Integrated Project: Maji Ndogo Part 1

- **Connecting to our MySQL database**

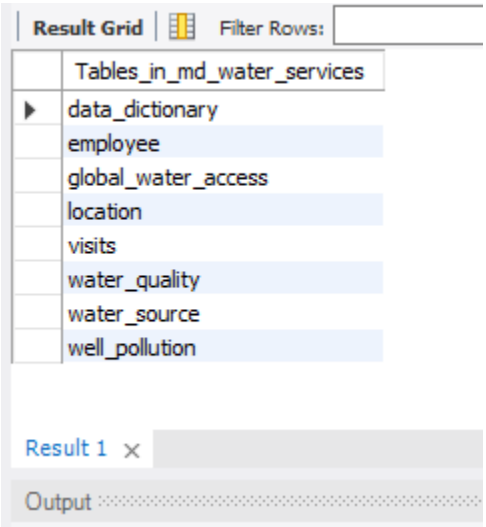
  Run the Database(md_water_services) on MySQL then run the query.

Using our database created in MySQL Workbench, we want to answer some questions on the range of our dataset. We can apply the same queries in MySQL Workbench and in this notebook if we connect to our MySQL server. Since we have a MySQL database, we can connect to it using mysql and pymysql.

Note

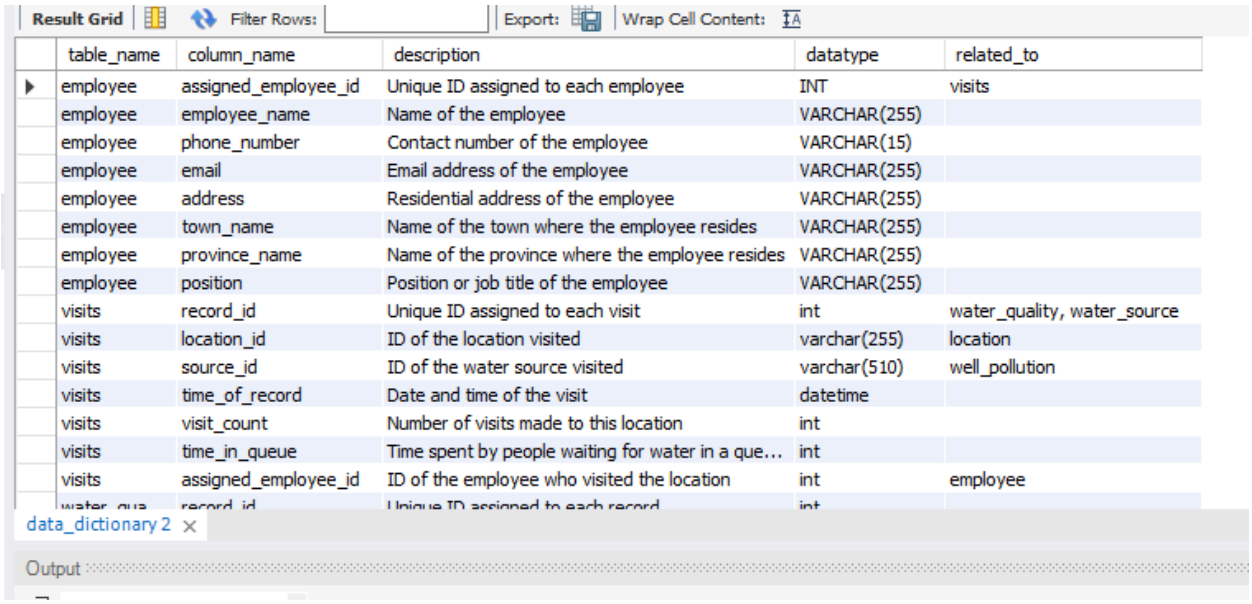
This will give you a list of all the tables in the database. SHOW TABLES - A data dictionary table has been embedded into the database. If you query the data_dictionary table, an explanation of each column is given there.

Show tables



SELECT * FORM

Data_dictionary;



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

table_name	column_name	description	datatype	related_to
employee	assigned_employee_id	Unique ID assigned to each employee	INT	visits
employee	employee_name	Name of the employee	VARCHAR(255)	
employee	phone_number	Contact number of the employee	VARCHAR(15)	
employee	email	Email address of the employee	VARCHAR(255)	
employee	address	Residential address of the employee	VARCHAR(255)	
employee	town_name	Name of the town where the employee resides	VARCHAR(255)	
employee	province_name	Name of the province where the employee resides	VARCHAR(255)	
employee	position	Position or job title of the employee	VARCHAR(255)	
visits	record_id	Unique ID assigned to each visit	int	water_quality, water_source
visits	location_id	ID of the location visited	varchar(255)	location
visits	source_id	ID of the water source visited	varchar(510)	well_pollution
visits	time_of_record	Date and time of the visit	datetime	
visits	visit_count	Number of visits made to this location	int	
visits	time_in_queue	Time spent by people waiting for water in a que...	int	
visits	assigned_employee_id	ID of the employee who visited the location	int	employee

