



Lebanese American University

School of Arts and Sciences

Department of Computer Science and Mathematics



A Database Design for Ageless Museum

By Group #4 “Recursive Query Backtracking”

# A Report Done By

**Mahmoud Joumaa** (Phase 4 Group Leader)

[mahmoud.joumaa@lau.edu](mailto:mahmoud.joumaa@lau.edu)

**Omar Fakhredin**

[omar.fakhredin@lau.edu](mailto:omar.fakhredin@lau.edu)

**Hussein Ali Ahmad**

[hussein.aliahmed@lau.edu](mailto:hussein.aliahmed@lau.edu)

And submitted to Dr. Hussein K. Bakri in partial fulfillment of the requirements of phase 3 of the database project for the CSC375 (Database Management Systems) course during the semester of Fall 2022

December 2022

## Table of Contents

I.	Introduction.....	13
II.	Copyright – MIT License.....	14
III.	Tool(s) used throughout this project .....	15
I.	To draw the Entity Relationship Model .....	15
I.	What is Microsoft Visio? .....	15
II.	Why Microsoft Visio?.....	15
II.	To generate the fictitious logo .....	15
IV.	System Description and Requirements .....	16
V.	Entity Relationship Model Diagram Legend .....	19
VI.	Complete ER Diagram for Ageless Museum.....	21
VII.	Entities and Their Attributes .....	22
01.	Piece .....	22
02.	Artist .....	23
03.	Section.....	24
04.	Provisioner .....	25
05.	Mediator .....	26
06.	Supplier.....	27
07.	Equipment .....	28
08.	Partner .....	29
09.	Sponsor .....	30
10.	Event .....	31
11.	Department.....	32
12.	Employee .....	33
13.	Reward .....	35
14.	Medical Record.....	36
15.	Police Record.....	37
16.	Emergency Contact.....	38

17.	Dependent .....	39
18.	Feedback Program.....	40
19.	Visitor .....	41
VIII.	Relationships and Their Explanations.....	43
01.	provides.....	43
02.	supplies .....	44
03.	creates .....	45
04.	is displayed.....	46
05.	is overseen.....	47
06.	earns .....	47
07.	maintains .....	48
08.	possesses .....	49
09.	has an .....	50
10.	has a .....	51
11.	is governed .....	52
12.	is managed .....	52
13.	visits .....	53
14.	buys.....	54
15.	provides feedback .....	55
16.	is hosted .....	56
17.	donates .....	56
18.	sponsors.....	57
19.	supervision .....	57
IX.	ER to Relational Mapping.....	58
01.	Mapping Strong Entity Types.....	58
01.	Piece.....	58

02.	Artist .....	58
03.	Section.....	59
04.	Provisioner .....	59
05.	Mediator .....	59
06.	Supplier.....	59
07.	Equipment.....	60
08.	Partner.....	60
09.	Sponsor .....	60
10.	Event .....	60
11.	Department.....	61
12.	Employee .....	61
13.	Feedback Program.....	61
14.	Visitor .....	61
02.	Mapping of Weak Entity Types .....	62
01.	Reward .....	62
02.	Medical Record.....	62
03.	Police Record.....	62
04.	Emergency Contact.....	63
05.	Dependent .....	63
03.	Mapping of Binary 1:1 Relationship Types .....	63
01.	maintains.....	63
02.	possesses .....	63
04.	Mapping of Binary 1:N Relationship Types .....	64
01.	is displayed.....	64
02.	earns .....	64
03.	has an .....	64

04.	has a .....	64
05.	buys .....	65
06.	provides feedback .....	65
05.	Mapping of Binary M:N Relationship Types.....	65
01.	creates .....	65
02.	is overseen.....	65
03.	is governed.....	66
04.	is managed .....	66
05.	visits .....	66
06.	donates .....	66
07.	sponsors.....	67
08.	supervision .....	67
06.	Mapping of Multivalued Attributes .....	68
01.	Employee Nationality .....	68
02.	Interests.....	68
03.	Comments .....	68
04.	Tags.....	69
05.	Culture.....	69
06.	Extension.....	69
07.	Mapping of N-ary Relationships.....	70
01.	provides.....	70
02.	supplies .....	70
03.	is hosted .....	70
X.	Comprehensive Overview of the Established Relations .....	71
XI.	Creating Tables on Oracle 11g Express Edition .....	76
01.	Creating Table: Piece.....	76
02.	Creating Table: Artist.....	77

03.	Creating Table: Section.....	77
04.	Creating Table: Provisioner .....	78
05.	Creating Table: Mediator .....	78
06.	Creating Table: Supplier .....	79
07.	Creating Table: Equipment .....	79
08.	Creating Table: Partner .....	80
09.	Creating Table: Sponsor.....	80
10.	Creating Table: Event .....	81
11.	Creating Table: Department.....	82
12.	Creating Table: Employee .....	83
13.	Creating Table: Feedback Program.....	84
14.	Creating Table: Visitor.....	84
15.	Creating Table: Reward .....	85
16.	Creating Table: Medical Record .....	85
17.	Creating Table: Police Record .....	86
18.	Creating Table: Emergency Contact .....	87
19.	Creating Table: Dependent .....	88
20.	Creating Table: Creates.....	89
21.	Creating Table: Is Overseen.....	89
22.	Creating Table: Is Governed .....	90
23.	Creating Table: Is Managed .....	90
24.	Creating Table: Visits .....	91
25.	Creating Table: Donates.....	92
26.	Creating Table: Sponsors .....	92
27.	Creating Table: Supervision.....	93
28.	Creating Table: Employee Nationality.....	93

29.	Creating Table: Interests .....	94
30.	Creating Table: Comments .....	94
31.	Creating Table: Tags .....	95
32.	Creating Table: Cultures .....	95
33.	Creating Table: Department Extension.....	96
34.	Creating Table: Provides.....	97
35.	Creating Table: Supplies.....	98
36.	Creating Table: Is Hosted .....	99
<b>XII.</b>	<b>Inserting Data Using SQL.....</b>	<b>100</b>
01.	Inserting Data into Piece .....	100
02.	Inserting Data into Artist.....	103
03.	Inserting Data into Section.....	104
04.	Inserting Data into Provisioner .....	105
05.	Inserting Data into Mediator .....	106
06.	Inserting Data into Supplier .....	107
07.	Inserting Data into Equipment .....	108
08.	Inserting Data into Partner .....	109
09.	Inserting Data into Sponsor.....	110
10.	Inserting Data into Event .....	111
11.	Inserting Data into Department.....	112
12.	Inserting Data into Employee .....	113
13.	Inserting Data into Feedback Program.....	115
14.	Inserting Data into Visitor.....	116
15.	Inserting Data into Reward .....	117
16.	Inserting Data into Medical Record .....	117
17.	Inserting Data into Police Record .....	118

18.	Inserting Data into Emergency Contact .....	119
19.	Inserting Data into Dependent .....	120
20.	Inserting Data into Creates.....	121
21.	Inserting Data into Is Overseen.....	122
22.	Inserting Data into Is Governed .....	123
23.	Inserting Data into Is Managed.....	124
24.	Inserting Data into Visits .....	125
25.	Inserting Data into Donates.....	126
26.	Inserting Data into Sponsors .....	127
27.	Inserting Data into Supervises .....	128
28.	Inserting Data into Employee Nationality.....	129
29.	Inserting Data into Interests .....	130
30.	Inserting Data into Comments .....	131
31.	Inserting Data into Tags.....	132
32.	Inserting Data into Cultures .....	133
33.	Inserting Data into Extension.....	134
34.	Inserting Data into Provides.....	135
35.	Inserting Data into Supplies .....	136
36.	Inserting Data into Is Hosted .....	137
<b>XIII.</b>	<b>Querying the Database.....</b>	<b>138</b>
01.	Education and Animals .....	138
02.	An Important Impression .....	139
03.	An Average Raise .....	140
04.	Bonus Promotion.....	141
05.	The Artist with Most Paintings .....	142
06.	Coordination Meeting .....	143

07.	Prison Break.....	144
08.	Grand Theft Sneak .....	145
XIV.	Normalization Up to Boyce-Codd Normal Form.....	146
01.	First Normal Form .....	146
02.	Second Normal Form.....	146
03.	Third Normal Form.....	146
04.	Boyce-Codd Normal Form.....	146
	Normalizing Each Relation:.....	147
01.	Piece.....	147
02.	Artist .....	147
03.	Section.....	148
04.	Provisioner .....	148
05.	Mediator.....	149
06.	Supplier.....	149
07.	Equipment.....	150
08.	Partner.....	150
09.	Sponsor .....	151
10.	Event .....	151
11.	Department.....	152
12.	Employee .....	152
13.	Feedback Program.....	153
14.	Visitor .....	153
15.	Reward .....	154
16.	Medical Record.....	154
17.	Police Record.....	155
18.	Emergency Contact.....	155
19.	Dependent .....	156

20.	creates .....	156
21.	is overseen.....	156
22.	is governed .....	157
23.	is managed .....	157
24.	visits .....	158
25.	donates .....	158
26.	sponsors.....	159
27.	supervision .....	159
28.	Employee Nationality .....	159
29.	Interests.....	160
30.	Comments .....	160
31.	Tags.....	160
32.	Culture.....	160
33.	Extension.....	161
34.	provides.....	161
35.	supplies .....	162
36.	is hosted .....	162
XV.	Conclusion .....	163
XVI.	References.....	164
XVII.	Instructor Feedback.....	165

*This page is intentionally left blank.*

## I. Introduction

Group #4 “Recursive Query Backtracking” is proud to present the fourth, and final, phase of the CSC375 (Database Management Systems) project for the Fall 2022 semester, entitled “A Database Design for Ageless Museum”, through which we aim to design a novel database for a newly established museum so various entities, such as employees and art pieces, are easily tracked by the Database Administrator (DBA). This database also helps naive, or parametric, users at the entrance of the museum easily track the visits of all the customers who can optionally provide feedback through the organized feedback program. Sophisticated users also benefit from this database design as it facilitates the collection and aggregation of data regarding the visitors and their interests as well as statistics that reveal which sections or departments attract the greatest amount of donations or revenue.

This database design not only aims to facilitate communication across the numerous aspects of this museum, but to securely archive these records as well. The different constraints set by this design contribute to limiting potential errors and mistakes related to data entry so the data aggregation process is more streamlined.

Sections II and III discuss the copyrights related to this project and the tools utilized throughout this phase of the project, respectively. The following section (IV) describes the system and its corresponding constraints after which a detailed illustration of the notations used is represented in the form of a legend based on the Peter-Chen notation (section V). A complete visualization of the ER Model for this database is then clearly depicted in section VI reflecting the logical design of this database. A detailed explanation of the entities, their respective attributes, and the relationships among them is, then, stated throughout sections VII and VIII respectively. Section IX of this report discusses the mapping process of the established ER diagram into relations based on a 7-step algorithm, as discussed in the 7<sup>th</sup> edition of the Fundamentals of Database Systems (Elmasri, R. & Navathe, S. B.) book, which is then reflected in section X, providing a comprehensive overview of all relations. Sections XI, XII, and XIII represent the implementation of the database on Oracle DBMS using SQL, including creating the tables, inserting data (forming a sample database state), and querying the database respectively. Section XIV then deals with normalizing, or decomposing, relations to meet the standards of all normal forms ranging from the first normal form to the Boyce-Codd normal form. Wrapping up this phase of the project is a conclusion (section XV) discussing a summary of the updated report and potentially beneficial use cases for end-users.

## II. Copyright – MIT License

Copyright (c) 2022 “Recursive Query Backtracking”

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

### **III. Tool(s) used throughout this project**

#### **I. To draw the Entity Relationship Model**

The tool used for completing the ER of this project is “Microsoft Visio Professional 2019”. This diagramming tool allows the creation of complex models and diagrams through a fairly simple and user-friendly interface.

##### **I. What is Microsoft Visio?**

Microsoft Visio is an application used to help digitally generate professional diagrams for several purposes, including business management, analysis, design, and presentation. It helps visualize complex sets of data into a more comprehensive layout, which renders it a potentially reliable choice for creating ER models.

##### **II. Why Microsoft Visio?**

Microsoft Visio offers preset ER component templates compatible with the Peter-Chen ER notation we opted for (refer to page 8 for the detailed legend). Another advantage of using Microsoft Visio are the various exporting options which have allowed us to export our Visio drawing (.vsdx) to both image (.png) and portable document (.pdf) formats. One limitation, however, is that the provided templates do not offer a preset “partial key” attribute. This is solved by creating a custom “partial key” attribute in Microsoft Visio itself, which provides enough flexibility to adapt to any notation. Another useful feature of Microsoft Visio is the ability to design elements and customize colors and effects which are more visually appealing so elements are more easily distinguished.

#### **II. To generate the fictitious logo**

Another tool which advertises as “easy-to-navigate” is the online website “Brand Crowd”. This tool allows users to generate different icons and logos.

(Refer to

<https://www.brandcrowd.com/maker/logo/f8248023-0f9a-4225-ae5a-495d6d7230a8/draft/cf47097c-d1fb-4dfc-ae2f-4dd996986b74>

to check the generated logo on the website)

## IV. System Description and Requirements

Ageless Museum is a museum set to eventually expand into the world's most comprehensive showcase of paintings, sculptures, relics, artifacts, stuffed animals, scientific exhibitions, and historical records.

A museum is a building in which objects of artistic, cultural, historical, or scientific interest are kept and shown to the public (Oxford Dictionary). It consists of several departments, such as HR and the business office, to manage the different aspects of the museum which is divided into sections, such as astronomy or historical weapons, so that visitors can plan their stay according to their interests. Each department has a name and is uniquely identified by an ID. Every department can only be operating at a single location and managed by a single employee. A department has a single email address, a single fax, but several extensions to facilitate contact. Every department has several employees working for it, and a department may receive donations from certain sponsors. A section determines where different art pieces are stored. Each section has a unique ID, a name, and a location. Every piece should have a name and be identified with a unique ID. A description of the piece, including the date at which someone started working on the piece, the date the piece was finished, and the inspiration that motivated the artist to create the art, is also to be stored. Other dates to keep track of are the date of purchase, the date the piece entered the museum, and the date a piece was sold if a piece is marked as sellable. More generally, the type of the piece, such painting or sculpture, must also be recorded along with the cultures that correlate to the corresponding piece and tags which represent important search keywords to help visitors find the piece they are looking for. Another element to track is the artist who invented the piece, so a short biography of that author, including some of his accomplishments, must be stored along with the artist's name and the corresponding dates of birth and death. Such artists are generally of only one nationality.

