EXPLORE AI ACADEMY

Maji Ndogo: From analysis to action

Beginning Our Data-Driven Journey in Maji Ndogo

Aziza Naledi
Online

Dear Team,

Congratulations on the successful completion of our extensive survey. Your dedication and hard work have resulted in an invaluable asset - a database of 60,000 records, meticulously collected by our devoted team of engineers, field workers, scientists, and analysts.

Now, the next crucial phase of our mission begins. We need to make sense of this immense data trove and extract meaningful insights. We must breathe life into these records and listen to the story they are telling us.

I urge you to load this database and thoroughly acquaint yourselves with it. Dive deep, explore its structure, understand the variables and the connections between them. Each record is a chapter of our story; each query you run is a thread weaving that story together. This is a process of discovery - to uncover the patterns and nuances in our data. It's a chance to ask the right questions, to identify the pressing problems, and to set the course for our data-driven solutions.

As you proceed, always remember that every bit of information is a piece of the bigger puzzle. Every insight, no matter how small, brings us one step closer to solving our water crisis. Together, we have embarked on this journey to bring about change. Let's continue to march ahead with the same determination and resolve.

I believe in our collective potential. Together, we will unravel the secrets held within these records, and use this knowledge to shape a brighter future for Maji Ndogo.

Best regards,
Aziza Naledi

06:26

Chidi Kunto
Online

Hey Team,

Hope this message finds you well. I'm Chidi Kunto, and I'm really excited to be working with all of you on this upcoming project.

Our president, Aziza Naledi, has given us a monumental task. We've got this mountain of data, and it's our job to sift through it, finding those nuggets of insight that will help us solve our water crisis here in Maji Ndogo.

But you won't be doing this alone. As your mentor, I'll be here not just to guide you but also to show you firsthand how a real data project kicks off. I'll be taking Naledi's instructions and breaking them down into clear, manageable tasks. And of course, I'll be sharing a few valuable tricks of the trade along the way.

I'll help you navigate through this data landscape, turning those raw numbers into meaningful insights. Just remember, every data point can be part of our solution, and every step you take in this project helps us get closer to our goal.

I'm looking forward to working closely with each of you, learning about your unique perspectives, and figuring out how we can best collaborate as a team. With your skills and dedication, I'm confident we can unlock the potential of this data and make a real difference to the people of Maji Ndogo.

Please know that my virtual door is always open. If you need some guidance, want to clarify something, or just have something cool to share, feel free to reach out.

Here's to an exciting journey ahead and the valuable lessons we'll learn together!

Cheers,
Chidi

07:54

Chidi Kunto
Online

Hey there,

You've probably seen President Naledi's message by now. She has emphasised the importance of our newly collected survey data and how vital it is for us to dive in and start making sense of it. As the senior data analyst, I've taken a close look at her message and have broken it down into a series of tasks that we need to tackle. So, let's roll up our sleeves and get started!

08:33

1. Get to know our data: Before we do anything else, let's take a good look at our data. We'll load up the database and pull up the first few records from each table. It's like getting to know a new city - we need to explore the lay of the land before we can start our journey.

08:37

2. Dive into the water sources: We've got a whole table dedicated to the types of water sources in our database. Let's dig into it and figure out all the unique types of water sources we're dealing with.

08:42

3. Unpack the visits to water sources: The 'visits' table in our database is like a logbook of all the trips made to different water sources. We need to unravel this logbook to understand the frequency and distribution of these visits. Let's identify which locations have been visited more than a certain number of times.

08:43

Chidi Kunto
Online

4. Assess the quality of water sources: The quality of water sources is a pretty big deal. We'll turn to the `water_quality` table to find records where the `subjective_quality_score` is within a certain range and the `visit_count` is above a certain threshold. This should help us spot the water sources that are frequently visited and have a decent quality score.

08:50

5. Investigate any pollution issues: We can't overlook the pollution status of our water sources. Let's find those water sources where the pollution_tests result came back as 'dirty' or 'biologically contaminated'. This will help us flag the areas that need immediate attention.

08:57

By working through these tasks, we'll not only be answering President Naledi's call to explore the database and extract meaningful insights, but we'll also be honing our SQL skills. It's a win-win situation! So, are you ready to dive in and start exploring with me?

