Aryaman Babber
CS 301

Sql Project Report

QUESTION 2

Question 1

id_Number	city	latitude	longitude	age
16024200205307139	Gatka	30.739949	68.010362	18
44249200704305798	Gatka	25.785895	55.07127	13
55271198705043506	Gatka	33.88641	68.995303	33
68526192306294851	Gatka	44.371403	68.002847	97
66731200711072335	Gatka	32.570533	51.436001	13
82900200301287423	Gatka	32.173595	64.251142	17
12191194012265053	Gatka	26.73436	52.028566	80
31125194110233960	Gatka	31.699652	56.750231	79
43617195312302408	Gatka	40.616965	58.893848	67
73816193612114485	Gatka	20.769137	58.556864	84
73647196802132510	Gatka	42.491744	63.810624	52
22762199109017707	Gatka	42.42619	53.150076	29
2895219520913161l	Gatka	23.574158	51.655536	68
13885193010114585	Gatka	23.294542	61.609247	90
40221197901308548	Gatka	43.323244	57.940717	41
53784196008266031	Gatka	33.637394	63.530661	60
82435198702237271	Gatka	28.355252	54.254846	33
73397196303245461	Gatka	34.138144	58.983797	57
83561193711051890	Gatka	33.073447	60.67555	83
10128195108299770	Gatka	22.694073	68.414813	69
92207195204237081	Gatka	48.952213	58.706606	68
94052194310286552	Gatka	31.51662	50.950716	77
63174198901161954	Gatka	47.824754	54.998942	31
50979194407205938	Gatka	38.796979	65.520042	76
90651196703087287	Gatka	49.899931	53.223468	53
92752196208144076	Gatka	39.913075	62.044917	58
32320200610182156	Gatka	34.254198	61.57343	14
80158193304253332	Gatka	40.393454	68.596821	87
55064200412132335	Gatka	30.993385	60.686369	16
77292201601056219	Gatka	28.544424	67.100134	4
76742202002283842	Gatka	37.64	57.232226	0
57455200501159807	Gatka	37.23079	61.300373	15
28389197002049699	Gatka	20.366805	65.388437	50
57957198606169849	Gatka	29.198979	65.043608	34
60812196712142078	Gatka	40.783897	60.556095	53
48451196102074549	Gatka	35.167073	56.420798	59
94663201812053351	Gatka	35.790828	58.431253	2
67809200805032979	Gatka	35.481906	69.44036	12

I first added an int variable called “age” to the census table. Then, I updated the variable “age” to equal 2020 minus the substring of the id number starting at index 6, of length 4.

Question 2

id_Number	city	latitude	longitude	age	bmi
10000196802294053	Georgopol	23.41177	102.008813	52	41.279
10003199201278064	Quarry	47.350345	27.078591	28	18.124
10027193608073352	Stalber	117.170615	124.095724	84	18.643
10030198301247150	Pochinki	67.281503	57.600325	37	19.918
10044197303055782	Mylta	110.588872	54.615626	47	26.605
10044199401112091	Zharki	32.883722	123.100291	26	25.773
10047197606034833	Pochinki	64.536014	65.41443	44	NULL
10048198004082379	Rozhok	74.350242	72.542774	40	37.877
10049196807229654	Sosnovka	61.752682	20.441515	52	12.591
10049201311171882	Gatka	45.835306	67.401906	7	31.634
10053197903285415	Zharki	32.284369	120.752658	41	NULL
10056193208168183	Sosnovka	79.682199	29.508383	88	NULL
10056201403156171	Novorep…	113.723677	16.073736	6	23.81
10058194410318103	Yasnaya	100.065575	94.760358	76	22.515
10058196701136852	Sosnovka	95.394466	5.963522	53	31.554
10062199212178190	Severny	79.924443	116.178299	28	15.432
10063196801273093	Severny	72.633226	129.251605	52	30.116
10064201309191745	Stalber	114.690916	134.449753	7	32.489
10066196606022032	Novorep…	129.898949	28.788841	54	33.51
10068194312215794	Mylta	121.065788	34.752327	77	15.95
10069194909077899	Zharki	25.902246	112.643894	71	33.36
10072194411174451	Yasnaya	113.5132	96.774822	76	30.586
10079196805246645	Stalber	114.4181	111.655922	52	27.115
10082198806204502	Stalber	100.425839	121.957957	32	27.127
10086199301019511	Sosnovka	94.078092	20.515058	27	15.37
10087200008044904	Yasnaya	119.890139	93.381211	20	15.255
10091201901094324	Novorep…	128.117718	27.009377	1	45.523
10096193108266083	Mylta	119.492793	53.314404	89	NULL
10096201306233339	Quarry	31.216872	32.374405	7	30.701
10101200012298295	Sosnovka	91.089294	26.609842	20	NULL
10108195801264053	Novorep…	111.707681	17.863966	62	24.994
10109201608289559	Quarry	40.130484	32.391732	4	13.445
10113198901073920	Kameshki	129.957836	106.716467	31	24.61
10114196610224585	Severny	87.269882	100.746887	54	16.62
10122194110206313	Kameshki	124.081526	117.529343	79	NULL
10128195108299770	Gatka	22.694073	68.414813	69	28.569
10130195403106078	Gatka	34.730697	60.957688	66	30.102
10147195810014427	Gatka	31.294889	64.274773	62	31.673

