Using MySQL to Reconstruct Malaria Epidemiologic Datasets from ClinEpiDB

Weilu Song

BMIN5020, April 2023

Contents

List of Tables

List of Figures

List of mySQL Codes

1 Project Descriptions (Part 1)

1.1 Introduction

Malaria is a significant global health challenge that causes morbidity and mortality, particularly in sub-Saharan Africa. According to the World Health Organization (WHO), an estimated 229 million cases of malaria occurred worldwide in 2020, leading to 409,000 deaths. The majority of these deaths occurred in children under the age of five.[1] In sub-Saharan Africa, malaria accounts for about 5% of all deaths in children under the age of five, and it is estimated that a child dies from malaria every two minutes.[2]

In Latin America, malaria remains a significant public health concern, with an estimated 0.5 million cases reported in the region in 2019. Brazil accounted for the highest number of malaria cases in the region, with over 300,000 cases reported.[2] Peru also reported a significant number of cases, with over 60,000 cases reported in 2019.[3] Malaria poses a significant burden on the health systems of these countries, with increased healthcare costs and loss of productivity due to illness and death.[4]

Epidemiological research is essential for understanding the transmission, distribution, and risk factors associated with malaria. However, previous studies may have been limited by small sample sizes and data collected from only a few areas. Small sample sizes and limited data collection areas can be major obstacles to obtaining representative and generalizable results in epidemiological research on malaria.[5]. For instance, a study conducted in Peru alone may not be representative of the entire Latin American region, as the epidemiology of malaria can differ significantly across different countries and regions. Furthermore, small sample sizes can lead to insufficient statistical power and lack of precision, limiting the reliability of the study findings.[6]

The integration of different malaria datasets using mySQL can provide several potential benefits for epidemiological research. Firstly, by combining data from different sources, researchers can explore associations and potential risk factors that may not be evident from individual datasets alone, leading to more accurate and robust findings and ultimately advancing our understanding of the transmission and pathology of malaria. Secondly, mySQL is a powerful tool for managing and analyzing large datasets, which is crucial in epidemiological research. The ability to store and manipulate data using SQL queries can enable researchers to more efficiently explore large and complex datasets, ultimately leading to more reliable and accurate results. Thirdly, integrating datasets using mySQL can help to overcome limitations of individual studies, such as small sample sizes and data collected from only a few areas. This can increase the sample size and representativeness of the study population, improving the reliability and generalizability of the study findings. Finally, the use of mySQL can facilitate collaboration among researchers by integrating data from multiple studies into a single dataset, which allows for greater opportunities for scientific exchange and collaboration.[7] Overall, the use of mySQL to construct malaria datasets has the potential to advance our understanding of the disease, improve the accuracy and reliability of study findings, and facilitate collaboration among researchers.[8]

In this project aims to showcase the versatility of mySQL in epidemiological research by demonstrating how different malaria datasets from various entity level datasets can be integrated and reshaped using advanced query codes to fit users' research interests or to be shared among coworkers. By overcoming the limitations of small datasets from single sources, integrated datasets from multiple sources can provide access to larger and more diverse datasets. For simplicity, this project will explore the association between exposures and outcomes using cross-sectional or survey datasets. We expect that the integrated datasets will benefit the study in several ways, including

facilitating more comprehensive analyses, increasing statistical power, and enabling comparisons across different studies.In addition, we also expect the integrated dataset can improve study results by increasing statistical power, accuracy and reliability of the study findings as well as the generalization and representation of outcome.

1.2 Method

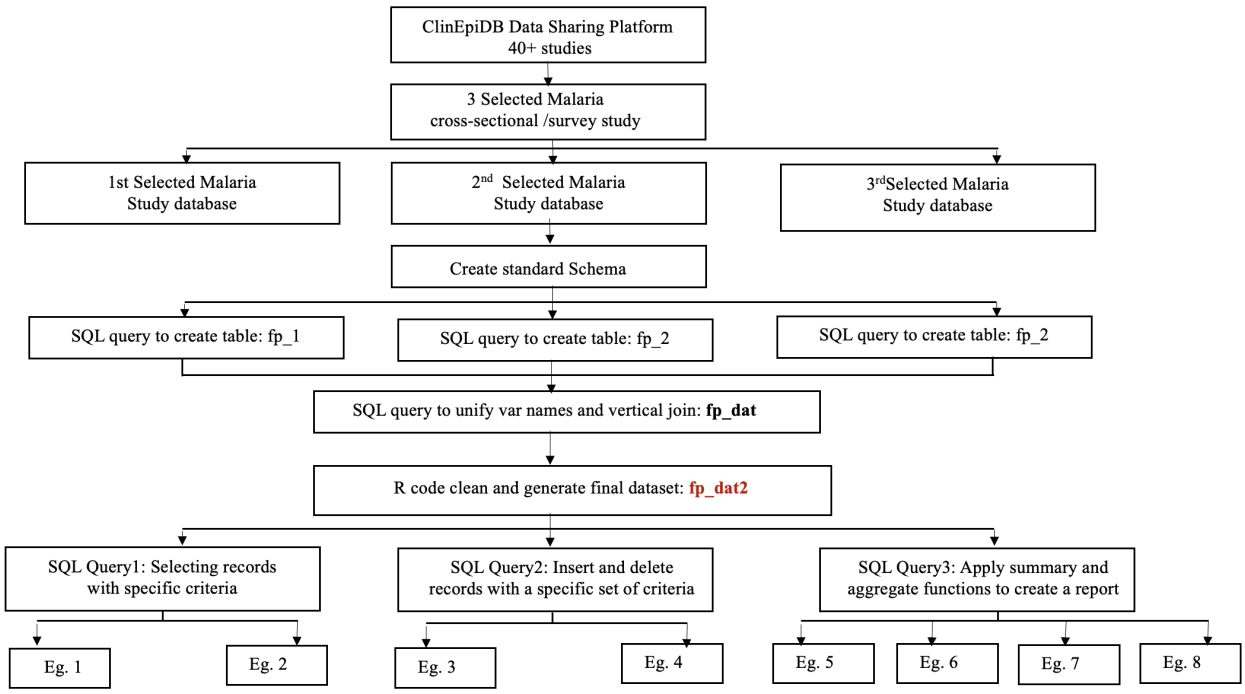

Figure 1: Project Framework for SQL Analysis

1.3 Analysis Plan

The analysis plan for this project involves four main steps. ***(a.)*** we will select study datasets that meet specific criteria, including a focus on malaria, a cross-sectional or survey study design, and availability of key malaria pathology variables such as demographic and sample test results. ***(b.)*** we will create a research reference table to guide variable selection, based on the infectious disease elements of host, agent, and environment. This will help us identify 19 key variables for the analysis. ***(c.)*** we will retrieve and subset the datasets using mySQL code, cleaning variable names and formatting as necessary. We will then vertically join the sub-datasets to construct a new, integrated dataset. ***(d.)***We will use R codes to clean this new dataset and Use SQL codes to provide 8 query examples that address different feature aspects for a basic dataset subset and malaria prevalence analysis.

The ER diagram for all three selected studies can be found in Part 2 (Database processing with mySQL), section 2.3.3. ***Click here to see Figure 10*** .

2 Database Processing with MySQL (Part 2)

2.1 Dataset Resources and Access

We used a combination of mySQL queries and Python scripts to extract and transform data from ClinEpiDB, an open-access exploratory data analysis platform launched in February 2018. ClinEpiDB integrates data from high-quality epidemiological studies and provides tools and visualizations to explore the data within the browser through a point-and-click interface. The platform enables investigators to maximize the utility and reach of their data, and also benefits other researchers who want to reuse the datasets to explore their own hypotheses for free.

2.2 Preparation for Data Extraction

2.2.1 Inclusion Criteria for the Dataset

→ Study is categorized as malaria;

→ Study design is either cross-sectional or survey type;

→ Study contain the main pathology variables of malaria such as demographic and sample test results;

→ Datasets are public.

2.2.2 "Study Summarized Table" Features

Here's a possible revision:

ClinEpiDB offers a wide range of datasets related to epidemiological studies, but for this particular project, we will only be using two of its features. The first one is the 'Study summarized table', which provides an overview of all the available datasets and allows users to filter them based on their specific interests. This feature is illustrated in figure 1 of our project report, and it will help us to identify the studies that are relevant to our research question.

Study summaries

Search Studies | Showing 1 - 47 of 47 Studies | Download | Add / Remove Columns

Study	Explore & analyze	Disease	Investigation type	Study design	Participants	Country	Years	Data accessibility
India ICEMR Fever Surveillance		Malaria	Surveillance		954	India	2016, 2017	Public
Namibia rfMDA RAVC Cluster Randomized Trial					17,461			Prerelease
GEMS1 Case Control		Diarrheal disease	Observational	Case-control	22,567	The Gambia, Mali, Bangladesh, Kenya, India, Pakistan, Mozambique	2007, 2008, 2009, 2010, 2011	Protected
GEMS1A Case Control		Diarrheal disease	Observational	Case-control	14,242	The Gambia, Mali, Bangladesh, Kenya, India, Pakistan, Mozambique	2011, 2012, 2013, 2014	Protected
PERCH Case Control		Respiratory condition	Observational	Case-control	9,557	Kenya, The Gambia, Mali, Zambia, South Africa, Thailand, Bangladesh	2011, 2012, 2013, 2014	Prerelease
VIDA Case Control		Diarrheal disease	Observational	Case-control	11,053	Kenya, Mali, The Gambia	2015, 2016, 2017, 2018	Prerelease
MORDOR Cluster Randomized Trial			Experimental	Cluster-randomized controlled trial	132,188	Niger	2014, 2015, 2016, 2017, 2018, 2019, 2020	Protected
Amazonia ICEMR Brazil Cohort		Malaria	Observational	Cohort	640	Brazil	2010, 2011, 2012, 2014	Protected
Amazonia ICEMR Peru Cohort		Malaria	Observational	Cohort	2,447	Peru	2012, 2013, 2014, 2015	Protected

Figure 2: Study Summarized Table on ClinEpiDB Website

For example, in this project, we follow the step: click "study sumamrize table" $\rightarrow$ filter "Data accessibility" $\rightarrow$ filter "investigation type", then we will see the result as showed in figure 2.

