

# Creating a Runs Database Under MIDAS and MySQL

Amira Malik  
Dr. Mandy Rominsky

## Abstract

At the Fermilab Test Beam facility, scientists test their equipment against the provided test beams. Since the equipment needs to be accurately tested, the parameters of the beam used must be documented extensively. The Test Beam Facility has a data acquisition system that can record the nature of each beam. However, the state of the beam and facility instruments needs to be recalled at a later time during analysis. . In order to recall the conditions, Test Beam Facility needs a runs database: a way to store and recall information by assigning tags or descriptors to each run.

During the summer, I was tasked with learning about data acquisition and recollection in order to set up a runs database. I had to learn the coding language SQL and how to use applications like MIDAS and MySQL as well as how the two programs interact with each other. In the following paper I discuss the concepts of how data is acquired, how databases store and recall information, and how MySQL and MIDAS interface with each other.

## Introduction

### Context of this work

The Test Beam Facility produces high energy charged particle beams with a variety of particle types and range of energies for users to test their equipment. Since scientists need to test their equipment, everything about the nature and type of the beam with which the scientists are working must be known and recallable. Therefore, the Test Beam Facility uses data acquisition to collect data from facility instrumentation to observe such beam parameters.

However, the Test Beam Facility needed a way to search for the beam parameters of previous runs.

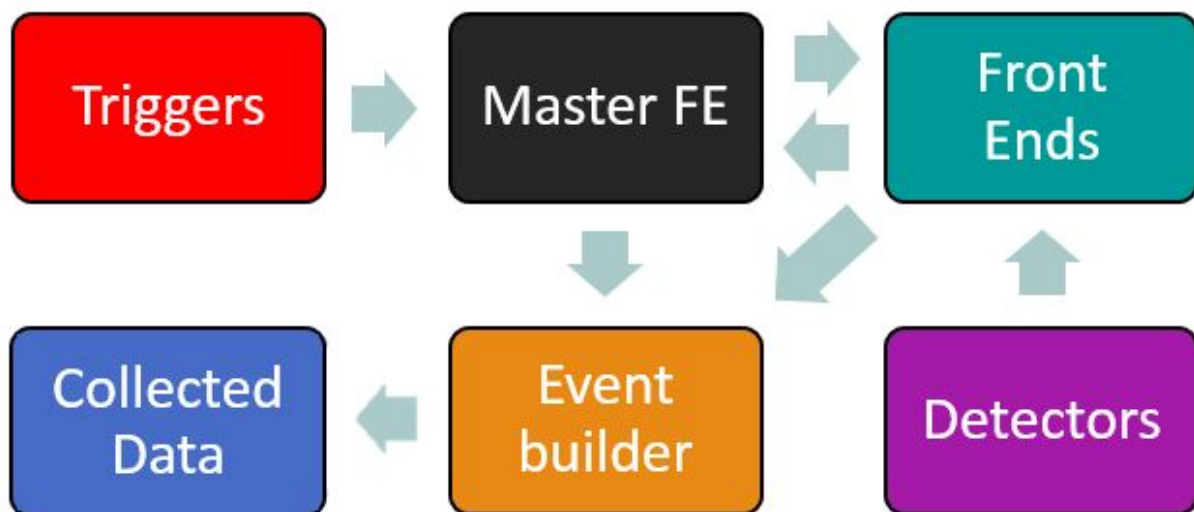
## Concepts

### Data Acquisition

Data acquisition is just that: acquiring data. It's essentially hardware that converts physical events into data that can be manipulated and stored into a computer. An example would be some sort of detector or PMT creating an analog signal from something like a muon passing through it. This detector would then send that analog signal to an ADC, or analog to digital converter, which then creates a digital signal to go to a front end, a program contained in a computer that interfaces with a specific detector.

Data acquisition can be as simple as a detector that changes an electrical signal into code for a computer to store. But as the data gets more complex, so should the way data is acquired.

A general model of data acquisition at the Fermilab Test Beam Facility is shown below.



**Figure 1.** A basic visual representation of how parameters of a run is collected through a data acquisition system.

With this model, the Test Beam Facility can collect integrated data from all the instruments. A special trigger server receives signals from the accelerator division clock and trigger paddles to indicate beam is starting. The Master frontend sends the trigger signal to the other frontends to prepare them for data taking. The detectors then record data to send to their

respective frontends. Finally, all the frontends send their data to the event builder, which puts all the information into a package, which becomes a single event in the run.

