Jennifer Der and Colin McLean
CS 3200 Summer 1 2017
Art History Database

README

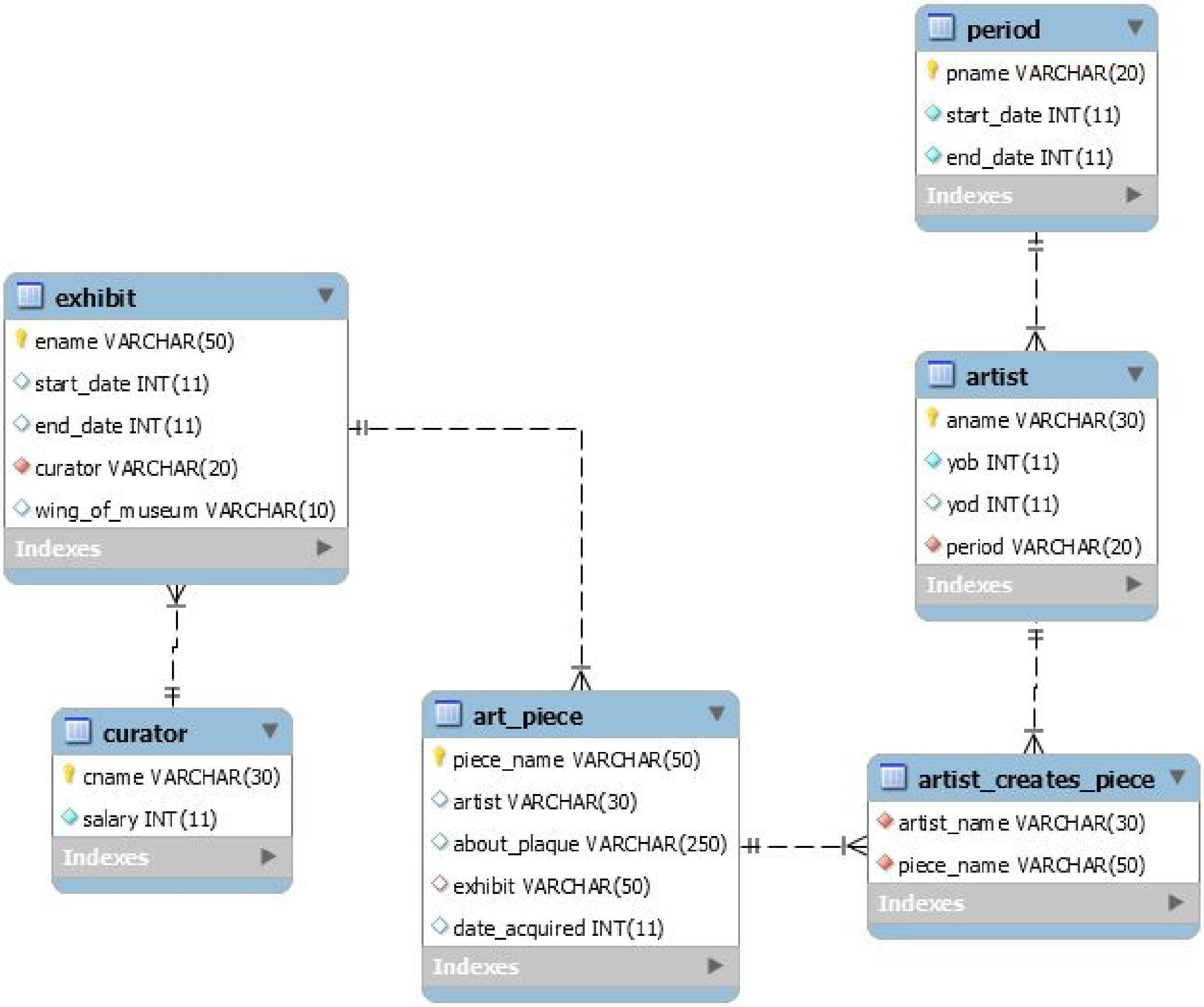
To Download:

- Applications needed:
  - MySQL Workbench
  - Eclipse (Though it should run from the command line)
    - java libraries that we used should already be imported via "import" statement
  - Java
- Steps:
  - 1. Open the arthistorydb sql file in MySQL, execute it
  - 2. Open the procedures sql file in MySQL, execute it
  - 3. Open the javaDB file in Eclipse, change username and password in java code to match your personal mySQL username and password, they are the first two variables in the ArtDB class run the code
    - Instructions about how to do CRUD operations should pop up onto the console, this is more deeply explained in the user flow diagram

Technical Specifications:

- For our frontend connection created a command line application/java frontend prompt that poses questions to the user via mySQL. The backend of the database will be in mySQL. We will be using the connectors that can be found at https://dev.mysql.com/downloads/connector/
- There are not many existing APIs that we can use for art or art history in general so we input a few example data points by hand via comma separated values using the Data Import Wizard in mySQL. After importing the data, we exported the entirety of the database as a self contained sql file so that the initial user has some initial data to view and work with. This seems to be something a museum would do by hand mostly anyway, since curators, artists, and pieces are added slowly to a museum collection, and not pulled right off a single source at once.

EER:



period
pname VARCHAR(20)
start_date INT(11)
end_date INT(11)
Indexes
exhibit
ename VARCHAR(50)
start_date INT(11)
end_date INT(11)
curator VARCHAR(20)
wing_of_museum VARCHAR(10)
Indexes
artist
aname VARCHAR(30)
yob INT(11)
yod INT(11)
period VARCHAR(20)
Indexes
curator
cname VARCHAR(30)
salary INT(11)
Indexes
art_piece
piece_name VARCHAR(50)
artist VARCHAR(30)
about_plaque VARCHAR(250)
exhibit VARCHAR(50)
date_acquired INT(11)
Indexes
artist_creates_piece
artist_name VARCHAR(30)
piece_name VARCHAR(50)
Indexes

UML:

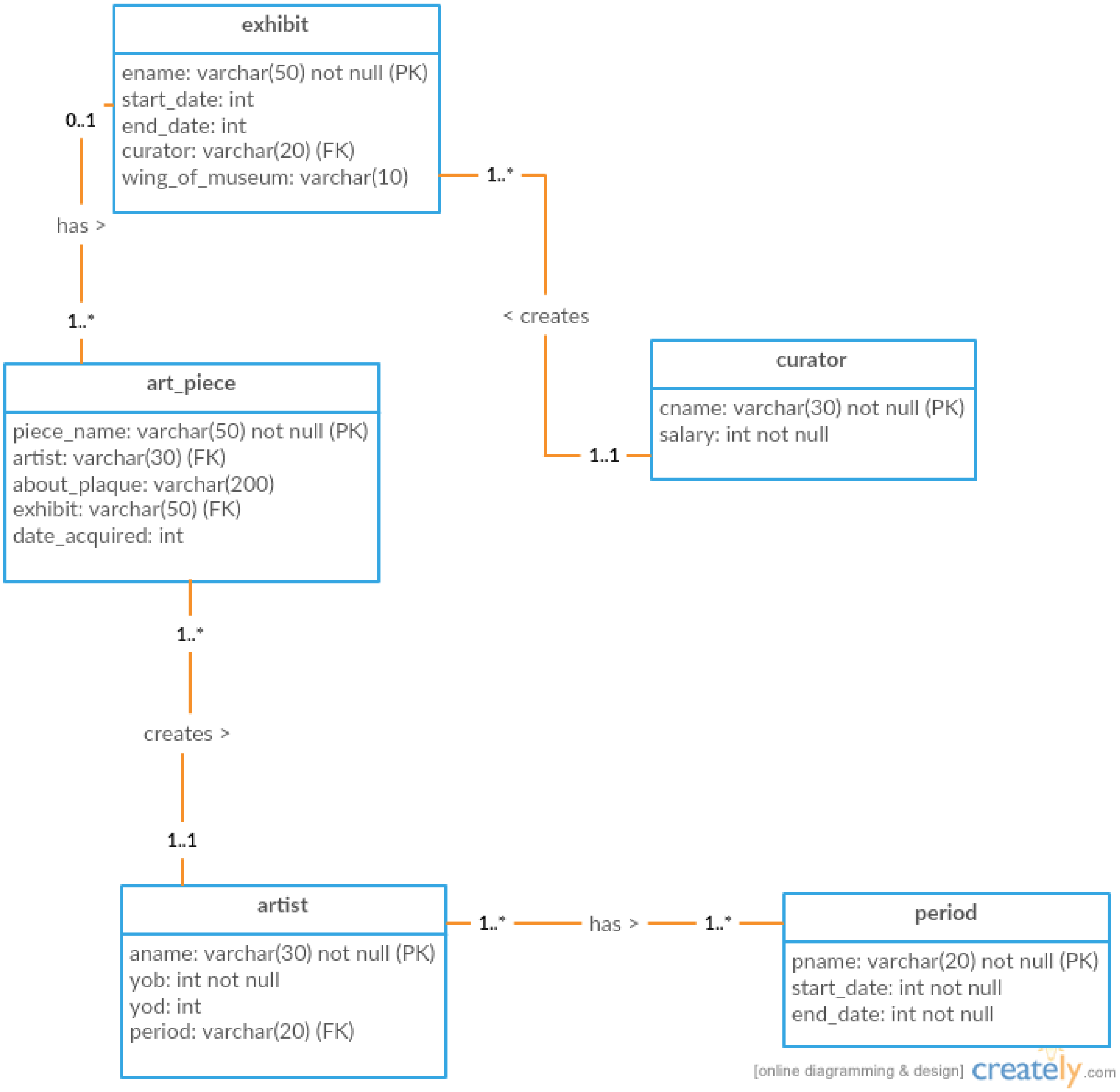
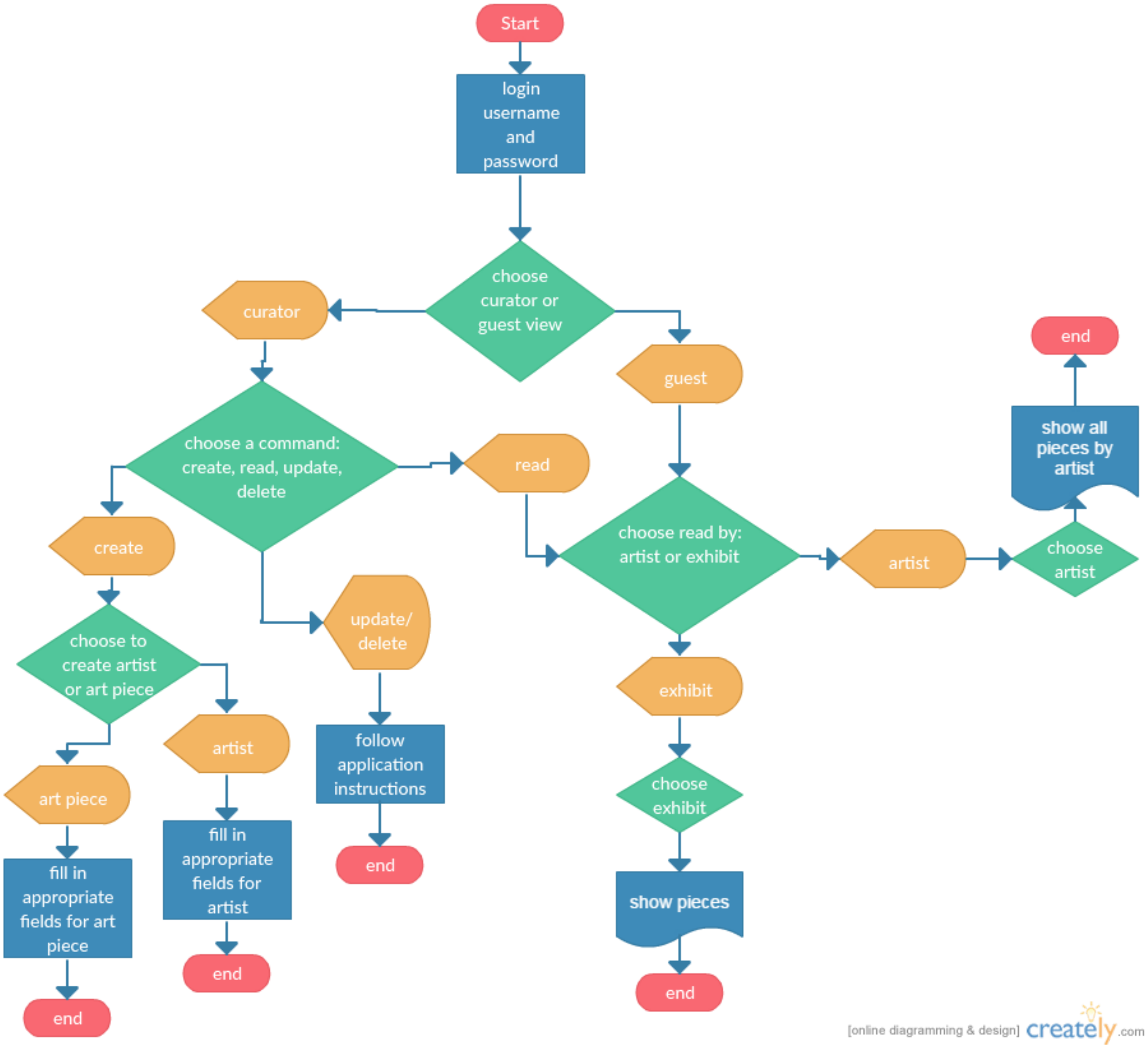