I added a float variable “bmi” to census.
I created a temp table where I left joined census and patient_info, with all the attributes from census, and the attributes height_cm and weight_kg from patient_info.
Then, I updated bmi in the temp table using the equation given in the guidelines.
I dropped the height_cm and weight_kg columns in the temp table, and dropped the original census table. I then created a new census table to replace the old one, which was equal to the temp table.

Question 3

id_Number	city	latitude	longitude	age	bmi	score
10082198806204502	Stalber	100.425839	121.957957	32	27.127	89.0112
10189193408204896	Novorepnoye	121.506277	28.353089	86	14.945	66.7172
10191193703022620	Yasnaya	107.01696	83.38315	83	30.422	66.4542
10232196901069519	Mylta	118.484584	58.861882	51	13.127	68.1992
10253195505249705	Severny	98.165108	118.622572	65	43.98	61.298
10254200105202453	Pochinki	73.514157	46.218303	19	18.544	86.1321
10264193712308570	Stalber	111.559918	134.151206	83	23.801	64.4497
10308199401166943	Georgopol	44.163273	102.206695	26	18.102	88.1736
10316198706087928	Primorsk	29.145734	23.100697	33	16.075	87.1745
10319201008255893	Lipovka	130.417667	82.303626	10	15.899	78.3792
10321195903136191	Stalber	100.038404	130.7281	61	29.037	79.4829
10342198501019513	Georgopol	36.968354	72.132293	35	22.204	93.5301
10453201303028336	Gatka	22.609959	64.532879	7	30.506	71.9723
10477197108084314	Mylta	111.946917	48.784558	49	43.409	62.5619
10486196901227051	Lipovka	132.824929	80.620879	51	24.997	88.0675
10547194307156189	Gatka	41.603517	61.7602	77	15.403	65.127
10553196106227158	Quarry	35.769376	26.979942	59	15.274	72.1377
10608194202136359	Kameshki	136.615535	119.078671	78	30.793	66.3632
10638194803243643	Georgopol	56.880371	94.224623	72	18.904	62.8108
10642195304159559	Zharki	37.21283	121.339939	67	36.513	57.5407
10703195805201638	Gatka	24.864417	64.32697	62	26.025	74.3601
10717200801173036	Georgopol	52.680669	80.978212	12	20.444	91.922
10736193403312032	Mylta	102.620492	59.404604	86	23.039	67.5518
10748193905037588	Severny	72.463563	129.86951	81	31.604	61.1448
10754194801169815	Primorsk	22.0294	22.81276	72	33.834	63.1972
10767195405063695	Pochinki	65.207341	46.821503	66	36.111	57.027
10777197812167334	Yasnaya	99.000142	86.200338	42	24.99	78.5019
10779196410233203	Rozhok	84.55996	85.450599	56	42.72	64.9923
10787196006131968	Pochinki	84.930609	66.49846	60	22.352	78.5557
10803198907179664	Lipovka	142.383632	83.726918	31	33.498	76.8815
10853201702267753	Yasnaya	102.105612	83.468877	3	11.455	80.1468
10858199809041983	Lipovka	147.618757	74.568584	22	30.524	79.5164
10861196103277506	Mylta	117.9207	42.474182	59	31.774	77.7616
10899201106013728	Sosnovka	69.459563	22.096537	9	15.873	73.7467
10999199002168457	Mylta	128.617818	41.395394	30	26.581	86.3705