data_dictionary 2 ×

Output

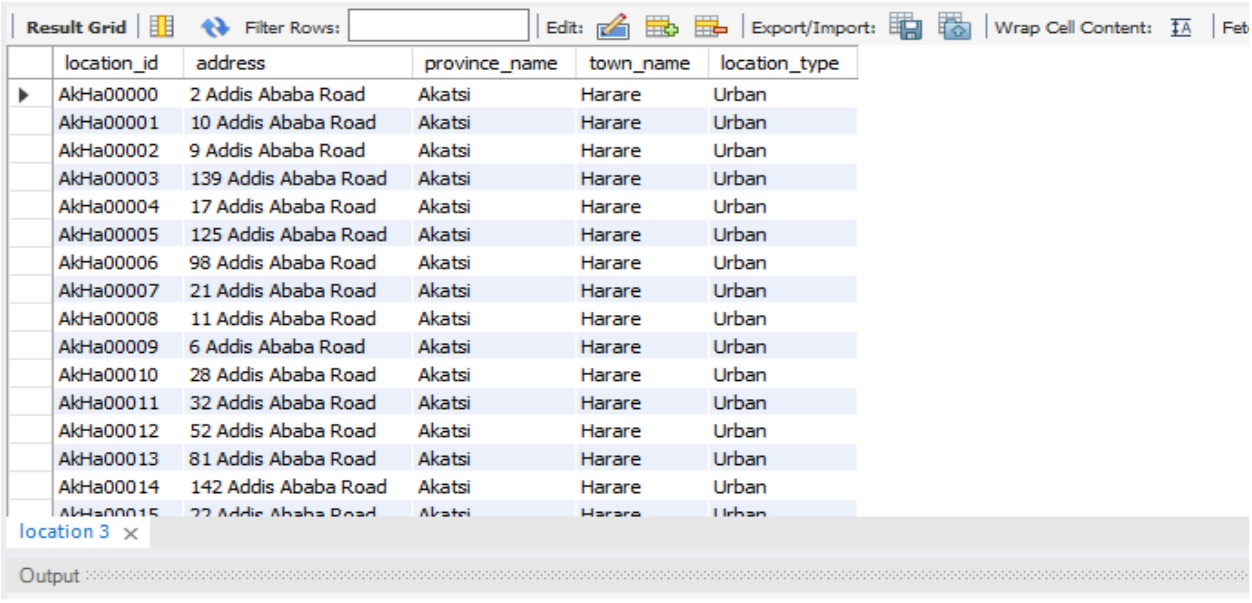
-- So let's have a look at one of these tables, Let's use location so we can use that killer query,

-- SELECT * but remember to limit it and tell. . PAGE(6)

SELECT * FROM

location

LIMIT 50;



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fet

location_id	address	province_name	town_name	location_type
AkHa00000	2 Addis Ababa Road	Akatsi	Harare	Urban
AkHa00001	10 Addis Ababa Road	Akatsi	Harare	Urban
AkHa00002	9 Addis Ababa Road	Akatsi	Harare	Urban
AkHa00003	139 Addis Ababa Road	Akatsi	Harare	Urban
AkHa00004	17 Addis Ababa Road	Akatsi	Harare	Urban
AkHa00005	125 Addis Ababa Road	Akatsi	Harare	Urban
AkHa00006	98 Addis Ababa Road	Akatsi	Harare	Urban
AkHa00007	21 Addis Ababa Road	Akatsi	Harare	Urban
AkHa00008	11 Addis Ababa Road	Akatsi	Harare	Urban
AkHa00009	6 Addis Ababa Road	Akatsi	Harare	Urban
AkHa00010	28 Addis Ababa Road	Akatsi	Harare	Urban
AkHa00011	32 Addis Ababa Road	Akatsi	Harare	Urban
AkHa00012	52 Addis Ababa Road	Akatsi	Harare	Urban
AkHa00013	81 Addis Ababa Road	Akatsi	Harare	Urban
AkHa00014	142 Addis Ababa Road	Akatsi	Harare	Urban
AkHa00015	22 Addis Ababa Road	Akatsi	Harare	Urban

location 3 ×

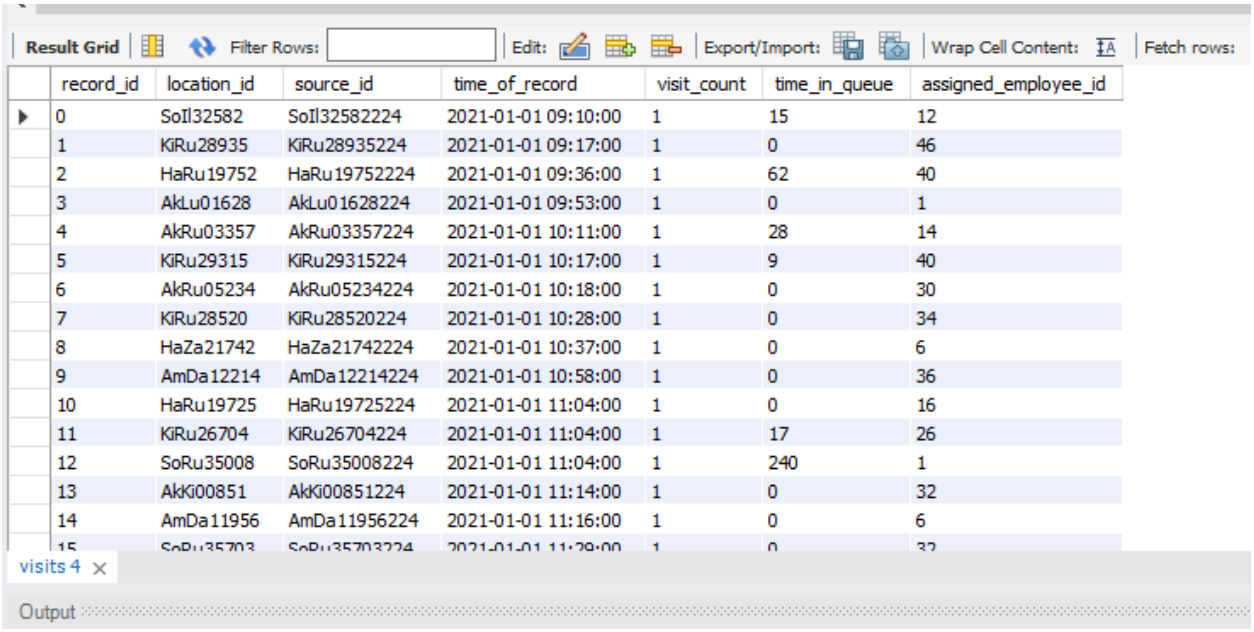
Output

Ok, so let's look **at** the visits **table**. PAGE(7)

SELECT * **FROM**

visits

LIMIT 50;



