Alyce Harlan
Ashley Folse
Ashley Morrison
The ‘A’ Team - Group 97

URL: http://classwork.engr.oregonstate.edu:9865/index.html

Feedback by the TAs and peer reviewer Version 3

TA FeedBack

Hi group 97,

To fully meet project requirements, each of your entities need to have their own CRUD operations. I only see the add functionality for each page right now.

- Madelyn Lazar

Peer FeedBack

From: Rachel Gee
Does the UI show where it will utilize a SELECT for every table in the schema?

1. Companies - From the main page, Select is utilized under “Manage Company.” The company name listed under this column header is the hyperlink to the Companies table display. The same attributes exist in the UI as the schema.

2. Users - From the main page, “Manage Users” hyperlink performs Select to display the Users table. The FK attribute userscompany_id does not display in table. I might add this to the columns of the table and in the add new user you could potentially add a field to associate a company name with a new user.

3. Forms - From the main page, “Manage Forms” hyperlink performs Select to display the Forms table. The FK attribute file_id is not displayed on the table but this may not be strictly necessary.

4. Files - From the main page, “Manage Files” hyperlink performs Select to display the Files table. The FK filescompany_id does not appear in the table. This could potentially be added

to the attribute columns of the displayed table, especially if you wanted to add a company name along with a new file as its added.

5. Signatures - From the main page, “Manage Signatures” hyperlink performs Select to display the Signatures table. “Signature Name” column in displayed table may encompass date_signed and foreign key attributes not explicitly listed.

6. user_file_access - Table not displayed with a Select in UI

7. form_signatures - Table not displayed with a Select in UI

Does at least one form utilize a search/filter/dropdown with a dynamically populated list of properties?

There are no search/filter/dropdowns featured in any of the forms. In Manage Users there is the potential to add another field (company name) to the add new user form and use a drop down for associating a company name from the list of companies in the database with a new user that is being added. You could also use this company name dropdown in the files form to associate a referenced company to a file. However, I do see some code in the DML file that is applicable to showing this on the UI forms.

Does the UI implement an INSERT for every table in the schema?

1. Companies - “Add Company” form for Insert implementation with name attribute input field

2. Users - “Add New User” form for Insert implementation. “User Name” and “Email” input fields present; may need another input field for the user’s company.

3. Forms - “Add Form” form for Insert implementation. “Name” input field appears; may need to associate a file_id with another input field in this form as well.

4. Files - “Add File” form for Insert implementation. Form contains “Name” input field for file name attribute; may need to add a company name input filed as well.

5. Signatures - Insert implementation achieved with “Add New Signature” form. “Signature Name” (~signature_id attribute; potential spot for a drop down?) and Email input fields

present. The attribute date_signed is not included - it looks like from the DDL file the date value is automatically generated? If you wanted to let the user have the ability to adjust the date you could add this as an input field. User form attribute should probably be included.

6. user_file_access - No insert form present

7. form_signatures - No insert form present

Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship?

Entities with FK attributes:

1. Users: userscompany_id (company) - Yes, the sql insert statement includes the foreign key

2. Forms: file_id (file) - I don't see an Insert statement for this table in the DML file

3. Files: filescompany_id (company) - Yes, the sql insert statement includes the foreign key

4. Signatures: userssig_id (user), usersform_id (form) - Yes, the sql insert statement includes the foreign keys

5. user_file_access (intersection table): usersfile_id (user), uniquefile_id (file) - Yes, the Insert statement includes both foreign keys; this should satisfy the insert requirement for a M:M relationship

6. form_signatures (intersection table): user_id(user), form_id (form) - There is not an Insert statement for this table in the DML file

Is there at least one DELETE, and does at least one DELETE remove things from a M:M relationship?

The DML file contain a sql statement to delete a user from the Users table. Users is in a M:M relationship with Files connected by the intersection table user_file_access. In the DDL file there is appropriate sql statements in the intersection table to delete on cascade so that when a user is deleted they will be deleted from the intersection table without deleting files from the Files table. The UI does not display a delete implementation.

Is there at least one UPDATE for any one entity, with fields for the corresponding attributes for that entity?

There is an Update implementation for editing a user's email within the DML file. Within the UI there needs to be a form to update/edit an existing user where the new email could be entered.

Is at least one relationship NULLable?

Nullable relationships are observed in several places, including in the user_file_access table. This makes sense because many users may have files associated with them but it is not absolutely necessary for a user to have a file associated with them to exist. Conversely, the table shows that a file can exist without a user and is also defined as a nullable relationship. It would be feasible to edit entries in the user_file_access table and change either a file or user to Null in a given entry.

Do you have any other suggestions for the team to help with their HTML UI? For example, using AS aliases to replace obscure column names such as fname with First Name.

Column names on the UI are readable. Adding the edit/update and delete options under "Actions" looks like a good next step and well as including the drop downs on the UI that you have already started the code for in the DML file. Great job!

From: Alexander Ngo

- *Does the UI show where it will utilize a SELECT for every table in the schema?* In other words, a data table for each table in the schema should be displayed on the UI (which are not required to be populated with sample data, but should at least have column names). If yes, which tables from the schema do you see fully represented in the UI with a SELECT? If not, which tables and/or attributes are missing? Note: it is generally not acceptable for just a single query to join all tables and display them.

The UI does show where SELECT will be utilized. From the main page, we have the option to click on the corresponding company name under Manage Company to get SELECT data from the companies table. Similarly, Manage Users, Manage Files, Manage Forms, and Manage Signatures can be used to SELECT data from the corresponding entity tables. The user_file_access and form_signatures tables don't seem to be accessible with the UI at the moment, but I noticed that besides the Manage Company page, the tables aren't yet populated, so I assume that the 2 Many to Many relationships will be viewable once those tables are populated.

- *Does at least one form utilize a search/filter/dropdown with a dynamically populated list of properties?* If yes, which form(s) have which features incorporated? For which attribute(s)? If not, where are a couple places this could be implemented?

There does not seem to be any search/filter/dropdown usage in any forms. I assume this is because the tables aren't yet populated. I think that some of the tables are missing attributes that are mentioned in the Schema/outline. For instance, the outline and Schema show that there should be a relationship between company and users. To reflect this, the UI for manage users should include an extra column labeled something like "Company" that should store the company affiliated with the user. A dropdown could be used here to make sure that the user is linked to an actual supported company. The same can be applied to any other case where there is a M:1 relationship between two entities (Files and Companies, Forms, and Files, etc.)

- *Does the UI implement an INSERT for every table in the schema?* In other words, there should be UI input fields that correspond to each table and attribute in that table. If yes, which tables from their schema do you see a complete INSERT for? If not, which tables either do not have INSERTs, or have attributes missing from their INSERT? In general, do you have any suggestions for their INSERT forms?