Art pieces are provided for the museum by a provisioner, and sometimes through a mediator. Each of the two has a name and must be identified by a unique ID. Certain contact information, such as phone numbers and emails, are to be maintained (for casual users) to easily access when needed. The country from which an art piece is being transported, along with the possible costs, are to be recorded as well as whether the acquisition of the art piece is temporary. If the acquisition is temporary, the remaining time is to be also recorded (such recorded data may help sophisticated users deal with more accurate statistics). Other suppliers supply the museum with equipment or hardware, like lighting or glass cases. Similar to the provisioner, a supplier has a name, a phone number, an email address, and is uniquely identified by an ID. As for the equipment, each piece of equipment has a name, falls under a certain category like lighting or staircases, and is uniquely identified by a corresponding serial number.

Occasionally, events are hosted by certain departments in various sections of the museum. Such events are attended by numerous partners and generous sponsors. Each event has a special name, but is identified by an ID. The exact date and time of the event are also important to keep track of. However, as the museum hosts various events, the type of the event should also be securely stored. This is to distinguish between casual school trips and VIP parties or sponsorship ceremonies. A brief description of the event should also be stored and provide ease of access. Sponsors are contributors who sponsor certain events or donate to certain departments of the museum. As for the partners, some are stakeholders who have invested their money in buying museum stocks while others are schools who frequently organize field trips to the museum, which is why each partner, uniquely identified by an ID, is to be given a class depending on the corresponding percentage share and a type which indicates whether the partner's field is education, technology, or other options. (Note that such a design would benefit sophisticated users who make use of such percentages).

As for the work force, each employee is assigned a unique ID. The employee's image (size of 4x6), first, last, and middle names, corresponding SSN, nationalities, and role are to be stored. The employee's sex can either be "male" or "female". Several dates, including the employee's date of birth, the date of commencing work, and the date of termination, are to be recorded as well so the employee's age (at least 18) and years worked are tracked. An employee also has an address composed of a country, a street, and the building in which the employee resides. As for the contact information, each employee must have one personal phone number and is provided with one departmental extension, if available, and one email address. The salary of the employee varies according to the employee's role and the corresponding number of years.

Every employee should have at least one emergency contact to be communicated with if need be. The name of the contact, the relationship with the corresponding employee, and one phone number must be stored. Important records to keep track of include the employee's medical record and police record. A typical medical record includes the employee's blood type and pressure, height, weight, BMI, and any other potentially relevant information. Certain properties of the medical record, such as the date the record is initialized and last updated, are also important to retain. A typical police record contributes to assigning each employee to an appropriate position. The date the record was issued should be tracked. The information stored for each record should mainly include if an employee has been part of any arrests, convictions, or criminal proceedings. An employee may also have people who are dependent upon the corresponding income. Information regarding such dependents, including their full name, date of birth, relationship with the employee, and sex should be stored. Only the following relationships are to be considered valid: parent, sibling, spouse, or child. If a parent, the dependent's age must be greater than or

equal to 64 years. On the other hand, the dependent must be less than 18 years old if the relationship is a sibling or a child.

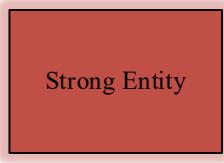
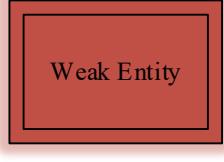
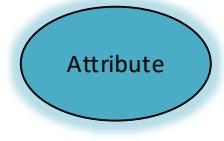
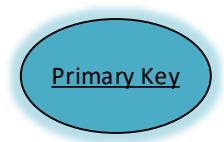
Employees can also earn different types of rewards which help motivate them to work harder, such as a vacation or a raise. Each reward must be quantified in terms of money (bonus) or time (vacation). Each recorded reward must have a name composed of a type and a number. There should also be a number tracking the number of times an employee has been granted the type of reward. Each reward should also have a date on which it is issued as well as any relevant details, including a proper justification for providing such reward.

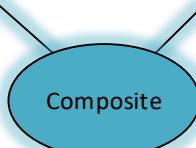
Visitors are another crucial aspect to track through the different tickets which are categorized as either kids' tickets or otherwise and are identified by their IDs. The visitors are tracked through their first, middle, and last names, as well as their age which identifies if they get a kid entry and their contact information, consisting of a phone number or an email address, which are used to send out promotional messages or important notices. Each visitor must be associated with a certain "gift point system" account which identifies how many points each visitor has so they can be exchanged for special entries. As some customers visit the museum more frequently than others, or pay more money than others, a specific class will be associated with each visitor according to how frequently they've visited the museum and how much money they've spent. The date and time each visitor enters and leaves the museum should also be recorded.

Finally, a feedback program must be established to further enhance and improve the museum. Each feedback entry will be uniquely identified by an ID and contain two main sections: the satisfaction rating on a scale from 0 to 10 and a comments section. A single entry may contain several comments. This program is strictly for the use of visitors as employees may directly communicate with their colleagues or supervisors to resolve any issues or voice any concerns.

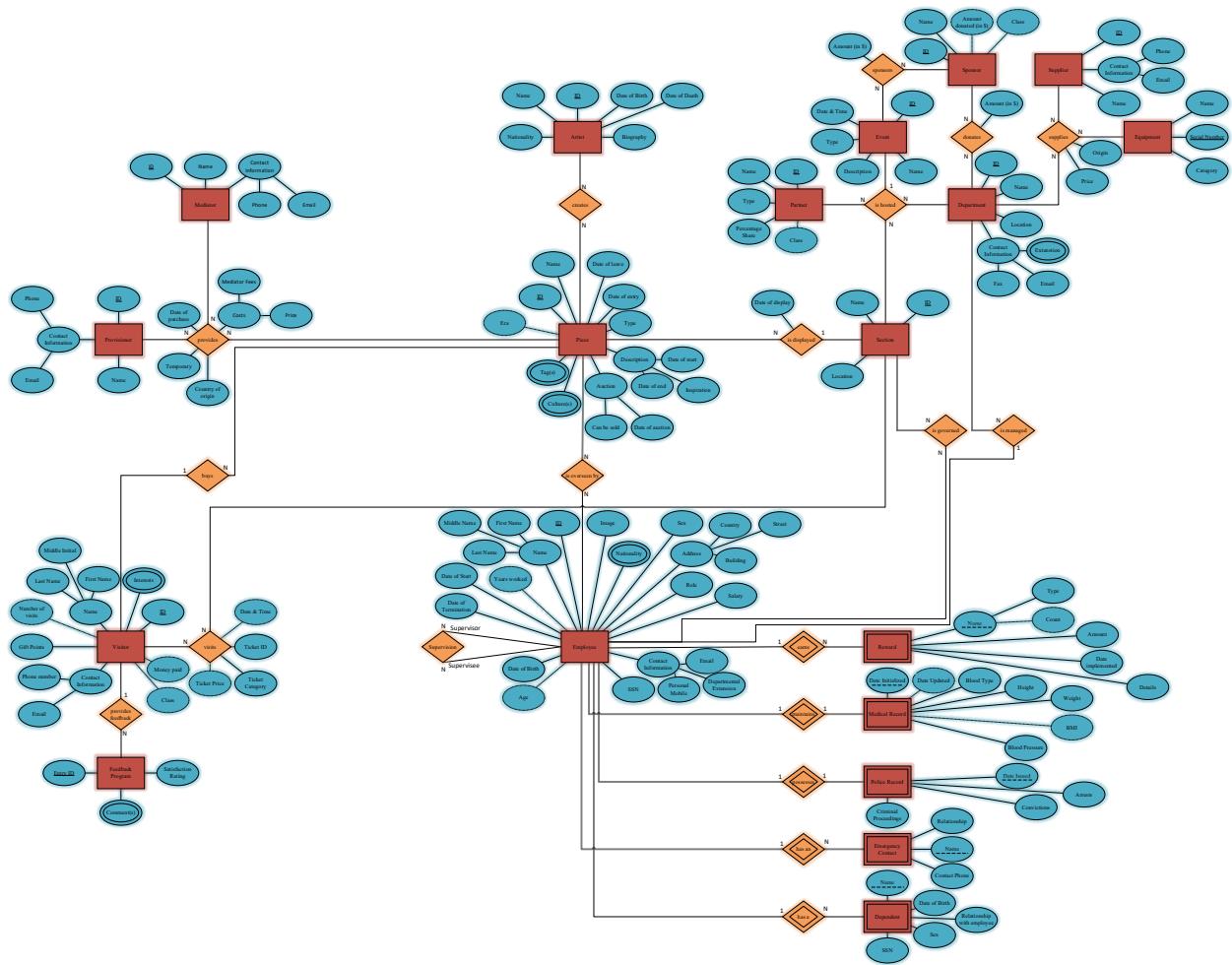
## V. Entity Relationship Model Diagram Legend

This section presents the diagrammatic notation used for ER according to the Peter-Chen notation as described in 7<sup>th</sup> edition of “Fundamentals of Database Systems” (Elmasri, R. & Navathe, S. B.):

Name	Symbol
Regular (Strong) Entity	 Strong Entity
Weak Entity	 Weak Entity
Regular (Simple) Attribute	 Attribute
Primary Key	 Primary Key
Partial Key	 Partial Key

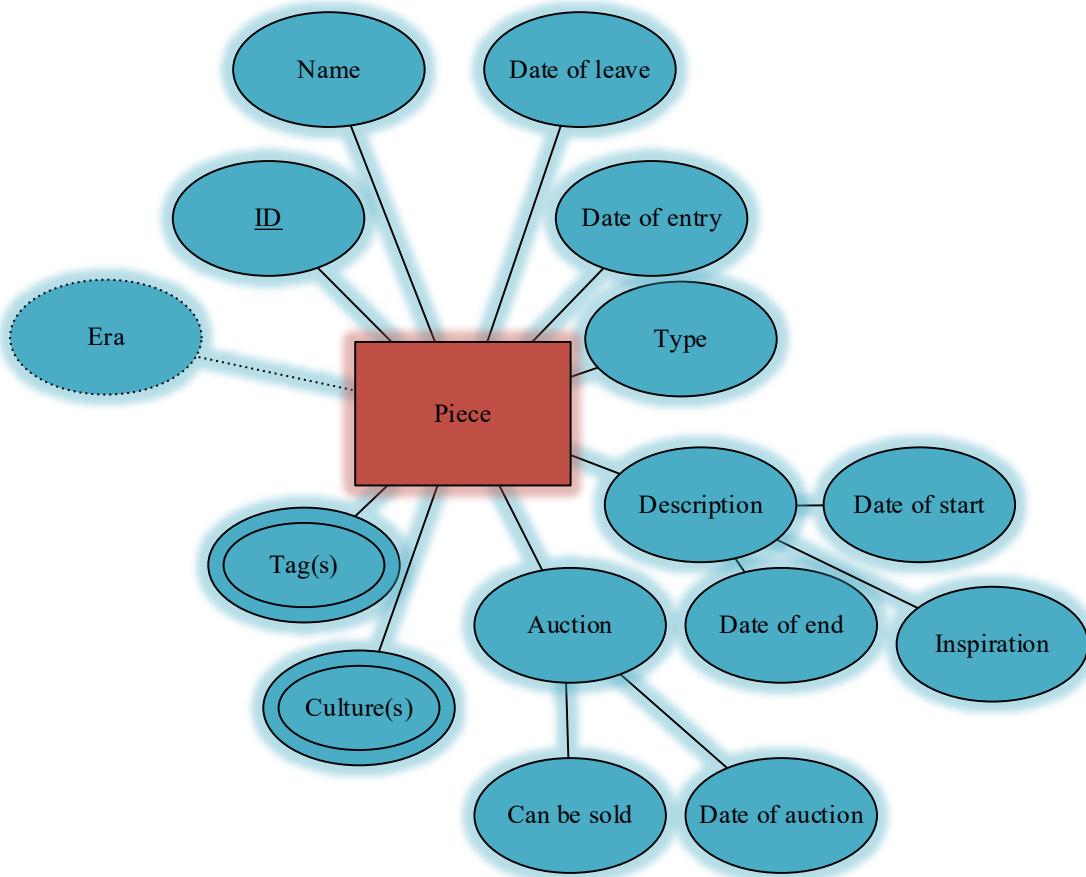
Multivalued Attribute	
Composite Attribute	
Derived Attribute	
Relationship	
Identifying Relationship	
Partial Participation	_____
Total Participation	_____

## VI. Complete ER Diagram for Ageless Museum



## VII. Entities and Their Attributes

### 01. Piece

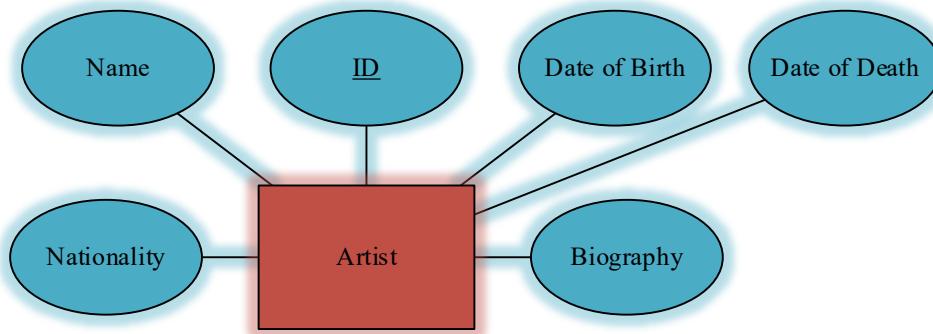


Piece is a strong entity that represents the art pieces stored in the museum.