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows:

record_id	location_id	source_id	time_of_record	visit_count	time_in_queue	assigned_employee_id
0	SoIl32582	SoIl32582224	2021-01-01 09:10:00	1	15	12
1	KiRu28935	KiRu28935224	2021-01-01 09:17:00	1	0	46
2	HaRu19752	HaRu19752224	2021-01-01 09:36:00	1	62	40
3	AkLu01628	AkLu01628224	2021-01-01 09:53:00	1	0	1
4	AkRu03357	AkRu03357224	2021-01-01 10:11:00	1	28	14
5	KiRu29315	KiRu29315224	2021-01-01 10:17:00	1	9	40
6	AkRu05234	AkRu05234224	2021-01-01 10:18:00	1	0	30
7	KiRu28520	KiRu28520224	2021-01-01 10:28:00	1	0	34
8	HaZa21742	HaZa21742224	2021-01-01 10:37:00	1	0	6
9	AmDa12214	AmDa12214224	2021-01-01 10:58:00	1	0	36
10	HaRu19725	HaRu19725224	2021-01-01 11:04:00	1	0	16
11	KiRu26704	KiRu26704224	2021-01-01 11:04:00	1	17	26
12	SoRu35008	SoRu35008224	2021-01-01 11:04:00	1	240	1
13	AkKi00851	AkKi00851224	2021-01-01 11:14:00	1	0	32
14	AmDa11956	AmDa11956224	2021-01-01 11:16:00	1	0	6
15	SoRu35703	SoRu35703224	2021-01-01 11:29:00	1	0	32

visits 4 ×

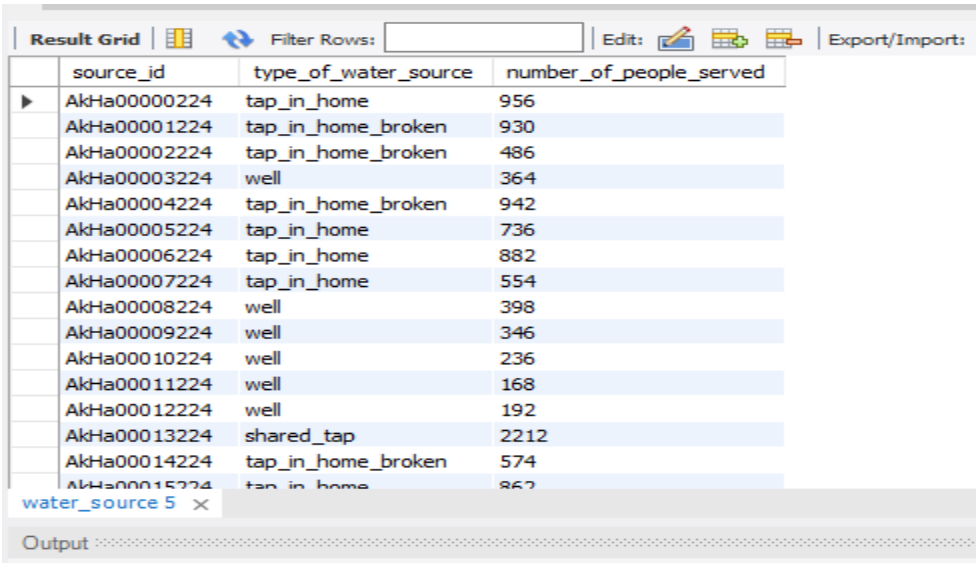
Output

Ok, so let's look at the water_source table to see what a '**source**' **is**. Normally "_id" columns **are** related **to** another **table**.

SELECT * **FROM**

water_source

LIMIT 50;



Result Grid | Filter Rows: | Edit: | Export/Import:

source_id	type_of_water_source	number_of_people_served
AkHa00000224	tap_in_home	956
AkHa00001224	tap_in_home_broken	930
AkHa00002224	tap_in_home_broken	486
AkHa00003224	well	364
AkHa00004224	tap_in_home_broken	942
AkHa00005224	tap_in_home	736
AkHa00006224	tap_in_home	882
AkHa00007224	tap_in_home	554
AkHa00008224	well	398
AkHa00009224	well	346
AkHa00010224	well	236
AkHa00011224	well	168
AkHa00012224	well	192
AkHa00013224	shared_tap	2212
AkHa00014224	tap_in_home_broken	574
AkHa00015224	tap_in_home	862

water_source 5 ×

Output

2. Dive **into** the water sources:

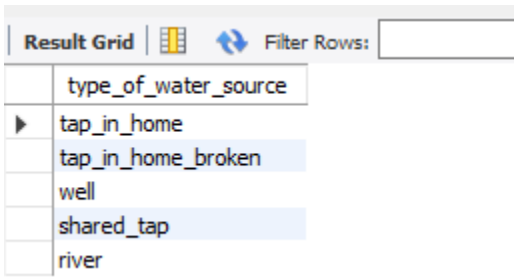
- Now that you're familiar with the structure of the tables, let's dive deeper. We need **to** understand the types **of** water sources
- We're dealing with it. Can you figure out which table contains this information**?**

SELECT DISTINCT

type_of_water_source

FROM

water_source;



So I get this when I run it: <<<refer to the PDF

- type_of_water_source 1- tap_in_home 2- tap_in_home_broken 3- well 4- shared_tap 5- river Let me quickly bring you up to speed on these water source types:

1. River - People collect drinking water along a river. This is an open water source that millions of people use in Maji Ndogo. Water from -- a river has a high risk of being contaminated with biological and other pollutants, so it is the worst source of water possible.
2. Well - These sources draw water from underground sources, and are commonly shared by communities. Since these are closed water -- sources, contamination is much less likely compared to a river. Unfortunately, due to the ageing infrastructure and the corruption of officials in the past, many of our wells are not clean.
3. Shared tap - This is a tap in a public area shared by communities
4. Tap in home - These are taps that are inside the homes of our citizens. On average about 6 people live together in Maji Ndogo, so -- each of these taps serves about 6 people.
5. Broken tap in home - These are taps that have been installed in a citizen's home, but the infrastructure connected to that tap is not -- functional. This can be due to burst pipes, broken pumps or water treatment plants that are not working.