09:04

Let's do this!

09:05

1. Get to know our data:
Start by retrieving the first few records from each table. How many tables are there in our database? What are the names of these tables? Once you've identified the tables, write a SELECT statement to retrieve the first five records from each table. As you look at the data, take note of the columns and their respective data types in each table. What information does each table contain?

09:06

I don't think they showed you this at the Academy, but when you access a new database in MySQL, a handy initial query to run is `SHOW TABLES`. This will give you a list of all the tables in the database.

09:12

You should see something like this:

Tables_in_md_water_services_student
data_dictionary
employee
global_water_access
location
water_quality
visits
water_source
well_pollution

09:16

It looks like someone took the time to name all of these tables pretty well because we can kind of figure out what each table is about without really having to think too hard. `water_source` probably logs information about each source like where it is, what type of source it is and so on.

09:19

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 it which table we are looking at.

09:21

You should get something like this:

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

09:26

So we can see that this table has information on a specific location, with an address, the province and town the location is in, and if it's in a city (`Urban`) or not. We can't really see what location this is but we can see some sort of identifying number of that location.

09:28

Ok, so let's look at the `visits` table.

09:31

Here's what I get:

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

09:37

Yeah, so this is a list of `location_id`, `source_id`, `record_id`, and a date and time, so it makes sense that someone (`assigned_employee_id`) visited some location (`location_id`) at some time (`time_of_record`) and found a 'source' there (`source_id`). Often the "_id" columns are related to another table. In this case, the `source_id` in the `visits` table refers to `source_id` in the `water_source` table. This is what we call a foreign key, but we'll get more into this next time.

09:40

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

08:47

I get:

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	clean_well	364
AkHa00004224	tap_in_home_broken	94

08:48

Nice! Ok, we're getting somewhere now... Water sources are where people get their water from! Ok, this database is actually complex, so maybe a good idea for you is to look at the rest of the tables quickly. You can just select them, but remember in good SQL there would be a data dictionary somewhere that documents all of this information, so you should read that as well, and even keep a copy of that close if we need to find information quickly.

09:54

A data dictionary has been embedded into the database. If you query the `data_dictionary table`, an explanation of each column is given there.

09:54

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. Can you figure out which table contains this information?

10:00

Once you've identified the right table, write a SQL query to find all the **unique** types of water sources.

10:00

So I get this when I run it:

type_of_water_source
tap_in_home
tap_in_home_broken
well
shared_tap
river

10:06

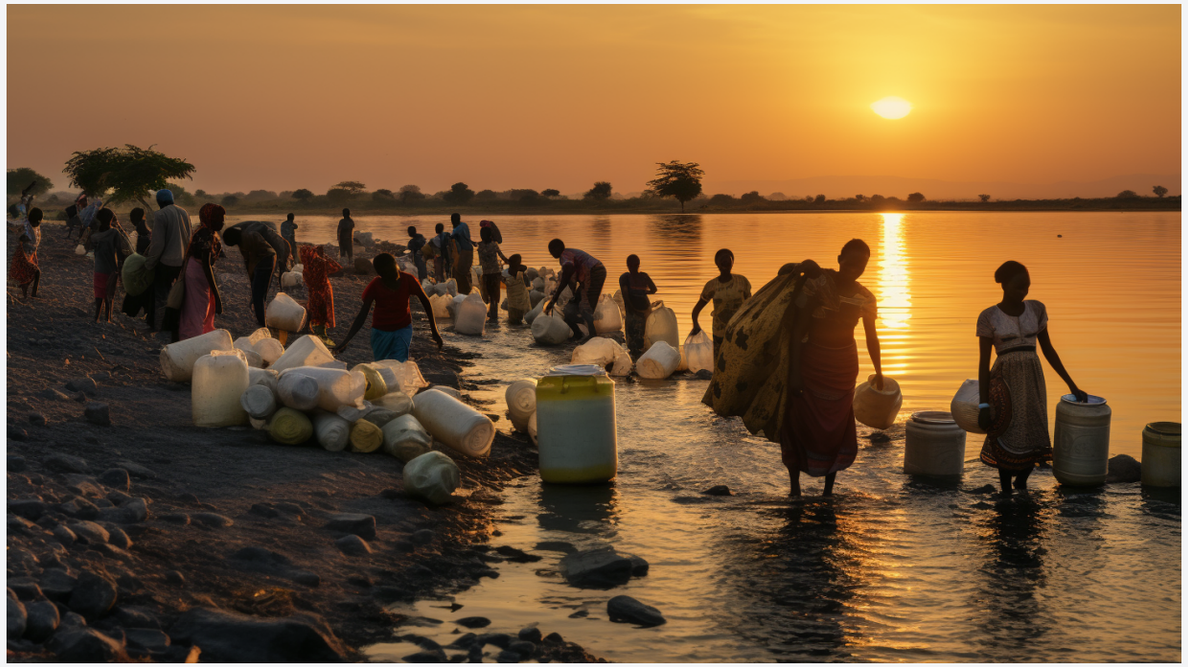
Let me quickly bring you up to speed on these water source types:

10:09

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.

10:15

This is a river in the province of Sokoto:

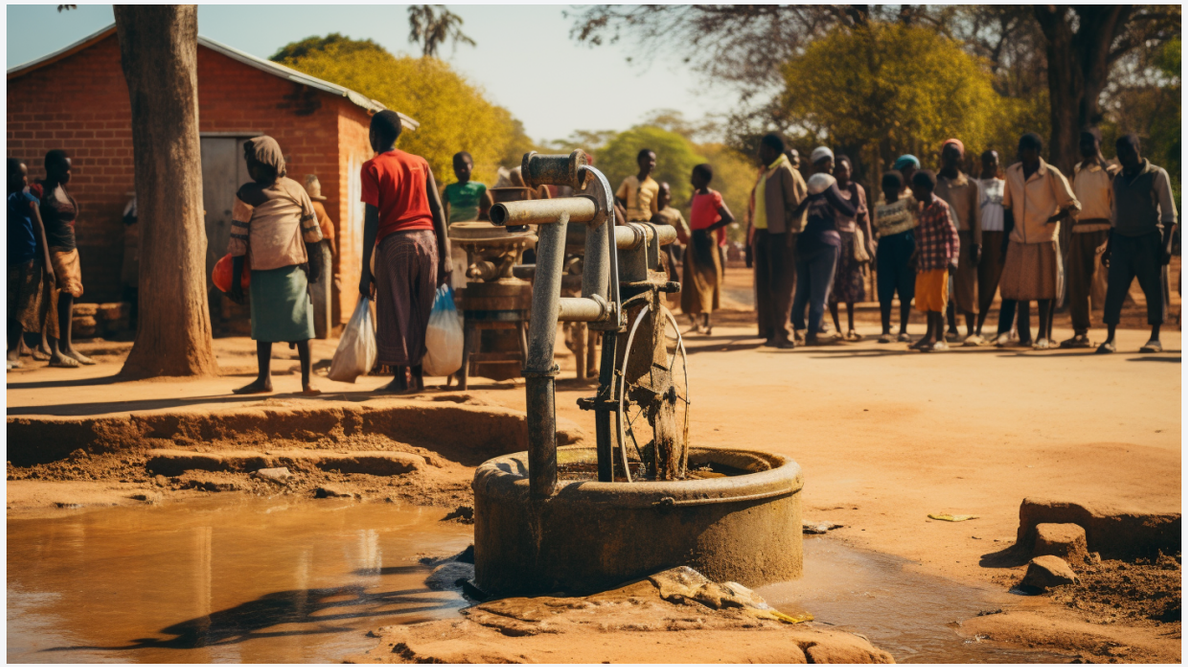


10:16

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 aging infrastructure and the corruption of officials in the past, many of our wells are not clean.