Each entity does seem to have forms at the bottom of their generated tables to add new values to the table. The forms seem to be lacking the fields needed to match the attributes mentioned in the outline. I suggest adding all fields so that they match the attributes mentioned in the outline. For example, Manage Users only shows User Name and Email. I suggest adding in a field for Company that has a drop down menu to establish the relationship between user and company.

- *Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship?* In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total). Or, alternatively, there should be an INSERT for INSERTing into the intersection table(s) directly. If yes, list all the table INSERTs that correctly add their corresponding FK attributes, and describe the group's implementation for INSERTing into the intersection of their M:M relationship. If not, which INSERTs need to be altered, and in what way?

While the UI does not include the relationships, the SQL file does properly include the foreign key needed to establish the M:M relationships. This is done by using SELECT, FROM, and WHERE.This can be seen for the INSERT INTO for Users, Forms, Files, and Signatures entities. As an example, for Users, (SELECT company_id FROM Companies

WHERE company_name = “McDonalds”) is used to link a company ID as a foreign key.
There are 2 M:M relationships managed by User_file_access and Form_signatures. Both properly add 2 foreign keys corresponding to the linked entities. I.e., ((SELECT user_id FROM Users WHERE user_name = "Grimace"), (SELECT file_id FROM Files WHERE file_name = "Menus")).

- *Is there at least one DELETE, and does at least one DELETE remove things from a M:M relationship?* In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers. If yes, describe all the DELETEs you see. If not, what’s missing in terms of this requirement?

There is a DELETE command in the inkedin_dml.sql file that removes something from the Users table which is in M:M relationships with Forms and Signatures. Since the DDL has ON DELETE CASCADE for Users, this should result in the deletion of any rows from any tables that reference that user including the intersection tables. There seems to be some cases (based on the outline) where it should be possible to delete a value without deleting the row from the child tables. For these cases, consider using ON DELETE SET NULL. For example, in Signatures, fk_usersig_id.

- *Is there at least one UPDATE for any one entity, with fields for the corresponding attributes for that entity?* In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record? If yes, describe all the UPDATEs you see. If not, where would you suggest adding an UPDATE, or what’s missing from the UPDATE(s) you see?

In inkedin_dml.sql, there is one UPDATE query that is used to update a user’s email. The query does properly reference the user_email attribute. The UI does not yet implement editing, but there does seem to be a space saved for it in actions.

- *Is at least one relationship NULLable?* In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus, it should be feasible to edit an Order and change the value of Employee to be empty. If yes, which NULLable relationship(s) do you see, and does it seem to make sense? If not, where would you suggest adding a NULLable relationship and why?

There seems to be several NULLable relationships throughout the schema. This makes sense as there are several relationships that aren’t required (i.e., a signature should still exist on a form even if the user is deleted). As another example, a signature can also exist without being on a form. If a form that the signature is on is deleted, the signature will still exist and can be later used for another form.

- *Do you have any other suggestions for the team to help with their HTML UI? For example, using AS aliases to replace obscure column names such as fname with First Name.*

It could potentially be useful to be able to view all users instead of just one company at a time. This could allow for implementation of filters and dropdowns (for adding a new user). This, in turn, would make it faster to add users from different companies without having to navigate to a new page each time.

From: Jimena Noa-Guevara

- *Does the UI show where it will utilize a SELECT for every table in the schema?* In other words, a data table for each table in the schema should be displayed on the UI (which are not required to be populated with sample data, but should at least have column names). If yes, which tables from the schema do you see fully represented in the UI with a SELECT? If not, which tables and/or attributes are missing? Note: it is generally not acceptable for just a single query to join all tables and display them.

Yes, all tables will use SELECT. This includes Companies, Users, Files. Forms, and Signatures. In their page, they all seem to be able to display data. Some of them don't have data yet, but it is set up such that when there is data, it is clearly shown.

- *Does at least one form utilize a search/filter/dropdown with a dynamically populated list of properties?* If yes, which form(s) have which features incorporated? For which attribute(s)? If not, where are a couple places this could be implemented?

I did not find a search/filter/dropdown menu, but I have noticed this too in other teams' UI since not all tables have been populated with data. This could be implemented under companies or really any table where it fits.

- *Does the UI implement an INSERT for every table in the schema?* In other words, there should be UI input fields that correspond to each table and attribute in that table. If yes, which tables from their schema do you see a complete INSERT for? If not, which tables either do not have INSERTs, or have attributes missing from their INSERT? In general, do you have any suggestions for their INSERT forms?

Yes, each table has an INSERT option including Companies, Users, Files, Forms, and Signatures.

- *Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship?* In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total). Or, alternatively, there should be an INSERT for INSERTing into the intersection table(s) directly. If yes, list all the table INSERTs that correctly add their corresponding FK attributes, and describe the group's implementation for INSERTing into the intersection of their M:M relationship. If not, which INSERTs need to be altered, and in what way?

Although not visible in the UI, the .sql file has INSERT functions that will add the corresponding FK attributes. For example, 'fk_userscompany_id' is an FK in Users, 'Fk_filescompany_id' is in Files, and 'Fk_file_id in Forms.

- *Is there at least one DELETE, and does at least one DELETE remove things from a M:M relationship?* In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers. If yes, describe all the DELETEs you see. If not, what's missing in terms of this requirement?

No, the UI does not yet have a DELETE function for any of the tables. However, I can see it in the .sql file. I would suggest to add it in for the next submission. The team uses ON CASCADES so you would need to adjust to set values to null when an item gets deleted.

- *Is there at least one UPDATE for any one entity, with fields for the corresponding attributes for that entity?* In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record? If yes, describe all the UPDATEs you see. If not, where would you suggest adding an UPDATE, or what's missing from the UPDATE(s) you see?

I could not find an UPDATE option, but I would suggest adding one in Files in case Companies or Users need to update a contract.

- *Is at least one relationship NULLable?* In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus, it should be feasible to edit an Order and change the value of Employee to be empty. If yes, which NULLable relationship(s) do you see, and does it seem to make sense? If not, where would you suggest adding a NULLable relationship and why?

Yes, Users and Files is a relationship that is NULLable because deleting one means that you can set its value to null in the other.

- *Do you have any other suggestions for the team to help with their HTML UI? For example, using AS aliases to replace obscure column names such as fname with First Name.*

Placeholders are a good idea!

From: Mackenzie Jackson

Does the UI show where it will utilize a SELECT for every table in the schema?

Based off the UI along with the corresponding query documentation, SELECT is utilized by giving the company name's on the homepage. It also appears that SELECt is used when clicking on "Manage Users", "Manage Files", "Manage Forms", and "Manage Signatures". In the DML doc, I see that you have also used SELECT to retrieve all users and all files names from a specific user, but I do not see the implementation of SELECT in regards to "Manage Forms" and "Manage Signatures." I do see it on the UI side, however.