id_Number	city	latitude	longitude	age	bmi	score
50728198802134505	Rozhok	74.852635	82.648446	32	NULL	73.4755
52238193605231657	Georgopol	22.931692	73.750901	84	NULL	73.4755
52637201211024355	Sosnovka	84.938515	13.600833	8	NULL	73.4755
53319199205294254	Gatka	36.191122	58.833878	28	NULL	73.4755
53846199007315313	Georgopol	55.912889	81.626718	30	NULL	73.4755
54138193002066540	Mylta	112.634621	36.47229	90	NULL	73.4755
54591201209059877	Gatka	45.067898	65.483691	8	NULL	73.4755
54724199011286704	Lipovka	140.452654	62.651782	30	NULL	73.4755
54949198305162002	Lipovka	147.998315	68.11706	37	NULL	73.4755
55405200005217017	Pochinki	76.533085	68.925683	20	NULL	73.4755
55496193008181888	Novorepnoye	124.204515	26.837034	90	NULL	73.4755
55796197709173135	Novorepnoye	116.495422	18.886505	43	NULL	73.4755
56059198804126390	Lipovka	143.730648	61.922285	32	NULL	73.4755
56248201204289956	Rozhok	60.882736	83.384905	8	NULL	73.4755
56776195901078949	Zharki	10.594603	121.452821	61	NULL	73.4755
56878195704153906	Sosnovka	97.484138	3.393281	63	NULL	73.4755
57077199207096871	Georgopol	40.535771	73.257051	28	NULL	73.4755
57222200709304687	Quarry	20.012446	36.767086	13	NULL	73.4755
57223197402228373	Stalber	117.214855	137.331533	46	NULL	73.4755
57385197109108032	Yasnaya	109.203038	87.200897	49	NULL	73.4755
57607198307107478	Georgopol	29.851728	91.282391	37	NULL	73.4755
57682193803112317	Sosnovka	98.709784	24.315927	82	NULL	73.4755
58480200406171598	Yasnaya	92.156739	99.369096	16	NULL	73.4755
59924195705035417	Kameshki	124.064311	121.646627	63	NULL	73.4755
60048201304124980	Kameshki	137.923027	124.536888	7	NULL	73.4755
60589194401269474	Lipovka	141.428941	66.634901	76	NULL	73.4755
61309197501301010	Rozhok	62.095426	72.55942	45	NULL	73.4755
61601200603103737	Gatka	34.822707	51.617994	14	NULL	73.4755
63057192903162142	Kameshki	132.207708	124.198499	91	NULL	73.4755
63087199306105058	Primorsk	20.635039	14.304805	27	NULL	73.4755
64005201608294910	Severny	63.745901	129.153203	4	NULL	73.4755
64477196909034736	Pochinki	80.699113	41.127693	51	NULL	73.4755
65084194201233893	Gatka	26.805105	67.263156	78	NULL	73.4755
65778195612058705	Georgopol	45.602191	83.665664	64	NULL	73.4755
66579197209151477	Quarry	37.00283	30.553637	48	NULL	73.4755

We create a temp_census table that left joins all of census to the needed attributes from patients_physical_exams.
We create temp_2 table that inner joins all of temp_census to the max(exam_date) of each id_Number in temp_census. Now, temp_2 has chosen the most recent exam dates for each id_Number (each person).
After the last step, there are 24 duplicates remaining, from people who took a physical exam twice on the same day. We create temp_3 table by inner joining temp_2 and max(metabolism). This gets rid of the 24 duplicates, and chooses the exam where the patient had a higher metabolism rate.
We create and set a fitness_score variable to temp_3 using the equation in the guidelines. We drop temp_census. We create a new temp_census where we left join census with temp_3. Now, temp_census contains all attributes from census, with a fitness score of everyone who has taken a physical exam.
We update fitness_score where fitness_score equals Null to the average score from temp_3. Now, all of temp_census has a fitness score, either from their physical exam, or the average fitness score. We see that in tables 1 and 2 shown above.