- Write an SQL query that retrieves all records from this table where the time_in_queue is more than some crazy time,

may 500 min. How would it feel to queue 8 hours for water? PAGE(16)

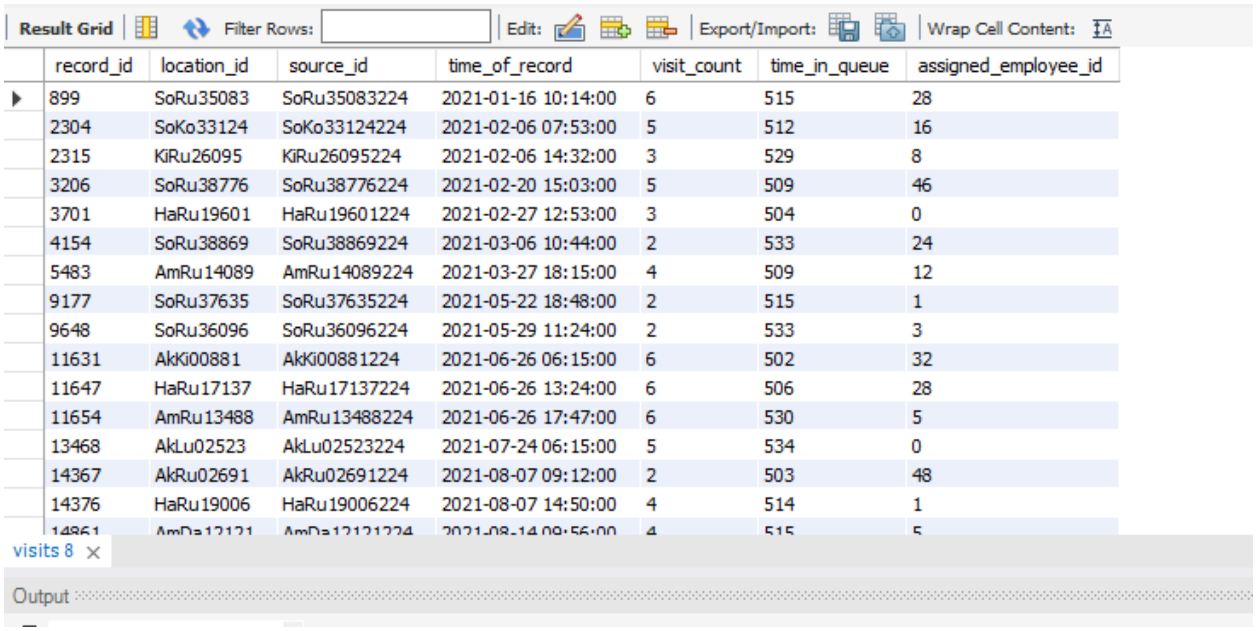
SELECT * FROM

visits

WHERE

time_in_queue > 500;

How is this possible? Can you imagine queueing 8 hours for water?



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

record_id	location_id	source_id	time_of_record	visit_count	time_in_queue	assigned_employee_id
899	SoRu35083	SoRu35083224	2021-01-16 10:14:00	6	515	28
2304	SoKo33124	SoKo33124224	2021-02-06 07:53:00	5	512	16
2315	KiRu26095	KiRu26095224	2021-02-06 14:32:00	3	529	8
3206	SoRu38776	SoRu38776224	2021-02-20 15:03:00	5	509	46
3701	HaRu19601	HaRu19601224	2021-02-27 12:53:00	3	504	0
4154	SoRu38869	SoRu38869224	2021-03-06 10:44:00	2	533	24
5483	AmRu14089	AmRu14089224	2021-03-27 18:15:00	4	509	12
9177	SoRu37635	SoRu37635224	2021-05-22 18:48:00	2	515	1
9648	SoRu36096	SoRu36096224	2021-05-29 11:24:00	2	533	3
11631	AkKi00881	AkKi00881224	2021-06-26 06:15:00	6	502	32
11647	HaRu17137	HaRu17137224	2021-06-26 13:24:00	6	506	28
11654	AmRu13488	AmRu13488224	2021-06-26 17:47:00	6	530	5
13468	AkLu02523	AkLu02523224	2021-07-24 06:15:00	5	534	0
14367	AkRu02691	AkRu02691224	2021-08-07 09:12:00	2	503	48
14376	HaRu19006	HaRu19006224	2021-08-07 14:50:00	4	514	1
14861	AmDa12121	AmDa12121224	2021-08-14 09:56:00	4	515	5

visits 8 ×

Output

I am wondering what type of water sources take this long to queue for.

- We will have to find that information in another table that lists
- the types of water sources. If I remember correctly, the table has type_of_water_source, and a source_id column.

So let's write down a couple of these source_id values from our results, and search for them in the other table. PAGE(17)

- AkKi00881224
- AkLu01628224
- AkRu05234224
- HaRu19601224
- HaZa21742224
- SoRu36096224
- SoRu37635224
- SoRu38776224

If we just select the first couple of records of the visits table without a WHERE filter,

we can see that some of these rows also have 0

mins queue time. So let's write down one or two of these too

select * from water_source

where

source_id = 'AkKi00881224'

OR

source_id = 'AkLu01628224'

OR

source_id = 'AkRu05234224'

OR

source_id = 'HaRu19601224'

OR

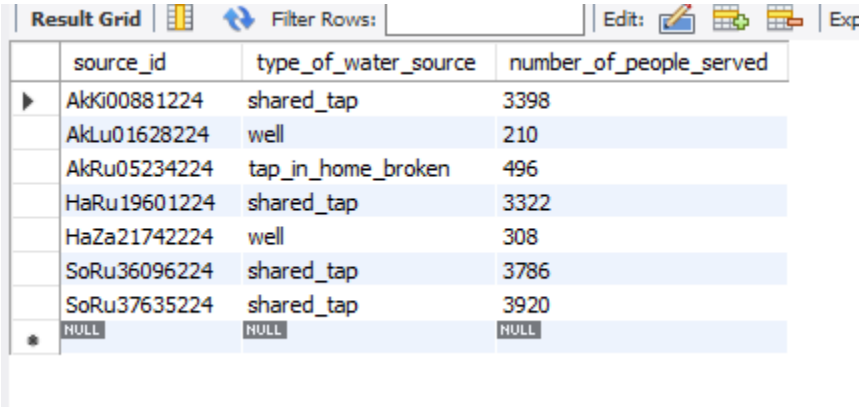
source_id = 'HaZa21742224'

OR

source_id = 'SoRu36096224'

OR

source_id = 'SoRu37635224';



Result Grid | Filter Rows: | Edit: | Exp

source_id	type_of_water_source	number_of_people_served
AkKi00881224	shared_tap	3398
AkLu01628224	well	210
AkRu05234224	tap_in_home_broken	496
HaRu19601224	shared_tap	3322
HaZa21742224	well	308
SoRu36096224	shared_tap	3786
SoRu37635224	shared_tap	3920
NULL	NULL	NULL

4. Assess the quality of water sources:

- The quality of our water sources is the whole point of this survey.
- We have a table that contains a quality score for each visit made
- about a water source that was assigned by a Field surveyor. They assigned a score to each source from 1,
- being terrible, to 10 for a
- good, clean water source in a home. Shared taps are not rated as high,
- and the score also depends on how long the queue times are.

