Institut für Informatik
Datenbanken und Informationssysteme
Prof. Dr. Stefan Conrad
Thomas Germer

WiSe 24/25
Assignment 6
Due November 26, 8:00 AM

Relational Databases and Data Analysis

- As usual, use `zipme.py` to create the archive to upload in ILIAS.
- Some exercises ask you to take a *screenshot*. Photos taken with phones or similar give 0 points.

Exercise 1 *Security* (2 + 1 + 2 + 1 Points)

Inform yourself about SQL injection and Cross-site Scripting (XSS). (← There are two links. Consider a better PDF reader if the links are not visible.)

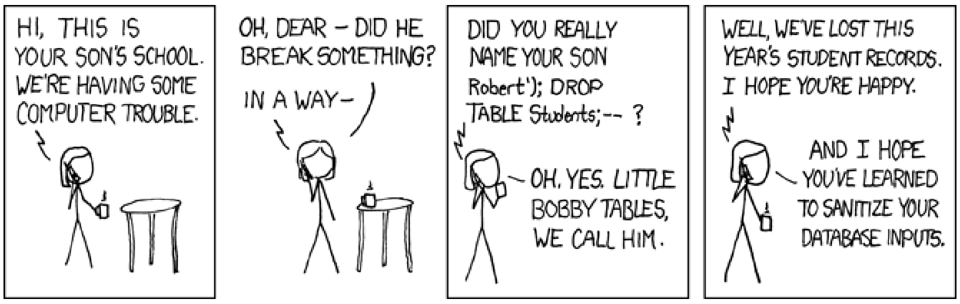
Businessman Elton Mollusk recently acquired the fictional social media platform “Tweeter”. You can find the source code in `exercise_1.py`. We have been tasked with discovering security vulnerabilities.

(a) The platform is vulnerable to SQL injection. Demonstrate this by posting a message as the user eltonmollusk without using his password. Explain your approach in `exercise_1_a.txt` and take a screenshot of your message named `exercise_1_a.png`.

(b) Describe how to fix the SQL injection vulnerability in `exercise_1_b.txt`. Disallowing certain passwords is not an option, since it decreases password entropy for no good reason.

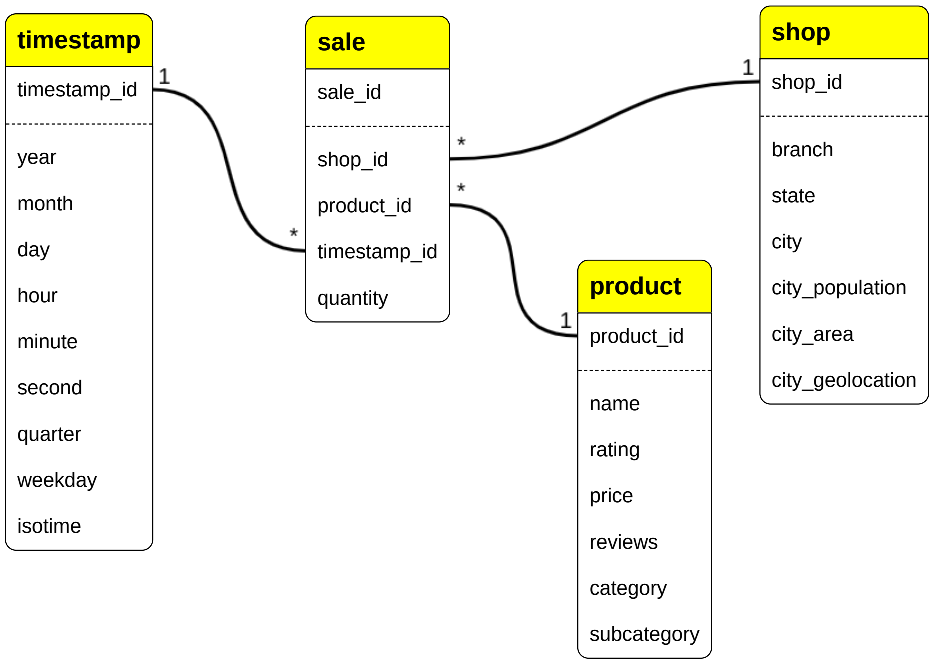
(c) The platform is vulnerable to XSS. Demonstrate this issue, take a screenshot of it named `exercise_1_c.png` and explain how your exploit works in `exercise_1_c.txt`.

