Project Step 3

CS340 Wedding Planners

Alex Wiedenhoeft

C. Kaye Dela Chica

Group 119

A) Fixes based on Feedback:

Feedback 1:

Does the UI show where it will utilize a SELECT for every table in the schema?
Yes; Viewing and updating Clients, Planned Weddings, Available Services, and Existing Payments are all possible through the user interface. These imply that data from related tables is displayed using SELECT queries. It's unclear, though, if every table's columns are displayed. Even if the table is empty, perhaps make sure that every attribute from the schema is visible.

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

No; The user interface lacks any obvious indication of search, filter, or dynamically populated dropdowns. Perhaps add a search box to the "View/Edit Planned Weddings" area that allows users to filter by date or client name. When planning weddings, dropdown menus in the "Available Services" section could be used to dynamically choose services.

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

Yes; The user interface implies INSERT capabilities by allowing users to add Clients, Weddings, Services, and Payments. It is unclear, nevertheless, if every attribute is present in every form. To guarantee thorough data entry, perhaps confirm that every field from the schema is included.

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

Partially; It appears that foreign keys are included because the user interface permits adding weddings and assigning clients. It's unclear, nevertheless, if the Wedding-Services (M:M) relationship is managed appropriately. Perhaps make sure that adding a wedding also adds pertinent rows to the intersection table that connects services and weddings.

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

No; DELETE options are not specifically available in the user interface. To ensure that related rows in the intersection table (Weddings-Services) are eliminated without erasing standalone Services or Clients, perhaps add a DELETE method for Weddings.

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

No; The UPDATE feature is not shown openly in the user interface. Perhaps include an update feature that would let consumers change service offerings, wedding dates, or client information.

Is at least one relationship NULLable?

Not Clear; The existence of optional associations is not indicated by the user interface. Perhaps think about allowing weddings to be scheduled but unpaid events by making payments optional.

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

No

Feedback 2:

- 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.
    - Each table in the schema is displayed in the UI. I can see the clients, weddings, services, and payments. The intersection table weddingServices is not present but that makes sense since it does not have any attributes of its own and is rather used only as an intersection table between services and weddings.
    - One thing I cannot find is that there is nowhere to see which wedding uses what service. So I would add a column in the Weddings table to display the services being used, and somehow make it so each wedding can have multiple services as well, since technically a wedding can have both catering and photography for example.
- 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?
    - At least one dropdown is used, and it is used in a very useful way. Service ID in the services table uses it, as well as the service type. Client ID in the client table uses it, invoice ID in the payment table,
- 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?
    - As mentioned above, there is a missing "services" option for when you add a wedding, and it also does not display the services applied to a wedding. Other than that, every table has the correct insert options
- 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?
    - The M:M relationship between weddings and services is not fully formed yet, as mentioned above again it is missing the service option in the insert section on the wedding page, and the service page does not display the weddings that use it, but honestly it makes sense that they would not display all the weddings that use a specific service since it is not necessary and can cause clutter.
- 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?
    - Each table has a delete option, and there is a delete option for an attribute involved in an M:M relationship, being the wedding service. If it is deleted, it does not delete the wedding.
- 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?
    - All tables have update options and they all are in proper places, too. The payment and wedding tables are missing one update option as they do not let you update the client ID,. I think unless maybe you by mistake assigned a wedding to a wrong client and want to update it, then otherwise this would not be necessary.
- 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?
    - The service is nullable in the wedding which makes sense since not every wedding will have a service provided.
- 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.
    - Other than what I mentioned before, I would suggest using aliases to change the column headers from names such as phoneNum to Phone Number, totalBudget to Total Budget, etc.

Feedback 3:
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, the tables displayed on the UI are clients, weddings, services, and payments. The intersection table (weddingServices) isn't included in the UI. One issue is that you can't see what services each wedding has. The intersection table facilitates the M:N relationship between weddings and services. You could join the intersection table to the weddings table in the UI so it could list information about which services the wedding has.

- *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?

Yes, the weddings form includes drop downs for the wedding type and service type, and the service form includes a dropdown for the service type. There are other drop downs for wedding id, payment id, and client id which should generally be avoided because the user may not know the id and may have to navigate between pages. For example, you can change it later so that it shows the names instead of the ids for client id. One feature you could add for the database administrator would be a search feature to search weddings by name, services, 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?

Yes for the most part, as there are the clients, weddings, services, and payments table, and each one has proper inserts. However, there isn't an ability to add data to the intersection table which is important for managing relationships.

- *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?