malaria

Showing 1 - 21 of 47 Studies

Download | Add / Remove Columns

Study	Explore & analyze	Disease	Investigation type	Study design	Participants	Country	Years	Data accessibility
India ICEMR Behavior Cross-sectional		Malaria	Observational	Cross-sectional	497	India	2017	Public
India ICEMR Cohort		Malaria	Observational	Cohort	397	India	2013, 2014, 2015	Public
India ICEMR Cross-sectional		Malaria	Observational	Cross-sectional	3,267	India	2012, 2013, 2014	Public
India ICEMR DAMaN Quasi-experimental Stepped-wedge		Malaria	Experimental	Non-randomized	2,470	India	2019, 2020	Public
India ICEMR Fever Surveillance		Malaria	Surveillance		954	India	2016, 2017	Public
India ICEMR Meghalaya Cross-sectional		Malaria	Observational	Cross-sectional		India	2018, 2019	Public
India ICEMR Severe *P. vivax* and *falciparum* Cohort		Malaria	Observational	Cohort	51	India	2016, 2017	Public
PRISM ICEMR Cohort		Malaria	Observational	Cohort	1,421	Uganda	2011, 2012, 2013, 2014, 2015, 2016, 2017	Public
PRISM2 ICEMR Cohort		Malaria	Observational	Cohort	531	Uganda	2017, 2018, 2019	Public
PROMOTE Birth Cohort 3 Randomized Controlled Trial		Malaria	Experimental	Randomized controlled/clinical trial	782	Uganda	2016, 2017, 2018	Public

Figure 3: Filtered Malaria Studies across Various Study Designs

Based on the information provided, we can infer that the project aims to investigate the relationship between malaria disease and various exposure variables. The figure2 indicates that there are 3 cross-sectional study datasets that meet the eligibility criteria for this project. These datasets are publicly available and labeled as malaria disease datasets. The specific studies that have been selected for this project are listed in figure 3. The cross-sectional study design suggests that data has been collected at a single time point from a sample of the population, allowing us to estimate the prevalence of malaria and associated risk factors.

India ICEMR Behavior Cross-sectional		Malaria	Observational	Cross-sectional	497	India	2017	Public
India ICEMR Cross-sectional		Malaria	Observational	Cross-sectional	3,267	India	2012, 2013, 2014	Public
India ICEMR Meghalaya Cross-sectional			Observational	Cross-sectional		India	2018, 2019	Public

Figure 4: Three Selected Malaria Datasets Based on Inclusion Criteria

2.2.3 "Download "Festure

Once we have identified the specific malaria datasets that we want to use, we can access the "Download" feature. This feature provides us with the ability to download either subset datasets or entire datasets. With subset datasets, we can choose only the variables (exposures) that we are interested in, while the entire datasets include all variables for each entity level, such as household,

participant, and participant repeated level. Figure 4 provides a visual representation of this feature.

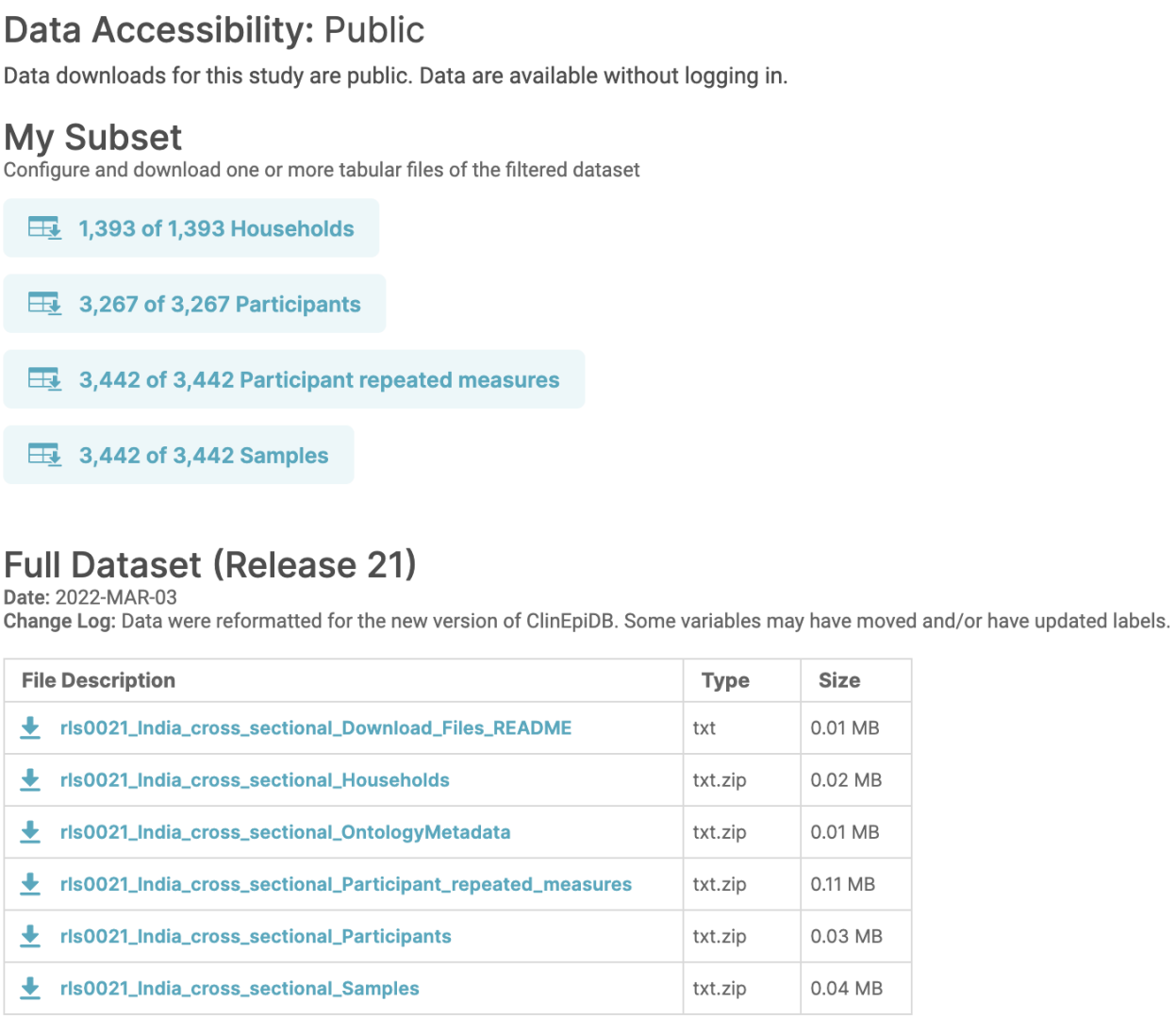

Figure 5: Downloading Selected Malaria Study Datasets

2.2.4 Creating a Standardized Schema

To create a customized integrated dataset and establish new exposure hypotheses, we will first need to download the full datasets for each selected study. We will then map our own schema based on the pathology of malaria, taking into account the host-agent-environment factors that are known to play a role in the disease. This schema will serve as a standardized framework for our data analysis, allowing us to efficiently search for and retrieve the relevant variables from each dataset. By creating a unified schema, we can ensure consistency and comparability across the datasets, which will ultimately facilitate our investigation into the complex interplay between various exposures and the risk of malaria.

Table 1: Host-Agent-Environment factors for Malaria

Host	**Agent**	**Environment**
Age, Sex, Genetics, Immune status, Behavior	Type of pathogen (e.g., bacteria, virus, parasite), Virulence, Infectivity, Resistance to treatment	Physical environment (e.g., temperature, humidity, altitude), Social environment (e.g., population density, access to healthcare), Biological

We have identified the key exposures that need to be extracted from the study datasets based on the information presented in table 1. To achieve this, a standardized schema must be developed to locate the required variables in the three selected datasets. Considering the host-agent-environment factors that contribute to malaria, a unified schema was created and is presented in table 2. Utilizing this schema, the datasets can be systematically searched for the relevant variables, facilitating a comprehensive analysis of the factors associated with malaria.

Table 2: Standardized Schema for Integrated Studies

No.	**Var name**	**Definition**	**Format**	**Length**
1	pid	Participant ID	char	NA
2	ani_indoor	Whether the participant's household keeps animals indoors	char	255
3	district	District in India where the participant lives	char	255
4	tn_bc	Number of ITN (insecticide-treated net) bednets owned by the participant's household	char	255
5	pov_line	Poverty line of the participant's household	char	255
6	dwell_type	Type of dwelling where the participant lives	char	255
7	age	Age of the participant	int	NA
8	fdiarrhea	Whether the participant had diarrhea	char	255
9	height	Height of the participant	char	255
10	education	Education level of the participant	char	255
11	mala_diag	Whether the participant was diagnosed with malaria	char	255
12	occupation	Occupation of the participant	char	255
13	sex	Biological gender of the participant	char	255
14	blood_ss	Blood sample results for sickle cell anemia	char	255
15	pqcr_neg	Results for Plasmodium falciparum PCR test	char	255
16	pviv_neg	Results for Plasmodium vivax PCR test	char	255
17	pf_neg	Results for Plasmodium falciparum rapid diagnostic test	char	255
18	pcr_source	Source of the PCR test	char	255

2.3 Data Transformation

2.3.1 Data Importation

To import the dataset, we need to find corresponding variables that we designed in table2 from different entity level dataset. Becasue for some variables such as INT use and bednet use, the study colleceted data from household level, but for demographic information such as age, sex, the study collected from participant level.

```
CREATE SCHEMA fp;
USE fp;
SET SQL_SAFE_UPDATES=0;
SET GLOBAL local_infile=1;

/* Datasets from selected study 1*/

/* Create household level table:hh using table wizard*/
SELECT * FROM hh;
UPDATE hh
SET Ani_indoor = COALESCE(Ani_indoor, 'Unknown'),
    Dwell_type = COALESCE(Dwell_type, 'Unknown'),
    ITN_bc = IFNULL(ITN_bc, 0),
    pov_line = COALESCE(pov_line, 'Unknown');

/*Create participant level table: par using table wizard*/
SELECT * FROM par;

/* Create participant repeated level table: pr using table wizard
   */
SELECT * FROM pr;

/*Create sample level table: sample*/
SELECT * FROM sample;
```

SQL Code 1: Creating Project Schema and Importing Raw Datasets

2.3.2 ER Model for Selected Study Dataset(s)

Revised: As each study comprises multiple datasets containing data at various entity levels (such as household, patient, patient repeated, and sample), it is crucial to create an E-R model to produce a clear chart mapping the relationships between these datasets and how to construct the new dataset based on the primary keys and variables provided in each dataset. In this project, we used mySQL to generate the initial E-R model without using SQL code to manage or modify any entity relationship.

