**1. Users Table**

* **Entity Type**: Strong Entity
* **Attributes**:
  + **user\_id** (Primary Key, Key Attribute)
  + name
  + email (Unique, Key Attribute)
  + password
  + role
  + created\_at
  + updated\_at
* **Key Attributes**:
  + **user\_id**, **email** (Unique)
* **Derived Attributes**: None

**2. Artists Table**

* **Entity Type**: Strong Entity
* **Attributes**:
  + **artist\_id** (Primary Key, Key Attribute)
  + **user\_id** (Foreign Key to Users)
  + portfolio\_link
  + skill\_tags
  + certifications
  + bio
  + created\_at
  + updated\_at
* **Key Attributes**:
  + **artist\_id**
  + **user\_id** (Foreign Key to Users)
* **Derived Attributes**: None

**3. Services Table**

* **Entity Type**: Strong Entity
* **Attributes**:
  + **service\_id** (Primary Key, Key Attribute)
  + **artist\_id** (Foreign Key to Artists)
  + title
  + description
  + price
  + category
  + created\_at
  + updated\_at
* **Key Attributes**:
  + **service\_id**
  + **artist\_id** (Foreign Key to Artists)
* **Derived Attributes**: None

**4. Bookings Table**

* **Entity Type**: Strong Entity
* **Attributes**:
  + **booking\_id** (Primary Key, Key Attribute)
  + **service\_id** (Foreign Key to Services)
  + **user\_id** (Foreign Key to Users)
  + booking\_date
  + status (Pending/Confirmed/Completed)
  + created\_at
  + updated\_at
* **Key Attributes**:
  + **booking\_id**
  + **service\_id** (Foreign Key to Services)
  + **user\_id** (Foreign Key to Users)
* **Derived Attributes**: None

**5. Blogs Table**

* **Entity Type**: Strong Entity
* **Attributes**:
  + **blog\_id** (Primary Key, Key Attribute)
  + **author\_id** (Foreign Key to Artists)
  + title
  + content
  + created\_at
  + updated\_at
* **Key Attributes**:
  + **blog\_id**
  + **author\_id** (Foreign Key to Artists)
* **Derived Attributes**: None

**6. Comments Table**

* **Entity Type**: Strong Entity
* **Attributes**:
  + **comment\_id** (Primary Key, Key Attribute)
  + **blog\_id** (Foreign Key to Blogs)
  + **user\_id** (Foreign Key to Users)
  + comment\_text
  + created\_at
  + updated\_at
* **Key Attributes**:
  + **comment\_id**
  + **blog\_id** (Foreign Key to Blogs)
  + **user\_id** (Foreign Key to Users)
* **Derived Attributes**: None

**7. Events Table**

* **Entity Type**: Strong Entity
* **Attributes**:
  + **event\_id** (Primary Key, Key Attribute)
  + **artist\_id** (Foreign Key to Artists)
  + title
  + description
  + event\_date
  + location
  + created\_at
  + updated\_at
* **Key Attributes**:
  + **event\_id**
  + **artist\_id** (Foreign Key to Artists)
* **Derived Attributes**: None

**8. Exhibitions Table**

* **Entity Type**: Strong Entity
* **Attributes**:
  + **exhibition\_id** (Primary Key, Key Attribute)
  + **event\_id** (Foreign Key to Events)
  + **artist\_id** (Foreign Key to Artists)
  + artwork\_title
  + artwork\_description
  + created\_at
  + updated\_at
* **Key Attributes**:
  + **exhibition\_id**
  + **event\_id** (Foreign Key to Events)
  + **artist\_id** (Foreign Key to Artists)
* **Derived Attributes**: None

**9. Ratings Table**

* **Entity Type**: Strong Entity
* **Attributes**:
  + **rating\_id** (Primary Key, Key Attribute)
  + **service\_id** (Foreign Key to Services)
  + **user\_id** (Foreign Key to Users)
  + rating\_value
  + review\_text
  + created\_at
  + updated\_at
* **Key Attributes**:
  + **rating\_id**
  + **service\_id** (Foreign Key to Services)
  + **user\_id** (Foreign Key to Users)
* **Derived Attributes**: None

**10. Payments Table**

* **Entity Type**: Strong Entity
* **Attributes**:
  + **payment\_id** (Primary Key, Key Attribute)
  + **booking\_id** (Foreign Key to Bookings)
  + amount
  + payment\_status (Success/Failed/Pending)
  + payment\_date
  + created\_at
  + updated\_at
* **Key Attributes**:
  + **payment\_id**
  + **booking\_id** (Foreign Key to Bookings)
* **Derived Attributes**: None

**11. Categories Table**

* **Entity Type**: Strong Entity
* **Attributes**:
  + **category\_id** (Primary Key, Key Attribute)
  + name
  + description
  + created\_at
  + updated\_at
* **Key Attributes**:
  + **category\_id**
* **Derived Attributes**: None