Let's check if this is true.

The surveyors only made multiple visits to shared taps and did not revisit other types of water sources. So there should be no records of second visits to locations where there are good water sources, like taps in homes.

So I will write a query to find records where the subject_quality_score is 10

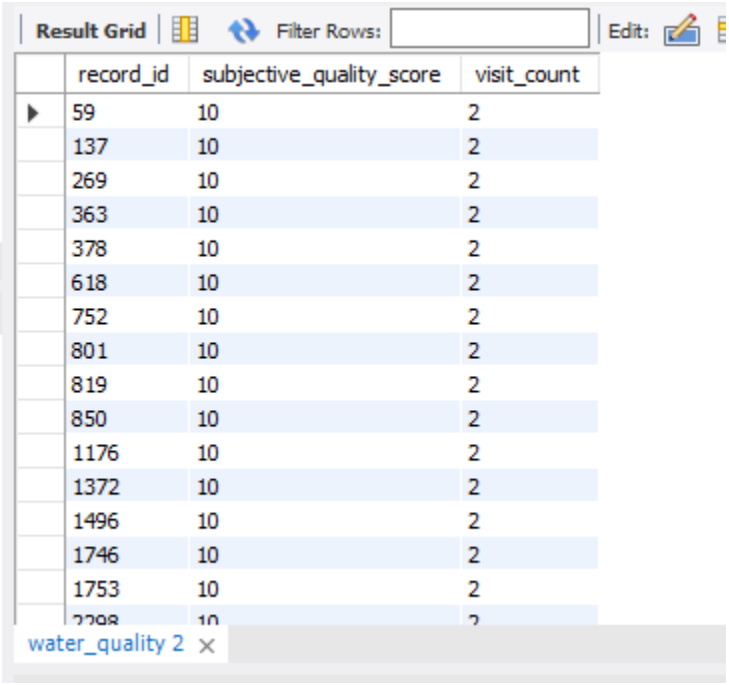
only looking for home taps -- and where the source

was visited a second time. What will this tell us? PAGE(18)

SELECT *FROM

water_quality

```
WHERE
  subjective_quality_score = 10
AND
  visit_count = 2
```



Result Grid | Filter Rows: | Edit:

record_id	subjective_quality_score	visit_count
59	10	2
137	10	2
269	10	2
363	10	2
378	10	2
618	10	2
752	10	2
801	10	2
819	10	2
850	10	2
1176	10	2
1372	10	2
1496	10	2
1746	10	2
1753	10	2
2298	10	2

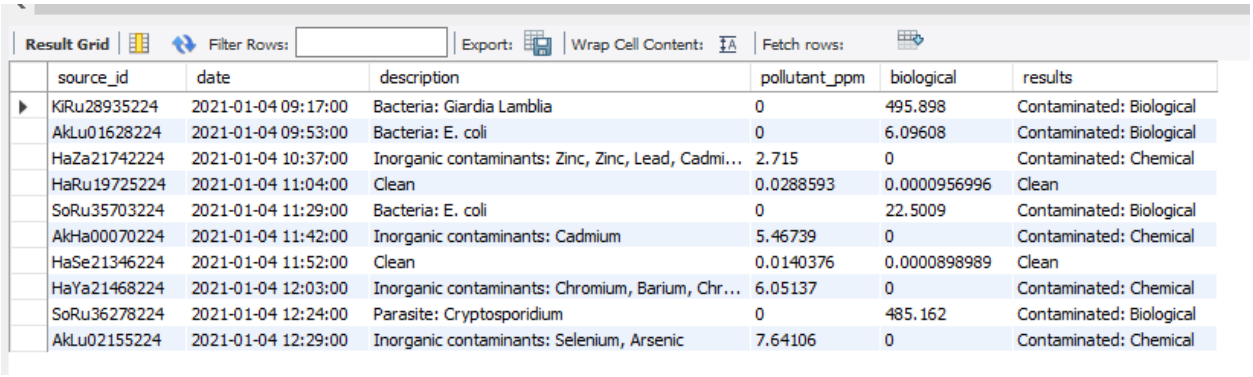
water_quality 2

- I get 218 rows of data. But this should not be happening!
- I think some of our employees may have made mistakes.
- To be honest, I'll be surprised if there are no errors in our data at this scale!
- I'm going to send Pres. Naledi a message that we have to recheck some of
- these sources. We can appoint an Auditor to check some of the data independently,
- and make sure we have the right information!