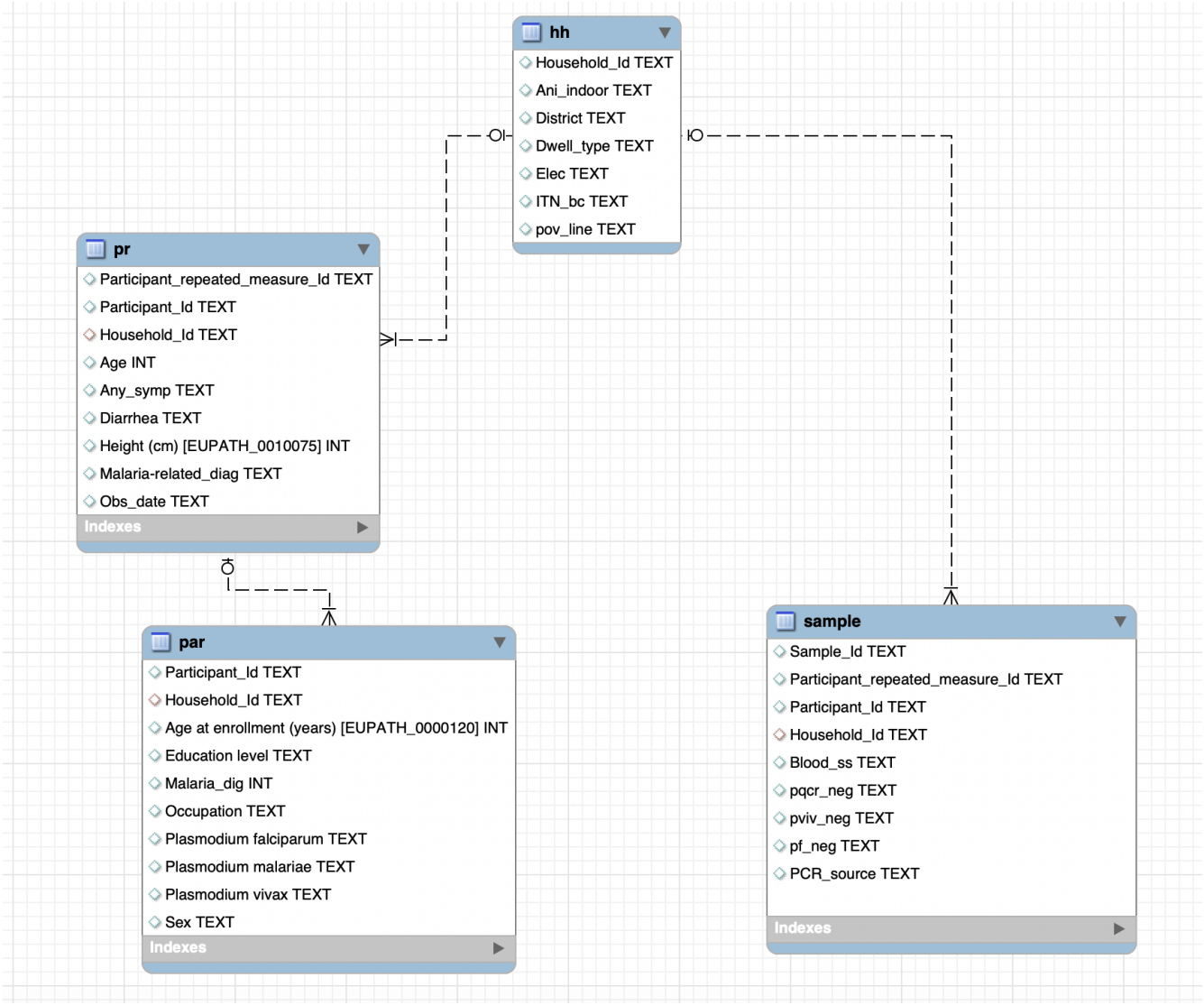

Figure 6: ER Diagram for the 1st Selected Study Database

The figure 5 shows an example of an E-R model for one of the selected studies in this project: ***India ICEMR cross-sectional study***. As shown in the figure, there are four different datasets in this study. The household-level dataset collected household information, the participant dataset collected non-repeated participant information. It should be noted that due to the long duration of this survey, some participants' information was repeatedly collected more than once, and the sample dataset collected lab test results based on all the repeated measured participants.

However, by using the EER diagram, it becomes easier to identify issues with the current relationships among all datasets and to reshape the datasets in mySQL to fit our project's purpose.

To simplify the project and focus only on participant-level data for calculating prevalence, we need to make some changes to the original data. Since the household table does not have a participant_Id, we cannot create a new table to get variables from the household table based on each participant. To overcome this, we will assign a participant_Id to the household table and then create a new table called fp1 based on the schema presented in Table 2. This will allow us to reconstruct the first dataset for the selected study 1 and work with participant-level data.

2.3.3 Reconstructing the Dataset (Using 1st Selected Study as an Example)

Step 1. Create new household level table called hh_new by using left join to assign the hh_new table Participant_Id

```
CREATE TABLE hh_new AS
SELECT h.*, p.Participant_Id
FROM household h
LEFT JOIN participant p
ON h.Household_Id = p.Household_Id;
```

SQL Code 2: Using the JOIN Function to Assign Participant_Id to the Household Table

Step 2.Now that all datasets contain the Participant_Id, we can use SQL code to create a new table called fp1 and extract variables based on the standardized schema (Table 2) that we previously created. The fp1 table will contain all the variables needed for our integrated table, including those from the different entity level datasets in the first selected study database.

```
CREATE TABLE fp1 AS
SELECT
    p.Participant_Id,
    hn.Ani_indoor,
    hn.District,
    hn.ITN_bc,
    hn.pov_line,
    hn.Dwell_type,
    p.`Age at enrollment (years) [EUPATH_0000120]`,
    pr.Diarrhea,
    pr.`Height (cm) [EUPATH_0010075]`,
    p.`Education level`,
    p.Malaria_dig,
    p.Occupation,
    p.Sex,
    s.Blood_ss,
    s.pqcr_neg,
    s.pviv_neg,
    s.pf_neg,
    s.PCR_source
FROM participant p
LEFT JOIN hh_new hn ON p.Participant_Id = hn.Participant_Id
LEFT JOIN pr ON p.Participant_Id = pr.Participant_Id
LEFT JOIN sample s ON p.Participant_Id = s.Participant_Id;
```

SQL Code 3: Creating the fp1 Table from the First Selected Study's Database

Step 3. Deduplicate repeated Participant_Id. To ensure that each row has a unique participant, we ran the following SQL code to check whether the Participant_Id is unique. If the Participant_Id occurs more than once, it indicates that there are repeated records, and we need to investigate why and remove them.

```sql
/*Check whether now in par table the Participant_Id is unqiue
   for each row*/
SELECT COUNT(*) as count, Participant_Id
FROM fp1
GROUP BY Participant_Id
HAVING count > 1;
```

SQL Code 4: Identifying Duplicate Participant_Ids in fp1 Table

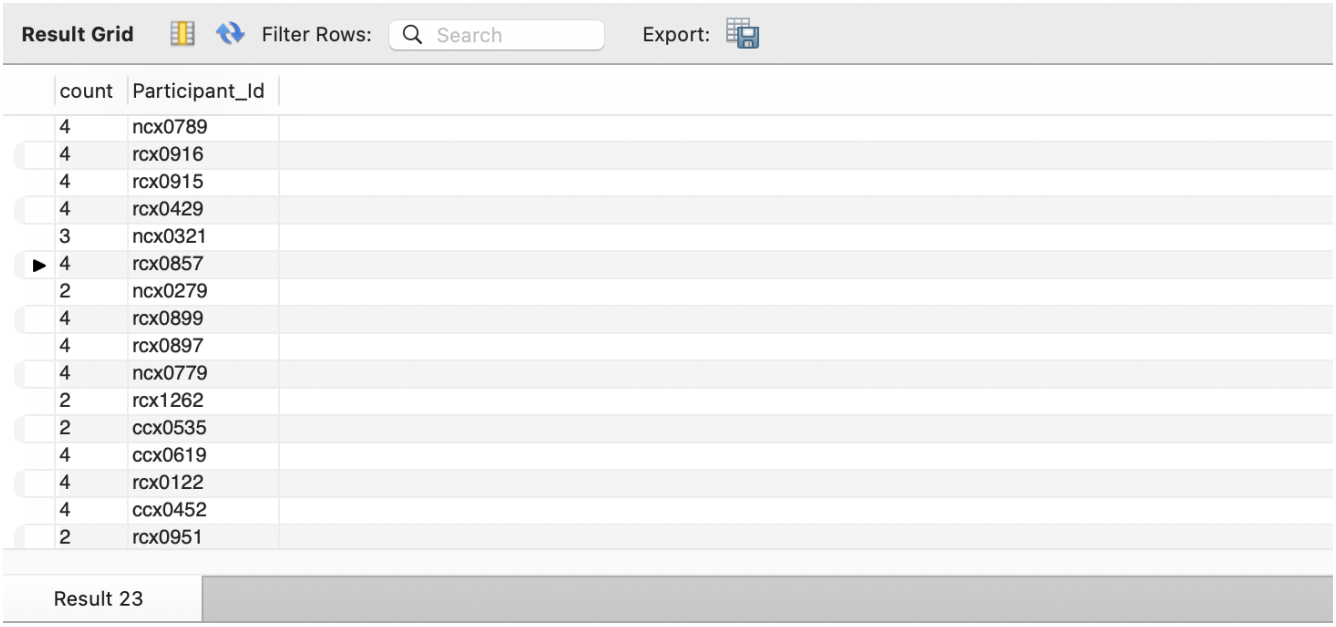

Figure 7: Repeated Primary Key in Reconstructed fp1 Table

```sql
CREATE TABLE fp_1 AS
SELECT *
FROM fp1
WHERE Participant_Id IN (SELECT DISTINCT Participant_Id FROM fp1);
```

SQL Code 5: Deduplicate Participant_Id for fp1 table

```sql
/*Check whether now in par table the Participant_Id is unqiue
   for each row*/
SELECT COUNT(*) as count, Participant_Id
FROM fp_1
GROUP BY Participant_Id
HAVING count > 1;
```

SQL Code 6: Checking for Duplicated Participant_Ids

As the result returns in mySQL, there is no repeated Paricipant_Id in par table, which is the expected result.