For the wedding table, there is an area to add the client id (FK). This also exists in the payments table, where there's an area to add the client id (FK). The intersection table, according to the schema, just includes a wedding id and the service id. That table facilitates the M:N relationship between weddings and services. There doesn't seem to be a way to insert into that table yet.

- *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 are delete forms for the clients, payments, weddings, and services tables. For the M:N relationship, when implemented, you can delete a service by ID and that would delete the record in the intersection table while leaving the wedding.

- *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?

There are updates for the clients, weddings, services, and payments tables. At the moment you can update these forms by the IDs, i.e. clientID, invoiceID, serviceID, weddingID. The wedding update form is missing a place to update the clientID if that ever changes.

- *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?

A wedding might not have any services and a service doesn't have to be connected to a wedding. There is an optional M:N relationship between weddings and services. It also

looks like the serviceType and serviceCost are optional, as well as the weddingType, invoiceDate, totalAmount, and paymentDate.

- *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.*

As stated above, the group can use aliases to replace the column names. For each delete form, there's two lines of whitespace between the form and the "Delete Payment" button. They appear as
 in the HTML and I'm not sure if it's on purpose but it's inconsistent with the add and update forms.

Feedback 4:

- *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 clearly displays a SELECT for every entity in the schema. I'm not entirely sure where the weddingServices junction table is displayed in the UI. The represented tables are Clients, Weddings, Services, and Payments.

- *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?

Every page has an update and delete form for entries in the table. Each of these forms has a dropdown menu which will dynamically populate with the ids of that table. Furthermore, weddings has an add Service dropdown which will dynamically contain the options for services available for the weddings.

- *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?

Every table has an INSERT implementation. For the Clients, Services, Weddings, and Payments tables, this is an add form on the associated page. For the weddingServices table, this is included in the Update Wedding add service dropdown. As such, every table has a method to INSERT from the UI.

- *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?

Every INSERT appears to be set up correctly to add the corresponding FK attributes. For the M:M relationship between Weddings and Services, the aforementioned add Service dropdown in the update wedding form should INSERT the FKs for the given Weddings entry and the associated Services entry into weddingServices upon submission.

- *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 for every table that is provided as a form on each corresponding page. However, I am unsure if there is an apparent way to remove an entry from weddingServices. The available DELETEs are for Services, Weddings, Clients, and Payments.

- *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?

There is an UPDATE for every table that is provided as a form on each corresponding page. However, I am unsure if there is an apparent way to update an entry from weddingServices. The available UPDATEs are for Services, Weddings, Clients, and Payments.

- *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?

The relationship between Services and Weddings is a M:M relationship, so it is naturally optional. However, I am unsure if this counts as a NULLable relationship. Otherwise, I do not see any obvious examples of a NULLable relationship in this UI. However, it does not seem that one is necessary.

- *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.*

This UI is incredibly clear and seems to fit the requirements well. As such, I have no other suggestions.

Our Changes:

The feedback we received from Project 1's draft centered around the lack of specifics. Our project was focused on event planning, where users can plan any event. This would've made it difficult to create an application, so as a result we decided to narrow down our focus to wedding planning. By clearly stating a problem, we are able to center our focus on one main project. With this, we have been able to establish a direct target audience, such as couples and wedding planners (clients) that can manage their weddings by overseeing their services, payments, and wedding details.

Another piece of feedback was specifying entities. Our project continues to have four entities: clients, weddings, services, and payments. Before, our attributes were vague, now they are much more descriptive, making it easier to understand.

We used feedback from the second draft of our project to make several changes as well. Firstly, we changed the phoneNum attribute's data type in the schema from varchar to int and fixed minor naming inconsistencies in the ERD and schema. We also added the intersection table weddingServices to our ERD. In addition, we added a nullable relationship in the weddingServices entity. Lastly, we made sure to include the INSERT statements in our data definition queries.

We used the feedback from the third draft to add a "Wedding Services" page to the UI. Users will be able to view, add, and remove services to/from weddings here. We also implemented aliases for the SELECT queries for each table (e.g. phoneNum to Phone Number). These changes helped to add clarity to the UI and to flesh out the M:M relationship between Weddings and Services.