- ID (key) uniquely identifies the piece
- Name of the artwork
- Type which indicates whether the piece is a painting, sculpture, stuffed animal, insect, or some other custom option
- Tags (multivalued) which function as search keywords to better categorize the database
- Date of entry which indicates the date on which the piece enters the museum
- Date of leave which indicates the date on which the piece leaves the museum
- Cultures (multivalued) which represent the different cultures which pertain to the art piece
- Auction (composite attribute) which consists of:
  - Can be sold (Boolean) attribute to determine whether a piece can be sold
  - Date of auction which indicates the date on which a certain piece is auctioned

- Description (composite) attribute consisting of:
  - The date a certain artist started working on the piece
  - The date a certain artist finished working on the piece
  - The inspiration which motivated the artist to work on the piece
- Era is a simple attribute which can be derived from the start and end dates

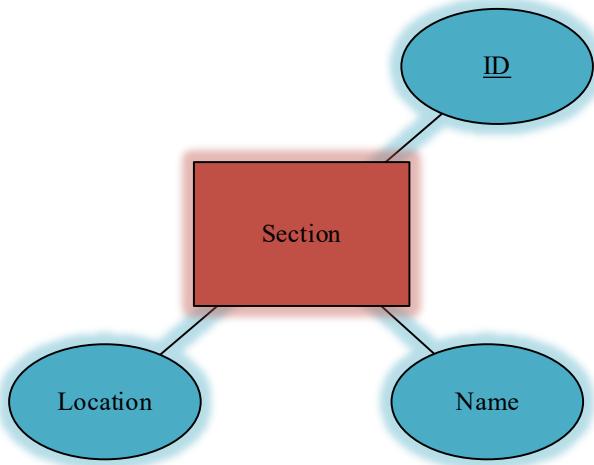
## 02. Artist



Artist is strong entity representing the creator of a certain art piece.

- ID is a key attribute and a positive integer uniquely identifying each artist
- Name is a simple attribute representing the full name of the artist
- Nationality is a simple attribute representing the nationality of the artist
- Biography is a simple attribute providing a brief biography of the artist
- Date of Birth is a simple attribute representing the date the artist was born on (cannot be a future date beyond the current one)
- Date of Death is a simple attribute representing the date on which an artist died (it may be NULL but cannot be a future date beyond the current one)

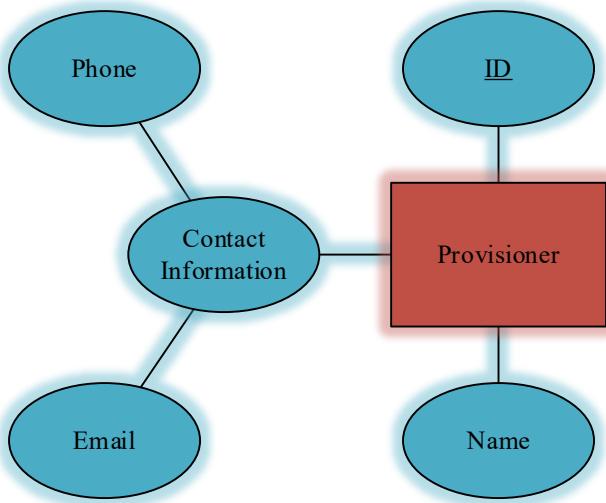
### 03. Section



Section is a strong entity which represents the various subdivisions of the museum in which the art pieces are displayed. Various sections are established according to common tags of the art pieces.

- ID is a positive integer which uniquely identifies each section (key).
- Name is a simple attribute which represents the name of each section.
- Location is another simple attribute which represents where the section is located.

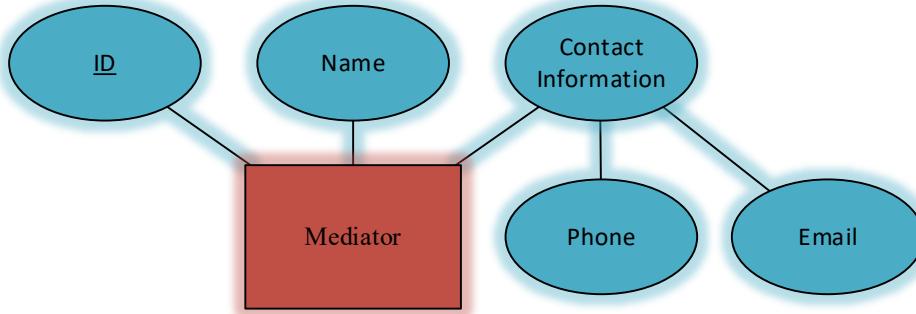
#### 04. Provisioner



Provisioner is a strong entity representing the contributors who provide the art pieces to the museum.

- ID is a positive integer which uniquely identifies each provisioner (key)
- Name is a simple attribute which represents the name of the provisioner
- Contact information is a composite attribute consisting of two simple components:
  - Phone (a single provisioner can have one phone number)
  - Email (a single provisioner can have one email address)

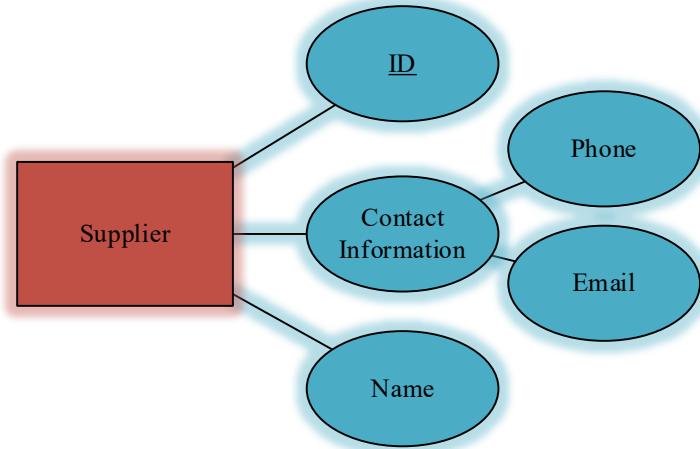
## 05. Mediator



Mediator is a strong entity which resembles a “middle-man” between the museum and the provisioner. Occasionally, transactions cannot be completed through direct communication between the museum and the provisioner and require, thus, the intervention of a mediator.

- ID is a positive integer which uniquely identifies each mediator (key).
- Name simple attribute which represents the name of the mediator.
- Contact information composite attribute which consists of two simple components:
  - Phone (each mediator has one phone number)
  - Email (each mediator has one email address)

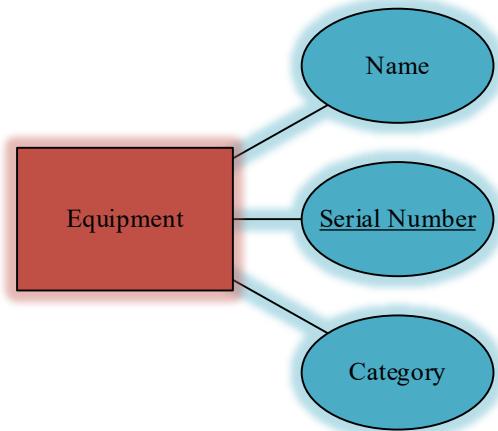
## 06. Supplier



Supplier is a strong entity type which represents contributors who supply the museum with various equipment or hardware.

- ID is a positive integer which uniquely identifies the supplier (key)
- Name which represents the name of the supplier
- Contact information which is a composite attribute consisting of two multivalued components:
  - Phone (a single supplier has one phone number)
  - Email (a single supplier has one email address)

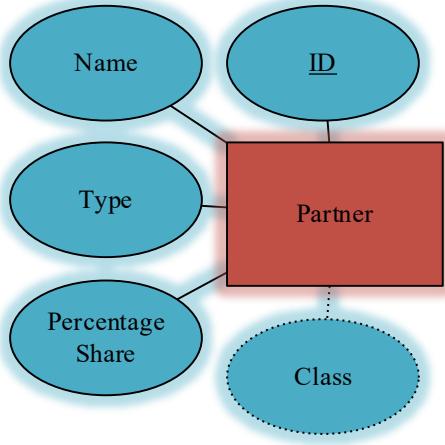
## 07. Equipment



Equipment is a strong entity which represents the various hardware across the museum.

- Serial number uniquely identifies the piece of equipment (key)
- Name is a simple attribute representing the name of equipment
- Category is a simple attribute representing the category to which the piece of equipment belongs (such as lighting or glass cases).

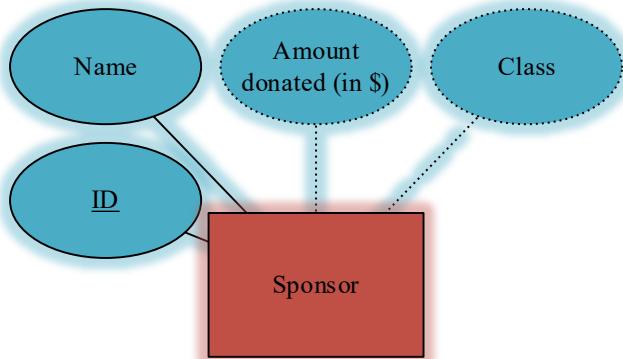
## 08. Partner



The partner entity represents stock investors who gain a share from the museum's profit in addition to occasional organized visitors (such as schools).

- ID is a positive integer which uniquely identifies each partner.
- Name is a simple attribute representing the name of the partner.
- Type is a simple attribute which indicates the partner's domain (such as technology or education)
- Percentage share is a simple attribute representing the percentage share of each partner. It could be zero.
- Class is an attribute derived from the percentage share which reflects different divisions based on the percentage share of each partner.

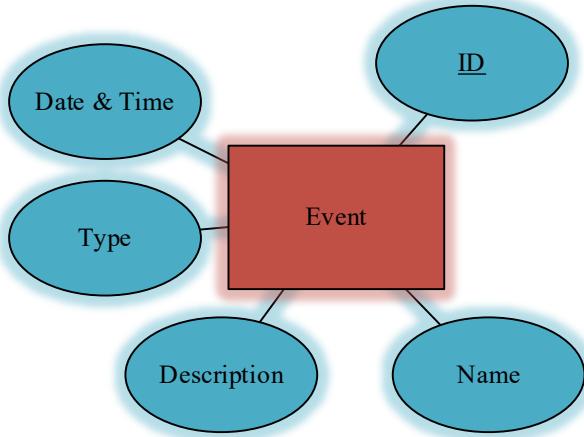
## 09. Sponsor



Sponsor is a strong entity which represents the different sponsors of the museum.

- ID is a positive integer which uniquely identifies each sponsor (key)
- Name is a simple attribute representing the name of each sponsor
- Amount donated is a derived attribute which calculates the sum of all the donations received from a sponsor
- Class is a derived attribute which classifies different partners based on the total amount donated

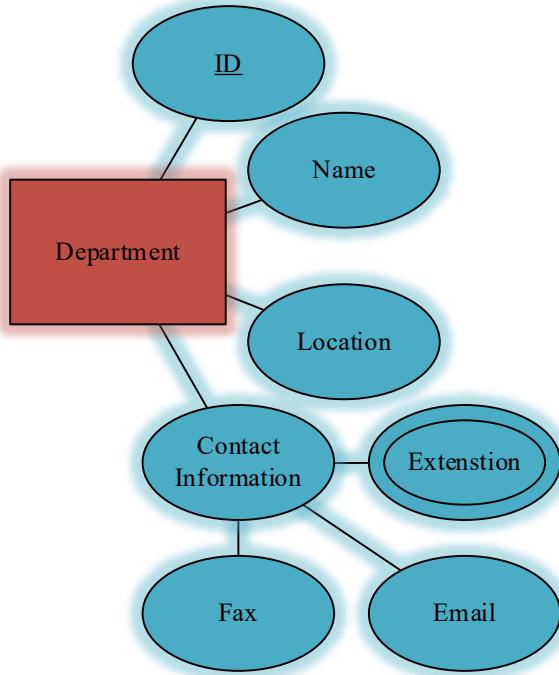
## 10. Event



Event is a strong entity that describes the various special events that the museum hosts to attract visitors.

- ID is a positive integer which uniquely identifies each event (key)
- Name is a simple attribute which represents the name of the event
- Type is a simple attribute which indicates the type (school trip, VIP party, or other custom options) of the event
- Date & Time of the event is a simple attribute which records the date on which the event is to occur and the time range during which it spans
- Description is a simple attribute which briefly illustrates some important details regarding the event

## 11. Department

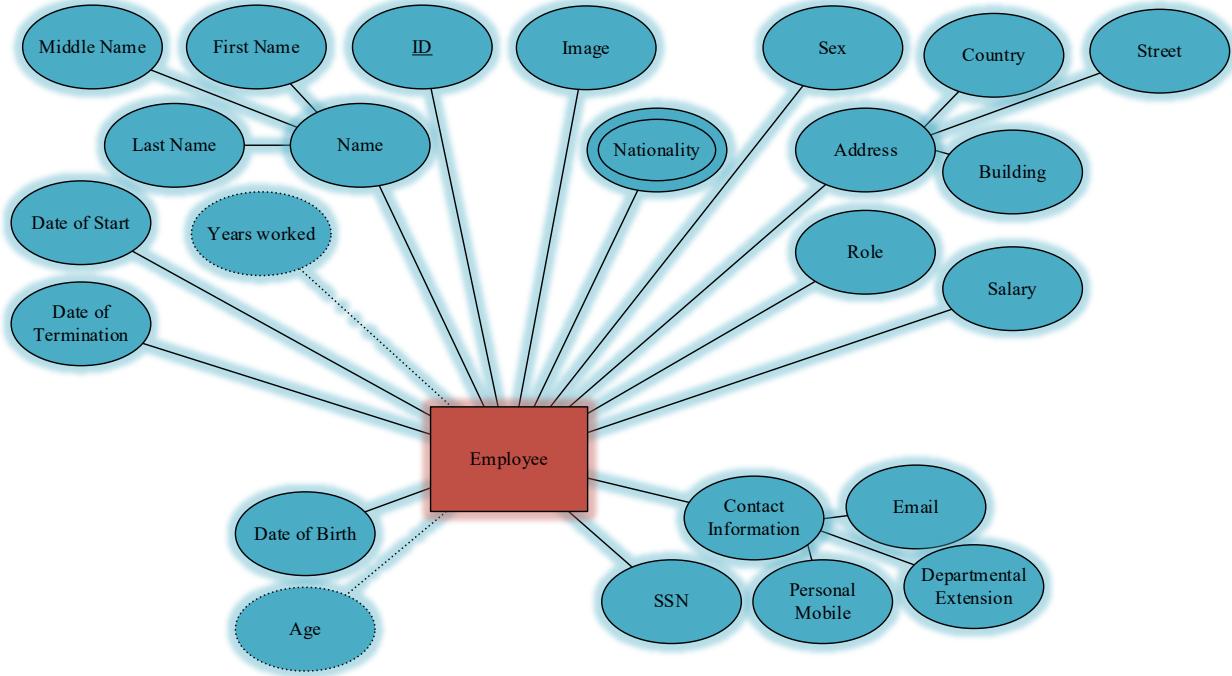


Department is a strong entity which represents the different departments which manage the various aspects of the museums.