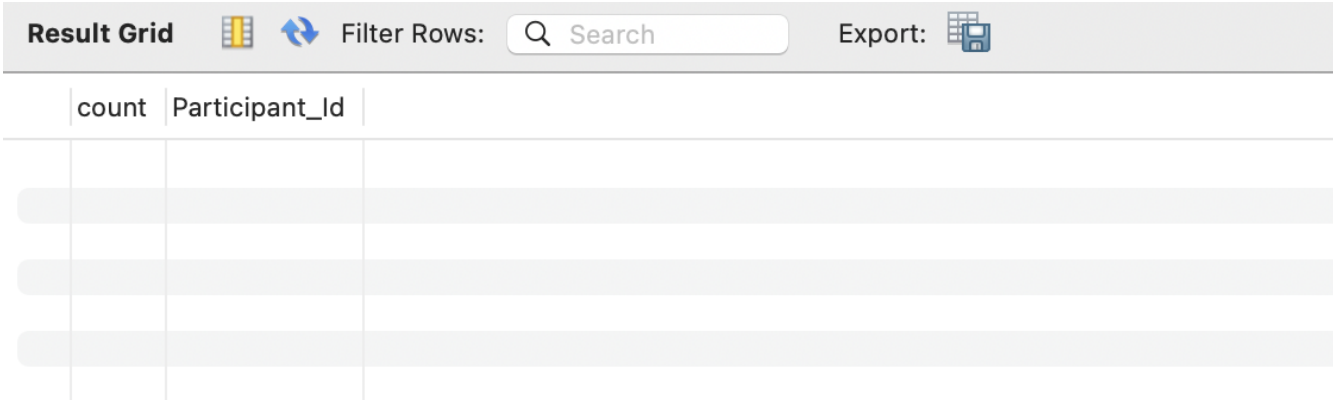

Figure 8: Results of Deduplication Check for Participant_Id

New Entity Relationship Diagram (ER) and Screenshot of Reconstructed Dataset for Selected Study 1.

After running the mySQL code, we were able to achieve the following: first, we assigned the Participant_Id to the household level dataset (hh_new), making this dataset usable in future queries to construct the standard table; second, we eliminated duplicate primary keys (PKs) and turned the 1:M relationship in the presented figure into a 1:1 relationship, generating a new reconstructed dataset that is better suited for our research purposes. Overall, the updated database enables us to perform queries at the participant level, using SQL code to reconstruct desired tables for prevalence analysis in the results section.

The updated ER diagram and a screenshot of the reconstructed dataset for Study 1 are shown below.

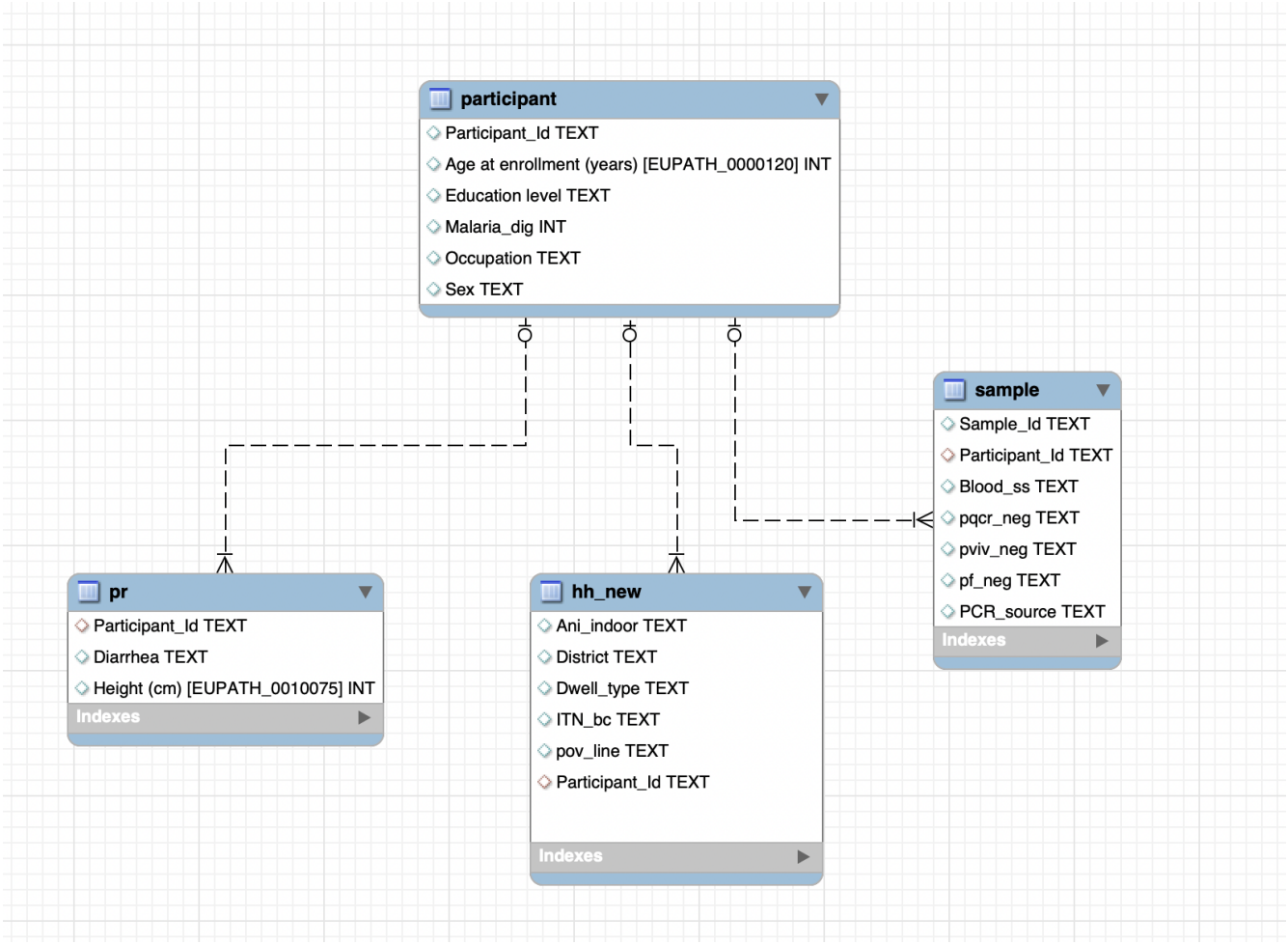

Figure 9: New ER for Updated 1st Selected Study Database

Step 4. We will repeat steps 1-3 to complete the standardization process for the remaining two tables: `fp_2` and `fp_3`. Using the same database processing method as described earlier, we will use `mySQL` to restructure the relationships among different datasets in each database for each selected study, and then create the tables `fp_2` and `fp_3` for further integration. Since the process is similar, we will skip the details to save space and only provide the `mySQL` codes in this project.

Below is the main mySQL code used to create the standard table **fp_2** based on the database of selected study-2:

```
/*Assigne PID to household dataset*/
CREATE TABLE d2_hh_new AS
SELECT h.*, p.pid
FROM d2_household h
LEFT JOIN d2_participant p
ON h.Household_Id = p.Household_Id;

/*Select variables from different entity level datasets in study
    2*/
CREATE TABLE fp2 AS
SELECT
```

```
    p.pid,
    hn.Ani_indoor,
    hn.District,
    hn.ITN_BC,
    hn.pov_line,
    hn.dwell_type,
    p.Age,
    p.Diarrhea,
    p.`Height (cm)`,
    p.`Education level`,
    p.malaria_diag,
    p.`Occupation [EUPATH_0000359]`,
    p.sex,
    s.Bss,
    s.pqcr_neg,
    s.pviv_neg,
    s.pf_neg,
    s.pcr_source
FROM d2_participant p
LEFT JOIN d2_hh_new hn ON p.pid = hn.pid
LEFT JOIN d2_sample s ON p.pid = s.pid;

/*Distinct Pid*/
   CREATE TABLE fp_2 AS
SELECT *
FROM fp2
WHERE Participant_Id IN (SELECT DISTINCT Participant_Id FROM fp2
   );

/*Check whether now in par table the PId is unqiue for each row
   */
SELECT COUNT(*) as count, pid
FROM fp_2
GROUP BY pid
HAVING count > 1;
```

SQL Code 7: Creating fp_2 Table Based on Standardized Schema

Below is the main mySQL code used to create the standard table **fp_3** based on the database of selected study-3:

```
/*Assigne PID to household dataset*/
CREATE TABLE d3_hh_new AS
SELECT h.*, p.Participant_Id
FROM d3_household h
LEFT JOIN d3_participant p
ON h.Household_Id = p.Household_Id;

```

```sql
/*Select variables from different entity level datasets in study 2*/
CREATE TABLE fp3 AS
SELECT
    p.Participant_Id,
    hn.animal_indoor,
    hn.district_india,
    hn.ITN_bednet_count,
    hn.poverty,
    hn.Dwell_type,
    p.`Age (years)`,
    p.Diarrhea,
    p.Height,
    p.`Education level`,
    p.malaria_dig,
    p.Occupation,
    p.Sex,
    s.Bss,
    s.pqcr_neg,
    s.pviv_neg,
    s.pf_neg,
    s.pcr_source
FROM d3_participant p
LEFT JOIN d3_hh_new hn ON p.Participant_Id = hn.Participant_Id
LEFT JOIN d3_sample s ON p.Participant_Id = s.Participant_Id;

/*Distinct Pid while keeping all other variables*/
   CREATE TABLE fp_3 AS
SELECT *
FROM fp3
WHERE Participant_Id IN (SELECT DISTINCT Participant_Id FROM fp3);

/*Check whether now in par table the PId is unqiue for each row */
SELECT COUNT(*) as count, Participant_Id
FROM fp_3
GROUP BY Participant_Id
HAVING count > 1;
```