10:21

This well is at 146 Okapi Road, in my home town of Yaounde:



10:22

3. Shared tap - This is a tap in a public area shared by communities.

10:27

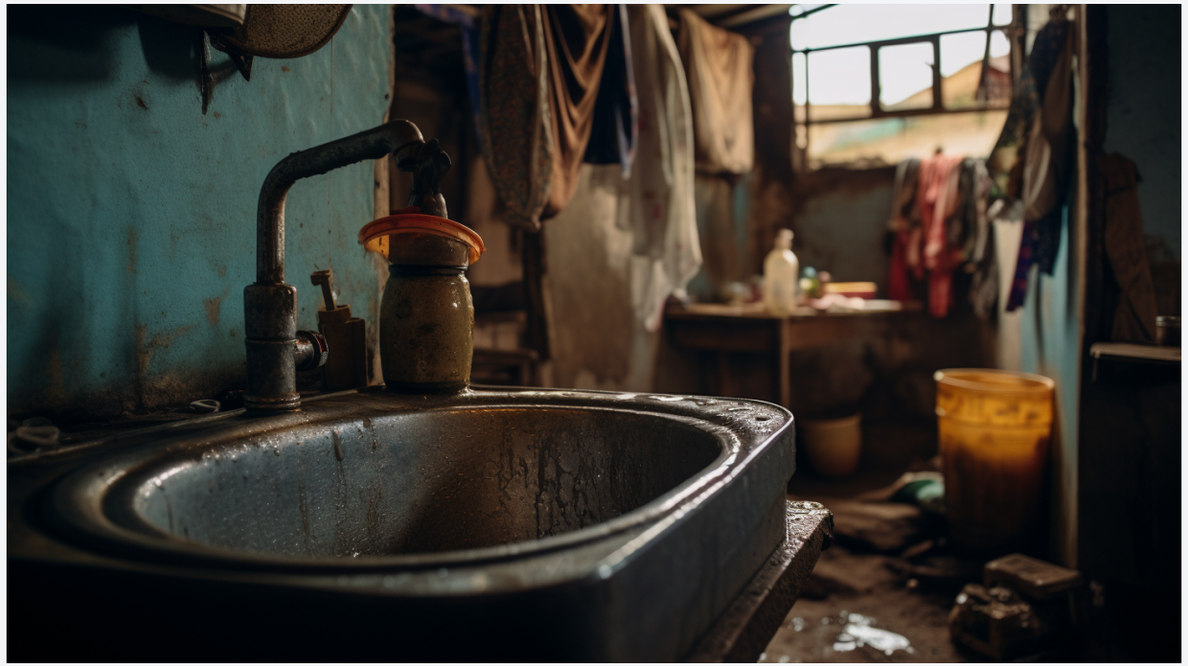
This is a shared tap from 18 Twiga Lane, Hawassa, that serves about 2700 people:

10:28

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.

10:28

This is a tap in my uncle's home in the capital city, Dahabu:

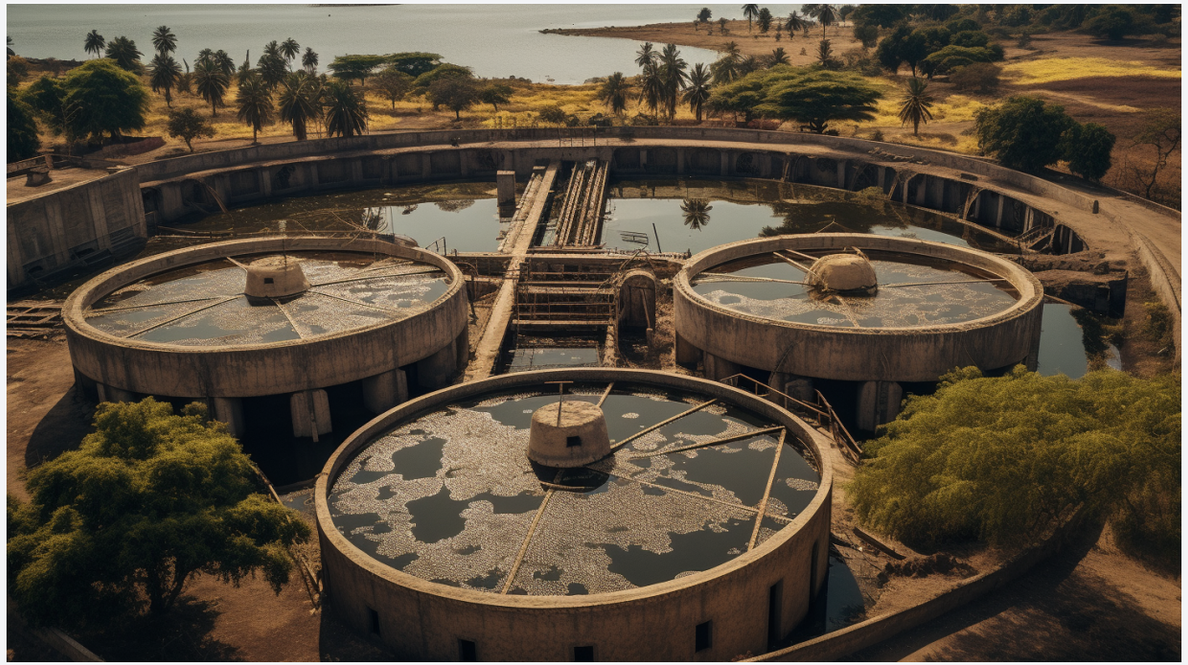


10:29

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.

10:35

This is a water treatment plant in the town of Kintampo that serves about 1000 people:



10:36

Chidi Kunto
Online

An important note on the home taps: About 6-10 million people have running water installed in their homes in Maji Ndogo, including broken taps. If we were to document this, we would have a row of data for each home, so that one record is one tap. That means our database would contain about 1 million rows of data, which may slow our systems down. For now, the surveyors combined the data of many households together into a single record.

10:37

For example, the first record, `AkHa00000224` is for a `tap_in_home` that serves 956 people. What this means is that the records of about 160 homes nearby were combined into one record, with an average of 6 people living in each house `160 x 6 ≈ 956`. So 1 `tap_in_home` or `tap_in_home_broken` record actually refers to multiple households, with the sum of the people living in these homes equal to `number_of_people_served`.

10:39

3. Unpack the visits to water sources:
We have a table in our database that logs the visits made to different water sources. Can you identify this table?

10:44

Write an SQL query that retrieves all records from this table where the `time_in_queue` is more than some crazy time, say 500 min. How would it feel to queue 8 hours for water?

10:47

This is the table I get:

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

10:54

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

10:57

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.
`AkKi00881224`
`SoRu37635224`
`SoRu36096224`
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.

10:58

I chose these two:
`AkRu05234224`
`HaZa21742224`
Ok, so now back to the `water_source` table. Let's check the records for those `source_ids`. You can probably remember there is a cool and a "not so cool" way to do it.

11:05

This is what I get.

source_id	type_of_water_source	number_of_people_served
AkKi00881224	shared_tap	3398
AkLu01628224	bio_dirty_well	210
AkRu05234224	tap_in_home_broken	496
HaRu19601224	shared_tap	3322
HaZa21742224	pol_dirty_well	308
SoRu36096224	shared_tap	3786
SoRu37635224	shared_tap	3920
SoRu38776224	shared_tap	3180

11:09

I added a couple of others... Sorry! Well, if you check them you will see which sources have people queueing. The field surveyors also let us know that they measured sources that had queues a few times to see if the queue time changed.

11:11

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.

11:14

Look through the table record to find the table.

11:17

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.

11:20

So please 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?

11:22

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!

11:23

5. Investigate pollution issues:
Did you notice that we recorded contamination/pollution data for all of the well sources? Find the right table and print the first few rows.

11:24

Find the right table and print the first few rows.

11:27

I get this:

source_id	date	description	pollutant_ppm	biological	results
KiRu28935224	2021-01-04 09:17:00	Bacteria: Giardia Lamblia	0.0	495.898	Contaminated: Biological
AkLu01628224	2021-01-04 09:53:00	Bacteria: Salmonella Typhi	0.0	376.572	Contaminated: Biological
HaZa21742224	2021-01-04 10:37:00	Inorganic contaminants: Zinc...	2.715	0.0	Contaminated: Chemical
HaRu19725224	2021-01-04 11:04:00	Clean	0.0288593	0.0	Clean
SoRu35703224	2021-01-04 11:29:00	Bacteria: E. coli	0.0	296.437	Contaminated: Biological

11:30

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.

11:36

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.