- ID is a positive integer which uniquely identifies each department (key)
- Name is a simple attribute which represent the name of each department
- Location is a simple attribute which represents where each department is located
- Contact information is a composite attribute consisting of two simple components and a multivalued attribute:
  - Email (a department has one email address though which it can be contacted)
  - Fax (a department has one fax through which it can be contacted)
  - Extension (a department may have more than one extension through which it can be contacted)

Assume that a department must have at least one extension, at least one email address, but it may not have a fax.

## 12. Employee

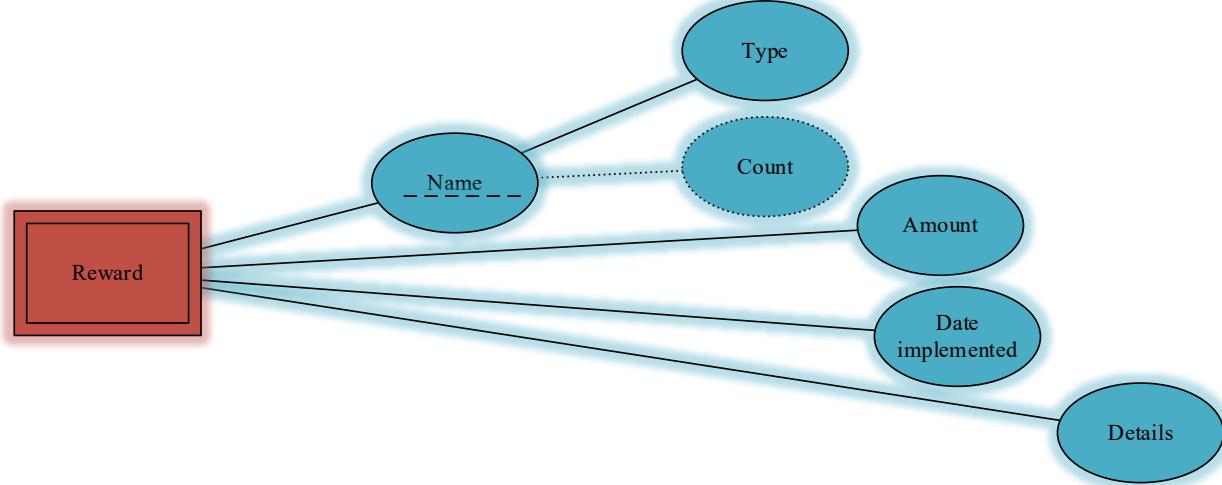


Employee is a strong entity type which represents the different employees across the museum.

- ID is a positive integer which uniquely identifies each employee (key)
- Name is a composite attribute formed of three simple components:
  - First Name of the employee
  - Middle Name of the employee
  - Last Name of the employee
- Image is a simple attribute which stores the image of the employee (the size of the image must be 4x6 according to the requirements)
- Sex is a simple attribute of values restricted to either “Male” or “Female”
- Nationality is a multivalued attribute as an employee could be of different nationalities
- Address is a composite attribute composed of three simple components:
  - Country in which the employee resides
  - Street on which the employee resides
  - Building in which the employee resides
- Role is a simple attribute which depicts the role an employee is responsible for in the museum
- SSN is a unique attribute to each employee representing the Social Security Number of each employee (not a key, but must be unique)
- Salary is a simple attribute indicating the salary of an employee. It must be an integer greater than 0.

- Contact information is a composite attribute formed of three simple components:
  - Email (a single employee has a single email address)
  - Personal Mobile (a single employee has one personal phone number)
  - Departmental extension (a single employee has a single departmental extension, if available)
- Several simple attributes in the form of dates are to be stored as well:
  - Date of birth (cannot be NULL)
  - Date of start representing the date on which an employee began working in the museum (cannot be NULL)
  - Date of termination representing the date on which an employee's contract had been terminated
- The following attributes are derived as they change frequently and can be concluded from other attributes:
  - The age is derived from the date of birth attribute (an employee must be at least 18 years old to work in the museum)
  - The years worked is derived from the Date of Start attribute

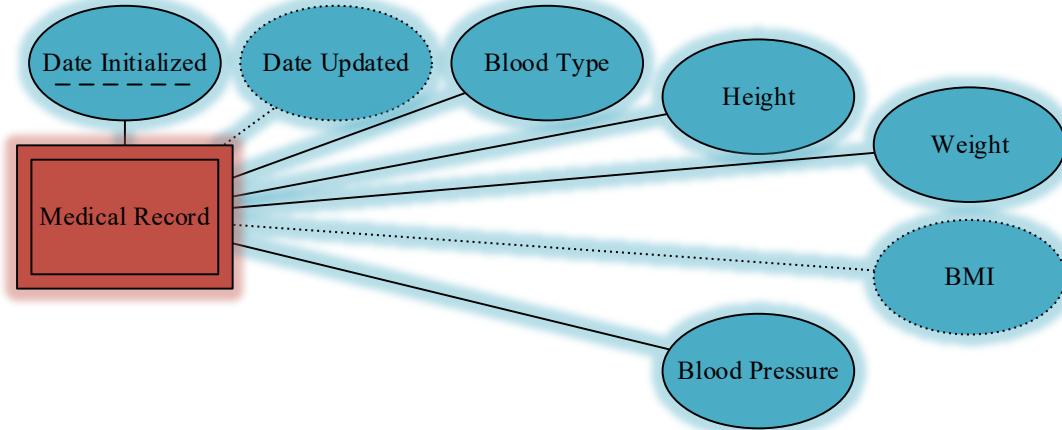
### 13. Reward



Reward is a weak entity owned by an employee who deserves a reward.

- The partial key of this entity is the composite attribute Name formed of the following:
  - Type is a simple attribute which indicates whether the reward is a bonus, vacation, or another custom option set by the DBA.
  - Count is a derived attribute indicating the number of times an employee has benefited from this type of reward.
- Date implemented is a simple attribute which records the date on which the reward is put into effect
- Details is a simple attribute which provides a brief clarification regarding why the employee is deserving of the granted reward in addition to any relevant information (cannot be NULL)
- Amount (in \$) is a simple attribute representing the amount of money the employee has been rewarded (if applicable)
- Time (in days) is a simple attribute representing the time off-work an employee is rewarded (if applicable)

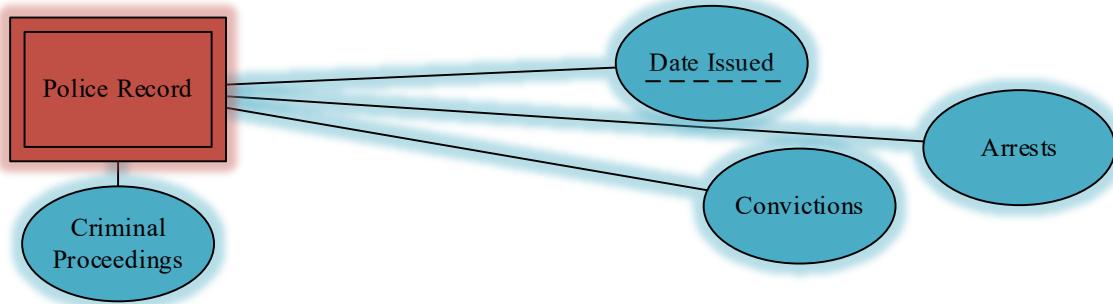
#### 14. Medical Record



Medical Record is a weak entity owned by an employee. It helps keep track of each employee's health status to ensure an appropriate work environment.

- Date initialized is a simple attribute representing the date on which a record had been created (partial key)
- Date updated is a derived attribute representing the date of the most recent update to the employee's medical record
- Blood Type is a simple attribute representing the blood type of each employee
- Height is a simple attribute representing the height of the employee
- Weight is a simple attribute representing the mass of each employee
- BMI is an attribute derived from height and weight. It is calculated using the following relation (according to the CDC):  $\frac{\text{weight (kg)}}{\text{height (m)}^2}$
- Blood Pressure is a simple attribute which indicates the typical pressure of someone's blood which allows certain follow-ups for those who suffer from low or high blood pressures.

## 15. Police Record

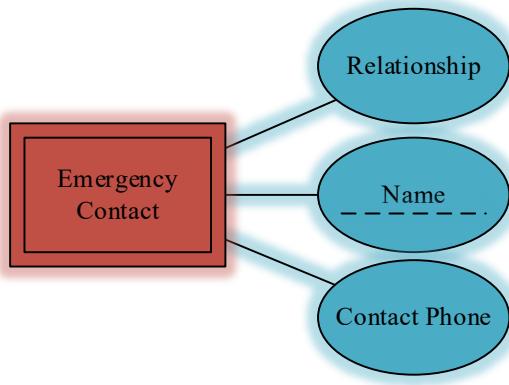


Police Record is a weak entity owned by Employee.

- Each record must have a date attribute indicating when it was issued (partial key)
- Each record consists of the following multivalued attributes which should preferably be False:
  - Arrests attribute representing a if the employee has been arrested
  - Convictions attribute representing a if the employee has been convicted of any crimes
  - Criminal Proceedings attribute representing if the employee had been involved in any criminal activity

(The default of these three attributes would be set to False)

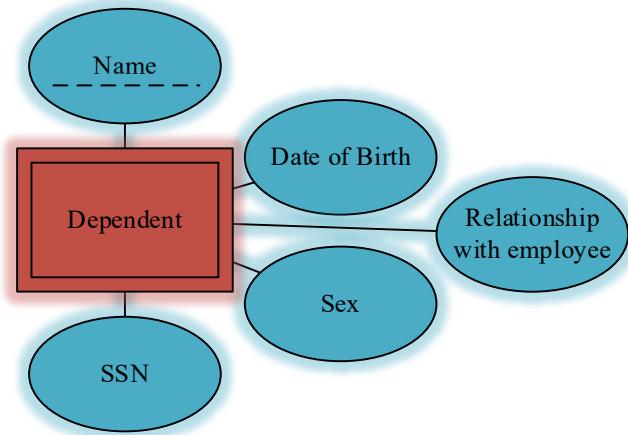
## 16. Emergency Contact



Emergency Contact is a weak entity owned by an employee. It stores information regarding the person to contact in case of an emergency related to a certain employee.

- Name which represents the name of the person to be contacted if need be (partial key)  
(Assuming that an employee does not provide several emergency contacts of the same name)
- Relationship which designates the relationship existing between the employee and the contact
- Contact Phone is a simple attribute representing the phone number via which the emergency contact can be contacted

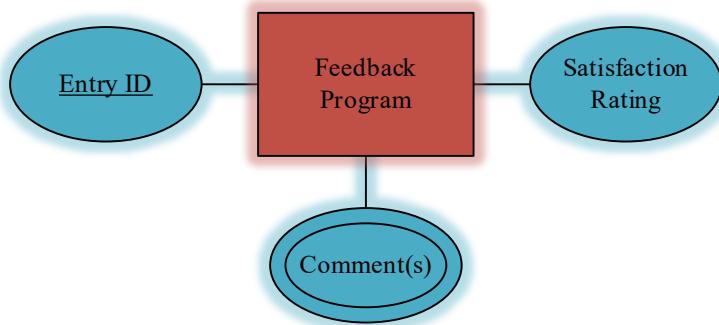
## 17. Dependent



Dependent is a weak entity type owned by an employee. It represents people who are financially dependent upon the employee.

- Name is a simple attribute representing the name of the dependent (partial key)  
(Assuming that it is unlikely an employee has several dependents of the same name)
- Relationship with employee is a simple attribute which represents the dependent's relationship with the employee (restricted to: parent, sibling, child, or spouse)
- SSN is a unique simple attribute (which can be NULL in the case of a dependent) representing the social security number of the dependent
- Sex is a simple attribute indicating whether the sex of the dependent is "Male" or "Female"

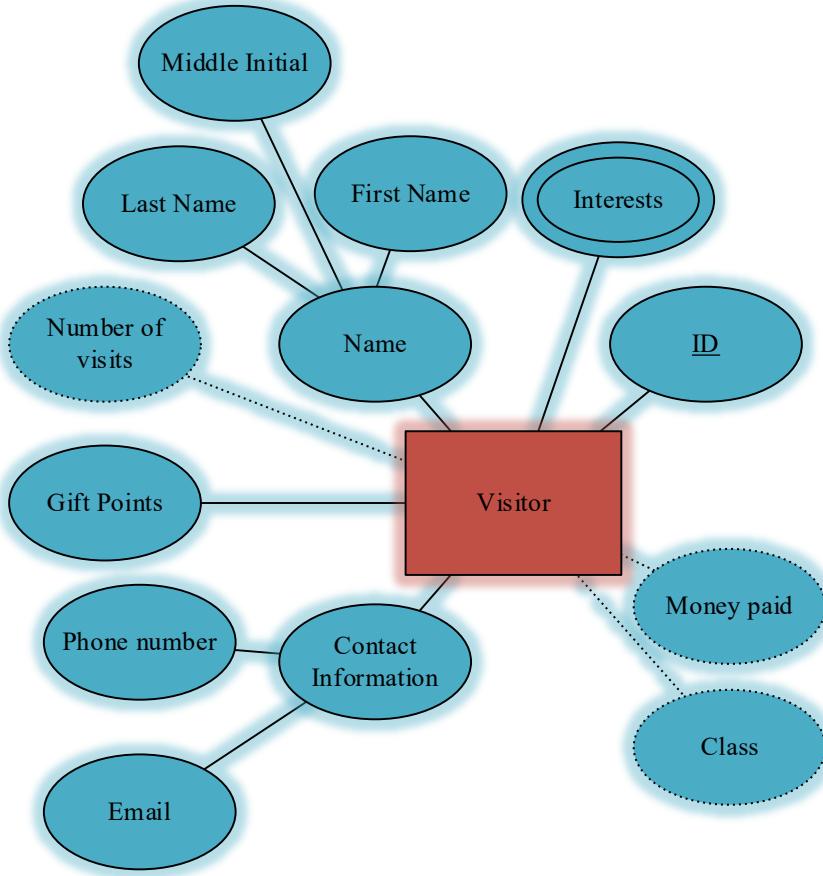
## 18. Feedback Program



Feedback Program is a strong entity type representing the newly founded feedback program which is utilized by the museum to gather data regarding the satisfaction of visitors' experiences as well as that of the employees. Such responses can be tracked through the source ID.