SQL Code 8: Creating fp_3 Table Based on Standardized Schema

The ER diagram after updated 3 selected study database are preseted below:

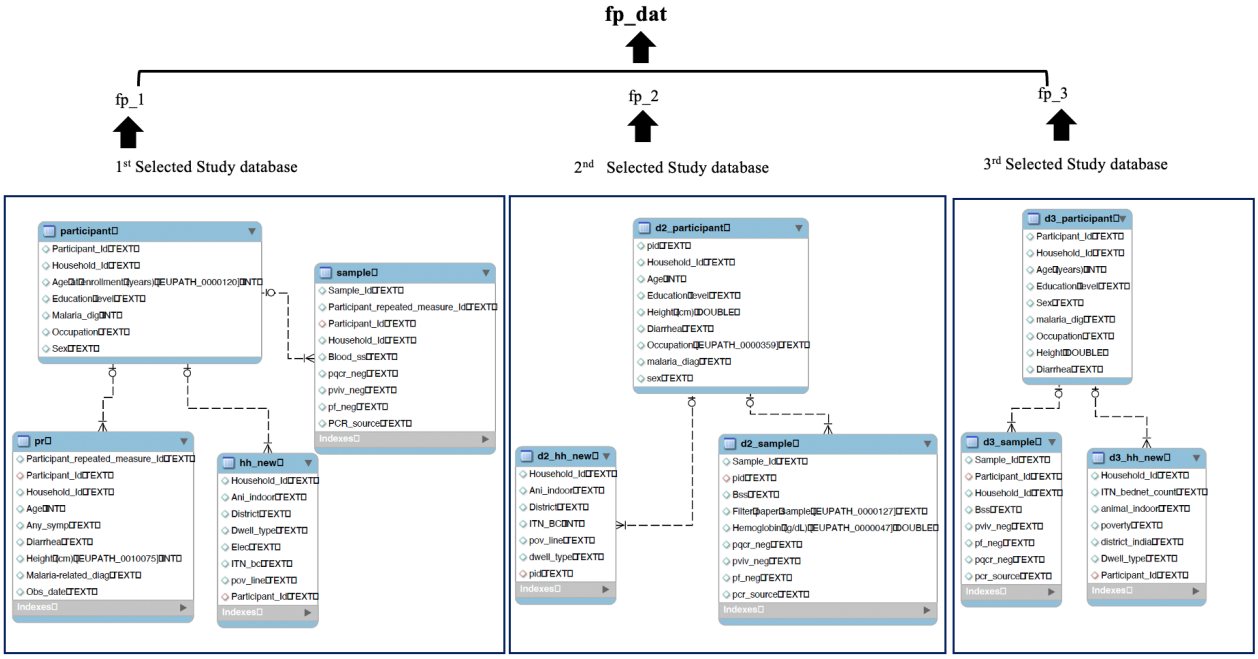

Figure 10: ER Diagram for 3 Selected Study Databases After Updates

Step 5. After generating fp_2 and fp_3 datasets using mySQL, we can combine them vertically to increase the sample size. In this step, we mainly focus on two tasks: a) unifying the variable names across different fp datasets. As shown in the code, the variable names are slightly different in each study's datasets, so we need to standardize them for better use in the future; b) vertically joining the three fp datasets using the 'UNION' command to create a single dataset. Finally, we will put all the queries in a new table named fp_dat, which will contain all the observations from the three pre-integrated datasets (fp_1, fp_2, fp_3) with consistent variable names for each column

```
CREATE TABLE fp_dat AS
SELECT
    Participant_Id AS pid,
    Ani_indoor AS ani_indoor,
    District AS district,
    ITN_bc AS itn_bc,
    pov_line AS pov_line,
    Dwell_type AS dwell_type,
    `Age at enrollment (years) [EUPATH_0000120]` AS age,
    Diarrhea AS diarrhea,
    `Height (cm) [EUPATH_0010075]`AS height,
    `Education level` AS education,
    Malaria_dig AS mala_diag,
    Occupation AS occupation,
    Sex AS sex,
    Blood_ss AS blood_ss,
    pqcr_neg AS pqcr_neg,
    pviv_neg AS pviv_neg,
```

```
    pf_neg AS pf_neg,
    PCR_source AS pcr_source
FROM fp_1
UNION
SELECT
    pid AS pid,
    Ani_indoor AS ani_indoor,
    District AS district,
    ITN_BC AS itn_bc,
    pov_line AS pov_line,
    dwell_type AS dwell_type,
    Age AS age,
    Diarrhea AS diarrhea,
    `Height (cm)` AS height,
    `Education level` AS education,
    malaria_diag AS mala_diag,
    `Occupation [EUPATH_0000359]` AS occupation,
    sex AS sex,
    Bss AS blood_ss,
    pqcr_neg AS pqcr_neg,
    pviv_neg AS pviv_neg,
    pf_neg AS pf_neg,
    pcr_source AS pcr_source
FROM fp_2
UNION
SELECT
    Participant_Id AS pid,
    animal_indoor AS ani_indoor,
    district_india AS district,
    ITN_bednet_count AS itn_bc,
    poverty AS pov_line,
    Dwell_type AS dwell_type,
    `Age (years)` AS age,
    Diarrhea AS diarrhea,
    Height AS height,
    `Education level` AS education,
    malaria_dig AS mala_diag,
    Occupation AS occupation,
    Sex AS sex,
    Bss AS blood_ss,
    pqcr_neg AS pqcr_neg,
    pviv_neg AS pviv_neg,
    pf_neg AS pf_neg,
    pcr_source AS pcr_source
FROM fp_3;
```

SQL Code 9: Integrating Three Standard Tables into One: fp_dat

Step 6. In this step, we will clean the integrated dataset (fp_dat) by checking for missing values, filling null cells, removing outliers, and ensuring record consistency across studies. Since MySQL is primarily designed for data storage and retrieval rather than data cleaning tasks, we will incorporate R to assist with the cleaning process. The resulting cleaned dataset, named fp_dat2, is presented below:

pid	ani_indoor	district	itn_bc	pov_line	dwell_type	age	diarrhea	height	education	mala_diag	occupation	sex	blood_ss	pqcr_neg	pviv_neg	pf_neg	pcr_source
ccx0364	No	Chennai	2	Above	House	26	No	168	High school	No	Housewife	Female	Yes	Negative	Negative	Negative	Small volume
ccx0365	No	Chennai	2	Above	House	29	No	168	High school	No	Service (salaried)	Male	Yes	Negative	Negative	Negative	Small volume
ccx0366	Yes	Chennai	2	Above	House	26	No	174	High school	No	Service (salaried)	Male	Yes	Negative	Negative	Negative	Small volume
ccx0367	Yes	Chennai	2	Above	House	23	No	148	High school	No	Housewife	Female	Yes	Negative	Negative	Negative	Small volume
ccx0368	No	Chennai	2	Above	House	26	No	164	Graduate or higher	No	Service (salaried)	Male	Yes	Negative	Negative	Negative	Small volume
ccx0369	No	Chennai	2	Above	House	45	No	154	Below primary	No	Housewife	Female	Yes	Negative	Negative	Negative	Small volume
ccx0370	No	Chennai	3	Above	House	66	No	148	Primary school	No	Service (salaried)	Female	Yes	Negative	Negative	Negative	Small volume
ccx0371	No	Chennai	0	Above	House	14	No	154	High school	No	Student	Female	Yes	Negative	Negative	Negative	Small volume
ccx0372	Yes	Chennai	2	Above	House	16	No	172	High school	No	Student	Male	Yes	Negative	Negative	Negative	Small volume
ccx0373	Yes	Chennai	2	Above	House	19	No	161	Graduate or higher	No	Student	Female	Yes	Negative	Negative	Negative	Small volume
ccx0374	Yes	Chennai	1	Above	House	65	No	165	Primary school	No	Trade (self-empl…	Male	Yes	Negative	Negative	Negative	Small volume
ccx0375	No	Chennai	1	Above	House	45	No	155	Below primary	No	Housewife	Female	Yes	Negative	Negative	Negative	Small volume
ccx0376	No	Chennai	1	Above	House	21	No	180	Graduate or higher	No	Service (salaried)	Male	Yes	Negative	Negative	Negative	Small volume
ccx0377	No	Chennai	0	Above	House	18	No	154	High school	No	Student	Female	Yes	Negative	Negative	Negative	Small volume
ccx0378	No	Chennai	2	Above	House	16	No	152	High school	No	Student	Female	Yes	Negative	Negative	Negative	Small volume
ccx0379	No	Chennai	5	Above	House	7	No	118	Primary school	No	Student	Male	Yes	Negative	Negative	Negative	Small volume
ccx0380	No	Chennai	3	Above	House	40	No	148	Primary school	No	Housewife	Female	Yes	Negative	Negative	Negative	Small volume
ccx0381	No	Chennai	4	Above	House	18	No	155	High school	No	Student	Female	Yes	Negative	Negative	Negative	Small volume
ccx0382	No	Chennai	3	Above	House	39	No	157	Primary school	No	Trade (self-empl…	Female	Yes	Negative	Negative	Negative	Small volume
ccx0383	No	Chennai	2	Above	House	32	No	143	High school	No	Housewife	Female	Yes	Negative	Negative	Negative	Small volume
ccx0384	No	Chennai	2	Above	House	5	No	106	Primary school	No	Student	Female	Yes	Negative	Negative	Negative	Small volume

Figure 11: Cleaned Dataset Generated by SQL Query: fp_dat2

2.4 Data Documentation

2.4.1 Updated Codebook for the Integrated Dataset

By using the code provided in this class, we can generate a dictionary based on the database of fp. Bascially, it is the same as standared schema(see table 2) we created before, but with more details.

```
SELECT t.table_schema AS Final_project,
t.table_name,
(CASE WHEN t.table_type = 'BASE TABLE' THEN 'table'
WHEN t.table_type = 'VIEW' then 'view'
ELSE t.table_type
END) AS table_type,
c.column_name,
c.column_type,
c.column_key,
c.is_nullable,
c.column_comment
FROM information_schema.tables AS t
INNER JOIN information_schema.columns AS c on t.table_name = c.
   table_name
AND  t.table_schema = c.table_schema
WHERE  t.table_type in('BASE TABLE', 'VIEW')
AND  t.table_schema = 'fp'
ORDER BY  t.table_schema, t.table_name, c.ordinal_position;
```

SQL Code 10: Creating a Dataset Dictionary for an Integrated Dataset

This code generates codebooks for all tables in the "fp" schema. We can manually select the level of codebook we want from the raw codebook, which contains all 10 tables from the 3 selected studies' databases. The cleaned codebook for the integrated dataset is represented by fp_dat. To save space, we have included only the final version of the fp_dat2 codebook, which displays information after the datasets have been extracted, subsetted, cleaned, joined, and integrated using SQL.Table 3 shows the updated codebook.

