Chris Mello
Scott Edwardsen
Project Group 53
CS 340 – Winter 2019 - 3/19/2019
Project and Database Outline

Chemistry Database

http://flip2.engr.oregonstate.edu:44551/

Chemistry Production Process Database

Project Outline

On January 1stScott was put in charge of the Chemistry Department at a food testing company. The chemistry department does a wide variety of testing including the nutritional labels you see in the grocery store. Most of the company's revenue comes from microbiological testing (micro department) and most of the systems in place do not capture the information needed to properly manage the chemistry department. At the company the operations department is judged by how well they control costs. This is calculated by dividing the number of test results by the number of labor hours. This calculation works well for the micro department but not for chemistry. In Chemistry the equation is much more complicated. Some of the additional variables include; how many similar tests are being run at the same time (batch size), how many retests are being done, how long does it take to set-up the instrument, etc. For example, if your batch size is 20 your cost per test is $2 but if your batch size is 5 then your cost per test is $55. This same cost model does not hold true for micro. The chemistry production process database is being developed to capture some of these variables to get a more accurate picture of the department's performance. We are calling this the Chemistry Metrics Database which will allow management to make decisions based on the performance of the department.

Database Outline, in Words

The entities in the database are:

- Certificate of Analysis (COA) – This is the document that customers receive with their test results. COA and report are terms that are used interchangeably. It has the following attributes:
  - **Report ID**: This number is automatically assigned to each COA when they are logged into the database. An auto-incrementing number which is the primary key
  - **Customer Name**: Name of the customer who requested the testing which is a string of maximum 200 characters. It cannot be blank and there is no default.
  - **Report Status**: The status of the COA which includes; created, received, in progress, complete, released. This is a string which a maximum of 25 characters. It cannot be blank, and the default is created.
  - **COA send date**: When the COA was emailed to the customer. This is a date time field. It can be blank if the Report Status is not released.
- Sample
  - **Sample ID**: This number is automatically assigned to each Sample when they are logged into the database. An auto-incrementing number based off of the report number which is the primary key. For example, if the report number is 999123 then the first sample will be 999123001 and the second sample will be 999123002, etc.
  - **Report ID**: This number defines which COA the samples have been assigned to. This cannot be blank and there isn't a default. A sample always has to be assigned to a report.
  - **Received date**: When the samples were received by the lab. This is a date time field. It can be blank if the samples have not been received yet.
  - **Test ID**: This number defines which testing has been assigned to the samples. This can be blank if there isn't any testing assigned and there isn't a default.
- Test
  - **Test ID**: This number is automatically assigned to each Test when they are logged into the database. An auto-incrementing number based off of the sample number which is the primary key. For example, if the sample number is 999123001 then the first test will be 999123001001 and the second test will be 999123001002, etc.
  - **Test Code**: Code that identifies the testing that needs to be done which is a string of maximum 200 characters. It cannot be blank and there is no default.
  - **Rush Code**: Code that identifies if rush testing is required. Customer can request a rush which increases their price but decreases the turnaround time.