Does at least one form utilize a search/filter/dropdown with a dynamically populated list of properties?

It's difficult to be able to tell if that is the case if look at the UI side. None of the tables have been populated with sample data so it does make it a big more confusing to see the how the properties will be presented logically. I suggest using a drop down list for "actions" in your users, files, forms, and signature pages.

Does the UI implement an INSERT for every table in the schema?

Yes, the UI does implement an INSERT for every table besides 'user_file_acces' and 'forms_signatures'. However, the UI implementation is not complete. When adding a company, I only see a box for "Name". With the Users, I only see a way to add User Name and Email but not "Actions". Same goes with the Files table, where you can oly add a Name and not "Actions".

Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship?

I am not seeing all of the relationships listed in the UI, as previously stated above. I do, however, see the FK attributes are states in the SQL. 'fk_userscompany_id' is listed as an FK in the table 'USERS'. 'Fk_filescompany_id' is listed under the table 'Files'. 'Fk_file_id' is listed under the table 'Forms'. However, I am not seeing these FK's represented in the tables on the UI side. I suggest adding those in. Your M:M relationship (I'm assuming is intersected by the form_signatures and user_file_acces) is also not represented in the UI, but I can see it in the SQL.

Is there at least one DELETE, and does at least one DELETE remove things from a M:M relationship?

No, there is no way to DELETE any of the entries on the UI side, though it does look like in the queries DELETE is implemented for Users. In the SQL doc, it appears that if you were to delete a user than they would also be deleted from the Files with any issues because you utilized “ON DELETE CASCADE” in all of your tables.

Is there at least one UPDATE for any one entity, with fields for the corresponding attributes for that entity?

Again, I am seeing an UPDATE query in your back end documentation, but I am not seeing an option for that on the UI side. It looks as if you are implementing the DELETE and UPDATE option for Users, so you just need to update the UI for the Users table.

Is at least one relationship NULLable?

Yes, as previously mentioned, I believe that you can sever the relationship between a user and files without deleting everything. I am also seeing several other NULLable occurrences. As another example, if a company is deleted, than a file will still remain. This seems beneficial, because even if a company is no longer operable, you do still want to keep important documents related to them.

Do you have any other suggestions for the team to help with their HTML UI?

The homepage’s table of contents is a little overwhelming and a bit redundant. I suggest creating clear links for “Browse Companies”, “Browse Users”, etc. You can always implement joins/references down the road in order to present the relationships of your tables clearly.

Actions Based on Feedback Version 3

- ☑ ~~Add edit buttons to each entity~~
    - ☑ ~~Companies~~

    - ☑ ~~Employees~~
    - ☑ ~~Forms~~
    - ☑ ~~Files~~
    - ☑ ~~Signatures~~
- ☑ ~~Add Delete buttons to each entity~~
    - ☑ ~~Companies~~
    - ☑ ~~Employees~~
    - ☑ ~~Forms~~
    - ☑ ~~Files~~
    - ☑ ~~Signatures~~
- ☑ ~~Standardize layout for each entity page~~
    - ☑ ~~Companies~~
    - ☑ ~~Employees~~
    - ☑ ~~Forms~~
    - ☑ ~~Files~~
    - ☑ ~~Signatures~~
- ☑ ~~Make splash page less redundant~~
    - ☑ ~~Remove "Manage" from headers~~
    - ☑ ~~Just one company column with the hyperlinks → remove the company column without hyperlinks~~
- ☑ ~~Add correct attributes to each entity to match schema diagram~~
    - ☑ ~~Companies~~
    - ☑ ~~Employees~~
    - ☑ ~~Forms~~
    - ☑ ~~Files~~
    - ☑ ~~Signatures~~
- ☑ ~~Add intersection tables to UI~~
    - ☑ ~~User file access (Would make sense under Users or Files)~~
    - ☑ ~~Form signatures (would make sense under Forms page)~~
- ☑ ~~Add a drop down menu or search option somewhere in the page~~
- ☑ ~~(Optional) It could potentially be useful to be able to view all users instead of just one company at a time. This could allow for implementation of filters and dropdowns (for adding a new user). This, in turn, would make it faster to add users from different companies without having to navigate to a new page each time.~~
  [Decided not to implement this suggestion at this time as it doesn't fit with our current layout. We may change this later on]
- ☑ ~~Use, ON DELETE SET NULL instead of CASCADE~~
- ☑ ~~Add more sample data for other entities~~
    - ☑ ~~Employees~~
    - ☑ ~~Forms~~

☑ ~~Files~~
☑ ~~Signatures~~

Feedback by the TAs and peer reviewer Version 1

TA FeedBack

Here are some suggestions:

-Keep the name of the tables consistent. For example, user_form is in **snake_case**, but the rest seem to be in **PascalCase**. Choose one and stick to it for consistency.

-In your ERD, try to avoid crossing lines. You also should label the FKs along with PKs in your diagram. This way we can more clearly see how some of the relationships work.

-You've done a great job explaining the many-to-many (M) relationships, but for tables like **user_form**, which appears to be your intersection table between Users and Forms, you should explicitly mention in your outline how it handles the M relationships. Also, describe how the foreign keys **(e.g., user_id and form_id)** in user_form connect to the primary keys in the Users and Forms tables.

- Madelyn Lazar

Peer Feedback

From: Danielle Chang
https://edstem.org/us/courses/67389/discussion/5495387?answer=12763301

- Does the overview describe what problem is to be solved by a website with DB back end? If yes, summarize. If not, what changes would better support describing the problem to be solved?

Yes, the overview properly describes the issue that the project will solve. Inked In will help companies manage large amounts of electronic files and keep track of e-signatures.

- Does the overview list specific facts? If yes, summarize what the facts illustrate about the proposed DB solution. If not, what facts would better support illustrating the scope and scale of the proposed DB solution?

Yes, the overview mentions that many companies manage over 10,000 electronic files and process around 200 signed forms daily. It also mentions that Inked In will allow companies to store up to 400,000 files and process 1,000 forms weekly. This demonstrates the large scale of the data that the database will be handling.

- Are at least four entities described, and does each one represent a single idea to be stored as a list? If yes, summarize. If not, based on the course material, what changes can you suggest to improve?

Yes, five entities are described: Company, User, Form, File, and Signature. Looking through the descriptions of each entity, it seems like each one represents a single idea, and every entity seems necessary for what the database hopes to achieve.

- Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints, and describe relationships between entities? If yes, summarize. If not, based on the course material, what changes can you suggest to improve?

Each entity has a short description except for Signature. Each entity lists attributes (with datatypes and constraints) and relationships for that entity. Company represents a company, User represent employees of a certain Company, Form represents a signable form, File represents an electronic file, and I would assume that Signature represents a signature on a form, although the description is missing. One question I had was whether user_name was the name of the user or the username chosen by the user. If it's the name of the user, I don't think it's necessary for the value to be unique, but if it's a username, then I understand.

- Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database? If yes, summarize. If not, based on the course material, what changes can you suggest to improve?

The 1:M relationships seem fine, but I am a bit confused on the purpose of the user_form table. It looks like it's meant to be an intersection table between User and Form, but it has been given a M:M relationship with User, and the connection to Form is missing on the ER diagram. From my understanding, intersection tables should go in between two entities with a M:M relationship to break that relationship into two 1:M relationships. So, user_form would have a M:1 relationship with User and a M:1 relationship with Form. A similar intersection table would likely need to be created between User and File since there is a M:M relationship there as well. I'm also wondering if it's necessary for forms and users to have a direct relationship at all. Since forms are within files and files already specify which users can access them, it seems a bit repetitive. I also noticed that there is no user foreign key in the form entity, which also applies that there isn't a direct relationship.

- Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming? If yes, summarize. If not, based on the course material, what changes can you suggest to improve?

Entities are not plural. I would rename the entities to Companies, Users, Forms, Files, and Signatures to fit with conventions. Additionally, the intersection table is in singular snake case, which does not follow entity naming conventions. Rather than naming the table user_form, it would probably be something like UserForms. Other than that, attributes are all singular and use snake case consistently.

From Kathryn Butler
https://edstem.org/us/courses/67389/discussion/5495387?answer=12770431
As a preface to this response, I want to say that I think your group name, the "A" team, is very clever. :)

Does the overview describe what problem is to be solved by a website with DB back end?

- Yes, it does. Inked In presents "a database-driven e-signature service that is designed to effectively and efficiently manage large volumes of files." It aims to manage the relationships between users, files, forms, and signatures.

Does the overview list specific facts?

- The overview does list specific and detailed facts about the objectives they plan to achieve. Their goals are to create a comprehensive file management system, a signature system with secure document handling, real-time tracking of the signature process and document status, and scalability for large organizations. It also lists specific amounts of documents companies can store securely.

Are at least four entities described, and does each one represent a single idea to be stored as a list?

- Yep, there's actually five entities described, and they each can represent a single idea. They have Company, User, Form, File and Signature. This group has gone above and beyond on their entities.

Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints, and describe relationships between entities?

- Most of them do, but I'm a little confused on the purpose of user_form. It seems as if User is connected to both Form and user_form. User_form "can be signed multiple times by the same User," but Form can be "signed (possibly multiple times) by User(s)." If I'm understanding this correctly, it seems as if user_form contains duplicate information which may not be needed for this database.

Are 1:M relationships correctly formulated?

- I think user_form should link to Form as well as User, which would create a simplified M:M connection between User and Form. Additionally, I see that Signature has both user_id and form_id, but in the ERD it is only connected to Form.

Is there at least one M:M relationship?

- Yes, there is a M:M connection between User and File, with File having User's foreign key. I did notice that User and Form are said to have a M:M connection as well, but in the ERD neither of them hold the other's foreign key.

Does the ERD present a logical view of the database?

- Yes it does, although there is a relationship connection that crosses over Form's diagram. I would recommend placing both Files and Signature

Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

- a) user_form should also be renamed to either UserForm or UserForms, depending on if you want to make your entities plural or not.
- b) I believe Files should be renamed to simply File, because all the other entities are singular. Either that or make the rest of them plural.
- c) Same as a), but every other entity and attribute seems to be correct.

Overall, this is an impressive rough draft. I especially like the detail in the Overview and Objective sections, and the logic behind the specific entities and attributes seems very clear. I look forward to seeing your final draft. Keep up the great work!

From: Ryan Orlando
https://edstem.org/us/courses/67389/discussion/5495387?answer=12770800
Hi group 97. Here's my review of your rough draft.

- Does the overview describe what problem is to be solved by a website with DB back end? If yes, summarize. If not, what changes would better support describing the problem to be solved?

Yes, the overview describes the problem to be solved, that being the handling of large amounts of files used by companies.

- Does the overview list specific facts?If yes, summarize what the facts illustrate about the proposed DB solution. If not, what facts would better support illustrating the scope and scale of the proposed DB solution?

The overview also lists specific facts, with the amount companies manage being over 10,000, around 200 processed signed forms. It also claims that this service can allow companies to store up to 400,000 documents and can process 1,000 forms per week.

- Are at least four entities described, and does each one represent a single idea to be stored as a list? If yes, summarize. If not, based on the course material, what changes can you suggest to improve?

There are at least four entities described in the overview, that being Company, User, Form, File, Signature. Each one clearly represents the idea of each entity and is easy to understand.

- Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints, and describe relationships between entities? If yes, summarize. If not, based on the course material, what changes can you suggest to improve?

The outline of each entity but one details properly describes the purpose, lists attribute datatypes and constraints, and describes relationships between entities. However, this is not the case for the Signature entity, as it's missing a description of the purpose. To summarize the rest, the Company is the base of which the entity with the most connections to other entities, User, is connected to. Most of the other entities are connected directly to User, with the exception of Signature, which branches off from Form.

- Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database? If yes, summarize. If not, based on the course material, what changes can you suggest to improve?

All the 1:M relationships are correctly formulated, and there are enough M:M relationships for a group of 3. However, it is a little confusing on what the what the purpose of the user_form table is. Other than that, the ERD looks fine and has the necessary details between entities.

- Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming? If yes, summarize. If not, based on the course material, what changes can you suggest to improve?

Everything is consistent between the entity names, all the attributes being singular, and the capitalization of the naming. However, each entity should be plural instead of singular, as this database would not be holding a single one of each, as described in the overview earlier. The only issue with the consistency between entities is user_form, which isn't consistent with the rest of the entities.

From: Joseph Castellano
https://edstem.org/us/courses/67389/discussion/5495387?answer=12791203

- **Does the overview describe what problem is to be solved by a website with DB back end?If yes, summarize. If not, what changes would better support describing the problem to be solved?**

Yes! The managing e-file storage and document signature processing is a great problem to address.

- **Does the overview list specific facts?If yes, summarize what the facts illustrate about the proposed DB solution. If not, what facts would better support illustrating the scope and scale of the proposed DB solution?**

The DB will store up to 400,000 digital documents, and can handle processing 1,000 forms/signatures per week (up to 200 per day). I'll note that (unless I am thinking about this the wrong way), it seems like the database will hit the 400,000 digital document limit in only about 8 years if it adds 1,000 documents per week.

- **Are at least four entities described, and does each one represent a single idea to be stored as a list? If yes, summarize. If not, based on the course material, what changes can you suggest to improve?**

Yes, the entities are "Company", "User", "Form", "File", and "Signature."