- Each feedback is uniquely identified by the Entry ID key attribute
- Satisfaction Rating is a simple attribute the value of which can only be within the range of integers 0 through 10
- Comments is a multivalued attribute representing any clarifications or notices a visitor or employee may like to add

## 19. Visitor



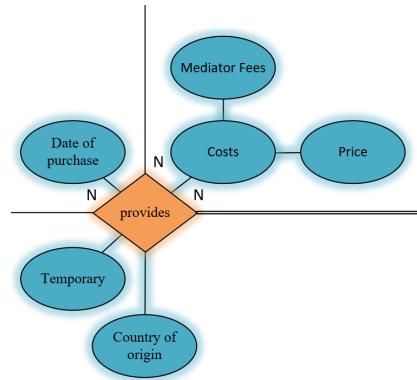
Visitor is a strong entity type representing people who visit the museum.

- ID is a key simple attribute and a positive integer representing the ID of each visitor.
- Name is a composite attribute formed of the three simple attributes:
  - First Name representing the first name of the visitor
  - Middle Initial representing the middle initial of the visitor
  - Last Name representing the last name of the visitor
- Contact Information is another composite attribute formed of the following two simple attributes:
  - Phone number (assuming each visitor can have at most only 1 phone number)
  - Email (assuming each visitor can have at most only 1 email)
- Gift Points is a simple attribute and a non-negative integer representing how much points a certain visitor has in their account
- Interests is a multivalued attribute representing the various interests of each visitor

- The following attributes are derived as they change frequently according to other factors:
  - Number of visits can be updated each time a visitor orders a ticket
  - Money paid can be calculated by adding each purchase a visitor makes
  - Class can be deduced based on the number of visits and amount of money spent per visitor

## VIII. Relationships and Their Explanations

### 01. provides



A provisioner provides a piece through a mediator to the museum.

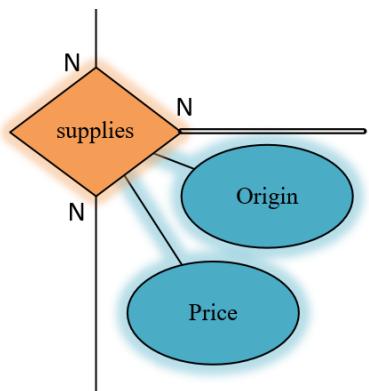
Many provisioners may provide many pieces through many mediators, which is why the cardinality is N to N to N.

As for the total participation, every piece must be provided by a certain provisioner, but not necessarily through a mediator or such that all provisioners participate in the relationship.

There exist several attributes on the relationship:

- Date of purchase indicating the date on which the piece is purchased from the provisioner
- Country of origin indicating where the piece shipment is coming from
- Temporary indicating whether the museum's acquisition of the piece is temporary
  - If null, then the acquisition is permanent
  - Otherwise, the date on which the piece must be returned to its respective provisioner must be entered
- Costs is a composite attribute formed of:
  - Price indicating the price of the piece
  - Mediator fees indicating the fees the mediator demands

## 02. supplies



A supplier supplies equipment to a department.

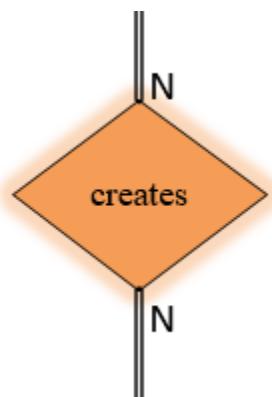
Many suppliers can supply many pieces of equipment to many departments, which is why the cardinality is N to N to N.

As for the total participation, every piece of equipment must be supplied by a supplier to a department.

Two attributes are present on this relationship:

- Origin indicating the country of origin of the hardware
- Price indicating the price of the equipment

03. creates

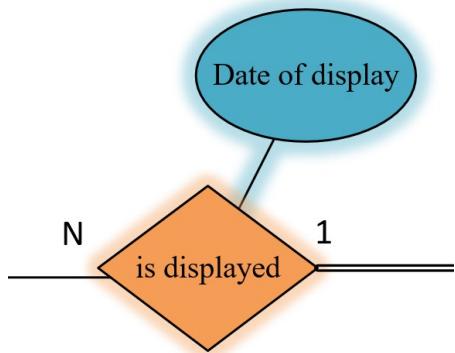


An artist creates a piece.

The cardinality is many to many because one artist can create one or more pieces and the pieces in the museum may be created by more than one artist.

The participation is total on both ends because each artist should have at least one piece and each piece is created by an artist.

04. is displayed



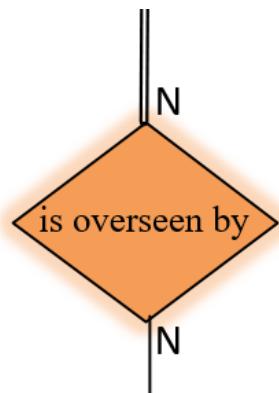
A piece is displayed in a section.

The cardinality is many to one because a section can have many pieces, but a piece may not be in more than one section.

The participation is partial on the piece's end because not all pieces may be displayed, but the participation is total for the section because each section must have one or more pieces displayed.

This relation has a date of display attribute depicting the date and time during which the piece was displayed in the section.

05. is overseen



A piece is overseen by an employee.

The cardinality is many to many because a piece may be overseen by one or more employee and an employee may oversee more than one piece.

The participation is total on the piece's end because every piece should be overseen by an employee, but the participation is partial on the employee's end because not all employees oversee pieces.

06. earns



"earns" illustrates the relationship between an employee and a reward. An employee earns a reward.

An employee participates in this relationship when he/she deserves to get a reward. Thus, the employee's participation in this relationship is partial because not every employee is required to participate in it. Moreover, the Reward participation in this relationship is total because all the rewards should be available for any employee who wants to work hard and grab the reward. Thus, we chose our cardinality as 1 to N since an employee can get every possible reward if he/she is willing to do that.

07. maintains



“maintains” illustrates the relationship between an employee and his/her medical record. An employee maintains a medical record.

Every employee totally participates in this relationship since every employee should have his/her medical record. Similarly, the medical record participates totally in this relationship, since a record must be maintained for every employee.

As a result, we chose 1 to 1 cardinality since every employee should have one medical record stored in our database.

08. possesses

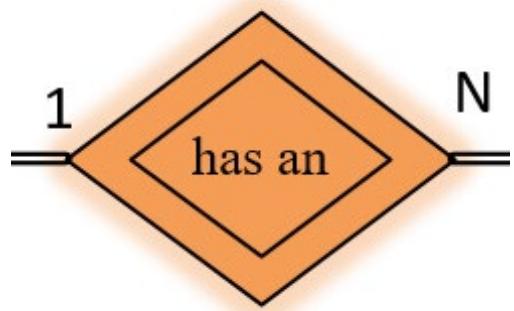


“possesses” illustrates the relationship between an employee and his/her Police record. An employee possesses a police record.

Every employee totally participates in this relationship since every employee should have his/her Police record available at the museum. Similarly, the Police record participates totally in this relationship, since a police record must be maintained for every employee at the museum.

As a result, we chose 1 to 1 cardinality since every employee should have one police record stored in our database.

09. has an

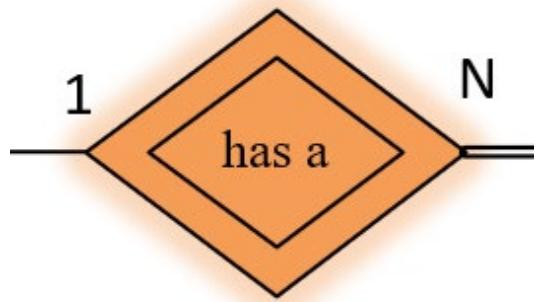


“has an” illustrates the relationship between an employee and his/her emergency contacts. An employee has an emergency contact.

Every employee totally participates in this relationship since every employee should provide the museum with a list of his emergency contact. Similarly, the Emergency contact participates totally in this relationship, since at least one emergency contact must be maintained for every employee at the museum. Also, the total participation of an emergency contact is because every emergency contact must be relevant for this employee.

As a result, we chose 1 to N cardinality since every employee should have at least 1 emergency contact such that all these contacts are relevant to the employee.

10. has a

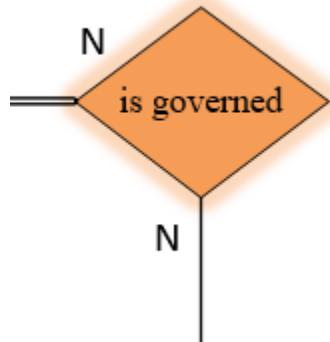


“has a” illustrates the relationship between an employee and his/her Dependents. An employee has a dependent.

Every employee partially participates in this relationship since an employee might have no dependents. However, the dependents participate totally in this relationship since every dependent should have an employee in order to exist.

Finally, we chose 1 to N cardinality since an employee can have up to N dependents.

## 11. is governed

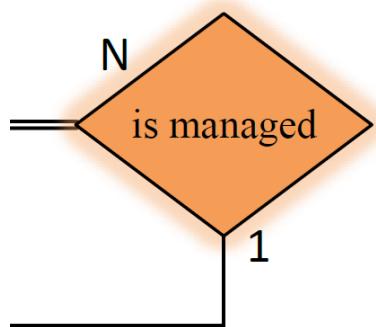


A section is governed by an employee.

Many employees can govern many sections, so the cardinality is N to N.

As for the total participation, each section must be governed by an employee, but an employee may not govern any sections, so the total participation is at the section's side.

## 12. is managed

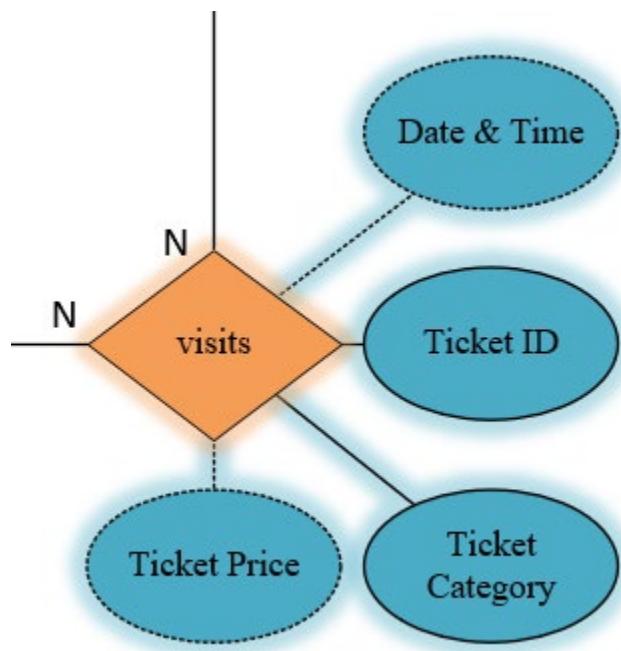


A department is managed by an employee,

A single employee may manage multiple departments, but a department is managed by a single employee. Thus, the cardinality is 1 to the employee's side but N to the department's side.

As for the participation, a department must be managed by an employee, but an employee may not manage any departments. Thus, the participation is total on the department's end, but partial on the employee's end.

### 13. visits



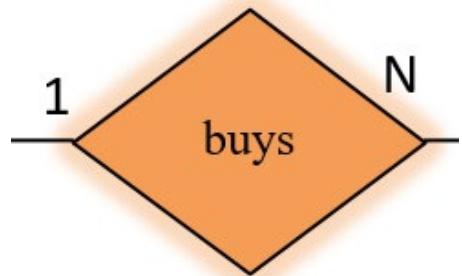
A visitor visits a section of the museum.

The cardinality is many to many because there may be a lot of visitors in the different sections of the museum.

However, the participation is not total because it is not necessary for every visitor to visit every section or for every section to be visited by the visitors.

Attributes on this relation include the ticket ID, category, price, and the date and time it was purchased.

14. buys

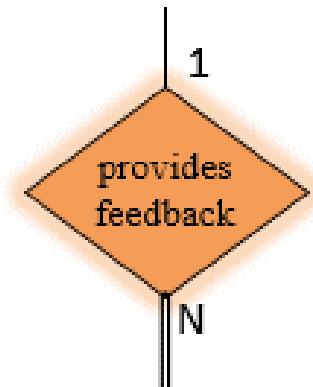


A visitor buys a piece.

The cardinality is one to many because one person can buy more than one piece and each bought piece is owned by one visitor.

The participation is not total on both sides because not all the pieces in the museum are sold and not all visitors buy pieces.

15. provides feedback

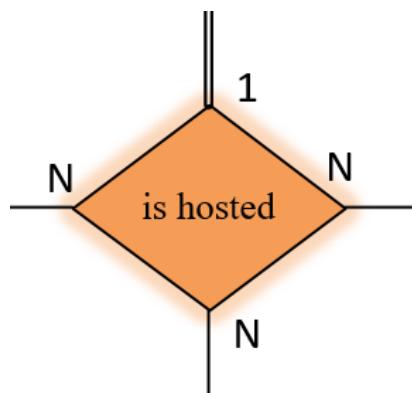


A visitor provides feedback to the feedback program.

The cardinality is one to many because each visitor can provide one or more feedbacks and every feedback is related to one visitor.