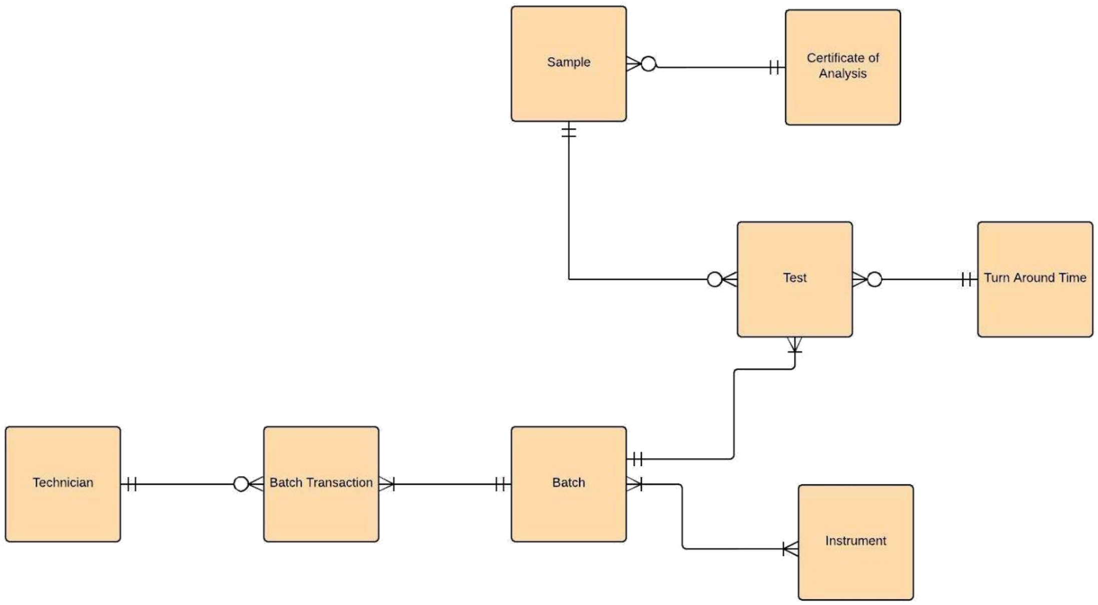
    This is a string of maximum 50 characters. It cannot be blank, and the default is no rush.
    - **Result**: Test result from the instrument which is a string of a maximum of 50 characters. It can be blank and there is no default.
    - **Test Reason Code**: Code that identifies why the testing is being done. Some of the reason codes include; initial, retest, qc failure, instrument failure, customer spec. This is a string of a maximum of 50 characters. It cannot be blank and there isn't a default.
    - **Start of Testing**: When the tests were started by the lab. This is a date time field. It can be blank if the testing hasn't been started.
    - **End of Testing**: When the tests were completed by the lab. This is a date time field. It can be blank if the testing hasn't been completed.
    - **Batch ID**: This number defines which batch (group of testing) the tests have been assigned to. This can be blank if the batch hasn't been created and there isn't a default.
- Turn Around Time
    - **Test Code**: Code that identifies the testing that needs to be done which is a string of maximum 200 characters. It cannot be blank and there is no default. The combination of test code and rush code create the primary key.
    - **Rush Code**: Code that identifies if rush testing is required. Customer can request a rush which increases their price but decreases the turnaround time. This is a string of maximum 50 characters. It cannot be blank, and the default is no rush. The combination of test code and rush code create the primary key.
    - **Due Date Offset**: The amount of time in minutes the lab has to send results to a customer before the results are considered late. This is a number that cannot be blank and there isn't a default.
- Batch
    - **Batch ID**: This number defines how the tests are grouped together. A test can only be a member of one batch, and this is the primary key and it cannot be blank and there isn't a default.
    - **Create Date**: When the batch was created by the lab. This is a date time field. It cannot be blank.
    - **Complete Date**: When the batch was completed by the lab. This is a date time field. It can be blank.
    - **Serial Number**: Identifies which instruments the batch was run on. This is a number and it cannot be blank.
- Instrument
    - **Serial Number**: Identifies the instrument where the test was running. This is a number and it cannot be blank because this is the primary key.
    - **Instrument Name**: Name of the instrument where the test was running which is a string of maximum 200 characters. It cannot be blank and there is no default.
- Batch Transaction
    - **Transaction Number**: This number is automatically assigned to each transaction when they are logged into the database. An auto-incrementing number which is the primary key.

    - **Batch ID**: This number defines which batch this transaction is associated with. It cannot be blank and there isn't a default.
    - **Technician ID**: This defines which technician did the work. This is a number that cannot be blank and doesn't have a default value.
    - **Batch Status**: The status of the batch which includes; created, in progress and complete. This is a string which a maximum of 25 characters. It cannot be blank, and the default is created.
- Technician
    - **Technician ID**: This number is automatically assigned to each technician when they become an employee at the company. An auto-incrementing number which is the primary key. This defines which technician did the work.
    - **First Name**: First name of the technician who did the testing which is a string with a maximum length of 50 characters. It cannot be blank and there is no default.
    - **Last Name**: Last name of the technician who did the testing which is a string with a maximum length of 50 characters. It cannot be blank and there is no default.