Table 3: Codebook for fp_dat

DB name	**table_type**	**Var_name**	**Definition**	**Format**	**Value**
fp	fp_dat	pid	Participant ID	char	NA
fp	fp_dat	ani_indoor	Whether the participant's household keeps animals indoors	char	No, Yes
fp	fp_dat	district	District in India where the participant lives	char	NA
fp	fp_dat	itn_bc	Number of ITN (insecticide-treated net) bednets owned by the participant's household	char	0-100
fp	fp_dat	pov_line	Poverty line of the participant's household	char	Below, Above
fp	fp_dat	dwell_type	Type of dwelling where the participant lives	char	House, Apartment, Unknown
fp	fp_dat	age	Age of the participant	int	0-100
fp	fp_dat	diarrhea	Whether the participant had diarrhea	char	No, Yes
fp	fp_dat	height	Height of the participant	char	0-100
fp	fp_dat	education	Education level of the participant	char	Primary,Highschool, Undergraduate,Graduate
fp	fp_dat	mala_diag	Whether the participant was diagnosed with malaria	char	No, Yes
fp	fp_dat	occupation	Occupation of the participant	char	Open text
fp	fp_dat	sex	Biological gender of the participant	char	Female, Male
fp	fp_dat	blood_ss	Blood sample results for sickle cell anemia	char	Negative,Positive
fp	fp_dat	pqcr_neg	Results for Plasmodium falciparum PCR test	char	Negative,Positive
fp	fp_dat	pviv_neg	Results for Plasmodium vivax PCR test	char	Negative,Positive,Not done
fp	fp_dat	pf_neg	Results for Plasmodium falciparum rapid diagnostic test	char	Negative, Positive
fp	fp_dat	pcr_source	Source of the PCR test	char	Small volume, Blood Spot

2.5 Results: Queries and Other Findings

2.5.1 Comparing Sample Sizes between Integrated Datasets and Individual Studies

The aim of this project is to increase the study sample size by integrating datasets from different study databases. A comparison of the sample size between the three selected studies and the newly reconstructed study dataset is shown in the table.

Table 4: Comparison of Sample Sizes Across Selected Studies and Reconstructed Dataset

Study Name	Sample Size	Study Name	Sample Size
India ICEMR Behavior Cross-sectional	490	Reconstructed malaria dataset	4595
India ICEMR Cross-sectional	3265		
India ICEMR Meghalaya Cross-sectional	840		

2.5.2 SQL Query1: Selecting records with specific criteria

Example 1. Select record by setting up a nested specific critieria This SQL code retrieves the district-wise malaria positive proportion from the fp_dat2 dataset where the proportion is greater than 0.5. The subquery in this code calculates the malaria positive proportion for each district by dividing the number of positive malaria cases (based on the pqcr_neg column being 'Positive') by the total number of cases in that district. The outer query then selects the district and malaria positive proportion columns from the subquery and applies a filter to only include districts where the malaria positive proportion is greater than 0.5. Finally, the results are sorted in descending order by malaria positive proportion. From an epidemiological perspective, this code is useful for identifying districts with high malaria transmission rates. The malaria positive proportion is a key indicator of malaria transmission intensity and is often used to track changes in malaria incidence over time. By identifying districts with a high malaria positive proportion, public health officials can target interventions to reduce malaria transmission in these areas, such as distributing insecticide-treated bed nets, conducting indoor residual spraying, and providing timely diagnosis and treatment to malaria patients. Additionally, the results of this code can be used to monitor changes in malaria incidence over time and evaluate the effectiveness of malaria control programs.

```
SELECT district, malaria_pos_proportion
FROM (
SELECT
    district,
    ROUND(SUM(pqcr_neg = 'Positive')/COUNT(*)*100, 2) AS
       malaria_pos_proportion
FROM
    fp_dat2
WHERE
    pqcr_neg IN ('Positive', 'Negative')
GROUP BY district
        ) AS subquery
```

```sql
WHERE malaria_pos_proportion > 0.5
ORDER BY malaria_pos_proportion DESC;
```

SQL Code 11: Nested select from dataset fp_dat2

```sql
SELECT district, malaria_pos_proportion
FROM (
    SELECT
        district,
        ROUND(SUM(pqcr_neg = 'Positive')/COUNT(*)*100, 2) AS malaria_pos_proportion
    FROM
        fp_dat2
    WHERE
        pqcr_neg IN ('Positive', 'Negative')
    GROUP BY district
) AS subquery
WHERE malaria_pos_proportion > 0.5
ORDER BY malaria_pos_proportion DESC;
```

100% 1:408 1 error found

Result Grid Filter Rows: Search Export:

district	malaria_pos_proport...
Sundergarh	2.86
Kheda	2.74
Chennai	1.74

Figure 12: Result: Top 3 Regions in India with Highest Malaria Prevalence

Example 2. Use the DELIMITER to set specific critieria.

```sql
DELIMITER //
CREATE PROCEDURE sex_restrict(IN sex_param CHAR(30))
BEGIN
SELECT *
FROM fp_dat2
WHERE sex= sex_param
AND pov_line = 'below'
AND age BETWEEN 15 AND 65;
END //
DELIMITER ;
CALL sex_restrict('Female')
```

SQL Code 12: Using DELIMITER to subset fp_dat2 by applying specific creteria

The output is presented below:

Result Grid Filter Rows: Search Export:

pid	ani_indoor	district	itn_bc	pov_line	dwell_type	age	diarrhea	height	education	mala_diag	occupation	sex	blood_ss	pqcr_neg	pviv_neg	pf_neg	pcr_source
ccx0526	No	Chennai	2	Below	House	32	No	153	Below primary	No	None	Female	Yes	Negative	Negative	Negative	Small volume
ccx0556	Yes	Chennai	2	Below	Apartment	25	No	164	Graduate or higher	No	Housewife	Female	Yes	Negative	Negative	Negative	Blood spot
ccx0618	Yes	Chennai	1	Below	Apartment	20	No	148	Primary school	No	Service (salaried)	Female	Yes	Positive	Positive	Positive	Blood spot
ccx0645	No	Chennai	0	Below	House	31	No	150	Below primary	No	Service (salaried)	Female	Yes	Negative	Negative	Negative	Blood spot
ccx0695	No	Chennai	1	Below	Apartment	21	No	143	High school	No	Housewife	Female	Yes	Negative	Negative	Negative	Blood spot
ncx0023	Yes	Kheda	1	Below	House	60	No	155	Below primary	No	None	Female	Yes	Negative	Negative	Negative	Blood spot
ncx0024	Yes	Kheda	1	Below	House	22	No	153	Primary school	No	Housewife	Female	Yes	Negative	Negative	Negative	Blood spot
ncx0025	Yes	Kheda	0	Below	House	30	No	148	Below primary	No	Housewife	Female	Yes	Negative	Negative	Negative	Blood spot
ncx0026	Yes	Kheda	2	Below	House	40	No	160	Below primary	No	Housewife	Female	Yes	Negative	Negative	Negative	Blood spot
ncx0027	Yes	Kheda	5	Below	House	15	No	156	Primary school	No	None	Female	Yes	Negative	Negative	Negative	Blood spot
ncx0031	No	Kheda	2	Below	House	28	No	165	Below primary	No	Housewife	Female	Yes	Negative	Negative	Negative	Blood spot
ncx0035	No	Kheda	0	Below	House	50	No	158	Below primary	No	Housewife	Female	Yes	Negative	Negative	Negative	Blood spot
ncx0036	No	Kheda	3	Below	House	18	No	162	Below primary	No	None	Female	Yes	Negative	Negative	Negative	Blood spot
ncx0037	No	Kheda	0	Below	House	29	No	147	Below primary	No	Housewife	Female	Yes	Negative	Negative	Negative	Blood spot
ncx0057	No	Kheda	0	Below	House	49	No	158	Below primary	No	Housewife	Female	Yes	Negative	Negative	Negative	Blood spot
ncx0059	No	Kheda	5	Below	House	25	No	156	Primary school	No	Housewife	Female	Yes	Negative	Negative	Negative	Blood spot

Figure 13: Result: Subset of dataset for malaria in women

2.5.3 SQL Query2: Insert and delete records with a specific set of criteria

Example 3. Insert and delete records with regular SQL

The SQL code below filters malaria status based on a specific set of criteria by creating a new table named fp_test. The mal_stat column is marked with malaria status using the following criteria: if all three malaria testing results are not equal, the malaria status is marked as Uncertain. If all testing results are negative, the malaria status is marked as negative. If all testing results are positive, the malaria status is marked as positive. By using these criteria, the code aims to determine how many participants have an Uncertain malaria status.

```
CREATE TABLE fp_test AS
SELECT pid, pqcr_neg, pviv_neg, pf_neg,
CASE
  WHEN pqcr_neg <> 'Positive' OR pviv_neg <> 'Positive' OR pf_neg
     <> 'Positive' THEN 'Uncertain'
  WHEN pqcr_neg = 'Negative' AND pviv_neg = 'Negative' AND pf_neg
     = 'Negative' THEN 'Negative'
ELSE 'Positive'
END AS mal_stat
FROM fp_dat2
  WHERE pqcr_neg='Positive' OR pviv_neg='Positive' OR pf_neg='
     Positive';

SELECT * FROM fp_test;
```

SQL Code 13: Filtering Malaria Cases Based on Test Results

```
547    SELECT * FROM fp_test;
548
```