- **Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints, and describe relationships between entities? If yes, summarize. If not, based on the course material, what changes can you suggest to improve?**

Yes, Signatures doesn't have any descriptive text (though maybe it's obvious what a signature on a form is). See some additional suggestions below about the relationships.

- **Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database? If yes, summarize. If not, based on the course material, what changes can you suggest to improve?**

I believe "user_form" is meant to facilitate the M:M relationship between Users, and Forms, but it doesn't seem to be presented that way in the ERD. I believe User_Forms should be in between Users and Forms as a junction table.

Likewise, I think you need a junction table between Files and and Users to help facilitated the M:M relationship (in the ERD).

I will echo what others have said and suggest you spread out the tables a little in the ERD because the overlapping lines between Forms, Files, and Signature makes the ERD hard to read.

"1:M - a Form can have multiple Signature, but each Signature is unique to the date_signed and therefore can only be applied once in the Form" - I'm wondering if having signature_id as a foreign key in Form might make more sense to allow this functionality? I'm not sure.

- **Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming? If yes, summarize. If not, based on the course material, what changes can you suggest to improve?**

Entities I think should be plural (Files is but nothing else).

From Brenna Jennings
https://edstem.org/us/courses/67389/discussion/5495387?answer=12791673

Does the overview describe what problem is to be solved by a website with DB back end?

Yes, the overview clearly describes what is being solved. The overview presents the issues that Inked In are facing. The bullet point list at the end shows the expectations on how the back-end will solve issues of signature storing, document status tracking, and a large influx of activity in a single day.

Does the overview list specific facts?

The overview itself does a great job of not only listing the issues they are facing, but also the numerical stats of the influx that the company is/ will be facing. This allows the reader to clearly see the issue that is needing to be taken care of. It also provides an exact amount (200/day) of influx that is expected to be handled using the new database.

Are at least four entities described, and does each one represent a single idea to be stored as a list?

The team has surpassed the four entities required, and each one allows for an easy understanding of what it is used for, and clearly represents the idea.

Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints, and describe relationships between entities?

The signature entity is missing a description, so that adds to the lack of clarity for that one portion. The only other thing I see is that "user_form" seems unnecessary in the ERD. With User and Form both being their own entities, at first look through, I am not fully understanding its purpose. Other than that, all other entities are well described and seem to have an obvious purpose.

Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?

Yes, their 1:M relationships are formulated correctly. Be careful on the arrangement of the ERD. You have a connection crossing over the Form box. See if it is possible to rearrange, just to make it easier to follow for a reader who is new to the draft.

Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

The entities are not plural, so I would change that. You may also want to rename user_form into a PascalCase structure, just to follow the provided guidelines.

Great job team! Also your name is hilarious! Go "A Team"!

From: Urbano San Roman
https://edstem.org/us/courses/67389/discussion/5495387?answer=12798397
Hi Group97! Below are my notes on your proposal. Also attached for easier access possibly.

Let me know if there's any questions or anything I can clarify.

Urbano

Group 97 Proposal Review:

Does the overview describe what problem is to be solved by a website with DB back end?

I love the name Inked In! Your introduction perfectly describes the issue and perfectly describes the problem. I especially like the 4 bullet points at the end of the introduction, which serve as both a summary and a list of targets/things to be achieved by the database back end. In summary, I can’t find any fault with the overview.

Does the overview list specific facts?

The overview does list specific facts and figures. Namely,

o That companies manage 10k digital documents.

o Companies process 200 signed forms daily.

o Inked In will allow for the storage of up to 400k digital documents.

o Inked In will allow for the processing of 1k forms and signatures per week.

I believe team 97 gets Full marks here also.

Are at least four entities described, and does each one represent a single idea to be stored as a list?

A group of 3 requires 5 entities. Group 97 achieve that with ‘Company’, ‘Form’, ‘Files’, User’, ‘Signature’. I’m unsure about the difference between a ‘form’ and a ‘file’. That is not immediately clear to me, but I hope to get a better understanding after spending some time studying the proposal. Adding to the confusion ‘Form’ is described as “records the details of a form made up of File(s)…”

And “files” is described as “records the details of a file which can be made up of multiple Form(s)”

So, my critique would be to make the relationship between form and file clearer.

Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints, and describe relationships between entities?

Company:

o Is each company in the database an Inked In customer?

o If so, I don’t think the company_name should be required to be unique. There may be companies with the same name, for example. The unique company_id, which is auto incremented should take care of differentiation.

o For company_name, specify the max number of characters allowed. I think 50 would work fine. So, varchar(50).

User:

o I think I would have liked the user description to reference its relationship with the company in addition to the database. I think this would make things clearer. But that may be redundant since it’s clearly mentioned in the line above.

o Add max number of characters for the VARCHARs.

o Don’t require that names be unique I would argue.

o A user has a M:1 relationship with Company. (0, 1 or many users are employed by 1 company). I think there is a typo/error in the first relationship bullet point.

o User_form is a composite entity. It shouldn't be M:N. I would imagine a user_form is associated with 1 and only 1 user. So, the relation would be 1:M.

Forms:

o I don't like how similar the description is to File. A form has a signature and a File does not. That is clear. I think one should own the other and so the description should have different ?verbs?. Maybe: *'Form records the details of a form which may be included in many files and signed (possibly multiple times) by User(s)'* would be a better description.

o Varchar for title should specify the max number of characters: suggestion; VARCHAR (50)

o I think the relationship described in the second bullet point between File and Form is confusing.

o My attempt: *1:M A specific form can only be used once in a file, but that specific form can be included in many files.*

File:

o Use varchar(50)

Signature:

o I like the added details for the relationship to make things clear 😊

Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?

The 'A' team has multiple M:N relationship exceeding the requirement.

Below, is my comments of the ERD diagram

ERD

o The USER to COMPANY relationship is missing the ‘one and only one’ symbol:

o a user must be associated with a company and only one company.

o The USER_FORM to USER relationship is wrong I believe.

o A USER_FORM must be associated with 1 and only 1 USER, but a USER may be associated with 0, 1, or many USER_FORMS.

o There’s a missing relationship between the USER_FORM entity and FORM entity.

o It should be a 1:M relationship: a USER_FORM must be associate with 1 and only 1 form.

o Should there be a relationship drawn between USER and SIGNATURE? My thinking is that the line drawn between USER and FORM should be deleted, there should be a line drawn between USER and SIGNATURE, and there should be a line drawn between SIGNATURE and FORM. I think that would make sense but I’m not completely sure. How could there not be a relationship between SIGNATURE and USER if SIGNATURE has a FK from USER?

o The SIGNATURE to FORM relationship is missing the ‘1 and only 1’ symbol: a SIGNATURE must be associated with one FORM and only one FORM .

o The FILE to FORM relationship is missing details.

o Can a FILE include 0 FORMS? Or must a FILE include at least 1 FORM?

Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

o All the entities should be plural. Only Files is plural.

Actions based on the feedback on version 1/**TO DO LIST**

List briefly the actions that you chose to take based on the above feedback. If you decided not to act on a specific suggestion, you need to describe in detail your reasoning.

- ☑ ~~Change all entity names to be plural~~
  - ☑ ~~In Verbal Description~~
  - ☑ ~~In ERD~~
- ☑ ~~Change user_form to userForms for consistency~~
  - ☑ ~~In Verbal Description~~
  - ☑ ~~In ERD~~
- ☑ ~~Move lines in ERD so they are not crossing/touching~~
- ☑ ~~Explain userForm in greater detail~~
  - ☑ ~~How are foreign keys used~~
  - ☑ ~~Answer how does it handle M relationships~~
- ☑ ~~Add short description to Signature in verbal description~~
- ☑ ~~Clarify if user_name is the Name of the Employee or a username they picked~~
  - ☑ ~~If it is the employee name, remove Unique characteristic~~
  - ☑ ~~Else, keep as is~~
- ☑ ~~Clarify how userForm is used, may not be necessary?~~
  - ☑ ~~Connect to Forms in ERD~~
  - ☑ ~~Clean up possibly redundant relationship between user and file and also user and form~~
  - ☑ ~~M:M connection between User and Form.~~
- ☑ ~~Signature has both user_id and form_id, but in the ERD it is only connected to Form~~
  - ☑ ~~Add link between Signature and User in ERD~~
- ☑ ~~I did notice that User and Form are said to have a M:M connection as well, but in the ERD neither of them hold the other's foreign key. ((Ashley F answering here - there is a join table that manages this M:N relationship now))~~
  - ☑ ~~Add foreign key link between User and Form if necessary~~
- ☑ ~~Add a junction table between Files and and Users to help facilitate the M:M relationship (in the ERD).~~
- ☑ ~~Clarify relationship between form and file~~
- ☑ ~~Add max number of chars for each varchar attribute~~
- ☑ ~~Clarify relationship between User and Company~~
- ☑ ~~Decide if a user_form is associated with 1 and only 1 user:~~
  - ☑ ~~Change the relation to 1:M.~~

- ☑ ~~Remove Unique requirement from Company names~~
- ☑ ~~Add these changes to ERD as applicable:~~
    - ☑ ~~Add the 'one and only one' symbol between USER to COMPANY relationship~~
    - ☑ ~~a user must be associated with a company and only one company.~~
    - ☑ ~~USER_FORM must be associated with 1 and only 1 USER, but a USER may be associated with 0, 1, or many USER_FORMS.~~
    - ☑ ~~Add relationship between USER_FORM entity and FORM entity – 1:M relationship: a USER_FORM must be associated with 1 and only 1 form.~~
    - ☑ ~~Should there be a relationship drawn between USER and SIGNATURE? My thinking is that the line drawn between USER and FORM should be deleted, there should be a line drawn between USER and SIGNATURE, and there should be a line drawn between SIGNATURE and FORM. I think that would make sense but I'm not completely sure. How could there not be a relationship between SIGNATURE and USER if SIGNATURE has a FK from USER?~~
    - ☑ ~~Add in 1 and only 1 symbol between SIGNATURE to FORM relationship – a SIGNATURE must be associated with one FORM and only one FORM .~~
    - ☑ ~~FILE to FORM relationship is missing details – can a FILE include 0 FORMS? Or must a FILE include at least 1 FORM? (a file can contain 0 forms, at the beginning of a file, it can be named and saved without forms being filled yet - Ashley Folse)~~

Upgrades to the Draft version 2

If you are making any changes to the files based on your own changed design decisions, they should be listed under this section.

Ashley Folse - gave all of the foreign keys more unique names for purposes of forward engineering the schema from MySQL database, within the ERD: clarified how access will be given to Users as employees of a specific Company using the join table user_file_access, clarified how Users and their Signature will be linked to a form using the form_signatures join table making the form_signatures_id the primary key for it and adding the two specific foreign keys for a user's id and the form's id. Clarified the purpose of the relationships between Companies and their Users. Clarified the relationships between Files and Forms both in the outline and in the ERD, clarified relationships between users and signatures, and forms and signatures in the ERD and the outline.

Upgrades to the Draft version 3

In version 3, we:

1) Added commas between all the table rows in our .sql file as this was the biggest piece of peer feedback. The lack of commas meant that peers could not view our .sql file tables properly.

2) Moved our ERD (which was actually a Schema) into the Schema portion of the report

3) Created a new ERD that was much simpler than we thought it would be.

4) We also added titles above each section of sample data so it was clear which entity the sample data belonged to.

Inked In

An E-Signature Service

Inked In Overview

Inked In is a specialized e-Signature service that is designed to address the needs of companies with extensive e-file storage and high volumes of document processing requirements. Many companies manage over 10,000 digital documents and process around 200 signed forms daily. It is essential to implement an efficient and secure e-signature solution. With Inked In, companies can store up to 400,000 digital documents while processing 1,000 forms and signatures per week ensuring smooth document handling and efficient processing. This service allows companies to organize and track their files as well as securely collaborating with a team to streamline their e-filing processes, and enhance analytical capabilities.. The service will handle and manage relationships between Users to their Files, and Files to their Forms and Signatures, ensuring traceability.

To address the challenges presented currently by companies unable to keep up with digital documents, we present a database-driven e-signature service that is designed to effectively and efficiently manage large volumes of files. The proposed database solution will support Inked In in achieving the following goals:

- Comprehensive file management, allowing users to store, access, and organize large volumes of documents.
- A signature system that links forms to specific files, ensuring traceability and secure document handling.
- Real-time tracking of document statuses, user activity, and signature completion.
- The ability to process 200 or more signed forms per day, ensuring scalability for large organizations and efficient workflow.

Overall Objective

The goal is to create an easy-to-use and reliable e-signature system that helps companies manage large amounts of files while making it simpler to sign documents. This system will keep files organized to simplify the signing process while providing clear reports for businesses to easily manage.