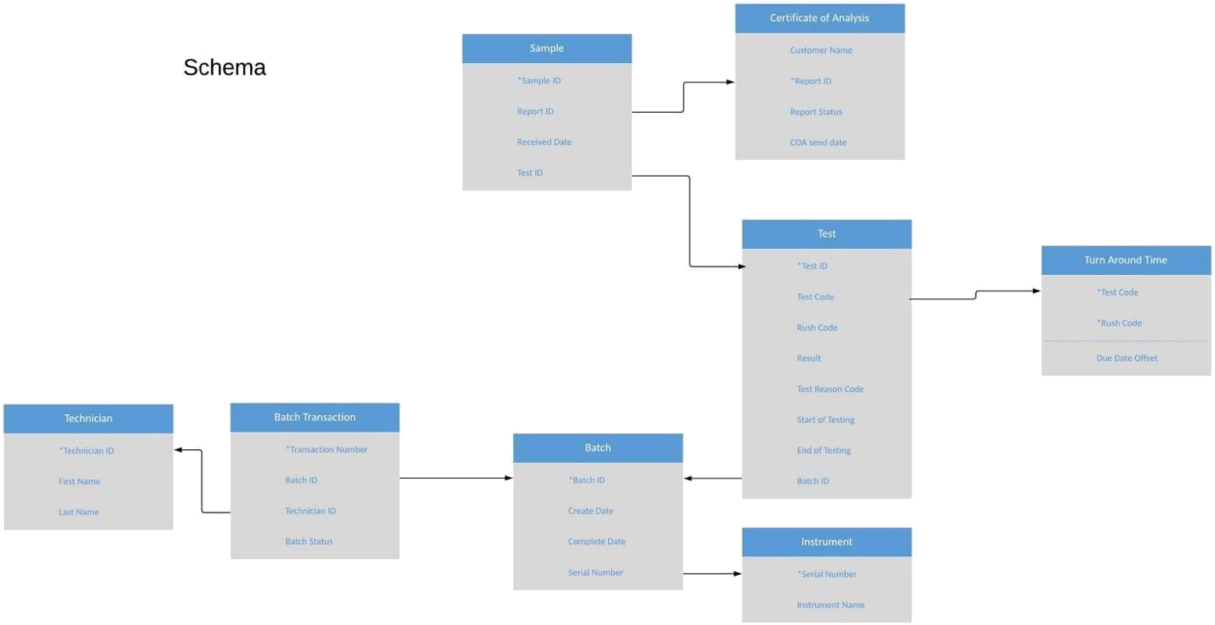
The relationships in my database are:

- Samples have a Certificate of Analysis (COA) – A samples can only be associated with one COA but a COA can have zero or many samples. So, Sample and COA are in a zero or many to one relationship.
- Tests have a Sample – A test can only be associated with one sample, but a sample can have zero or many tests. So, test and sample are in a zero or many to one relationship.
- A test has a turnaround time – A test can only be associated with one turnaround time, but a turnaround time can be associated with zero or many tests. So, test and turnaround time are in a zero or many to one relationship.
- A test has one batch – A test can only be associated with one and only one batch, but a batch can have one or many tests. So, test and batch are in a one or many to one relationship.
- A batch can be run on multiple instruments – A batch can be run on multiple instruments and an instrument can run multiple batches. Therefore, batch and instruments are in a many to many relationships.
- A batch transaction has one batch – A batch transaction can only be associated with one and only one batch, but a batch can have one or many batch transactions. So, batch transaction and batch are in a one or many to one relationship.
- A technician does work by completing a batch transaction – A batch transaction can only be associated with one technician, but a technician can be associated with zero or many batch transactions. So, batch transaction and technician are in a zero or many to one relationship.