At the Test Beam Facility, this model is facilitated by the Maximum Integrated Data Acquisition System (MIDAS). Users can control and monitor MIDAS through a command line terminal or via an online web control panel.

The acquired data needs to be organized using a database system. .

## Databases

A database is a collection or system of multiple related tables that stores rows of data with associated descriptors or tags in the form of columns in order to aid the storage and recollection of data. Therefore, databases circumnavigate an issue that would arise if data were simply stored by nothing but a file name. Without any descriptors, data would be vague and non-descriptive.

Databases are written in SQL, or Structured Query Language, to create tables. However, a table made by SQL is not necessarily the same as a table made in a spreadsheet. With SQL, you can make more complicated queries to get more specific data rather than having to look at data in a largely static data table.

As shown below, using SQL can grant you easier access when trying to find specific information in a set of data.

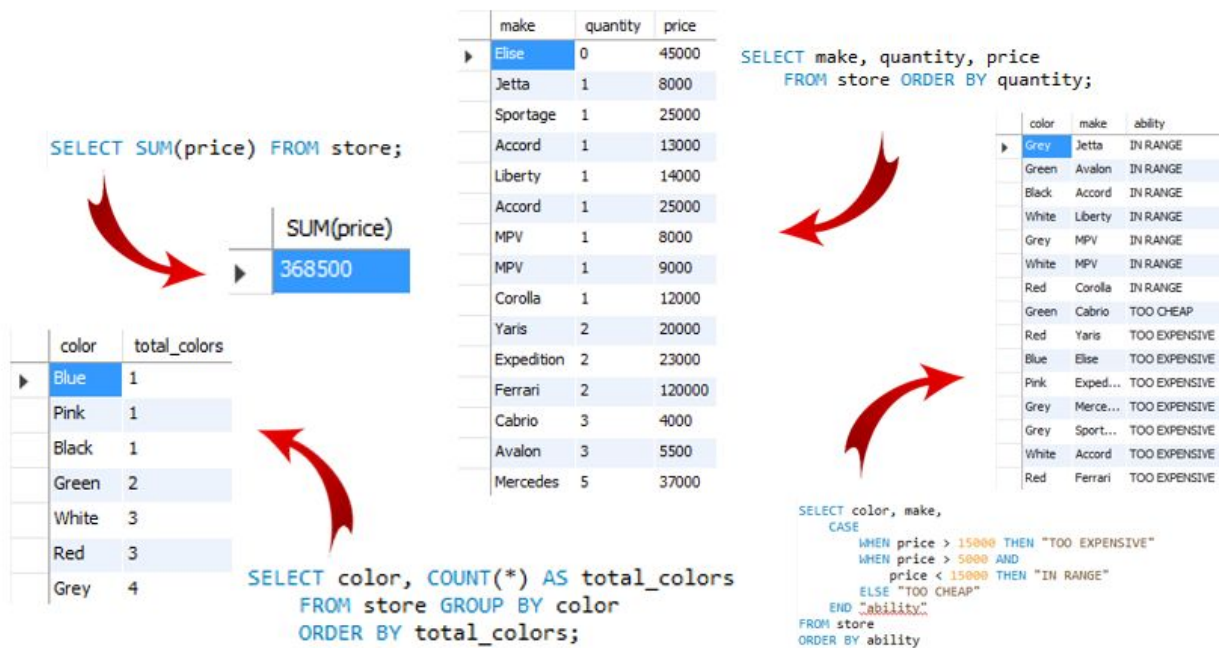
```
USE cars;
CREATE TABLE store (id INTEGER PRIMARY KEY, color TEXT,
    make TEXT, quantity TEXT, price INTEGER);
INSERT INTO store VALUES(1, "Red", "Yaris", 2, 20000);
INSERT INTO store VALUES(2, "Grey", "Jetta", 1, 8000);
INSERT INTO store VALUES(3, "Blue", "Elise", 0, 45000);
INSERT INTO store VALUES(4, "Green", "Cabrio", 3, 4000);
INSERT INTO store VALUES(5, "Pink", "Expedition", 2, 23000);
INSERT INTO store VALUES(6, "Green", "Avalon", 3, 5500);
INSERT INTO store VALUES(7, "Grey", "Mercedes", 5, 37000);
INSERT INTO store VALUES(8, "Grey", "Sportage", 1, 25000);
INSERT INTO store VALUES(9, "Black", "Accord", 1, 13000);
INSERT INTO store VALUES(10, "White", "Liberty", 1, 14000);
INSERT INTO store VALUES(11, "White", "Accord", 1, 25000);
INSERT INTO store VALUES(12, "Grey", "MPV", 1, 8000);
INSERT INTO store VALUES(13, "White", "MPV", 1, 9000);
INSERT INTO store VALUES(14, "Red", "Corolla", 1, 12000);
INSERT INTO store VALUES(15, "Red", "Ferrari", 2, 120000);

SELECT * FROM store;
```