(d) In `exercise_1_d.txt`, describe how to fix the vulnerability. Disallowing certain characters is not the right answer, since it unnecessarily limits what users can write


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.
https://xkcd.com/327/ by Randall Munroe licensed under CC BY-NC 2.5

Exercise 2 *Web Server* (2 + 2 Points)

Businessman Jefferson Bozo is using a database to store the sales of his online shopping platform called “Harz” (traditionally named after a forest, as is common for online shopping platforms). He now wants to display that data in his web browser.



Harz database schema.

In `exercise_2.py`, implement a web server using Flask and Psycopg. Start a PostgreSQL database, set the appropriate environment variables and then run `test_populate_database.py` to fill the database with data before you run the tests. This might take a minute. Unlike in the previous assignment, the tests will start the web server with `gunicorn` instead of Flask's default development server.

Your server should support the following functionality:

(a) The first n rows of the requested table should be displayed when visiting a URL in the form of:

`http://127.0.0.1:5000/table/<tablename>/<n>`

You can find a few examples on how those tables could look like on page 3. It is fine if your tables show the same data and look a bit different. This is not web design course after all.

(b) Under the following URL, the aggregated revenues for each combination of year and state should be shown.

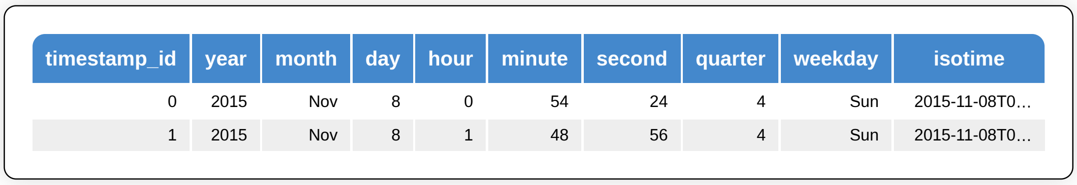
`http://127.0.0.1:5000/year-state`

You can find an example on page 4. You may use a little bit of Python to format the result, but all arithmetic operations should be done with SQL.

SQL and XSS injection vulnerabilities will cost you 2 points each.

(a)

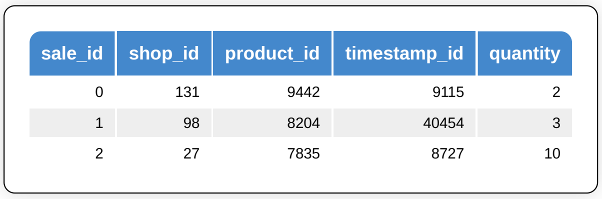
timestamp



timestamp_id	year	month	day	hour	minute	second	quarter	weekday	isotime
0	2015	Nov	8	0	54	24	4	Sun	2015-11-08T0…
1	2015	Nov	8	1	48	56	4	Sun	2015-11-08T0…

`http://127.0.0.1:5000/table/timestamp/2`

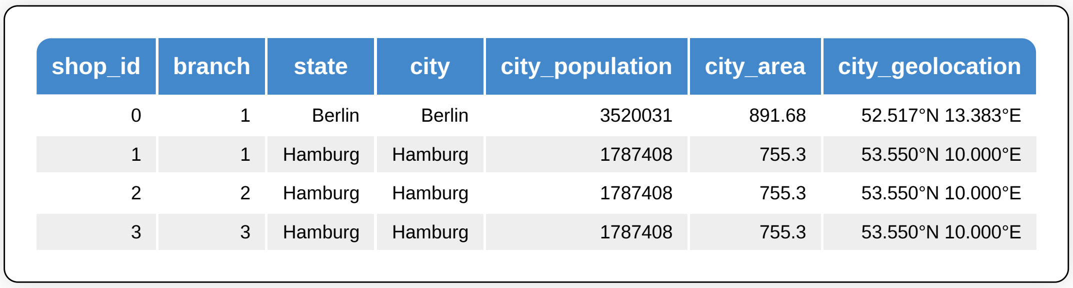
sale



sale_id	shop_id	product_id	timestamp_id	quantity
0	131	9442	9115	2
1	98	8204	40454	3
2	27	7835	8727	10