ER Diagram


Sample
Certificate of Analysis
Test
Turn Around Time
Technician
Batch Transaction
Batch
Instrument

Schema


Sample
*Sample ID
Report ID
Received Date
Test ID
Certificate of Analysis
Customer Name
*Report ID
Report Status
COA send date
Test
*Test ID
Test Code
Rush Code
Result
Test Reason Code
Start of Testing
End of Testing
Batch ID
Turn Around Time
*Test Code
*Rush Code
Due Date Offset
Technician
*Technician ID
First Name
Last Name
Batch Transaction
*Transaction Number
Batch ID
Technician ID
Batch Status
Batch
*Batch ID
Create Date
Complete Date
Serial Number
Instrument
*Serial Number
Instrument Name

Feedback by the peer reviewer:

Hi Christopher,

They still don't seem to work successfully, I believe you are working on it. You have made a clear framework and layout, you need to read the user's operation and the contents of the database. As I said in your step 5 post on the discussion board.

Here is the example code for display data from DB, hope it helps you:

```
router.get('/', function(req, res){
var callbackCount = 0;
var context = {};
context.jsscripts = ["deleteperson.js","filterpeople.js","searchpeople.js"];
var mysql = req.app.get('mysql');
getPeople(res, mysql, context, complete);
getPlanets(res, mysql, context, complete);
function complete(){
callbackCount++;
if(callbackCount >= 2){
res.render('people', context);
}

}
});
```

If you have problems, the piazza is a good platform for solving it.

Peng

Hello,

Looks like you are still getting some work done on this. It doesn't look like delete is set up on the front end, or at least I was not able to find it. You do have one Delete query set up in your manipulation file. That looks like it is set up right. From a front end perspective, they should all work the same or similar, so if you set up the rest similarly you should be in good shape.

On your database creation you might think about adding some constraints or callbacks. If you delete a sample for example, you probably want all the tests deleted to or an error to pop up that says you can't delete a sample while tests are still present. You can choose however you want to handle those cases, but something should probably happen. This is true for all of your tables that are connected.

Overall I like the idea and I think the website will look pretty good when you get some data in there.

Good luck with the rest of your project, Actions based on the feedback:

- Pengs response was very helpful
- We were able to get our handlebars working more interactively with the database.
- At this point our website was mildly interactive with the database, however, we began researching more into making multiple calls per request etc.

Feedback by the peer reviewer:

Hi Christopher,

I have encountered the same problem with you before. That is my page cannot show up all operator for add/delete/update, but it can work well in my database. I think you need to check the function that reads the data and whether it works successfully. If possible, check if your variable name is the same as the variable name in your HTML. This requires step-by-step checking, and you can temporarily ignore other functions to check a function. If you have problems with creating a function, You can read our class tutorial. it gave the example of that. https://oregonstate.instructure.com/courses/1727186/pages/week-8-learn-using-javascript-and-nodejs?module_item_id=18521066

Peng

I am not able to get anything working through the website. It looks like you have the basic CSS and HTML set up for the project, but I couldn't get any of your submission forms to work to test your update.

I looked at your manipulation file and it looks like you have a couple of the UPDATE queries set up. They look like they will work, so if you use the same methodology to create your other update queries I think you should be alright.

Good Luck with the rest of your project,

Actions based on the feedback:

- We worked to get our handlebars page to work properly. At the time of the comment, we were unable to create the functionality from the user to the database
- We used the source provided to research how to get our webpage to become more functional
- We were currently in the process of rolling out our function to more interactively work with the database, the comments relate directly to this issue.