The participation is partial on the visitor's end because not all visitors may provide feedbacks. However, the participation is total for the feedback program because every feedback is given by a visitor.

16. is hosted

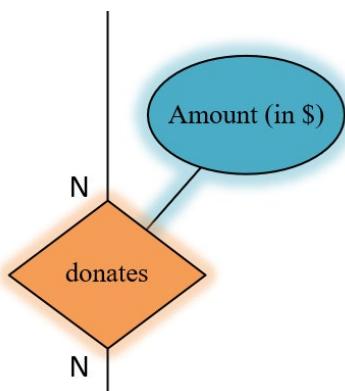


An event is hosted with a partner in a section by a department.

A single event can be hosted with many partners in many sections under the management of many departments.

Every event must be hosted. However, not all partners, or all departments, or all sections, must participate in this relationship.

17. donates



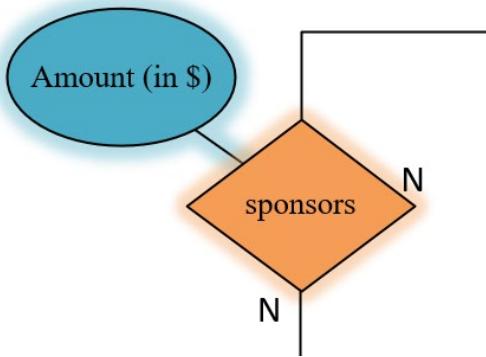
A sponsor donates a certain amount of money to a department.

Many sponsors can donate to many departments, so the cardinality is N to N.

However, not all sponsors must donate to all departments, which is why the participation is partial on both ends.

The Amount attribute indicates how much a sponsor has donated to a department.

## 18. sponsors



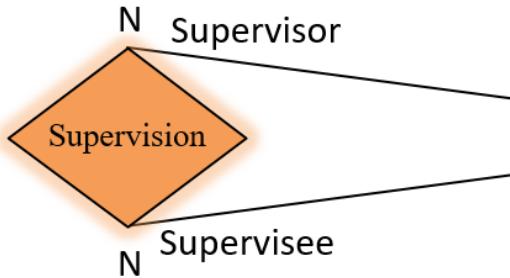
A sponsor sponsors an event. The sponsor provides a certain amount of money.

Many sponsors may sponsor many events, so the cardinality is N to N.

However, not all sponsors must sponsor all events, which is why the participation is partial on both ends.

The Amount attribute indicates how much a sponsor has contributed to an event.

## 19. supervision



An employee supervises an employee, so this relationship is recursive.

A supervisor may supervise many supervisees, and a single supervisee may be supervised by many supervisors, so the cardinality is N to N.

However, not all employees are supervised (such as the CEO or the founder of the museum), and not all employees are supervisors. Thus, the participation is partial on both ends.

## IX. ER to Relational Mapping

After defining a conceptual design of the Ageless Museum Database, a seven-step algorithm was followed to translate the conceptual ER design into a logical design represented through relations. This algorithm is described in the seventh edition of the Fundamentals of Database Systems (Elmasri, R. & Navathe, S. B.) book.

### 01. Mapping Strong Entity Types

Every strong entity type is mapped into a relation. A primary key is chosen from among the candidate keys of this strong entity type. This primary key will identify this relation. The simple attributes of the entity are mapped as an attribute into the relation. The composite attributes are mapped as atomic attributes in the relation.

#### 01. Piece

ID	Name	Date of leave	Date of entry	Type	Date of start	Date of end	Inspiration	Can be Sold	Date of auction
----	------	---------------	---------------	------	---------------	-------------	-------------	-------------	-----------------

Piece is a strong entity type; hence it will be mapped into a separate relation. It has the simple attribute ID as the only candidate key. Thus, ID will be the primary key for the Piece relation. Moreover, the relation includes all the simple attributes of the entity type Piece (Name, Date of leave, Date of entry, Type, Era). The composite attribute Description of the entity type is included as three simple attributes in the relation (Date of Start, Date of end, and Inspiration). Similarly, the composite attribute Auction is included as two simple attributes (Can be sold, Date of Auction). The derived Era attribute can be computed through SQL and is, thus, not mapped.

#### 02. Artist

ID	Name	Date of Death	Date of Birth	Biography	Nationality
----	------	---------------	---------------	-----------	-------------

Artist is a strong entity type; hence it will be mapped into a separate relation. It has the simple attribute ID as a candidate key. Thus, ID will be the primary key for the Artist relation. Moreover, the relation includes all the simple attributes of the entity type Artist (Name, Date of Birth, Date of Death, Biography, and Nationality).

### 03. Section

<u>ID</u>	Name	Location
-----------	------	----------

Section is a strong entity type; hence it will be mapped into a separate relation. It has the simple attribute ID as the only candidate key. Thus, ID will be the primary key for the Section relation. Moreover, the relation includes all the simple attributes of the entity type Section (Name, Location).

### 04. Provisioner

<u>ID</u>	Name	Phone	Email
-----------	------	-------	-------

This is a strong entity and thus the mapped relation includes all its simple attributes and ID (which is underlined as its primary key). Note that the simple attributes Phone & Email form the composite attribute “Contact Information”.

### 05. Mediator

<u>ID</u>	Name	Phone	Email
-----------	------	-------	-------

This is a strong entity and thus the relation includes all simple attributes of the mediator, and ID (which is underlined as its primary key). Note that the simple attributes Phone & Email form the composite attribute “Contact Information”.

### 06. Supplier

<u>ID</u>	Phone	Email	Name
-----------	-------	-------	------

Supplier is a strong entity type; hence it will be mapped into a separate relation. It has the simple attribute ID as the only candidate key. Thus, ID will be the primary key for the Supplier relation. Moreover, the relation includes the simple attribute- Name- of the entity type Supplier. The composite attribute Contact information of the entity type is included as two simple attributes in the relation (Phone and Email).

## 07. Equipment

<u>Serial Number</u>	Name	Category
----------------------	------	----------

Equipment is a strong entity type; hence it will be mapped into a separate relation. It has the simple attribute Serial Number as the only candidate key. Thus, the Serial Number will be the primary key for the Equipment relation. Moreover, the relation includes all the simple attributes of the entity type Equipment (Name, Category).

## 08. Partner

<u>ID</u>	Name	Type	Percentage Share
-----------	------	------	------------------

Partner is a strong entity type; hence it will be mapped into a separate relation. It has the simple attribute ID as the only candidate key. Thus, ID will be the primary key of the relation Partner. Moreover, the relation includes all the simple attributes of the entity type Partner (Name, Type, Percentage share). Class is a derived attribute and, thus, not mapped.

## 09. Sponsor

<u>ID</u>	Name
-----------	------

Sponsor is a strong entity type; hence it will be mapped into a separate relation. It has the simple attribute ID as the only candidate key. Thus, ID will be the primary key of the Sponsor relation. Moreover, the relation includes the simple attribute of this entity type (Name). The attributes Amount Donated and Class are derived and, thus, not mapped.

## 10. Event

<u>ID</u>	Name	Type	Description	Date & Time
-----------	------	------	-------------	-------------

Event is a strong entity type; hence it will be mapped into a separate relation. It has the simple attribute ID as the only candidate key. Thus, ID will be the primary key of the Event relation. Moreover, the relation includes all the simple attributes of the entity type Event (Name, Type, Description, Date &Time).

## 11. Department

ID	Name	Location	Fax	Email
----	------	----------	-----	-------

Department is a strong entity type; hence it will be mapped into a separate relation. It has the simple attribute ID as the only candidate key. Thus, ID will be the primary key for the Department relation. Moreover, the relation includes all the simple attributes of the entity type Department (Name, Location). The composite attribute Contact information of the entity type is included as two simple attributes in the relation (Fax and Email).

## 12. Employee

ID	Image	First_Name	Middle_Name	Last_Name	Sex	SSN
Date_of_Birth	Date_of_Start		Date_of_Termination			Role
Country	Street	Building	Email		Departmental_Extension	

This strong entity has a primary key of “ID” and its relation consequently contains all the simple attributes of this entity. Composite attributes are broken down into simple attributes and multivalued attributes are mapped into a separate relation.

## 13. Feedback Program

EntryID	Satisfaction Rating
---------	---------------------

This a strong entity, so include all its simple attributes and underlined the primary key which is the EntryID.

## 14. Visitor

ID	FirstName	MiddleInitial	LastName	Gift_Points	Phone_Number	Email
----	-----------	---------------	----------	-------------	--------------	-------

This is a strong entity uniquely identified by its primary key (ID), so include all of its corresponding simple attributes (FirstName, MiddleInitial, LastName, Gift\_Points, Phone\_Number, and Email).

## 02. Mapping of Weak Entity Types

Weak entities are existence-dependent on the owner strong entities. Hence, for each a weak entity relation, a partial key is defined to uniquely identify the entity when combined with the primary key of the corresponding owner entity (which is included as a foreign key as a consequence of mapping the corresponding relationships).

### 01. Reward

<u>EmployeeID</u> <u>(FK)</u>	Type	Count	Amount	Date implemented	Details
----------------------------------	------	-------	--------	------------------	---------

This is a weak entity, so include all its simple attributes. Its partial key is the combination of (Type, Count) which form the composite attribute Name. Its primary key is the combination of the owner primary key (EmployeeID) as a foreign key with its partial key. Note that count is a derived attribute as this facilitates the data entry process. The unit of amount (either \$ or days) can be derived through SQL depending on the Type of the reward.

### 02. Medical Record

<u>EmployeeID</u> <u>(FK)</u>	<u>Date</u> <u>Initialized</u>	Blood Type	Height	Weight	Blood pressure
----------------------------------	-----------------------------------	------------	--------	--------	----------------

Medical record is a weak entity, so include all its simple attributes. Its partial key is the Date Initialized attribute. Its primary key is the combination of its partial key with the owner's primary key (EmployeeID) as a foreign key.

### 03. Police Record

<u>EmployeeID (FK)</u>	<u>Date Issued</u>	Arrests	Convictions	Criminal Proceedings
------------------------	--------------------	---------	-------------	----------------------

Police record is a weak entity, so include all its simple attributes. Its partial key is the Date Issued attribute. Its primary key is the combination of its partial key with the owner primary key (EmployeeID) as a foreign key

#### 04. Emergency Contact

<u>EmployeeID</u> <u>(FK)</u>	<u>Name</u>	Relationship	Contact Phone
----------------------------------	-------------	--------------	---------------

Emergency contact is a weak entity, so include all its simple attributes. Its partial key is the contact's name and its primary key is the combination of its partial key with the owner primary key (EmployeeID) as a foreign key.

#### 05. Dependent

<u>EmployeeID</u> <u>(FK)</u>	<u>Name</u>	Relationship with Employee	Date of Birth	Sex	SSN
----------------------------------	-------------	-------------------------------	---------------	-----	-----

Dependent is a weak entity, so include all its simple attributes. Its partial key is the dependent's name and its primary key is the combination of its partial key with the owner primary key (EmployeeID) as a foreign key.

### 03. Mapping of Binary 1:1 Relationship Types

Following the algorithm described in the Fundamentals of Database Systems book, the foreign key approach is adopted to choose a primary from one participating entity and including it as an attribute in the relation of the other.

The following two relationships are the only exhibited 1:1 relationships. They do not require any mapping as their corresponding relations were established in the previous step while mapping weak entities.

01. maintains

02. possesses

## 04. Mapping of Binary 1:N Relationship Types

Following the general rule, the primary key of the relation of the participating entity type on the 1-side is included as a foreign key in the relation of the entity type on the N-side. Moreover, any attributes of the relationship are included in the relation of the participating entity type on the N-side.

### 01. is displayed

<u>ID</u>	Name	Date of leave	Date of entry	Type	Date of start	Date of end	Inspiration	Can be Sold	Date of auction	Section_ID (FK)	Date of Display
-----------	------	---------------	---------------	------	---------------	-------------	-------------	-------------	-----------------	-----------------	-----------------

“is displayed” is an 1:N relationship between a Piece and a Section. Thus, the primary key of the Section relation is included as a foreign key (Section\_ID) in the Piece relation. Moreover, the simple attribute of the relationship (Date of Display) is included in the relation of the entity type Piece.

### 02. earns

Type	Count	<u>EmployeeID (FK)</u>	Amount	Date implemented	Details
------	-------	------------------------	--------	------------------	---------

This 1:N relationship updates the Reward relation so now include the primary key of the owner relation as foreign key in this relation. The combination of the partial keys and the foreign key (Employee ID) constitute the primary key of this relation.

### 03. has an

This relationship requires no mapping as its corresponding relation was established while mapping weak entities (in step 2 of the discussed algorithm).

### 04. has a

This relationship requires no mapping as its corresponding relation was established while mapping weak entities (in step 2 of the discussed algorithm).

## 05. buys

ID	Name	Date of leave	Date of entry	Type	Date of start	Date of end	Inspirati on	Can be Sold	Date of auction	Section_ID (FK)	Date of Display	V_ID
----	------	---------------	---------------	------	---------------	-------------	--------------	-------------	-----------------	-----------------	-----------------	------

This 1:N relationship updates the Piece relation so that it now includes a foreign key which is the primary key of the visitor who bought the piece.

## 06. provides feedback

<u>EntryID</u>	Satisfaction Rating	Visitor_ID (FK)
----------------	---------------------	-----------------

This 1:N relationship updates the Feedback Program relation so that it now includes the foreign key of the visitor who had provided the feedback.

## 05. Mapping of Binary M:N Relationship Types

For each M: N binary relationship, we will create a new relation. This primary key of each relation of the participating entity type is included as a foreign key in the new relation. The combination of these foreign keys will make up the primary key of the new relation. Moreover, all the attributes of the relationship. Also, this relation will include any simple attribute (or any simple components of composite attributes) of the relationship.

## 01. creates

<u>Artist_ID (FK)</u>	<u>Piece_ID (FK)</u>
-----------------------	----------------------

“creates” is an N: N relationship between Artist and Piece. The relation “creates” contains Artist\_Name as a foreign key which references the primary key of the Artist relation. Similarly, Piece\_ID is a foreign key referencing the Piece relation. The combination of the 2 foreign keys makes up the primary key of the “creates” relation.