11:37

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

11:43

I got this:

source_id	date	description	pollutant_ppm	biological	results
AkRu08936224	2021-01-08 09:22:00	Bacteria: E. coli	0.0406458	35.0068	Clean
AkRu06489224	2021-01-10 09:44:00	Clean Bacteria: Giardia Lamblia	0.0897904	38.467	Clean
SoRu38011224	2021-01-14 15:35:00	Bacteria: E. coli	0.0425095	19.2897	Clean
AkKi00955224	2021-01-22 12:47:00	Bacteria: E. coli	0.0812092	40.2273	Clean
KiHa22929224	2021-02-06 13:54:00	Bacteria: E. coli	0.0722537	18.4482	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

11:45

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.

11:46

Chidi Kunto
Online

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.

11:50

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.

11:52

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.

11:59

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.

12:06

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.

12:09

Hint: 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".

12:14

The query should return 38 wrong descriptions.

12:20

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

12:21

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`

12:26

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.

12:31

Chidi Kunto
Online

Ok, so here is how I did it:

```
-- 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` to `Contaminated: Biological` where
`biological`  is greater than 0.01 plus current results is `Clean`
```

12:43

Before we make these changes, here is another nugget of advice: Begin complex queries by commenting on what you will do. This helps us to think through the problem before we write code to solve it, and once we're done, we have a well-documented code.

12:38

Then add how we would do it:

```
-- Case 1a
UPDATE
   -- Update well_pollution table
SET
   -- Change description to'Bacteria: E. coli'
WHERE
   -- Where the description is `Clean Bacteria: E. coli`

-- Case 1b
   -- Try to fill this in
-- Case 2
   -- Try to fill this in
```

12:46

Chidi Kunto
Online

And then fill in some details:

```
-- Case 1a
UPDATE
   well_pollution
SET
   description = 'Bacteria: E. coli'
WHERE
   description = 'Clean Bacteria: E. coli';

-- Case 1b
   -- Try to fill this in
-- Case 2`
   -- Try to fill this in
```

12:49

Ok, go ahead and fill in the rest.

12:53

Now, when we change any data on the database, we need to be SURE there are no errors, as this could fill the database with incorrect values. A safer way to do the UPDATE is by testing the changes on a copy of the table first.

12:54

The `CREATE TABLE new_table AS (query)` approach is a neat trick that allows you to create a new table from the results set of a query. This method is especially useful for creating backup tables or subsets without the need for a separate `CREATE TABLE` and `INSERT INTO` statement.

12:56

Chidi Kunto
Online

So if we run:

```
CREATE TABLE
   md_water_services.well_pollution_copy
AS (
    SELECT
        *
    FROM
        md_water_services.well_pollution
    );
```

13:02

We will get a copy of `well_pollution` called `well_pollution_copy`. Now we can make the changes, and if we discover there is a mistake in our code, we can just delete this table, and run it again.

13:06

Chidi Kunto
Online

So if we now run our query:

```
UPDATE
   well_pollution_copy
SET
   description = 'Bacteria: E. coli'
WHERE
   description = 'Clean Bacteria: E. coli';

UPDATE
   well_pollution_copy
SET
   description = 'Bacteria: Giardia Lamblia'
WHERE
   description = 'Clean Bacteria: Giardia Lamblia';

UPDATE
   well_pollution_copy
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
```

13 : 11

Chidi Kunto
Online

We can then check if our errors are fixed using a SELECT query on the `well_pollution_copy` table:

```
SELECT
    *
FROM
    well_pollution_copy
WHERE
    description LIKE "Clean_%"
   OR (results = "Clean" AND biological > 0.01);
```

13 : 13

Then if we're sure it works as intended, we can change the table back to the `well_pollution` and delete the `well_pollution_copy` table.

```
UPDATE
    well_pollution_copy
SET
    description = 'Bacteria: E. coli'
WHERE
    description = 'Clean Bacteria: E. coli';

UPDATE
    well_pollution_copy
SET
    description = 'Bacteria: Giardia Lamblia'
WHERE
    description = 'Clean Bacteria: Giardia Lamblia';

UPDATE
    well_pollution_copy
SET
    results = 'Contaminated: Biological'
WHERE
    biological > 0.01 AND results = 'Clean';

DROP TABLE
    md_water_services.well_pollution_copy;
```

13:19

So I hope that today helped you to see what the survey data is all about, I hope that you are as excited as I am to get stuck in! Until then, keep well!

13:33