100% | 23:545 | 1 error found

Result Grid | Filter Rows: Search | Export:

pid	pqcr_neg	pviv_neg	pf_neg	mal_stat
ccx0400	Negative	Positive	Negative	Uncertain
ccx0433	Negative	Positive	Negative	Uncertain
ccx0436	Negative	Positive	Negative	Uncertain
ccx0448	Positive	Positive	Negative	Uncertain
ccx0460	Negative	Positive	Negative	Uncertain
ccx0490	Negative	Positive	Negative	Uncertain
ccx0496	Negative	Positive	Negative	Uncertain
ccx0509	Negative	Positive	Negative	Uncertain
ccx0535	Negative	Negative	Positive	Uncertain
ccx0536	Negative	Positive	Positive	Uncertain
ccx0550	Negative	Positive	Negative	Uncertain
ccx0587	Negative	Positive	Negative	Uncertain

Figure 14: Result: Filtering Malaria Cases Based on Test Results

Example 4. Insert with TRIGGER. Suppose we want to study the influence of poverty on infant malaria. We have a dataset called fp_trig, but we need to add more observation values. However, some of the new data has input typos and is invalid. In this case, we can use the TRIGGER function to identify these invalid inputs. For example, we can use the values 999, 777, and 666 to represent invalid inputs for age, height, and pov_line, respectively.

```
1
2  DROP TABLE IF EXISTS fp_trig;
3  CREATE TABLE fp_trig AS
4  SELECT pid, pov_line,age,height,sex,pf_neg
5  FROM fp_dat2
6  WHERE pf_neg='Positive'
7    AND pov_line='Below'
8    AND age BETWEEN 0 AND 2;
9
10 DROP TRIGGER IF EXISTS inputcheck;
11 DELIMITER //
12 CREATE TRIGGER inputcheck BEFORE INSERT
13 ON fp_trig
14 FOR EACH ROW
15 BEGIN
16   IF NEW.age > 2 THEN
17   SET NEW.age = 999;
18     END IF;
19   IF NEW.height > 120 THEN
20     SET NEW.height = 777;
21     END IF;
22   IF NEW.pov_line = 'Above' THEN
23     SET NEW.pov_line = 666;
24     END IF;
25 END //
```

```
DELIMITER ;


INSERT INTO fp_trig (pid, age, pov_line, height, sex, pf_neg)
VALUES
('ccx1001', 2, 'Below', 170, 'M', 'Positive'),
('ccx1002', 2, 'Above', 55, 'F', 'Negative'),
('ccx1003', 19, 'Below', 175, 'M', 'Positive'),
('ccx1004', 31, 'Above', 20, 'F', 'Negative'),
('ccx1005', 23, 'Below', 180, 'M', 'Positive');

SELECT * FROM fp_trig;
```

SQL Code 14: Using TRIGGER to Insert Valid Data into a Dataset

Result Grid Filter Rows: Search Export:

pid	pov_line	age	height	sex	pf_neg
ccx1001	Below	2	777	M	Positive
ccx1002	666	2	55	F	Negative
ccx1003	Below	999	777	M	Positive
ccx1004	666	999	20	F	Negative
ccx1005	Below	999	777	M	Positive
rcx0190	Below	2	85	Male	Positive
rcx0276	Below	1	85	Male	Positive
rcx0341	Below	2	76	Female	Positive
rcx0343	Below	1	81	Male	Positive
rcx0378	Below	2	79	Female	Positive
rcx0521	Below	2	76	Male	Positive
rcx0727	Below	1	70	Male	Positive
rcx0812	Below	2	190	Female	Positive
rcx0818	Below	2	78	Male	Positive
rcx1014	Below	2	146	Male	Positive
rcx1049	Below	1	69	Female	Positive
rcx1389	Below	1	146	Male	Positive
rcx1643	Below	2	82	Male	Positive
rcx4425	Below	2	87	Female	Positive
rcx4430	Below	1	75	Male	Positive
rcx4466	Below	2	79	Male	Positive
rcx4470	Below	2	77	Male	Positive
rcx4493	Below	2	84	Female	Positive

Figure 15: Result: Using TRIGGER to Insert Valid Data into a Dataset

2.5.4 SQL Query3: Apply summary and aggregate functions to create a report

Example 5. Calculating Malaria Prevalence Across Different Areas of India. This SQL code retrieves information about malaria positivity rates for each district, stratified by sex. It first selects data from a table named fp_dat2, which presumably contains information about individuals who have been tested for malaria using a diagnostic test called PQCR. The WHERE clause specifies that only records with pqcr_neg values of 'Positive' or 'Negative' will be included in the analysis. It then groups the data by sex and district using the GROUP BY clause. The

SUM(pqcr_neg= 'Positive') function counts the number of individuals who tested positive for malaria in each group, and COUNT(*) counts the total number of individuals in each group. The proportion of individuals with a positive test result is calculated by dividing the number of positive test results by the total number of individuals and multiplying by 100. The ROUND() function is used to round the proportion to three decimal places.

The resulting table shows the proportion of individuals with a positive malaria test result for each sex-district combination, sorted by district and within each district, by the proportion of positive test results in descending order. This information can be used to identify areas with high malaria transmission and to target malaria control interventions.

```
1       SELECT
2       sex,
3       district,
4       ROUND(SUM(pqcr_neg= 'Positive')/COUNT(*)*100, 3) AS
          malaria_pos_proportion
5       FROM
6       fp_dat2
7       WHERE
8       pqcr_neg IN ('Positive', 'Negative')
9       GROUP BY
10      sex,
11      district
12      ORDER BY
13      district,
14      malaria_pos_proportion DESC;
```

SQL Code 15: Malaria Prevalence in India: A Geographical Analysis

```sql
SELECT
    sex,
    district,
    ROUND(SUM(pqcr_neg= 'Positive')/COUNT(*)*100, 3) AS malaria_pos_proportion
FROM
    fp_dat2
WHERE
    pqcr_neg IN ('Positive', 'Negative')
GROUP BY
    sex,
    district
ORDER BY
    district,
    malaria_pos_proportion DESC;
```

100% 2:409 1 error found

Result Grid Filter Rows: Search Export:

sex	district	malaria_pos_proport...
Male	Chennai	2.238
Female	Chennai	1.399
Male	Kheda	3.683
Female	Kheda	1.914
Female	Mahesana	0.000
Male	Mahesana	0.000
Male	Panchmahal	0.000
Female	Panchmahal	0.000
Male	Sundergarh	4.108
Female	Sundergarh	1.869

Figure 16: Result: Malaria Prevalence in India: A Geographical Analysis

Example 6. Calculating Malaria Prevalence Across Different Subgroup

```sql
SELECT sex, dwell_type, COALESCE(ani_indoor, 'Unknown') AS
   ani_indoor,
SUM(pf_neg='Positive') AS num_positive_cases,
COUNT(*) AS num_total_cases,
ROUND(SUM(pf_neg='Positive')/COUNT(*)*100,2) AS
   proportion_positive_cases
FROM fp_dat2
WHERE age BETWEEN 15 AND 65
AND   mala_diag = 'No'
GROUP BY dwell_type, sex, ani_indoor
ORDER BY proportion_positive_cases DESC;
```

SQL Code 16: Calculate Malaria Prevalance Based on Each Subgroups

```
461
462     SELECT sex, dwell_type, COALESCE(ani_indoor, 'Unknown') AS ani_indoor,
463       SUM(pf_neg='Positive') AS num_positive_cases,
464       COUNT(*) AS num_total_cases,
465       ROUND(SUM(pf_neg='Positive')/COUNT(*)*100,2) AS proportion_positive_cases
466     FROM fp_dat2
467     WHERE age BETWEEN 15 AND 65
468     AND   mala_diag = 'No'
469     GROUP BY dwell_type, sex, ani_indoor
470     ORDER BY proportion_positive_cases DESC;
471
```

100% 1:471 1 error found

Result Grid Filter Rows: Search Export:

sex	dwell_type	ani_indoor	num_positive_cas...	num_total_cas...	proportion_positive_cases
Female	House	Yes	162	650	24.92
Male	House	Yes	115	473	24.31
Male	Apartment	Yes	10	62	16.13
Male	House	No	60	428	14.02
Female	House	No	75	650	11.54
Female	Apartment	Yes	8	113	7.08
Female	Apartment	No	19	301	6.31
Male	Apartment	No	7	160	4.38
Male	Apartment	Unknown	0	25	0.00
Female	Apartment	Unknown	0	26	0.00

Figure 17: Result: Calculate Malaria Prevalance Based on Each Subgroup

The results show that women living in houses with animals indoors have the highest malaria prevalence (24.92%), followed by men living in houses with animals indoors (24.31%). On the other hand, men living in apartments without animals indoors have the lowest malaria prevalence (4.38%). This is expected as males usually have stronger immune systems than females, and animals can be an important carrier of the malaria bacteria. However, further rigorous statistical analyses (e.g., logistic regression models) should be conducted using professional software such as R or Python.

Example 7. Calculating Malaria Prevalence for a Specific Group with Multiple Criteria

```
SELECT
CASE
    WHEN Age BETWEEN 15 AND 25 THEN 'Teenager'
    WHEN Age BETWEEN 26 AND 35 THEN 'Younger Adult'
    WHEN Age BETWEEN 36 AND 55 THEN 'Adult'
    WHEN Age > 55 THEN 'Senior'
ELSE 'Unknown'
END AS age_group,
  ROUND(SUM(pqcr_neg='Positive')/COUNT(*)*100,2) AS
     pqcr_prop_positive,
  ROUND(SUM(pviv_neg='Positive')/COUNT(*)*100,2) AS
     pviv__prop_positive
FROM fp_dat2
WHERE sex = "Female"
```

```
  AND pov_line = 'Below'
  AND Age BETWEEN 15 AND 65
  GROUP BY age_group;
```

SQL Code 17: Malaria in Females Living Under Poverty Line in India

The output is presented below:

```
SELECT
  CASE
    WHEN Age BETWEEN 15 AND 25 THEN 'Teenager'
    WHEN Age BETWEEN 26 AND 35 THEN 'Younger Adult'
    WHEN Age BETWEEN 36 AND 55 THEN 'Adult'
    WHEN Age > 55 THEN 'Senior'
    ELSE 'Unknown'
  END AS age_group,
  ROUND(SUM(pqcr_neg='Positive')/COUNT(*)*100,2) AS pqcr_prop_positive,
  ROUND(SUM(pviv_neg='Positive')/COUNT(*)*100,2) AS pviv__prop_positive
FROM fp_dat2
WHERE sex = "Female"
AND pov_line = 'Below'
AND Age BETWEEN 15 AND 65
GROUP BY age_group;
```