## 02. is overseen

<u>ID_Piece (FK)</u>	<u>Emp_ID (FK)</u>
----------------------	--------------------

“is overseen by” is an N: N relationship between Piece and employee. The relation “Is overseen by” contains ID\_Piece as a foreign key that references the primary key of the Piece relation.

Similarly, Emp\_ID is a foreign key referencing the Employee relation. The combination of the 2 foreign keys makes up the primary key of the “Is overseen by” relation.

### 03. is governed

<u>Section_ID (FK)</u>	<u>Employee_ID (FK)</u>
------------------------	-------------------------

“is governed” is an N: N relationship between the Section entity type and the Employee entity type. The relation “is governed” contains Section\_ID as a foreign key that references the primary key of the Section relation. Similarly, Employee\_ID is a foreign key referencing the Employee relation. The combination of the 2 foreign keys makes up the primary key of the “is governed” relation.

### 04. is managed

<u>Dep_ID (FK)</u>	<u>Emp_ID (FK)</u>
--------------------	--------------------

“is managed” is an N: N relationship between the Section entity type and the Employee entity type. The relation “is managed” contains Dep\_ID as a foreign key that references the primary key of the Department relation. Similarly, Emp\_ID is a foreign key referencing the Employee relation. The combination of the 2 foreign keys makes up the primary key of the “is managed” relation.

### 05. visits

<u>Visitor_ID (FK)</u>	<u>Section_ID (FK)</u>	Ticket ID	Ticket Category
------------------------	------------------------	-----------	-----------------

This is a binary many-to-many relationship. Thus, create a new relation and include as foreign keys the primary keys of the visitor and the section entities. The combination of these foreign keys is the primary key for this relation.

### 06. donates

<u>Sponsor_ID (FK)</u>	<u>Department_ID</u>	Amount in (\$)
------------------------	----------------------	----------------

“donates” is an N: N relationship between the Sponsor entity type and the Department entity type. The relation “donates” contains Department\_ID as a foreign key that references the primary key of the Department relation. Similarly, Sponsor\_ID is a foreign key referencing the Sponsor relation. The combination of the 2 foreign keys makes up the primary key of the “donates”

relation. Moreover, the relation includes the simple attribute of the donates relationship (Amount in \$).

#### 07. sponsors

<u>Event_ID (FK)</u>	<u>Sponsor_ID (FK)</u>	Amount in (\$)
----------------------	------------------------	----------------

“sponsors” is an N: N relationship between the Sponsor entity type and the Event entity type. The relation “sponsors” contains Event\_ID as a foreign key that references the primary key of the Event relation. Similarly, Sponsor\_ID is a foreign key referencing the Sponsor relation. The combination of the 2 foreign keys makes up the primary key of the “sponsors” relation. Moreover, the relation includes the simple attribute of the relationship (Amount in \$).

#### 08. supervision

<u>Supervisor_ID</u>	<u>Supervisee_ID</u>
----------------------	----------------------

This N:M recursive relation is mapped into a single relation the primary key of which (Supervisor\_ID, Supervisee\_ID) uniquely identifies each tuple. This combination stores all the supervisors of a single employee as well as all the supervisees of a single employee.

## 06. Mapping of Multivalued Attributes

For each multivalued attribute, we will create a new relation. This relation will include the primary key of the relation of the entity type that the multivalued attribute belongs to. Also, the relation will include the multivalued attribute. The primary key of this relation will be the combination of the foreign key and the multivalued attribute.

### 01. Employee Nationality

<u>Employee_ID</u>	<u>Nationality</u>
--------------------	--------------------

This is a multivalued attribute for the Employee strong entity, so a new relation is created. Its primary key is the combination of (Employee\_ID, Nationality) and stores the multiple nationalities of a single employee.

### 02. Interests

<u>Visitor_ID (FK from visitor)</u>	<u>Interests</u>
-------------------------------------	------------------

This is a multivalued attribute and thus create a new relation for it. The primary key is the combination of the foreign keys from the relation visitor and the attribute itself. (The primary key for visitor is the GiftID, FirstName, Middle Initial, and LastName)

### 03. Comments

<u>EntryID (FK from feedback program)</u>	<u>Comment</u>
---	----------------

This is a multivalued attribute so we create a new relation for it. Include as primary key the primary key of the entity feedback program which is EntryID and the attribute itself which is comment.

#### 04. Tags

<u>Piece_ID (FK)</u>	<u>Piece_tags</u>
----------------------	-------------------

This relation represents the multivalued attribute Tags of the Piece entity type. This relation will include Piece\_ID as a foreign key that references the primary key of the Piece relation. Moreover, it will include the multivalued attribute (Tags) as a simple attribute in it (Piece\_tags). The primary key of this relation is a combination of the foreign key (Piece\_ID) and Piece\_tags.

#### 05. Culture

<u>Piece_ID (FK)</u>	<u>Piece_Culture</u>
----------------------	----------------------

This relation represents the multivalued attribute Culture(s) of the Piece entity type. This relation will include Piece\_ID as a foreign key that references the primary key of the Piece relation. Moreover, it will include the multivalued attribute (Culture(s)) as a simple attribute in it (Piece\_Culture). The primary key of this relation is a combination of the foreign key (Piece\_ID) and Piece\_Culture.

#### 06. Extension

<u>Dept_ID (FK)</u>	<u>Dept_Extension</u>
---------------------	-----------------------

This relation represents the multivalued attribute Extension of the Department entity type. This relation will include Dept\_ID as a foreign key that references the primary key of the Department relation. Moreover, it will include the multivalued attribute (Extension) as a simple attribute in it (Dept\_Extension). The primary key of this relation is a combination of the foreign key (Dept\_ID) and Dept\_Extension.

## 07. Mapping of N-ary Relationships

For every N-ary relationship, we map it into a new relation. We include the primary key of the relation of each participating entity type as a foreign key in this relation. These foreign keys, combined together, compose the Primary key of this new relation. Also, this relation will include any simple attribute (or any simple components of composite attributes) of the relationship.

### 01. provides

<u>ProvisionerID</u> <small>(FK from provisioner)</small>	<u>MediatorID</u> <small>(FK from mediator)</small>	<u>PieceID</u> <small>(FK from piece)</small>	Date of purchase	Mediator Fees	Price	Temporary	Country of origin
--	--	--	------------------	---------------	-------	-----------	-------------------

Provides is a tertiary relationship and thus include all the primary keys of the contributing entities participating in this relationship (Provisioner, mediator, piece) as foreign keys. The combination of these foreign keys is the primary key for the relation provides. Also, include all the simple attributes that this relationship has.

### 02. supplies

<u>Supplier_ID (FK)</u>	<u>Equipment_SR (FK)</u>	<u>Dept_ID (FK)</u>
-------------------------	--------------------------	---------------------

This relation represents the “supplies” relationship, which is a tertiary relationship among the entity types Supplier, Equipment, and Department. In this relation, we include Dept\_ID as a foreign key that references the primary key of the Department relation. Similarly, Supplier\_ID and Equipment\_SR are foreign keys referencing the Supplier relation and the Equipment Relation respectively. The combination of the 3 foreign keys makes up the primary key of the “supplies” relation.

### 03. is hosted

<u>Event_ID (FK)</u>	<u>Partner_ID (FK)</u>	<u>Department_ID (FK)</u>	<u>Section_ID (FK)</u>
----------------------	------------------------	---------------------------	------------------------

This relation represents the “is hosted” relationship, which is a quaternary relationship among the entity types Event, Partner, Section, and Department. In this relation, we include Department\_ID as a foreign key that references the primary key of the Department relation. Similarly, Event\_ID, Partner\_ID, and Section\_ID are foreign keys referencing the Event, Partner, and Sections respectively. The combination of the 4 foreign keys makes up the primary key of the “is hosted” relation.

## X. Comprehensive Overview of the Established Relations

Piece

ID	Name	Date of leave	Date of entry	Type	Date of start	Date of end	Inspiration	Can be Sold	Date of auction	Section_ID (FK)	Date of Display	V_ID
----	------	---------------	---------------	------	---------------	-------------	-------------	-------------	-----------------	-----------------	-----------------	------

Artist

ID	Name	Date of Death	Date of Birth	Biography	Nationality
----	------	---------------	---------------	-----------	-------------

Section

ID	Name	Location
----	------	----------

Provisioner

ID	Name	Phone	Email
----	------	-------	-------

Mediator

ID	Name	Phone	Email
----	------	-------	-------

Supplier

ID	Phone	Email	Name
----	-------	-------	------

Equipment

Serial Number	Name	Category
---------------	------	----------

Partner

<u>ID</u>	Name	Type	Percentage Share
-----------	------	------	------------------

Sponsor

<u>ID</u>	Name
-----------	------

Event

<u>ID</u>	Name	Type	Description	Date & Time
-----------	------	------	-------------	-------------

Department

<u>ID</u>	Name	Location	Fax	Email
-----------	------	----------	-----	-------

Employee

<u>ID</u>	Image	First_Name	Middle_Name	Last_Name	Sex	SSN
Date_of_Birth		Date_of_Start			Date_of_Termination	
Country	Street	Building	Email		Departmental_Extension	

Feedback Program

<u>EntryID</u>	Satisfaction Rating	Visitor_ID (FK)
----------------	---------------------	-----------------

Visitor

<u>ID</u>	FirstName	MiddleInitial	LastName	Gift_Points	Phone_Number	Email
-----------	-----------	---------------	----------	-------------	--------------	-------

Reward

<u>EmployeeID</u> <u>(FK)</u>	<u>Type</u>	<u>Count</u>	<u>Amount</u>	<u>Date implemented</u>	<u>Details</u>
----------------------------------	-------------	--------------	---------------	-------------------------	----------------

Medical Record

<u>EmployeeID</u> <u>(FK)</u>	<u>Date Initialized</u>	<u>Blood Type</u>	<u>Height</u>	<u>Weight</u>	<u>Blood pressure</u>
----------------------------------	-------------------------	-------------------	---------------	---------------	-----------------------

Police Record

<u>EmployeeID (FK)</u>	<u>Date Issued</u>	<u>Arrests</u>	<u>Convictions</u>	<u>Criminal Proceedings</u>
------------------------	--------------------	----------------	--------------------	-----------------------------

Emergency Contact

<u>EmployeeID</u> <u>(FK)</u>	<u>Name</u>	<u>Relationship</u>	<u>Contact Phone</u>
----------------------------------	-------------	---------------------	----------------------

Dependent

<u>EmployeeID</u> <u>(FK)</u>	<u>Name</u>	<u>Relationship with Employee</u>	<u>Date of Birth</u>	<u>Sex</u>	<u>SSN</u>
----------------------------------	-------------	-----------------------------------	----------------------	------------	------------

creates

<u>Artist_ID (FK)</u>	<u>Piece_ID (FK)</u>
-----------------------	----------------------

is overseen

<u>ID_Piece (FK)</u>	<u>Emp_ID (FK)</u>
----------------------	--------------------

is governed

<u>Section_ID (FK)</u>	<u>Employee_ID (FK)</u>
------------------------	-------------------------

is managed

<u>Dep_ID (FK)</u>	<u>Emp_ID (FK)</u>
--------------------	--------------------

visits

<u>Visitor_ID (FK)</u>	<u>Section_ID (FK)</u>	Ticket ID	Ticket Category
------------------------	------------------------	-----------	-----------------

donates

<u>Sponsor_ID (FK)</u>	<u>Department_ID</u>	Amount in (\$)
------------------------	----------------------	----------------

sponsors

<u>Event_ID (FK)</u>	<u>Sponsor_ID (FK)</u>	Amount in (\$)
----------------------	------------------------	----------------

supervision

<u>Supervisor_ID</u>	<u>Supervisee_ID</u>
----------------------	----------------------

Employee Nationality

<u>Employee_ID</u>	<u>Nationality</u>
--------------------	--------------------

Interests

<u>Visitor_ID</u> (FK from visitor)	<u>Interests</u>
-------------------------------------	------------------

Comments

<u>EntryID</u> (FK from feedback program)	<u>Comment</u>
---	----------------

Tags

<u>Piece_ID</u> (FK)	<u>Piece_tags</u>
----------------------	-------------------

Culture

<u>Piece_ID</u> (FK)	<u>Piece_Culture</u>
----------------------	----------------------

Extension

<u>Dept_ID</u> (FK)	<u>Dept_Extension</u>
---------------------	-----------------------

provides

<u>ProvisionerID</u> (FK from provisioner)	<u>MediatorID</u> (FK from mediator)	<u>PieceID</u> (FK from piece)	Date of purchase	Mediator Fees	Price	Temporary	Country of origin
---	---	-----------------------------------	------------------	---------------	-------	-----------	-------------------

supplies

<u>Supplier_ID</u> (FK)	<u>Equipment_SR</u> (FK)	<u>Dept_ID</u> (FK)
-------------------------	--------------------------	---------------------

is hosted

<u>Event_ID</u> (FK)	<u>Partner_ID</u> (FK)	<u>Department_ID</u> (FK)	<u>Section_ID</u> (FK)
----------------------	------------------------	---------------------------	------------------------

## XI. Creating Tables on Oracle 11g Express Edition

### 01. Creating Table: Piece

Column Name	Data Type	Nullable	Default	Primary Key
ID	NUMBER	No	-	1
NAME	VARCHAR2(100)	No	-	-
DATE_OF_LEAVE	DATE	Yes	-	-
DATE_OF_ENTRY	DATE	No	-	-
TYPE	VARCHAR2(255)	No	-	-
DATE_OF_START	VARCHAR2(255)	Yes	'UNAVAILABLE'	-
DATE_OF_END	VARCHAR2(255)	Yes	'UNAVAILABLE'	-
INSPIRATION	VARCHAR2(1500)	No	-	-
CAN_BE SOLD	VARCHAR2(5)	No	-	-
DATE_OF_AUCTION	DATE	No	-	-
SECTION_ID	NUMBER	Yes	1	-
DATE_OF_DISPLAY	DATE	Yes	-	-
V_ID	NUMBER	Yes	11	-
1 - 13				

```

CREATE TABLE Piece (
    ID INTEGER NOT NULL,
    Name      varchar(100)      NOT NULL,
    Date_of_leave DATE,
    Date_of_Entry DATE NOT NULL,
    Type varchar(255) NOT NULL ,
    Date_of_Start varchar(255) DEFAULT 'UNAVAILABLE',
    Date_of_end varchar(255) DEFAULT 'UNAVAILABLE',
    Inspiration varchar(1500) NOT NULL,
    Can_Be_Sold varchar(5) NOT NULL CHECK (Can_Be_Sold IN ('True','False')),
    Date_of_Auction DATE NOT NULL,
    PRIMARY KEY (ID),
    Section_ID INTEGER DEFAULT 1,
    Date_of_Display DATE,
    V_ID INTEGER DEFAULT 11,
    FOREIGN KEY(Section_ID) REFERENCES Section(ID),
    FOREIGN KEY(V_ID) REFERENCES Visitor(ID));