5. **Investigate pollution issues:**

1. Did you notice that we recorded contamination/pollution data for all of the well sources?
2. Find the right table and print the first few rows.
3. Find the right table and print the first few rows.

```
SELECT    * FROM
  well_pollution
LIMIT 10;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

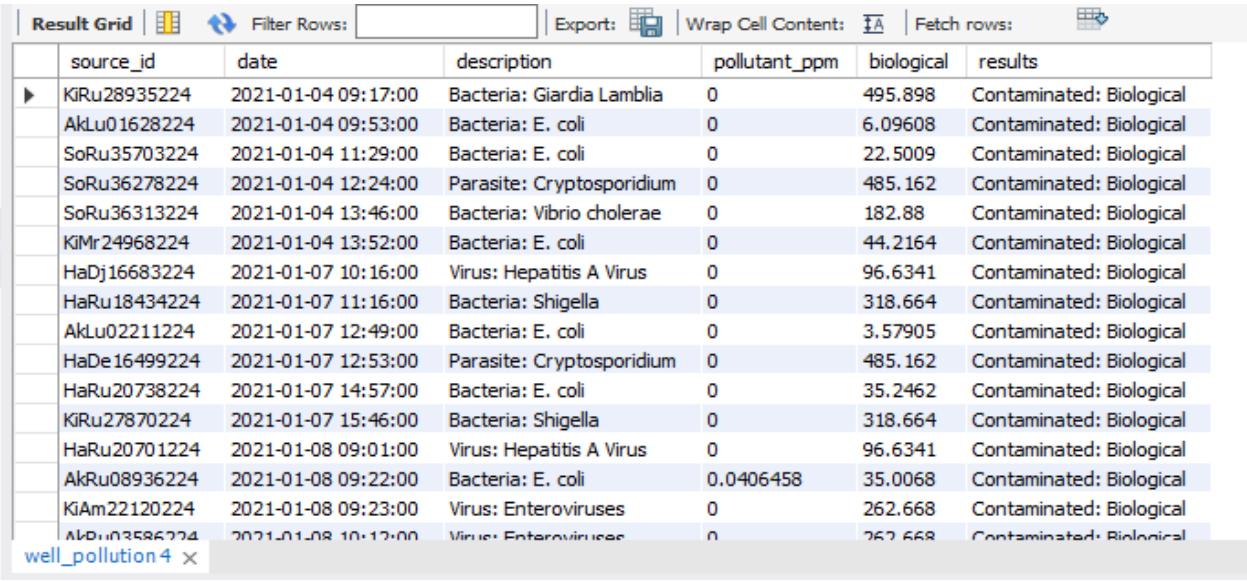
source_id	date	description	pollutant_ppm	biological	results
KiRu28935224	2021-01-04 09:17:00	Bacteria: Giardia Lamblia	0	495.898	Contaminated: Biological
AkLu01628224	2021-01-04 09:53:00	Bacteria: E. coli	0	6.09608	Contaminated: Biological
HaZa21742224	2021-01-04 10:37:00	Inorganic contaminants: Zinc, Zinc, Lead, Cadmi...	2.715	0	Contaminated: Chemical
HaRu19725224	2021-01-04 11:04:00	Clean	0.0288593	0.0000956996	Clean
SoRu35703224	2021-01-04 11:29:00	Bacteria: E. coli	0	22.5009	Contaminated: Biological
AkHa00070224	2021-01-04 11:42:00	Inorganic contaminants: Cadmium	5.46739	0	Contaminated: Chemical
HaSe21346224	2021-01-04 11:52:00	Clean	0.0140376	0.0000898989	Clean
HaYa21468224	2021-01-04 12:03:00	Inorganic contaminants: Chromium, Barium, Chr...	6.05137	0	Contaminated: Chemical
SoRu36278224	2021-01-04 12:24:00	Parasite: Cryptosporidium	0	485.162	Contaminated: Biological
AkLu02155224	2021-01-04 12:29:00	Inorganic contaminants: Selenium, Arsenic	7.64106	0	Contaminated: Chemical

Water Quality Integrity Check

It looks like our scientists diligently recorded the water quality of all the wells. Some are contaminated with biological contaminants, while others are polluted with an excess of heavy metals and other pollutants. Based on the results, each well was classified as Clean, Contaminated: Biological or Contaminated: Chemical. It is important to know this because wells that are polluted with bio or other contaminants are not safe to drink. It looks like they recorded the source_id of each test, so we can link it to a source, at some place in Maji Ndogo. In the well pollution table, the descriptions are notes taken by our scientists as text, so it will be challenging to process it. The biological column is in units of CFU/mL, so it measures how much contamination is in the water. 0 is clean, and anything more than 0.01 is contaminated. Let's check the integrity of the data. The worst case is if we have contamination, but we think we don't. People can get sick, so we need to make sure there are no errors here.

So, write a query that checks if the results is Clean but the biological column is > 0.01.

```
SELECT  * FROM
  well_pollution
WHERE
   biological > 0.01
LIMIT 50;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

source_id	date	description	pollutant_ppm	biological	results
KiRu28935224	2021-01-04 09:17:00	Bacteria: Giardia Lamblia	0	495.898	Contaminated: Biological
AkLu01628224	2021-01-04 09:53:00	Bacteria: E. coli	0	6.09608	Contaminated: Biological
SoRu35703224	2021-01-04 11:29:00	Bacteria: E. coli	0	22.5009	Contaminated: Biological
SoRu36278224	2021-01-04 12:24:00	Parasite: Cryptosporidium	0	485.162	Contaminated: Biological
SoRu36313224	2021-01-04 13:46:00	Bacteria: Vibrio cholerae	0	182.88	Contaminated: Biological
KiMr24968224	2021-01-04 13:52:00	Bacteria: E. coli	0	44.2164	Contaminated: Biological
HaDj16683224	2021-01-07 10:16:00	Virus: Hepatitis A Virus	0	96.6341	Contaminated: Biological
HaRu18434224	2021-01-07 11:16:00	Bacteria: Shigella	0	318.664	Contaminated: Biological
AkLu02211224	2021-01-07 12:49:00	Bacteria: E. coli	0	3.57905	Contaminated: Biological
HaDe16499224	2021-01-07 12:53:00	Parasite: Cryptosporidium	0	485.162	Contaminated: Biological
HaRu20738224	2021-01-07 14:57:00	Bacteria: E. coli	0	35.2462	Contaminated: Biological
KiRu27870224	2021-01-07 15:46:00	Bacteria: Shigella	0	318.664	Contaminated: Biological
HaRu20701224	2021-01-08 09:01:00	Virus: Hepatitis A Virus	0	96.6341	Contaminated: Biological
AkRu08936224	2021-01-08 09:22:00	Bacteria: E. coli	0.0406458	35.0068	Contaminated: Biological
KiAm22120224	2021-01-08 09:23:00	Virus: Enteroviruses	0	262.668	Contaminated: Biological
AkRu03586224	2021-01-08 10:12:00	Virus: Enteroviruses	0	262.668	Contaminated: Biological

