of rows: | 25 ⌄ | Filter rows: | Search this table |

Extra options

first_name	last_name	ip_Address
Della	Dansey	78.65.174.230
Fred	Eldershaw	47.216.221.116
Zach	Mundow	96.170.68.230
Ashton	Mathen	60.17.130.245
Evey	Stacy	10.220.145.72
Stanleigh	Clendennen	46.80.131.75
Elvis	Prendeville	87.154.233.216
Demetra	Propper	26.97.13.11
Erica	Czaple	15.142.30.64
Mikol	Creddon	10.205.142.134
Annissa	Addie	33.245.247.239
Turner	Twort	49.137.165.111
Tedi	Cobley	45.123.221.228
Henri	Thoresbie	50.172.32.229
Merilyn	Moorerud	84.210.149.31
Jerrie	Crang	99.212.29.170
Patience	Stribling	98.101.54.66
Archibald	Grasser	61.211.49.132
Odelle	Fontenot	7.169.91.172
Brook	Hollier	9.28.195.130
Joannes	Patnelli	62.129.146.213
Minerva	Francescotti	0.212.133.206
Tandy	Pagden	23.227.50.251
Brad	Vance	10.189.34.120
Wolfgang	Ormiston	91.112.185.170

5) Write a query that searches for any email address registered with the reddit.com domain (i.e., @reddit.com).

SELECT email FROM mockdata WHERE email LIKE '%@reddit.com';

6) Write a query that inserts four records with incorrectly formatted email addresses (e.g., aiden@univ.ca is a correctly formatted email address but aidenuniv.ca is not)

INSERT INTO mockdata(id, first_name, last_name, email, studentID, ip_Address) VALUES ('1800','Ravi','Chandan','ravichandantrentuca','0741418','99.26.35.12');

INSERT INTO mockdata(id, first_name, last_name, email, studentID, ip_Address) VALUES ('1802','cristiano','ronaldo','cr7@trentuca','0754863','99.86.96.12');

INSERT INTO mockdata(id, first_name, last_name, email, studentID, ip_Address) VALUES ('1803','lionel','messi','liomessi@redditca','0795684','98.63.12.69');
INSERT INTO mockdata(id, first_name, last_name, email, studentID, ip_Address) VALUES ('1804','martin','odegard','martinodreddit.ca','0785696','95.63.12.31');

a. Write a query with a regular expression that extracts all the email addresses that are not in the correct format. You can lookup the regular expression (cite your source in the PDF that you submit, you do not have to use a particular citation formation, the link is just fine).

Without regular expression:

SELECT * FROM mockdata WHERE email NOT LIKE '%_@_%._%';

id	first_name	last_name	email	studentID	ip_address
1800	Ravi	Chandan	ravichandantrentuca	741418	99.26.35.12
1802	cristiano	ronaldo	cr7@trentuca	754863	99.86.96.12
1803	lionel	messi	liomessi@redditca	795684	98.63.12.69
1804	martin	odegard	martinodreddit.ca	785696	95.63.12.31

With regular expression:

SELECT email FROM mockdata WHERE email NOT REGEXP '^[A-Z0-9+_.-]+@[A-Z0-9.-]+\.[A-Z0-9.-]+$';

email
ravichandantrentuca
martinodreddit.ca

b. Lookup what MySQL procedures are and write a MySQL procedure that deletes all the email addresses that are not in the correct format.

```
DELIMITER &&

CREATE PROCEDURE delete_incorrect_email ()

BEGIN

SELECT COUNT(email) AS CountBeforeDeletion FROM mockdata

WHERE email NOT LIKE '%_@_%._%';

DELETE FROM mockdata WHERE email NOT LIKE '%_@_%._%';

SELECT COUNT(email) AS CountAfterDeletion FROM mockdata WHERE

email NOT LIKE '%_@_%._%';

END &&
```

```
DELIMITER ;

CALL `delete_incorrect_email`();
```

Execution results of routine `delete_incorrect_email`

CountBeforeDeletion

4

CountAfterDeletion

0