Feedback by the peer reviewer:

I think most of what we can provide at this point is nit-picking, but here are my thoughts for what it's worth:

- I like the page style so far - very clean!
- General - the site seemed to be looking for some missing resources "GET http://flip2.engr.oregonstate.edu:45451/homepageStylesheet.css net::ERR_ABORTED 404 (Not Found)
  favicon.ico:1 GET http://flip2.engr.oregonstate.edu:45451/favicon.ico 404 (Not Found)"
- The 'Report ID Lookup' label is blending into the darker bar and butting up against the serach box
- Probably going to want some 'Home' content before too long...
- 'Lookup' - I couldn't find 'Report ID' that retruend data - maybe a summary report or something would be useful to drill down to that point (although it sounds from your write-up like the users will have something to start with so that may not be necessary)
- 'Input Data' - I used 'WLC' in all of the identifiers I made up (so if I created any junk rows that's how you can clean up after me). But I'm guessing one or more of those values is a foreign key and my row wasn't created. It will be good to see feedback on whether the row was inserted and any reasons why not in those cases
- 'Assign Batch To Instrument' - This is another one where production users will probably have additional data that will tell them the correct ID's but I haven't got it and could use a report or something to get me pointed in the right direction
  I guess my overall 'unfamiliar user' impression is that it needs a lot of information from me (specific ID's) to function, and might want to prompt me with lists or selects of these ID's if that's an appropriate user experience).

It's looking pretty slick, and seems to be doing a good job of capturing a very complex concept. I'm looking forward to seeing the next iteration. Thanks for sharing!

Hi Christopher! Your page looks really nice so far! The design is clean and easy to navigate. Some notes (though I know it's all still a work in progress):

-Some entries/pages might benefit from short instructions (i.e. should i enter an integer representing a certain number of days for Due Date Offset on the Turn Around Time page? I was also unsure what should be input for the fields in the Input Data page). If there is a required format/input, be sure to let the user know to help prevent user input errors.

- On the Turn Around Time page, Test Code and Rush Code might be easier to navigate as drop down menus since it's pulling from the Test table. Consider this for any field that is utilizing existing data from a table.

-You're probably still working on it, but remember to include delete and update functionality on your page.

Overall, it's looking good!

Hi Christopher,

Compared to the last time, your database changes very well, I like your layout. Here are some comments:

- Your project doesn't look complete, it doesn't read the database correctly, I didn't see any data information.
- Similarly, I can't insert any data, I don't know if it will be implemented in the back-end, but the front-end does not display data information.
- I didn't see any updates or deletes for the entire project, although this is the next project step, don't forget to add them. This will cause a deduction on your score.
- In addition, I can't see any relationship about some table at the front-end, I believe you can do better in your next step

Peng

Actions based on the feedback:

- We began changing the structure of our navbar to include more drop downs for ease ability.
- We began implementing the delete functionality for 2 of our foreseen most variable tables
- We didn't take any action on the instruction of the database as the users of the database will be the workers at the company, who will already have a knowledge of the inputs
- We began using handlebars to implement the form to be more dynamic and interactive

Feedback by the peer reviewer:

"In your project outline, I think you should be clear about what your database does. For example, what is recorded in your database, is it record a cost in the chemical production process? Or is it a record of the production and instruments in the chemical production process?

In your relationship of the database, "A test has only one sample and a sample has 0 or many tests." There is a small mistake in your ER diagram, the diagram shows the opposite with your described (it is a test has 0 or many tests and a sample has only one test).

I like your schema, it is clear to show their relationship with the arrow.

Peng“

Actions based on the feedback:

- We have altered the ERD to reflect our relationships , which was initially incorrect in regard to test and sample
- We have added additional details to the project outline

Upgrades to the Draft Version:

- Small changes were made throughout the draft however, the major ones are listed below:
- We have altered the ERD to reflect our relationships , which was initially incorrect in regard to test and sample
- Added the sentence: We are calling this the Chemistry Metrics Database which will allow management to make decisions based on the performance of the department.

Feedback by the peer reviewer:

I really like projects built on real-world cases, and this is a good one. They can be pretty complicated, and it feels like there were some nuances of this setup that were pretty tricky. Nice job capturing what is clearly a sophisticated system!