	id	color	make	quantity	price
▶	1	Red	Yaris	2	20000
	2	Grey	Jetta	1	8000
	3	Blue	Elise	0	45000
	4	Green	Cabrio	3	4000
	5	Pink	Expedition	2	23000
	6	Green	Avalon	3	5500
	7	Grey	Mercedes	5	37000
	8	Grey	Sportage	1	25000
	9	Black	Accord	1	13000
	10	White	Liberty	1	14000
	11	White	Accord	1	25000
	12	Grey	MPV	1	8000
	13	White	MPV	1	9000
	14	Red	Corolla	1	12000
	15	Red	Ferrari	2	120000

**Figure 2.** A set of SQL commands creating a sample table in an existing database, 'cars', and a query recalling the entire table (right), with the results shown (left).



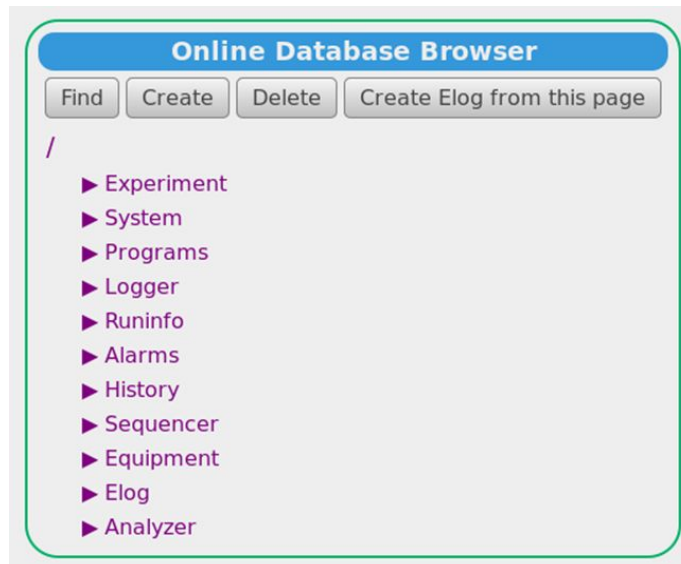
**Figure 3.** Different SQL queries of the table created in fig. 2; each query recalls and / or classifies different information with each respective result also shown.

Armed with SQL and the ability to store information, the Test Beam Facility could have simply made an SQL table by hand each time an experiment on the Test Beam was run. But to insert so many new rows with maybe even 100s of columns would be too time consuming. To solve this problem, MySQL was chosen because of it's existing implementation in MIDAS. MySQL will create a database and host a database server.

## MIDAS & MySQL

MIDAS collects data and MySQL can create and write to a database, the next step is to integrate the two applications. . Within MIDAS, we use the native online database editor,(ODBEdit), inside the Online Database Browser, which facilitates the creation of a runs database.