Inked In Database Outline

Entities:

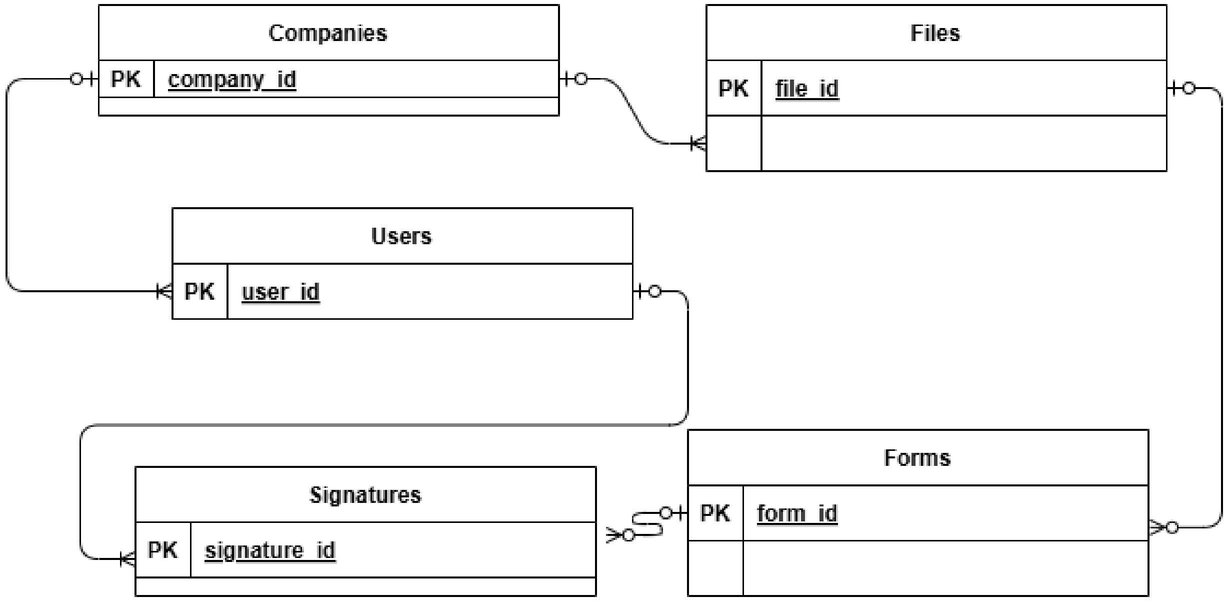
- Companies: records a company to which many employees (Users) belong
    - company_id: int, auto_increment, unique, not NULL, PK
    - company_name: varchar(145), not NULL
    - Relationship:
        - 1:M between Companies and its employees (User)
- Users: records the details about Users of the e-signature service
    - user_id: int, auto_increment, unique, not NULL, PK
    - user_name: varchar(50), not NULL
    - user_email: varchar(50), unique, not NULL
    - fk_userscompany_id: int, auto_increment, unique, not NULL, FK(References Companies)
    - Relationship:
        - 1:M One of the Users belongs to one of the Companies which can have many Users
        - M:N A user can sign multiple Forms and Forms can have Signatures from many Users
        - M:N A user can access multiple Files and Files can be used by many Users
- Forms: represent documents to be stored within Files and signed multiple times by Users
    - form_id: int, auto_increment, unique, not NULL, PK
    - form_name: varchar(145), not NULL
    - fk_file_id: int, auto_increment, unique, not NULL, FK(referencing Files)
    - Relationship:
        - 1:M - one to many, because a Forms can have multiple Signatures being signed by several Users
        - 1:M - one to many, because a Files can be made up of multiple Forms but Forms will only be used once in given Files
        - M:N - many to many, because Forms can be accessed by many Users and a Users can access multiple Forms
- Files: records the details of Files; files can have multiple Forms, multiple Users from the same Company can be given access to the same Files
    - file_id: int, auto_increment, unique, not NULL, PK
    - file_name: varchar(50), not NULL
    - fk_filescompany_id: int, unique, not NULL, FK(Referencing Companies)
    - Relationship:
        - 1:M - A File can contain multiple Forms, serves as storage for multiple documents
        - M:N - Users can access multiple files, and files can be accessed by multiple users.
- Signatures : Records an instance of a User signing a Form, includes date signed.

    - signature_id: int, auto_increment, unique, not NULL, PK
    - date_signed: timestamp, not NULL (Note: use timestamp instead of datetime or date so that each Signatures have a unique timestamp and cannot be duplicated)
    - fk_usersig_id: int, NULL, FK(referencing Users)
    - fk_usersform_id: int, NULL, FK(referencing Forms)
    - Relationship:
        - 1:M - Forms can have multiple Signatures, but each of the Signatures is a unique signing event with a secure timestamp

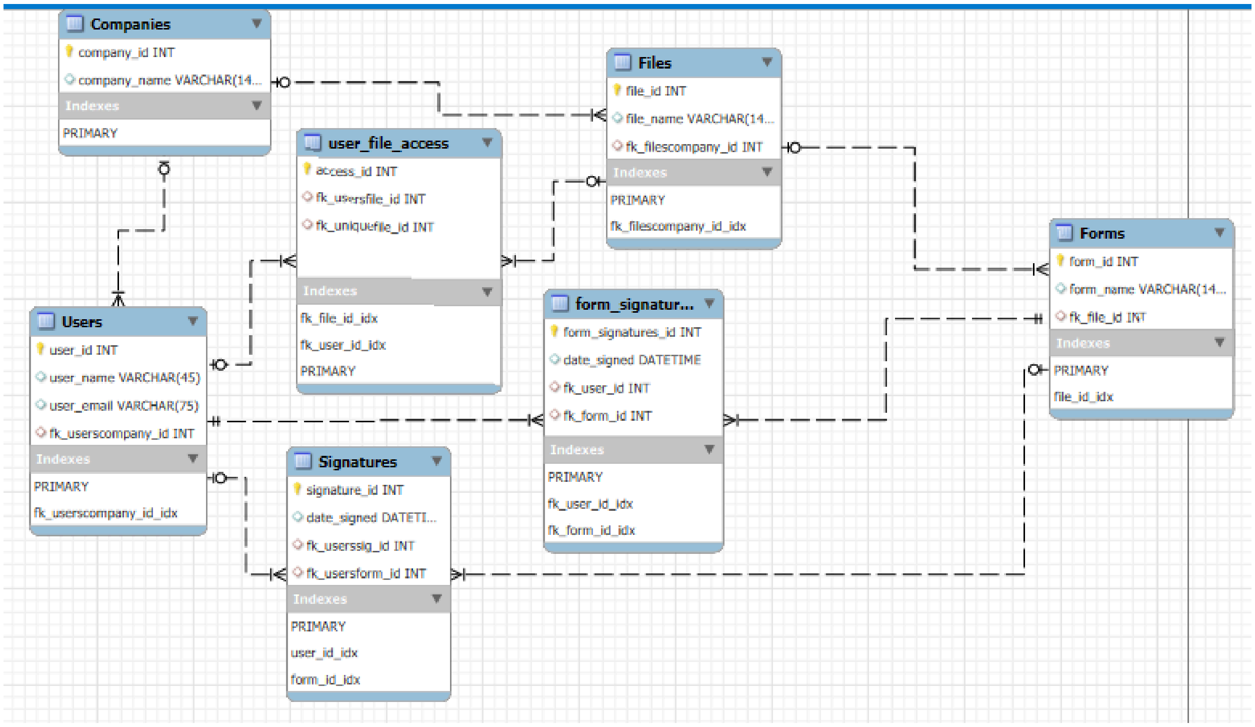
- user_file_access: A join table that will manage access rights for users to files
    - access_id: int, auto-increment, unique, not NULL, PK
    - fk_usersfile_id: INT, NULL, FK (referencing Users)
    - fk_uniquefile_id: INT, NULL, FK(referencing Files)
    - Relationships:
        - M:N - indicates which users can access which files