exhibit
ename: varchar(50) not null (PK)
start_date: int
end_date: int
curator: varchar(20) (FK)
wing_of_museum: varchar(10)
0..1
has >
1..*
1..*
< creates
1..1
curator
cname: varchar(30) not null (PK)
salary: int not null
art_piece
piece_name: varchar(50) not null (PK)
artist: varchar(30) (FK)
about_plaque: varchar(200)
exhibit: varchar(50) (FK)
date_acquired: int
1..*
creates >
1..1
artist
aname: varchar(30) not null (PK)
yob: int not null
yod: int
period: varchar(20) (FK)
1..*
has >
1..*
period
pname: varchar(20) not null (PK)
start_date: int not null
end_date: int not null
[online diagramming & design] creately.com

User Flow:


Start
login username and password
choose curator or guest view
curator
guest
choose a command: create, read, update, delete
read
create
update/ delete
choose read by: artist or exhibit
artist
choose artist
show all pieces by artist
end
choose to create artist or art piece
artist
art piece
fill in appropriate fields for artist
fill in appropriate fields for art piece
follow application instructions
end
end
end
exhibit
choose exhibit
show pieces
end
[online diagramming & design] creately.com

Lessons Learned:

- The technical experience that we both gained was in the form of learning more about mySQL and java and what it takes to connect the two. Also, neither of us had ever really worked with the command line before or creating a user application that had multiple steps that completely depended on user input. The project as a whole was definitely a steep learning curve for the two of us.
- We have worked together in the past but not while having to learn a substantial amount of technical abilities. In general, we worked well together and managed our time well. However, we wish there were more time in the semester so that we could implement more user access. We were able to implement the CRUD operations over several data types, in ways that we saw as beneficial and provided the most utility to the user but wish that the implementation itself was cleaner.

- We realized we did not have as much time to implement all of the views and utility that we initially set out to do, however implemented all the necessary functionality that would make the database useful. We did not figure out how to connect to the internet, or make a web application because connecting our database in mySQL to java was enough of a learning curve. We could not really figure out how to connect the data to the internet or even locally host a web application in time for us to implement the entire project on the application.
- We did not create a separate table to represent artist_creates_piece, or rather did not insert any data into this table because it was not a many to many relationship. Only a piece is created by an artist but the artist does not have direct access to the pieces they have created strictly through the artist table. In order to find the pieces created by a given artist, one must sort through the art_piece table where the artist name equals the given artist. Overall, all of the code should be functional.

Future work:

- The purpose of the database is to provide information about art history to the user. We hope to provide information sorted by time periods, art periods, and by artists. The user should be able to access basic information about the time period in which the artist lived, notable pieces, as well as location (ie exhibit) in which pieces are shown. Because of the design of the database, and its implementation it is a bit simplistic to be used completely by any exhibit and museum setting. However, it could be used as personal use in order to keep track of one’s own artwork, effectively sorting artwork by time/year it was created, location, collaborations etc.
- Added functionality might come along with more security for the curator access point because as of now, the curator can just access the database by saying that they are a curator, and does not need to provide any password. We could have chosen to add a hardcoded “password” that the curator would have to provide in order to be able to create, update or delete rows from the database, however this seemed like a cheap fix and not essential to our objectives with creating this database.