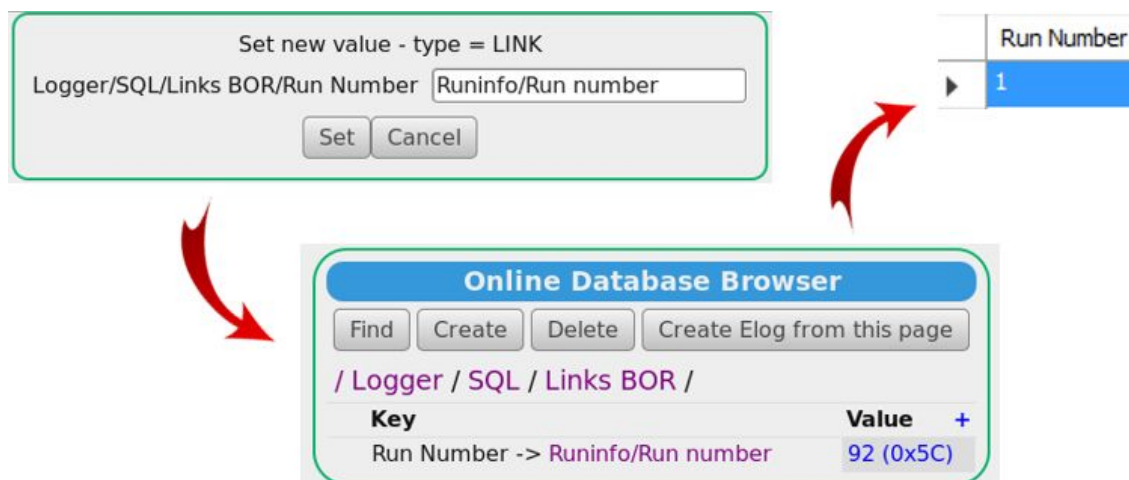
The ODB is structured like a directory with subfolders and various information within each folder.



**Figure 4.** The interface of the Online Database Browser within the MIDAS web control panel. Each string is representative of a different directory.

All the data collected by MIDAS for the run is present in the ODB, we need MIDAS to write select data into an accessible table.

The application mlogger takes the data found in MIDAS and writes it to a selected database. We need to point mlogger to the information found MIDAS via a symbolic link. Once mlogger knows where the specific information associated with a run exists, it can automatically write it once a run has started. All we need to do to see the table is to log into MySQL and recall the information to where we linked mlogger.



**Figure 5.** Linking mlogger to information found in a folder, 'Runinfo,' (top left), the link shown in the ODB (bottom), and the resulting table after one run (top right).

And with each new experiment, there will be a new run number, and another row of information. And for each data point, or "column" of information we would want, we'd just point mlogger to where MIDAS collects that data in order for said data to be written to an accessible table.

## **The Future**

Creating a database will help the scientists who come to the Test Beam Facility track the conditions of the beam. . Currently, there is a lot of debugging to work on regarding the interactions between MIDAS and MySQL. During my time I worked mostly on trying to understand the concepts of data acquisition and subsequent storage and installing the needed applications.

After installation, MIDAS was indeed recording data and MySQL was writing test data and parameters. However, after a day or so, both programs started to have complications. The data being written with MySQL is labeled as `unknown` and other technical errors are coming up. These errors will have to be addressed before a working database is set up.

This database will give scientists easier access to the parameters of each run in order for them to fully understand the beam with which they were testing their equipment. I started the process of setting up a database by installing MIDAS and understanding the basics of mlogger's implementation, my documentation of using MIDAS in conjunction with MySQL will guide future scientists to implement fully the database at the Test Beam Facility.

## **Acknowledgements**

Fermilab is operated by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the United States Department of Energy. QuarkNet is an educational program sponsored by the National Science Foundation and the Department of Energy whose aim is to support science education in schools by establishing a nationwide network of science teachers. This work was supported by CRADA FRA-2014-0022-02 agreement between Fermi Research Alliance LLC and the University of Notre Dame for the 2015-16 QuarkNet summer program.

I would also like to thank the following individuals who made my time at Fermilab enjoyable and educational:

George Dzuricksos  
Robb Gotlund  
Brendan Kiburg  
Mishelle Mironov  
Kathleen Murphy  
Todd Nebel  
Paul Nebres  
Mandy Rominsky  
Geoff Savage  
Eugene E Schmidt Jr.  
Kevin Shuman  
Chris Stoughton  
Megan Szubert  
Laura Thorpe

## **References**

- Heng, C. (2015, December 16). What is MySQL? What is a Database? What is SQL?  
Retrieved July 25, 2016.
- Scientific DB Applications » ConDB. (2014, September 16). Retrieved July 25, 2016.
- Axani, S. N., Kirby, C., & Conrad, J. M. (2016, June 6). The Desktop Muon Detector: A simple,  
physics-motivated machine- and electronics-shop project for university students.  
Retrieved July 25, 2016.
- MIDAS Documentation. (n.d.). Retrieved July 25, 2016, from  
[https://midas.triumf.ca/MidasWiki/index.php/Main\\_Page](https://midas.triumf.ca/MidasWiki/index.php/Main_Page)