**12. Followers Table**

* **Entity Type**: Weak Entity (because it depends on both Users and Artists)
* **Attributes**:
  + **follower\_id** (Primary Key, Key Attribute)
  + **artist\_id** (Foreign Key to Artists)
  + **user\_id** (Foreign Key to Users)
  + followed\_at
* **Key Attributes**:
  + **follower\_id**
  + **artist\_id** (Foreign Key to Artists)
  + **user\_id** (Foreign Key to Users)
* **Derived Attributes**: None

**13. Notifications Table**

* **Entity Type**: Strong Entity
* **Attributes**:
  + **notification\_id** (Primary Key, Key Attribute)
  + **user\_id** (Foreign Key to Users)
  + message
  + is\_read
  + created\_at
* **Key Attributes**:
  + **notification\_id**
  + **user\_id** (Foreign Key to Users)
* **Derived Attributes**: None

**14. Portfolios Table**

* **Entity Type**: Strong Entity
* **Attributes**:
  + **portfolio\_id** (Primary Key, Key Attribute)
  + **artist\_id** (Foreign Key to Artists)
  + portfolio\_url
  + description
  + created\_at
  + updated\_at
* **Key Attributes**:
  + **portfolio\_id**
  + **artist\_id** (Foreign Key to Artists)
* **Derived Attributes**: None

**15. Messages Table**

* **Entity Type**: Strong Entity
* **Attributes**:
  + **message\_id** (Primary Key, Key Attribute)
  + **sender\_id** (Foreign Key to Users)
  + **receiver\_id** (Foreign Key to Users)
  + content
  + sent\_at
* **Key Attributes**:
  + **message\_id**
  + **sender\_id** (Foreign Key to Users)
  + **receiver\_id** (Foreign Key to Users)
* **Derived Attributes**: None

Here are the relationships between the tables in your **Artistry Hub** application:

**1. Users - Artists**

* **Relationship**: 1-to-1 (One user can have one artist profile)
* **Foreign Key**: **user\_id** in **Artists** table references **user\_id** in **Users** table.

**2. Artists - Services**

* **Relationship**: 1-to-Many (One artist can offer multiple services)
* **Foreign Key**: **artist\_id** in **Services** table references **artist\_id** in **Artists** table.

**3. Users - Bookings**

* **Relationship**: 1-to-Many (One user can make multiple bookings)
* **Foreign Key**: **user\_id** in **Bookings** table references **user\_id** in **Users** table.

**4. Services - Bookings**

* **Relationship**: 1-to-Many (One service can have multiple bookings)
* **Foreign Key**: **service\_id** in **Bookings** table references **service\_id** in **Services** table.

**5. Users - Comments**

* **Relationship**: 1-to-Many (One user can post multiple comments)
* **Foreign Key**: **user\_id** in **Comments** table references **user\_id** in **Users** table.

**6. Blogs - Comments**

* **Relationship**: 1-to-Many (One blog can have multiple comments)
* **Foreign Key**: **blog\_id** in **Comments** table references **blog\_id** in **Blogs** table.

**7. Artists - Blogs**

* **Relationship**: 1-to-Many (One artist can write multiple blogs)
* **Foreign Key**: **author\_id** in **Blogs** table references **artist\_id** in **Artists** table.

**8. Artists - Events**

* **Relationship**: 1-to-Many (One artist can host multiple events)
* **Foreign Key**: **artist\_id** in **Events** table references **artist\_id** in **Artists** table.

**9. Events - Exhibitions**

* **Relationship**: 1-to-Many (One event can have multiple exhibitions)
* **Foreign Key**: **event\_id** in **Exhibitions** table references **event\_id** in **Events** table.

**10. Artists - Exhibitions**

* **Relationship**: 1-to-Many (One artist can have multiple exhibitions)
* **Foreign Key**: **artist\_id** in **Exhibitions** table references **artist\_id** in **Artists** table.

**11. Services - Ratings**

* **Relationship**: 1-to-Many (One service can have multiple ratings)
* **Foreign Key**: **service\_id** in **Ratings** table references **service\_id** in **Services** table.

**12. Users - Ratings**

* **Relationship**: 1-to-Many (One user can give multiple ratings)
* **Foreign Key**: **user\_id** in **Ratings** table references **user\_id** in **Users** table.

**13. Bookings - Payments**

* **Relationship**: 1-to-1 (One booking can have one payment)
* **Foreign Key**: **booking\_id** in **Payments** table references **booking\_id** in **Bookings** table.

**14. Users - Notifications**

* **Relationship**: 1-to-Many (One user can receive multiple notifications)
* **Foreign Key**: **user\_id** in **Notifications** table references **user\_id** in **Users** table.

**15. Artists - Portfolios**

* **Relationship**: 1-to-Many (One artist can have multiple portfolios)
* **Foreign Key**: **artist\_id** in **Portfolios** table references **artist\_id** in **Artists** table.

**16. Users - Followers**

* **Relationship**: Many-to-Many (Users can follow multiple artists and vice versa)
* **Join Table**: **Followers** (with **user\_id** and **artist\_id** as Foreign Keys, creating a Many-to-Many relationship)

**17. Users - Messages**

* **Relationship**: Many-to-Many (Users can send messages to each other)
* **Join Table**: **Messages** (with **sender\_id** and **receiver\_id** as Foreign Keys, creating a Many-to-Many relationship)

**DFD**