well_pollution 4

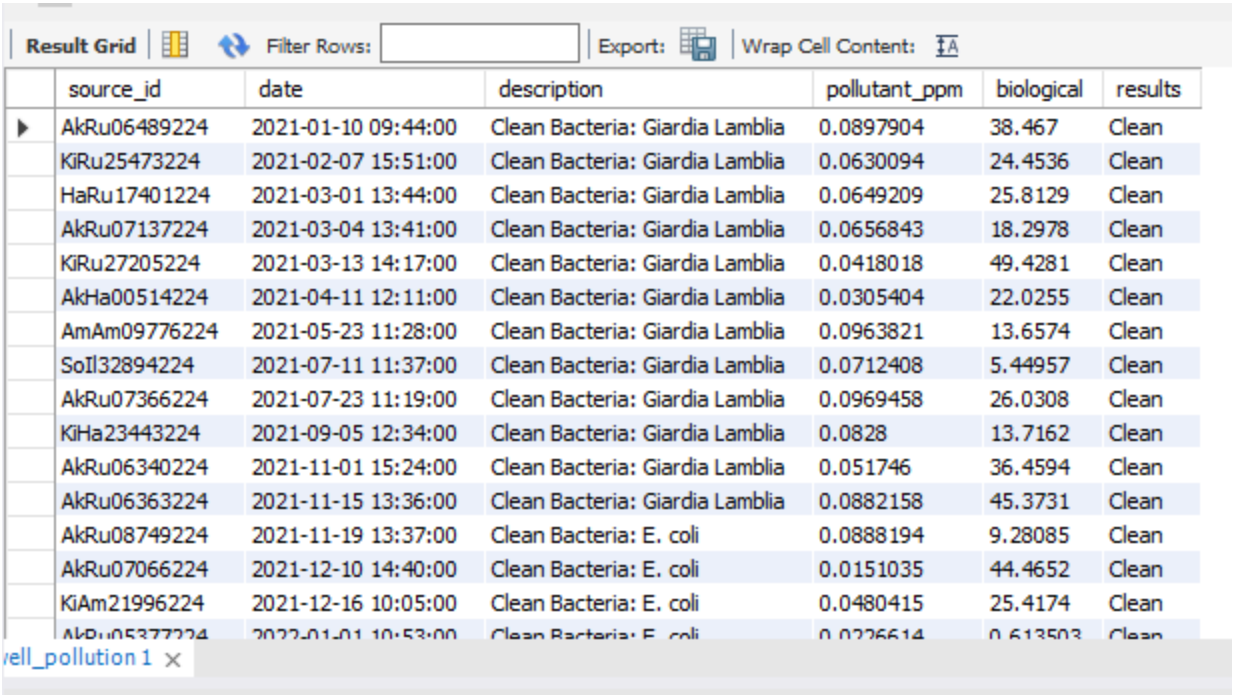
- If we compare the results of this query to the entire table,
- it seems like we have some inconsistencies in how the well statuses are
- recorded. Specifically, it seems that some data input personnel
- might have mistaken the description field for determining the cleanliness of the water.

Biological Contamination Data Integrity Analysis

1. It seems like, in some cases, if the description field begins with the word “Clean”, the results have been classified as “Clean” in the results column, even though the biological column is > 0.01
2. When we work with real-world data we may find inconsistencies due to data being misinterpreted based on a description rather than its actual values. Let’s dive deeper into the cause of the issue with the biological contamination data.
3. Vuyisile has told me that the descriptions should only have the word “Clean” if there is no biological contamination (and no chemical pollutants). Some data personnel must have copied the data from the scientist's notes into our database incorrectly. We need to find and remove the “Clean” part from all the descriptions that do have a biological contamination so this mistake is not made again.
4. The second issue has arisen from this error, but it is much more problematic. Some of the field surveyors have marked wells as Clean in the results column because the description had the word “Clean” in it, even though they have a biological contamination. So we need to find all the results that have a value greater than 0.01 in the biological column and have been set to Clean in the results column.
5. First, let's look at the descriptions. We need to identify the records that mistakenly have the word Clean in the description. However, it is important to remember that not all of our field surveyors used the description to set the results – some checked the actual data

- To find these descriptions, search for the word Clean with additional characters after it.
- As this is what separates incorrect descriptions from the records that should have "Clean".

```
SELECT  * FROM
  well_pollution
WHERE
  description LIKE 'Clean_%';
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

source_id	date	description	pollutant_ppm	biological	results
AkRu06489224	2021-01-10 09:44:00	Clean Bacteria: Giardia Lamblia	0.0897904	38.467	Clean
KiRu25473224	2021-02-07 15:51:00	Clean Bacteria: Giardia Lamblia	0.0630094	24.4536	Clean
HaRu17401224	2021-03-01 13:44:00	Clean Bacteria: Giardia Lamblia	0.0649209	25.8129	Clean
AkRu07137224	2021-03-04 13:41:00	Clean Bacteria: Giardia Lamblia	0.0656843	18.2978	Clean
KiRu27205224	2021-03-13 14:17:00	Clean Bacteria: Giardia Lamblia	0.0418018	49.4281	Clean
AkHa00514224	2021-04-11 12:11:00	Clean Bacteria: Giardia Lamblia	0.0305404	22.0255	Clean
AmAm09776224	2021-05-23 11:28:00	Clean Bacteria: Giardia Lamblia	0.0963821	13.6574	Clean
SoIl32894224	2021-07-11 11:37:00	Clean Bacteria: Giardia Lamblia	0.0712408	5.44957	Clean
AkRu07366224	2021-07-23 11:19:00	Clean Bacteria: Giardia Lamblia	0.0969458	26.0308	Clean
KiHa23443224	2021-09-05 12:34:00	Clean Bacteria: Giardia Lamblia	0.0828	13.7162	Clean
AkRu06340224	2021-11-01 15:24:00	Clean Bacteria: Giardia Lamblia	0.051746	36.4594	Clean
AkRu06363224	2021-11-15 13:36:00	Clean Bacteria: Giardia Lamblia	0.0882158	45.3731	Clean
AkRu08749224	2021-11-19 13:37:00	Clean Bacteria: E. coli	0.0888194	9.28085	Clean
AkRu07066224	2021-12-10 14:40:00	Clean Bacteria: E. coli	0.0151035	44.4652	Clean
KiAm21996224	2021-12-16 10:05:00	Clean Bacteria: E. coli	0.0480415	25.4174	Clean
AkRu05377224	2022-01-01 10:53:00	Clean Bacteria: E. coli	0.0226614	0.613503	Clean

ell_pollution 1 ×

The query should return 38 wrong descriptions.

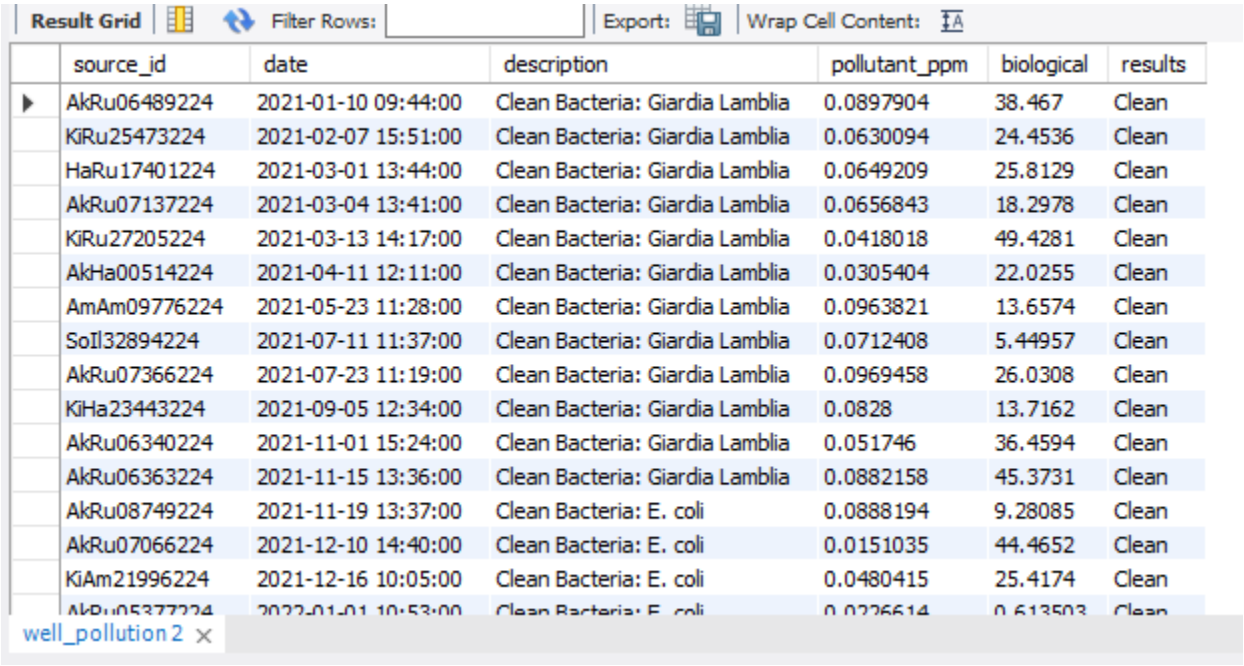
Now we need to fix these descriptions so that we don't encounter this issue again in the future.

Looking at the results we can see two different descriptions that we need to fix:

1. All records that mistakenly have Clean Bacteria: E. coli should updated to Bacteria: E. coli
2. All records that mistakenly have Clean Bacteria: Giardia Lamblia should updated to Bacteria: Giardia Lamblia

- The second issue we need to fix is in our results column.
- We need to update the results column from Clean to Contaminated: Biological
- Where the biological column has a value greater than 0.01.