`http://127.0.0.1:5000/table/sale/3`

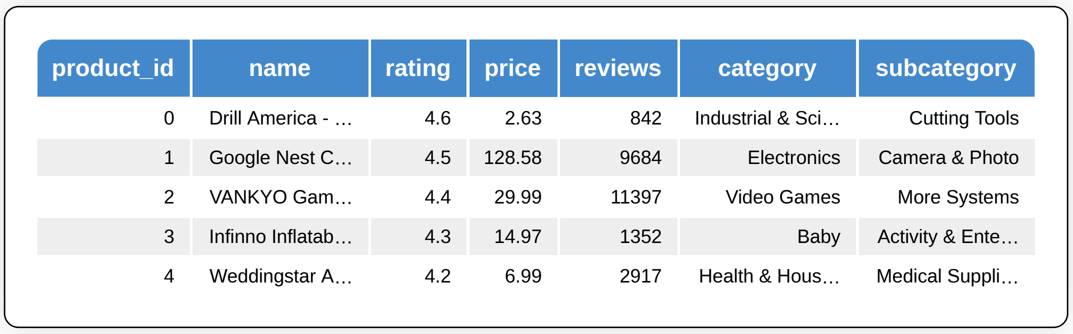
shop



shop_id	branch	state	city	city_population	city_area	city_geolocation
0	1	Berlin	Berlin	3520031	891.68	52.517°N 13.383°E
1	1	Hamburg	Hamburg	1787408	755.3	53.550°N 10.000°E
2	2	Hamburg	Hamburg	1787408	755.3	53.550°N 10.000°E
3	3	Hamburg	Hamburg	1787408	755.3	53.550°N 10.000°E