Here are my thoughts - for what they're worth:

- It felt like CUSTOMER ought to be an entity. Of course there is nothing that says it has to be, and you needed to stay on 3 pages, but it seems highly likely you'll want to associate multiple tests for a single customer, and probably capture contact details and other customer attributes as well
- I wonder if REPORT STATUS might do better as a simple code that your software then interprets to a human-readable string, but the way you've got it can be displayed as-is
- That's a pretty sophisticated system of using the parent RECORD key and an auto-incrementing local number. I don't know enough to know how tricky that will be, but it sounds sophisticated!
- Should TEST ID be "...automatically assigned to each TEST..." rather than "...each SAMPLE..." as written?
- TEST.TEST_CODE says "...needs to be bone..." instead of "...needs to be done..."
- TURN_AROUND_TIME.TEST_CODE also has the word "bone" in it, and the double use of the word "that" in the first sentence might be what you meant but could be more clear with slightly different wording

- I struggled to understand why TURN AROUND TIME is an entity and not an attribute of TEST
- I wonder if the single BATCH attribute SERIAL NUMBER will be adequate to represent the many-to-many relationship between BATCHes and INSTRUMENTs
- I had a hard time figuring out the purpose of BATCH TRANSACTION entities
- BATCH_TRANSACTION.BATCH_STATUS should probably be BATCH_TRANSACTION.BATCH_TRANSACTION_STATUS, right?
- Your ER diagram shows a one-to-one-or-many relationship between SAMPLE and TEST. I think the arrow in the diagram might be reversed...?

It was fun picturing the actual procedures and processed represented, and trying to wrap my brain around everything. Great job capturing a complex subject. I hope that my comments are helpful!

Actions based on the feedback:

1. We reviewed the changes suggested for customer by creating it as its own entity. We decided not to make any changes because the database is a Chemistry Metrics database and we only need the customer's name
2. Report status wasn't changed from a string to a code because in LIMS (Laboratory Information Management System) this information is stored as a string. We don't want LIMS to store it as a string and the Chemistry Metrics database store it in a different format because it is likely both databases will be imported into a data warehouse.
3. "Test ID: This number is automatically assigned to each Sample when they are logged into the database" Changed the word Sample to test
4. Test.TestCode: Code that identifies that testing that needs to be bone which is a string of maximum 200 characters. Change that to the and bone to done.
5. TurnAroundTime.TestCode: Code that identifies that testing that needs to be bone which is a string of maximum 200 characters. Change that to the and bone to done.
6. Reviewed comments and no action was taken. This is due to customers having different turn around times. "I struggled to understand why TURN AROUND TIME is an entity and not an attribute of TEST"
7. Reviewed comments and no action was taken. There is a many to many relationships because a batch can run on multiple instruments and an instrument can have multiple batches. "I wonder if the single BATCH attribute SERIAL NUMBER will be adequate to represent the many-to-many relationship between BATCHes and INSTRUMENTs"
8. Reviewed comments and no action was taken. The purpose is to create an audit as a batch changes. "Ihad a hard time figuring out the purpose of BATCH TRANSACTION entities"
9. Reviewed comments and no action was taken. No we want to capture the batch status not the transaction status. "BATCH_TRANSACTION.BATCH_STATUS should probably be BATCH_TRANSACTION.BATCH_TRANSACTION_STATUS, right?"
10. Reviewed comment and not action taken. A sample can have one or many tests but a test can only be associated with one sample. "Your ER diagram shows a one-to-one-or-many relationship between SAMPLE and TEST. I think the arrow in the diagram might be reversed...?"

Upgrades to the Draft Version:

1. "Test ID: This number is automatically assigned to each Sample when they are logged into the database" Changed the word Sample to test
2. TurnAroundTime.TestCode: Code that identifies that testing that needs to be bone which is a string of maximum 200 characters. Change that to the and bone to done.

Feedback by the peer reviewer:

Hi Chris and Scott:

I did go back and look at the only example Schema given so far in the course material and it showed just arrows for relationships, not the 0 or 1 to many kind of lines. The **View the Slides PDF**within **Week 3 - Learn**shows that approach.

When I Google search the topic, I see it both ways, so it may not matter....

Actions and Upgrades based on the feedback:

We updated the Schema to fit the single direction arrow format.