```
SELECT * FROM
  Well_pollution
WHERE
   pollutant_ppm > 0.01
AND description LIKE 'Clean_%';
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

source_id	date	description	pollutant_ppm	biological	results
AkRu06489224	2021-01-10 09:44:00	Clean Bacteria: Giardia Lamblia	0.0897904	38.467	Clean
KiRu25473224	2021-02-07 15:51:00	Clean Bacteria: Giardia Lamblia	0.0630094	24.4536	Clean
HaRu17401224	2021-03-01 13:44:00	Clean Bacteria: Giardia Lamblia	0.0649209	25.8129	Clean
AkRu07137224	2021-03-04 13:41:00	Clean Bacteria: Giardia Lamblia	0.0656843	18.2978	Clean
KiRu27205224	2021-03-13 14:17:00	Clean Bacteria: Giardia Lamblia	0.0418018	49.4281	Clean
AkHa00514224	2021-04-11 12:11:00	Clean Bacteria: Giardia Lamblia	0.0305404	22.0255	Clean
AmAm09776224	2021-05-23 11:28:00	Clean Bacteria: Giardia Lamblia	0.0963821	13.6574	Clean
SoIl32894224	2021-07-11 11:37:00	Clean Bacteria: Giardia Lamblia	0.0712408	5.44957	Clean
AkRu07366224	2021-07-23 11:19:00	Clean Bacteria: Giardia Lamblia	0.0969458	26.0308	Clean
KiHa23443224	2021-09-05 12:34:00	Clean Bacteria: Giardia Lamblia	0.0828	13.7162	Clean
AkRu06340224	2021-11-01 15:24:00	Clean Bacteria: Giardia Lamblia	0.051746	36.4594	Clean
AkRu06363224	2021-11-15 13:36:00	Clean Bacteria: Giardia Lamblia	0.0882158	45.3731	Clean
AkRu08749224	2021-11-19 13:37:00	Clean Bacteria: E. coli	0.0888194	9.28085	Clean
AkRu07066224	2021-12-10 14:40:00	Clean Bacteria: E. coli	0.0151035	44.4652	Clean
KiAm21996224	2021-12-16 10:05:00	Clean Bacteria: E. coli	0.0480415	25.4174	Clean
AkRu05377224	2022-01-01 10:53:00	Clean Bacteria: E. coli	0.0226614	0.613503	Clean

well_pollution 2

- Case 1a: Update descriptions that mistakenly mention -- Clean Bacteria: E. coli to Bacteria: E. coli
- Case 1b: Update the descriptions that mistakenly mention -- Clean Bacteria: Giardia Lamblia to Bacteria: Giardia Lamblia
- Case 2: Update the result Contaminated: Biological Where --biological is greater than 0.01 plus current results isClean`

- **So if we now run our query:**

```
SET
  sql_safe_updates = 0;
UPDATE
    well_pollution
SET
  description = 'Bacteria: E. coli'
WHERE
  description = 'Clean Bacteria: E. coli' ;
UPDATE
```

```
    well_pollution
SET
  description = 'Bacteria: Giardia Lamblia'
WHERE
    description = 'Clean Bacteria: Giardia Lamblia' ;
UPDATE
    well_pollution
SET
  results = 'Contaminated: Biological'
WHERE
     biological > 0.01 AND results = 'Clean';
```

- **Put a test query here to make sure we fixed the errors.**

Use the query we used to show all of the erroneous rows

```
SELECT * FROM
  well_pollution
WHERE
   pollutant_ppm > 0.01
AND description LIKE 'Clean_%';
```