`http://127.0.0.1:5000/table/shop/4`

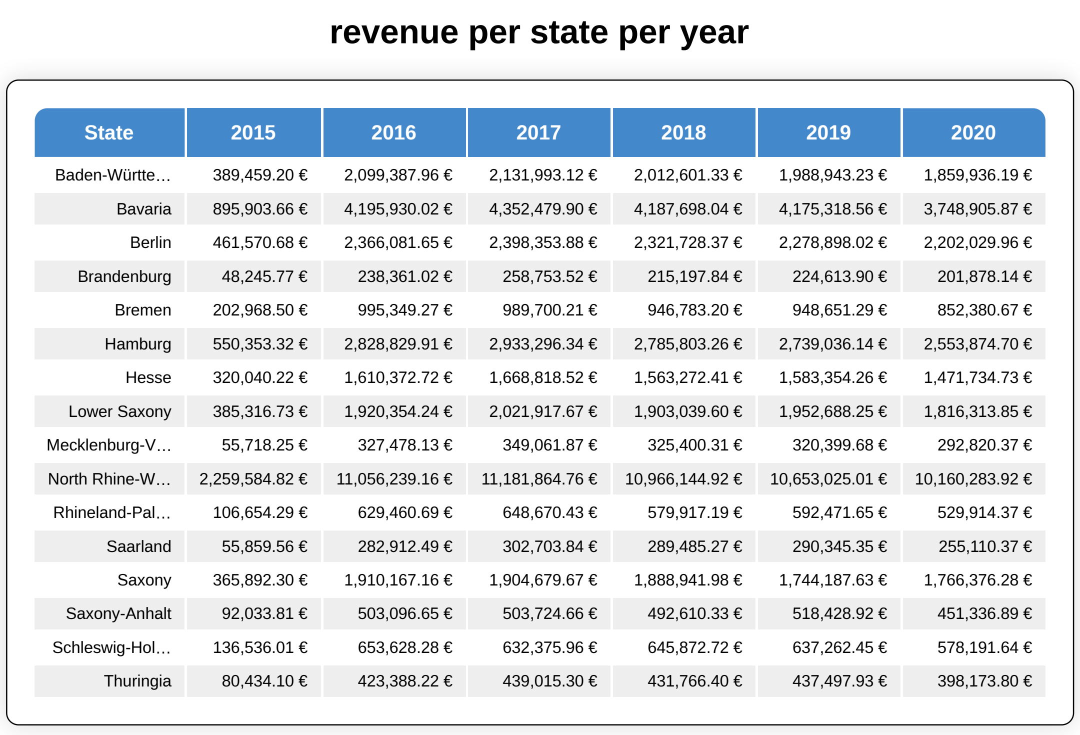
product



product_id	name	rating	price	reviews	category	subcategory
0	Drill America - …	4.6	2.63	842	Industrial & Sci…	Cutting Tools
1	Google Nest C…	4.5	128.58	9684	Electronics	Camera & Photo
2	VANKYO Gam…	4.4	29.99	11397	Video Games	More Systems
3	Infinno Inflatab…	4.3	14.97	1352	Baby	Activity & Ente…
4	Weddingstar A…	4.2	6.99	2917	Health & Hous…	Medical Suppli…

`http://127.0.0.1:5000/table/product/5`

(b)



revenue per state per year

State	2015	2016	2017	2018	2019	2020
Baden-Württe…	389,459.20 €	2,099,387.96 €	2,131,993.12 €	2,012,601.33 €	1,988,943.23 €	1,859,936.19 €
Bavaria	895,903.66 €	4,195,930.02 €	4,352,479.90 €	4,187,698.04 €	4,175,318.56 €	3,748,905.87 €
Berlin	461,570.68 €	2,366,081.65 €	2,398,353.88 €	2,321,728.37 €	2,278,898.02 €	2,202,029.96 €
Brandenburg	48,245.77 €	238,361.02 €	258,753.52 €	215,197.84 €	224,613.90 €	201,878.14 €
Bremen	202,968.50 €	995,349.27 €	989,700.21 €	946,783.20 €	948,651.29 €	852,380.67 €
Hamburg	550,353.32 €	2,828,829.91 €	2,933,296.34 €	2,785,803.26 €	2,739,036.14 €	2,553,874.70 €
Hesse	320,040.22 €	1,610,372.72 €	1,668,818.52 €	1,563,272.41 €	1,583,354.26 €	1,471,734.73 €
Lower Saxony	385,316.73 €	1,920,354.24 €	2,021,917.67 €	1,903,039.60 €	1,952,688.25 €	1,816,313.85 €
Mecklenburg-V…	55,718.25 €	327,478.13 €	349,061.87 €	325,400.31 €	320,399.68 €	292,820.37 €
North Rhine-W…	2,259,584.82 €	11,056,239.16 €	11,181,864.76 €	10,966,144.92 €	10,653,025.01 €	10,160,283.92 €
Rhineland-Pal…	106,654.29 €	629,460.69 €	648,670.43 €	579,917.19 €	592,471.65 €	529,914.37 €
Saarland	55,859.56 €	282,912.49 €	302,703.84 €	289,485.27 €	290,345.35 €	255,110.37 €
Saxony	365,892.30 €	1,910,167.16 €	1,904,679.67 €	1,888,941.98 €	1,744,187.63 €	1,766,376.28 €
Saxony-Anhalt	92,033.81 €	503,096.65 €	503,724.66 €	492,610.33 €	518,428.92 €	451,336.89 €
Schleswig-Hol…	136,536.01 €	653,628.28 €	632,375.96 €	645,872.72 €	637,262.45 €	578,191.64 €
Thuringia	80,434.10 €	423,388.22 €	439,015.30 €	431,766.40 €	437,497.93 €	398,173.80 €

`http://127.0.0.1:5000/year-state`