Question 4

Name	id_Number	row_Number	seat_Number	auditorium_Numb...	score	test_result
William Pugliares	49432193611075033	8	36	4	67.9829	False
Zada Romeo	88165201602123657	62	42	3	73.9469	True
Ada Robey	93136196911109053	59	26	3	81.5715	False
Adam Santillan	19630198310151209	34	26	3	77.2453	False
Addie Pfost	43387196212059489	18	13	5	65.0174	False
Adele Newsome	94516196507108821	25	26	1	68.6594	True
Albert Borrego	71877202005268066	39	24	6	83.6807	NULL
Alejandro Ortega	28428198812138627	29	47	3	81.1709	False
Alexander Wagner	79062197508308270	53	8	2	82.0229	False
Alicia Coke	17136197410157102	29	56	5	77.6427	True
Alison Davis	56250195006232771	22	47	4	71.8847	False
Allison Picard	28135195611115643	59	20	2	76.0074	False
Alphonse Sturges	93796194403111352	32	23	4	61.4176	False
Alvin Skimehorn	44198196810031436	59	20	3	62.3086	True
Amanda Jimenez	95133200408156003	24	18	6	76.097	False
Amber Hammond	91682195203255473	43	28	1	61.0785	False
Amelia Bermudez	72693199806015718	12	56	6	77.4992	False
Amy Crum	38838201504141604	13	1	4	83.778	False
Amy Poindexter	10779196410233203	20	57	2	64.9923	NULL
Ana Bonin	41641193108263960	24	58	2	61.34	False
Andrew Bagshaw	65320201804279107	25	22	4	73.9417	False
Angel Hawthorne	98350194811058966	33	49	2	74.9309	False
Angelita Linkovich	10999199002168457	13	27	3	86.3705	False
Anibal Denison	91473195903192519	43	38	3	79.4334	False
Ann Bonner	38814199910266113	49	14	3	78.3287	False
Annie Lesperance	49349199006082878	5	54	6	82.5108	False
Anthony Hundley	29485195105217255	50	40	1	60.3837	False
Arthur Loney	91370197909216423	7	1	1	68.0948	NULL
Arthur Scharich	10608194202136359	10	30	3	66.3632	NULL
Ashlee Hazen	30291194802231568	19	16	3	66.4044	False
Audrey Blakely	31463196704291159	2	25	6	68.1372	True
Barbara Miller	25964200803224587	10	9	4	85.0962	NULL
Barbara Wyatt	29426200106277626	30	10	6	77.4993	False
Benjamin Calais	94077199508022023	25	3	3	80.1185	False
Bernadine Beal	34871195404021691	38	42	5	76.21	False
Bertha Levels	12143198710063133	44	43	6	72.5118	True
Bertha Samantha	44949195408034742	27	13	2	72.333	NULL
Betty Alvarez	17919198408291523	4	38	3	93.3375	False

We create a temp_1 table that left joins the census fitness score to all of audiences.
We create a temp_2 table that left joins all of temp_1 and the zovid12test test_result.
We drop audiences, and create a new audiences set to temp_2.

QUESTION 3

Question 1

```
FAILED TO INFECT ALL. LAST INFECTED TIME:  41
FAILED TO INFECT ALL. LAST INFECTED TIME:  71
FAILED TO INFECT ALL. LAST INFECTED TIME:  35
FAILED TO INFECT ALL. LAST INFECTED TIME:  34
FAILED TO INFECT ALL. LAST INFECTED TIME:  39
FAILED TO INFECT ALL. LAST INFECTED TIME:  55
```

Question 2

id_Number	Name	ExposureTime
66053200803043147	Shiela Cardello	70
56248201204289956	Rosa Fite	70
55887198004123682	Douglas Snelling	70
46712201208108443	Mark Morales	67
73408199201225701	Charles Pigeon	67
...	...	...
37116193309082648	Reba Boland	2
87163193004201477	Melvin Kobayashi	2
97534193406175172	James Haase	2
39862194209173962	Scott Galbo	2
84658193401264079	Kathryn Robinson	2

15945 rows × 2 columns

Integration Explanation

First, it is important to note that I used Jupyter Notebook (Python 3 notebook) for this project.

Using pip install, I downloaded the sqlalchemy library, from which I could import sqlalchemy.engine.

```
# Imports sqlalchemy for sql access
from sqlalchemy.engine import create_engine
```

I then pip installed my_sql_ext for Python, which allows for sql magic in a Jupyter notebook. Using this, I loaded the sql extension into the notebook. I then used a sql query to access the mysql database.

```
# ACCESSES MYSQL DATABASE
%load_ext sql
%sql mysql://root:testing@localhost/zovid
```

This connects my sql database to python for the rest of the notebook, so I can use sql queries later on.