100% 1:438 1 error found

Result Grid Filter Rows: Search Export:

age_group	pqcr_prop_positive	pviv__prop_positi...
Younger Adult	1.09	2.55
Teenager	1.67	1.67
Senior	0.00	2.25
Adult	1.49	2.23

Figure 18: Result: Malaria in Females Living Under Poverty Line in India

Example 8. Comparison of Malaria Testing Methods

```
SELECT
sex,
  CASE
    WHEN Age BETWEEN 15 AND 25 THEN 'Teenager'
    WHEN Age BETWEEN 26 AND 35 THEN 'Younger Adult'
    WHEN Age BETWEEN 36 AND 55 THEN 'Adult'
    WHEN Age > 55 THEN 'Senior'
  ELSE 'Unknown'
END AS age_group,
ROUND(SUM(pf_neg='Positive')/COUNT(*)*100,2) AS
   proportion_pf_positive,
ROUND(SUM(pqcr_neg='Positive')/COUNT(*)*100,2) AS
   proportion_pqcr_pos_cases,
ROUND(SUM(pviv_neg='Positive')/COUNT(*)*100,2) AS
   proportion_pviv_pos_cases
```

```sql
FROM fp_dat2
WHERE age BETWEEN 15 AND 65
GROUP BY sex, age_group
ORDER BY sex, proportion_pf_positive;
```

SQL Code 18: Comparison of Different Malaria Testing Methods Among Subgroups

```sql
SELECT
sex,
    CASE
        WHEN Age BETWEEN 15 AND 25 THEN 'Teenager'
        WHEN Age BETWEEN 26 AND 35 THEN 'Younger Adult'
        WHEN Age BETWEEN 36 AND 55 THEN 'Adult'
        WHEN Age > 55 THEN 'Senior'
    ELSE 'Unknown'
END AS age_group,
ROUND(SUM(pf_neg='Positive')/COUNT(*)*100,2) AS proportion_pf_positive,
ROUND(SUM(pqcr_neg='Positive')/COUNT(*)*100,2) AS proportion_pqcr_pos_cases,
ROUND(SUM(pviv_neg='Positive')/COUNT(*)*100,2) AS proportion_pviv_pos_cases
FROM fp_dat2
WHERE age BETWEEN 15 AND 65
GROUP BY sex, age_group
```

100% 2:469 1 error found

Result Grid Filter Rows: Search Export:

sex	age_group	proportion_pf_positive	proportion_pqcr_pos_ca...	proportion_pviv_pos_ca...
Female	Senior	15.50	1.50	3.50
Female	Teenager	16.50	1.17	1.17
Female	Adult	17.50	2.21	2.94
Female	Younger Adult	17.53	1.20	2.41
Male	Senior	14.05	1.08	1.62
Male	Younger Adult	16.29	2.24	5.11
Male	Adult	19.78	3.11	5.56
Male	Teenager	22.77	5.19	5.76

Figure 19: Result: Comparison of Different Malaria Testing Methods Among Subgroups

Example5. Summary of Malaria Cases by Sex and Poverty Line Status This SQL code provides a summary statistic on malaria cases among individuals living below the poverty line in India. The results include the count of malaria cases detected by any testing method, as well as the average, maximum, and minimum age of the cases. The results are stratified by sex, as it is a common confounding factor in epidemiological studies.

```sql
DROP TABLE  fp_agg;
CREATE TABLE fp_agg AS
SELECT
pid,
sex,
age,
district,
pov_line,
```

```
  CASE
    WHEN pqcr_neg <> 'Positive' OR pviv_neg <> 'Positive' OR
       pf_neg <> 'Positive' THEN 'Uncertain'
    WHEN pqcr_neg = 'Negative' AND pviv_neg = 'Negative' AND
       pf_neg = 'Negative' THEN 'Negative'
  ELSE 'Positive'
  END AS mal_stat
  FROM fp_dat2
  WHERE
    pqcr_neg='Positive' OR pviv_neg='Positive' OR pf_neg='Positive
       ';

  SELECT
      sex,
      pov_line,
      COUNT(*) AS count,
      MAX(age) AS max_age,
      MIN(age) AS min_age,
      ROUND(AVG(age), 1) AS avg_age
  FROM fp_agg
  WHERE mal_stat = 'Positive'
    AND pov_line = 'Below'
  GROUP BY sex, pov_line;
```

SQL Code 19: Summary of Malaria Cases by Sex and Poverty Line Status

```sql
DROP TABLE  fp_agg;
CREATE TABLE fp_agg AS
SELECT
    pid,
    sex,
    age,
    district,
    pov_line,
    CASE
        WHEN pqcr_neg <> 'Positive' OR pviv_neg <> 'Positive' OR pf_neg <> 'Positive' THEN 'Uncertain'
        WHEN pqcr_neg = 'Negative' AND pviv_neg = 'Negative' AND pf_neg = 'Negative' THEN 'Negative'
        ELSE 'Positive'
    END AS mal_stat
FROM fp_dat2
WHERE
    pqcr_neg='Positive' OR pviv_neg='Positive' OR pf_neg='Positive';

SELECT
    sex,
    pov_line,
    COUNT(*) AS count,
    MAX(age) AS max_age,
    MIN(age) AS min_age,
    ROUND(AVG(age), 1) AS avg_age
FROM fp_agg
WHERE mal_stat = 'Positive'
    AND pov_line = 'Below'
```

100% 1:623 2 errors found

Result Grid Filter Rows: Search Export:

sex	pov_line	count	max_age	min_age	avg_age
Female	Below	13	51	3	20.4
Male	Below	21	50	3	22.0

Figure 20: Result: Summary of Malaria Cases by Sex and Poverty Line Status

The analysis indicates that there are no significant age differences between males and females living under the poverty line, as well as no significant differences in maximum and minimum age. Additionally, it appears that males have a higher number of total malaria positive cases compared to females (21 vs. 13). However, it is important to note that further careful analysis is required in professional statistical software such as R or SAS before drawing any conclusive results..

3 Database Administration (Part 3)

3.1 Data Audit

To ensure the accuracy and reliability of the fp database, a data audit plan will be implemented. Regularly scheduled audits will be conducted using log files generated by Rmarkdown, as well as commands such as "table" and "summary" to check for completeness, accuracy, and consistency. Any discrepancies or errors identified during the audit process will be documented and addressed accordingly. In addition, the data audit plan will include procedures for data quality control, such as reviewing the data collection process, data entry procedures, and data cleaning procedures. These measures will help ensure that the data in the fp database is accurate and reliable, and can be used with confidence for analysis and reporting. The audit plan will be reviewed and updated on a regular basis to ensure that it remains effective and relevant over time.

3.2 Data Security

To protect the confidentiality and privacy of research participants, the fp database will be secured through physical and computer hardware measures. Physical files will be stored in locked cabinets or secure offices with limited access, and computer hardware will have strong passwords and firewalls, with two-step verification and access rights based on staff roles. Sensitive variables such as SSN will be removed from the dataset and participant data such as DOB will be obfuscated. The system will be monitored for security breaches, and staff will be trained on data protection policies and procedures.

3.3 Dataset Back up

In fp database, all collected data, associated files, scripts, and documentation are regularly backed up using a dynamic approach that efficiently captures changes while minimizing storage requirements. Backups are stored both in the cloud (such as Dropbox or AWS) and on physical media like external hard drives to ensure redundancy and accessibility. To determine the appropriate length of time to store backups, the value and sensitivity of the data are considered. In case there are highly sensitive datasets in fp database, it should be backed up for several years. A comprehensive backup strategy that considers the dataset's unique characteristics ensures that the data is safe, secure, and available for future use.

3.4 Disaster Recovery Plan

In case of unpreventable natural disasters, unforeseen accidents or data corruption issues with fp database, a comprehensive disaster recovery plan is in place. The premises are evacuated immediately, all personnel involved in managing the dataset are notified, and physical backups are transferred to a secure off-site location. Backups from cloud storage are also restored to a new server or instance as soon as possible. To create a complete snapshot of the dataset in a structured format that is easily restorable, a MySQL dump is used with the code "sudo mysql dump -u [user] -p [fp] ¿ [filename].sql". It provides an additional layer of protection against data loss or corruption, particularly for sensitive datasets in fp database. By implementing this strategy, the risk of data loss is minimized, and the fp database remains available and functional.

4 Reference

1. Al-Awadhi, M., Ahmad, S., & Iqbal, J. (2021). Current status and the epidemiology of malaria in the Middle East Region and beyond. Microorganisms, 9(2), 338.

2. de Andrade Schramm, J. M., Campos, M. R., Emmerick, I. C. M., Pereira Mendes, L. V., Mota, J. C., & Junior, S. H. A. S. (2016). Spatial analysis of neglected diseases in Brazil, 2007 to 2009. Tempus–Actas de Saúde Coletiva, 10(2), ág-119.

3. Yuen, C. M., Puma, D., Millones, A. K., Galea, J. T., Tzelios, C., Calderon, R. I., ... & Keshavjee, S. (2021). Identifying barriers and facilitators to implementation of community-based tuberculosis active case finding with mobile X-ray units in Lima, Peru: a RE-AIM evaluation. BMJ open, 11(7), e050314.

4. Malaria, R. B. (2005). World malaria report 2005. World Health Organization and UNICEF.

5. Pull, J. H., & World Health Organization. (1972). Malaria surveillance methods, their use and limitations (No. WHO/MAL/72.767). World Health Organization.

6. Oduro, A. R., Maya, E. T., Akazili, J., Baiden, F., Koram, K., & Bojang, K. (2016). Monitoring malaria using health facility based surveys: challenges and limitations. BMC Public Health, 16(1), 1-9.

7. Ziegler, P., & Dittrich, K. R. (2007). Data integration—problems, approaches, and perspectives. Conceptual modelling in information systems engineering, 39-58.

8. Ruhamyankaka, E., Brunk, B. P., Dorsey, G., Harb, O. S., Helb, D. A., Judkins, J., ... & Tomko, S. S. (2019). ClinEpiDB: an open-access clinical epidemiology database resource encouraging online exploration of complex studies. Gates Open Research, 3.