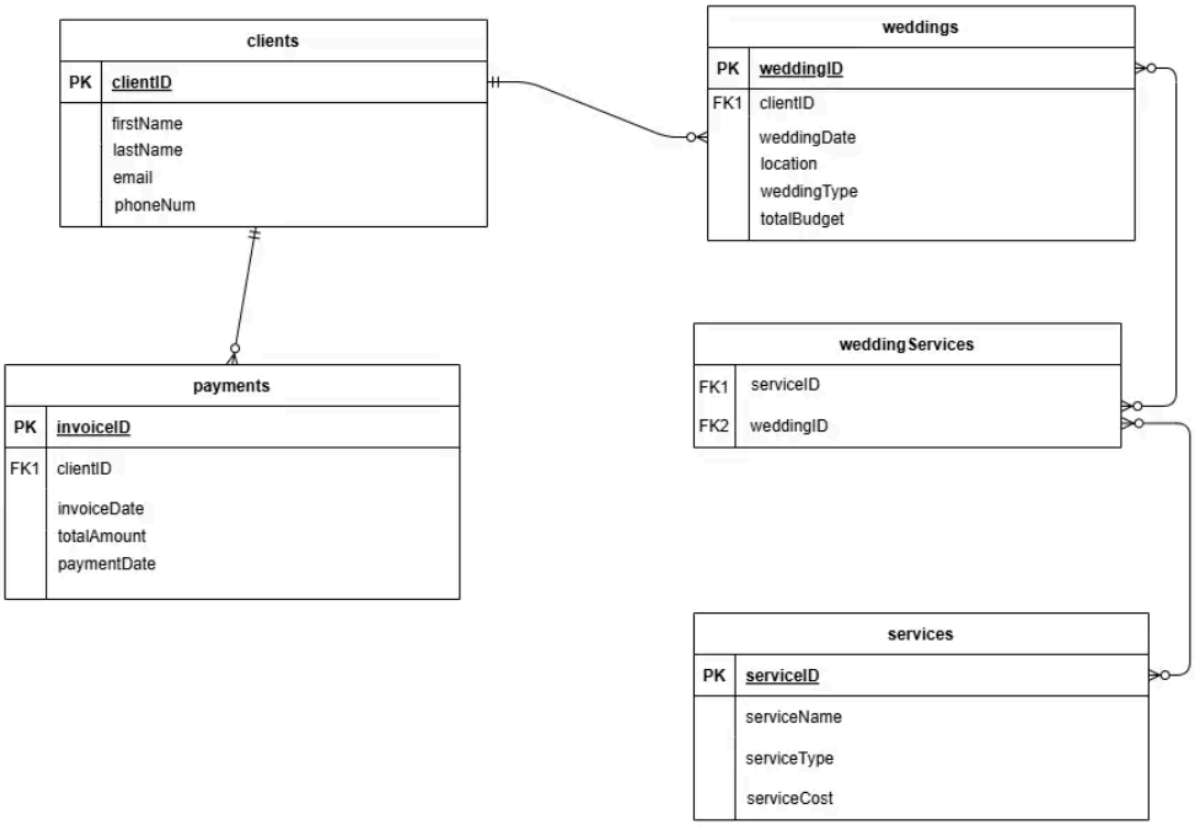
B) Project Outline and Database Outline - Updated Version:

Database Outline

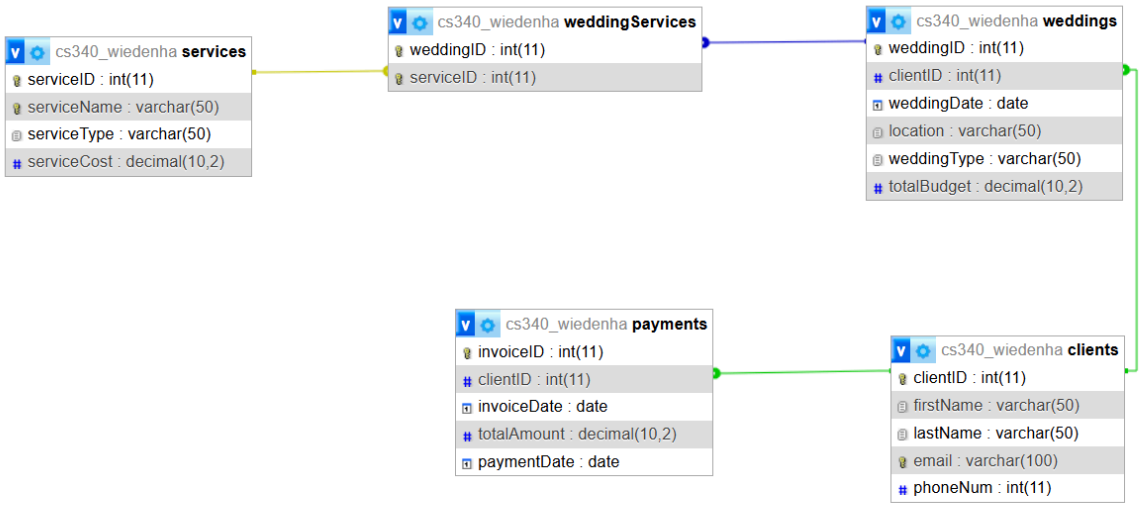
- Clients: Record information of the clients who plan the weddings
    - clientID: int(11) , auto_increment, unique, not NULL, PK
    - firstName: varchar(50), not NULL
    - lastName: varchar(50), not NULL
    - email: varchar, not NULL
    - phoneNum: int, not NULL
    - Relationship(s): A 1:M relationship with the Weddings entity will be implemented with clientID as a FK inside Weddings. A 1:M relationship with the Payments entity will be implemented with clientID as a FK inside Payments.
- Weddings: Base information on the event
    - weddingID: int, auto_increment, unique, not NULL, PK
    - clientID: int, not NULL, FK
    - weddingDate: date, not NULL
    - location: varchar (50), not NULL
    - weddingType: varchar(50)
    - totalBudget: decimal, not NULL
    - Relationship(s): A M:1 relationship with the Clients entity will be implemented with clientID as a FK inside Wedding. An optional M:M relationship with Services will be created with an intersection table called Wedding Services, which will have the serviceID from the Services entity and the weddingID from the Weddings entity as foreign keys.
- Services: Keep track of the different services that the wedding may need (bands, food, decor)
    - serviceID: int, auto_increment, unique, not NULL, PK
    - serviceName: varchar(50), unique, not NULL
    - serviceType: varchar(50)
    - serviceCost: decimal(10, 2)
    - Relationship(s): An optional M:M relationship with Weddings will be created with an intersection table called Wedding Services, which will have the serviceID from the Services entity and the weddingID from the Weddings entity as foreign keys.
- Payments: Records total amount charged to client’s account

- invoiceID: int, auto_increment, unique, not NULL, PK
- clientID: int, not NULL, FK
- invoiceDate: date
- totalAmount: decimal(10, 2)
- paymentDate: date
- Relationship(s): A M:1 relationship with the Clients entity will be implemented with clientID as a FK inside Payments.

C) Entity-Relationship Diagram:



D) Schema:


cs340_wiedenha services
serviceID : int(11)
serviceName : varchar(50)
serviceType : varchar(50)
serviceCost : decimal(10,2)
cs340_wiedenha weddingServices
weddingID : int(11)
serviceID : int(11)
cs340_wiedenha weddings
weddingID : int(11)
clientID : int(11)
weddingDate : date
location : varchar(50)
weddingType : varchar(50)
totalBudget : decimal(10,2)
cs340_wiedenha payments
invoiceID : int(11)
clientID : int(11)
invoiceDate : date
totalAmount : decimal(10,2)
paymentDate : date
cs340_wiedenha clients
clientID : int(11)
firstName : varchar(50)
lastName : varchar(50)
email : varchar(100)
phoneNum : int(11)

E) Example Data:

Clients				
clientID	First Name	Last Name	email	Phone num
1	Alex	Wiedenhoeft	wiedenhoeft@osu.com	(503) 473-1234
2	Kaye	DelaChica	delachica@osu.com	(808) 398-1234
3	Tony	James	james@osu.com	(619) 845-0040
4	Gina	Lee	lee@osu.com	(971) 570-1234
5	Henry	Adams	adams@osu.com	(626) 432-4321

Weddings					
weddingID	clientID	weddingDate	location	weddingType	totalBudget
101	1	2025-12-03	Japan	Garden	20000
102	2	2027-11-26	Hawaii	Traditional	25000
103	3	2026-07-23	Hawaii	Beach	18000
104	4	2026-04-14	Chicago	City	30000
105	5	2026-01-01	Oregon	Black Tie	7500

Services			
serviceID	serviceName	serviceType	serviceCost
201	Live Band	Music	2500
202	Catering	Food	7000
203	Photographer	Photography	6000
204	Stylist	Decor	4000
205	Bartender	Food	1500

Payments				
invoiceID	clientID	invoiceDate	totalAmount	paymentDate
301	1	2025-01-03	20000	2025-07-30
302	2	2027-03-26	25000	2027-05-24
303	3	2026-01-23	18000	2026-04-02
304	4	2025-12-14	30000	2026-01-18
305	5	2025-10-01	7500	2025-12-15

Wedding Services	
WeddingID	serviceID
101	201
102	202
103	203
104	204
105	205