- form_signatures: A join table that establishes the relationship between a form and the signatures on it.
    - form_fignatures_id: INT, NOT NULL, PK
    - date_signed: timestamp, not NULL
    - fk_user_id: INT, NULL, FK (referencing Users)
    - fk_form_id: INT, NULL, FK (referencing Forms)
    - Relationship:
        - 1:M - indicates that a form can have signatures from multiple users

Entity Relationship Diagram

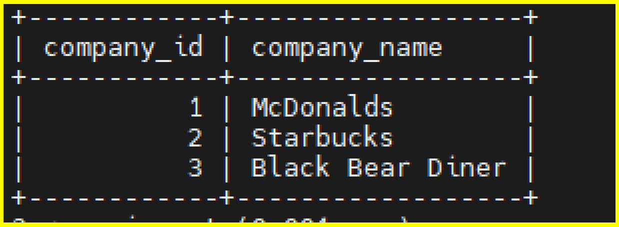


Schema



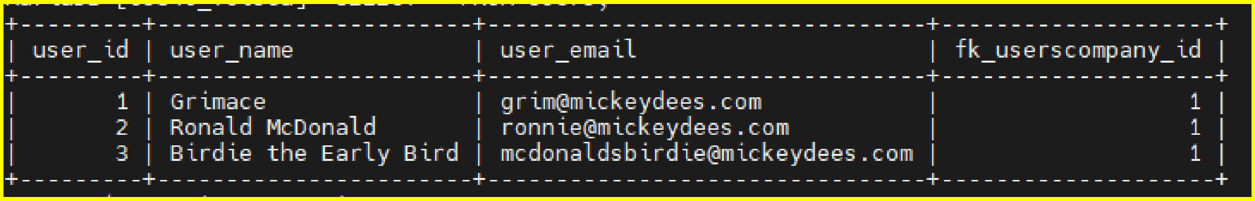
Example Data

Sample Companies Data



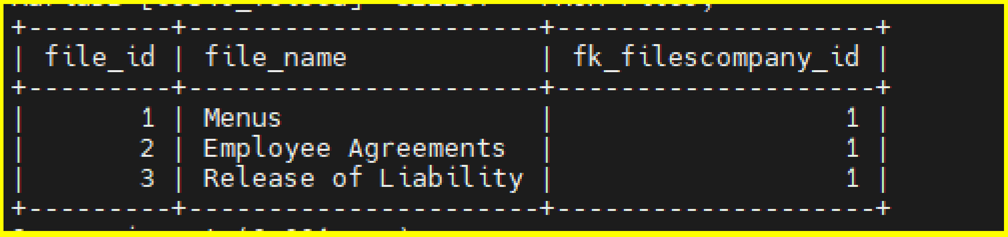
```
+------------+-------------------+
| company_id | company_name      |
+------------+-------------------+
|          1 | McDonalds         |
|          2 | Starbucks         |
|          3 | Black Bear Diner  |
+------------+-------------------+
```

Sample Users Data



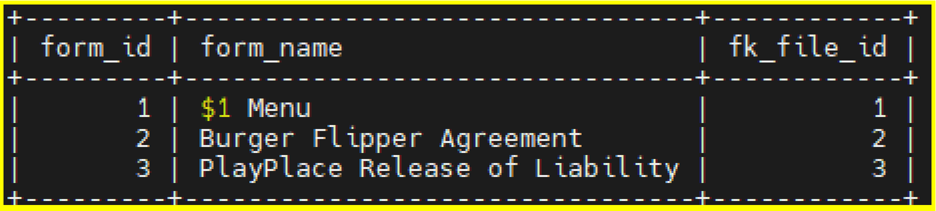
```
+---------+----------------------+-------------------------------+--------------------+
| user_id | user_name            | user_email                    | fk_userscompany_id |
+---------+----------------------+-------------------------------+--------------------+
|       1 | Grimace              | grim@mickeydees.com           |                  1 |
|       2 | Ronald McDonald      | ronnie@mickeydees.com         |                  1 |
|       3 | Birdie the Early Bird | mcdonaldsbirdie@mickeydees.com |                  1 |
+---------+----------------------+-------------------------------+--------------------+
```

Sample Files Data



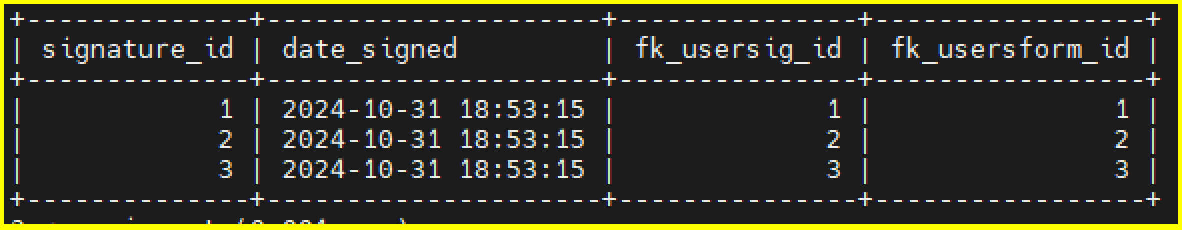
```
+---------+----------------------+--------------------+
| file_id | file_name            | fk_filescompany_id |
+---------+----------------------+--------------------+
|       1 | Menus                |                  1 |
|       2 | Employee Agreements  |                  1 |
|       3 | Release of Liability |                  1 |
+---------+----------------------+--------------------+
```

Sample Forms Data



```
+---------+--------------------------------+------------+
| form_id | form_name                      | fk_file_id |
+---------+--------------------------------+------------+
|       1 | $1 Menu                        |          1 |
|       2 | Burger Flipper Agreement       |          2 |
|       3 | PlayPlace Release of Liability |          3 |
+---------+--------------------------------+------------+
```

Sample Signatures Data



```
+--------------+---------------------+---------------+-----------------+
| signature_id | date_signed         | fk_usersig_id | fk_usersform_id |
+--------------+---------------------+---------------+-----------------+
|            1 | 2024-10-31 18:53:15 |             1 |               1 |
|            2 | 2024-10-31 18:53:15 |             2 |               2 |
|            3 | 2024-10-31 18:53:15 |             3 |               3 |
+--------------+---------------------+---------------+-----------------+
```