```

## 02. Creating Table: Artist

Column Name	Data Type	Nullable	Default	Primary Key
ID	NUMBER	No	-	1
NAME	VARCHAR2(255)	No	-	-
DATE_OF_DEATH	DATE	Yes	-	-
DATE_OF_BIRTH	DATE	Yes	-	-
BIOGRAPHY	VARCHAR2(2300)	No	-	-
NATIONALITY	VARCHAR2(255)	No	-	-
				1 - 6

```
CREATE TABLE Artist (
    ID INTEGER NOT NULL CHECK(ID>999),
    Name  varchar(255)  NOT NULL,
    Date_of_Death Date,
    Date_of_Birth Date,
    Biography varchar(2300) NOT NULL,
    Nationality varchar(255) NOT NULL,
    PRIMARY KEY (ID));
```

## 03. Creating Table: Section

Column Name	Data Type	Nullable	Default	Primary Key
ID	NUMBER	No	-	1
NAME	VARCHAR2(255)	No	-	-
LOCATION	VARCHAR2(255)	Yes	'Main Department'	-
				1 - 3

```
CREATE TABLE Section (
    ID INTEGER NOT NULL CHECK(ID>0),
    Name  varchar(255)  NOT NULL,
    Location varchar(255) DEFAULT 'Main Department',
    PRIMARY KEY (ID));
```

#### 04. Creating Table: Provisioner

Column Name	Data Type	Nullable	Default	Primary Key
ID	NUMBER	No	-	1
NAME	VARCHAR2(255)	No	-	-
PHONE	VARCHAR2(50)	Yes	-	-
EMAIL	VARCHAR2(250)	Yes	-	-
				1 - 4

```
CREATE TABLE Provisioner (
    ID INTEGER NOT NULL CHECK(ID>1000),
    Name varchar(255) NOT NULL,
    Phone varchar(50),
    Email varchar(250),
    PRIMARY KEY (ID));
```

#### 05. Creating Table: Mediator

Column Name	Data Type	Nullable	Default	Primary Key
ID	NUMBER	No	-	1
NAME	VARCHAR2(255)	No	-	-
PHONE	VARCHAR2(50)	Yes	-	-
EMAIL	VARCHAR2(250)	Yes	-	-
				1 - 4

```
CREATE TABLE Mediator(
    ID INTEGER NOT NULL CHECK(ID>10000),
    Name varchar(255) NOT NULL,
    Phone varchar(50),
    Email varchar(250),
    PRIMARY KEY (ID));
```

## 06. Creating Table: Supplier

Column Name	Data Type	Nullable	Default	Primary Key
ID	NUMBER	No	-	1
PHONE	VARCHAR2(50)	Yes	-	-
EMAIL	VARCHAR2(250)	Yes	-	-
NAME	VARCHAR2(255)	No	-	-
				1 - 4

CREATE TABLE Supplier(

```

ID INTEGER NOT NULL CHECK(ID>0),
Phone varchar(50),
Email varchar(250),
Name  varchar(255)  NOT NULL,
PRIMARY KEY (ID));
```

## 07. Creating Table: Equipment

Column Name	Data Type	Nullable	Default	Primary Key
SERIAL_NUMBER	NUMBER	No	-	1
NAME	VARCHAR2(255)	No	-	-
CATEGORY	VARCHAR2(255)	Yes	'General Uses'	-
				1 - 3

CREATE TABLE Equipment(

```

Serial_Number INTEGER NOT NULL,
Name  varchar(255)  NOT NULL,
Category varchar(255) DEFAULT 'General Uses',
PRIMARY KEY(Serial_Number));
```

## 08. Creating Table: Partner

Column Name	Data Type	Nullable	Default	Primary Key
ID	NUMBER	No	-	1
NAME	VARCHAR2(255)	No	-	-
TYPE	VARCHAR2(255)	No	-	-
PERCENTAGE_SHARE	FLOAT	Yes	0.1	-
				1 - 4

```
CREATE TABLE Partner (
    ID INTEGER NOT NULL CHECK(ID>0),
    Name varchar(255) NOT NULL,
    Type varchar(255) NOT NULL,
    Percentage_Share FLOAT DEFAULT 0.1 CHECK(Percentage_Share BETWEEN 0.1 AND 50),
    PRIMARY KEY (ID));
```

## 09. Creating Table: Sponsor

Column Name	Data Type	Nullable	Default	Primary Key
ID	NUMBER	No	-	1
NAME	VARCHAR2(255)	No	-	-
				1 - 2

```
CREATE TABLE Sponsor (
    ID INTEGER NOT NULL,
    Name varchar(255) NOT NULL,
    PRIMARY KEY (ID));
```

## 10. Creating Table: Event

Column Name	Data Type	Nullable	Default	Primary Key
ID	NUMBER	No	-	1
NAME	VARCHAR2(255)	No	-	-
TYPE	VARCHAR2(255)	No	-	-
DESCRIPTION	VARCHAR2(255)	No	-	-
DATE_AND_TIME	VARCHAR2(255)	No	-	-
				1 - 5

CREATE TABLE Event(

    ID INTEGER NOT NULL CHECK(ID>0),

    Name varchar(255) NOT NULL,

    Type varchar(255) NOT NULL ,

    Description varchar(255) NOT NULL,

    Date\_and\_time varchar(255) NOT NULL,

PRIMARY KEY (ID));

## 11. Creating Table: Department

Column Name	Data Type	Nullable	Default	Primary Key
ID	NUMBER	No	-	1
NAME	VARCHAR2(255)	Yes	'MUSEUM CENTRAL DEPARTMENT'	-
LOCATION	VARCHAR2(255)	Yes	'BLISS STREET'	-
FAX	VARCHAR2(255)	Yes	'7 000 011'	-
EMAIL	VARCHAR2(255)	Yes	'Museum.main@gmail.com'	-
EMP_ID	NUMBER	Yes	1000001	-
1 - 6				

```

CREATE TABLE Department (
    ID INTEGER NOT NULL CHECK(ID>0),
    Name  varchar(255) DEFAULT 'MUSEUM CENTRAL DEPARTMENT',
    Location varchar(255) DEFAULT 'BLISS STREET',
    Fax varchar(255) DEFAULT '7 000 011',
    Email varchar(255) DEFAULT 'Museum.main@gmail.com',
    PRIMARY KEY (ID),
    Emp_ID INTEGER DEFAULT 1000001,
    FOREIGN KEY(Emp_ID) REFERENCES Employee(ID));

```

## 12. Creating Table: Employee

Column Name	Data Type	Nullable	Default	Primary Key
ID	NUMBER	No	-	1
IMAGE	NUMBER	No	-	-
FIRST_NAME	VARCHAR2(100)	No	-	-
MIDDLE_NAME	VARCHAR2(2)	No	-	-
LAST_NAME	VARCHAR2(100)	No	-	-
SEX	VARCHAR2(2)	No	-	-
SSN	NUMBER	No	-	-
DATE_OF_BIRTH	DATE	Yes	-	-
DATE_OF_START	DATE	No	-	-
DATE_OF_TERMINATION	DATE	Yes	-	-
ROLE	VARCHAR2(50)	Yes	-	-
COUNTRY	VARCHAR2(255)	Yes	'LEBANON'	-
STREET	VARCHAR2(255)	No	-	-
BUILDING	VARCHAR2(255)	No	-	-
EMAIL	VARCHAR2(255)	No	-	-
DEPARTMENTAL_EXTENSION	NUMBER	Yes	1111	-
1 - 16				

CREATE TABLE EMPLOYEE (

```

ID INTEGER NOT NULL CHECK(ID>1000000),
Image INTEGER NOT NULL,
First_Name      varchar(100)      NOT NULL,
Middle_Name     varchar(2)       NOT NULL,
Last_Name       varchar(100)      NOT NULL,
Sex varchar(2) NOT NULL CHECK (Sex IN ('M','F','NULL')),
SSN INTEGER NOT NULL CHECK(SSN>0),
Date_of_Birth DATE,
Date_of_Start DATE NOT NULL,
Date_of_termination DATE,
Role varchar(50),
Country varchar(255) DEFAULT 'LEBANON',
Street varchar(255) NOT NULL ,
Building varchar(255) NOT NULL,
Email varchar(255) NOT NULL,
Departmental_Extension INTEGER DEFAULT 1111 CHECK(Departmental_Extension>1100),
PRIMARY KEY (ID));

```

### 13. Creating Table: Feedback Program

Column Name	Data Type	Nullable	Default	Primary Key
ENTRYID	NUMBER	No	-	1
SATISFACTION_RATING	NUMBER	No	-	-
VISITOR_ID	NUMBER	Yes	1	-
				1 - 3

```
CREATE TABLE FeedbackProgram (
    EntryID INTEGER NOT NULL CHECK(EntryID>0),
    Satisfaction_Rating INTEGER NOT NULL CHECK(Satisfaction_Rating BETWEEN 0 AND 10),
    PRIMARY KEY (EntryID),
    Visitor_ID INTEGER DEFAULT 1,
    FOREIGN KEY (Visitor_ID) REFERENCES Visitor(ID) );
```

### 14. Creating Table: Visitor

Column Name	Data Type	Nullable	Default	Primary Key
ID	NUMBER	No	-	1
FIRST_NAME	VARCHAR2(255)	No	-	-
MIDDLE_INITIAL	VARCHAR2(2)	No	-	-
LAST_NAME	VARCHAR2(255)	No	-	-
GIFT_POINTS	NUMBER	Yes	0	-
PHONE_NUMBER	VARCHAR2(100)	Yes	-	-
EMAIL	VARCHAR2(255)	No	-	-
				1 - 7

```
CREATE TABLE Visitor (
    ID INTEGER NOT NULL CHECK(ID>0),
    First_Name      varchar(255)      NOT NULL,
    Middle_Initial varchar(2) NOT NULL,
    Last_Name varchar(255) NOT NULL,
    Gift_Points INTEGER DEFAULT 0 CHECK(Gift_Points>=0),
    Phone_Number varchar(100),
    Email varchar(255) NOT NULL,
    PRIMARY KEY (ID));
```

## 15. Creating Table: Reward

Column Name	Data Type	Nullable	Default	Primary Key
EMPLOYEEID	NUMBER	No	10000001	1
TYPE	VARCHAR2(255)	No	-	2
AMOUNT	FLOAT	Yes	0.0	-
TIME	NUMBER	Yes	-	-
DATE_IMPLEMENTED	DATE	No	-	-
DETAILS	VARCHAR2(255)	No	-	-
1 - 6				

CREATE TABLE REWARD(

```

EmployeeID INTEGER DEFAULT 10000001,
Type varchar(255) NOT NULL,
Amount FLOAT DEFAULT 0.0,
Time INTEGER,
Date_implemented DATE NOT NULL,
Details varchar(255) NOT NULL,
PRIMARY KEY(EmployeeID,Type),
FOREIGN KEY (EmployeeID) REFERENCES EMPLOYEE(ID));

```

## 16. Creating Table: Medical Record

Column Name	Data Type	Nullable	Default	Primary Key
EMPLOYEEID	NUMBER	No	10000001	1
DATE_INITIALIZED	DATE	No	-	2
BLOOD_TYPE	VARCHAR2(3)	Yes	'O-'	-
WEIGHT	FLOAT	Yes	55	-
HEIGHT	FLOAT	Yes	135	-
BLOOD_PRESSURE	VARCHAR2(255)	No	-	-
1 - 6				

CREATE TABLE MEDICAL\_RECORD(

```

EmployeeID INTEGER DEFAULT 10000001,
Date_initialized DATE NOT NULL,
Blood_Type VARCHAR(3) DEFAULT 'O-' CHECK (Blood_Type IN ('A+','A-','B+','B-','AB+','AB-','O+','O-')),
Weight FLOAT DEFAULT 55 CHECK (Weight>=3.5),
Height FLOAT DEFAULT 135 CHECK(Height>=45),
Blood_Pressure varchar(255) NOT NULL,
FOREIGN KEY (EmployeeID) REFERENCES EMPLOYEE(ID),
PRIMARY KEY(EmployeeID,Date_initialized));

```

## 17. Creating Table: Police Record

Column Name	Data Type	Nullable	Default	Primary Key
EMPLOYEEID	NUMBER	No	10000001	1
DATE_ISSUED	DATE	No	-	2
ARRESTS	VARCHAR2(6)	No	-	-
CONVICTIONS	VARCHAR2(6)	No	-	-
CRIMINAL_PROCEEDINGS	VARCHAR2(6)	No	-	-
1 - 5				

CREATE TABLE POLICE\_RECORD(

```

EmployeeID INTEGER DEFAULT 10000001,
Date_Issued DATE NOT NULL,
Arrests varchar(6) NOT NULL CHECK (Arrests IN ('True','False')),
Convictions varchar(6) NOT NULL CHECK (Convictions IN ('True','False')),
Criminal_Proceedings varchar(6) NOT NULL CHECK (Criminal_Proceedings IN ('True','False')),

PRIMARY KEY(EmployeeID,Date_Issued),
FOREIGN KEY (EmployeeID) REFERENCES EMPLOYEE(ID));

```

## 18. Creating Table: Emergency Contact

Column Name	Data Type	Nullable	Default	Primary Key
EMPLOYEEID	NUMBER	No	10000001	1
NAME	VARCHAR2(255)	No	-	2
RELATIONSHIP	VARCHAR2(255)	No	-	-
CONTACT_PHONE	VARCHAR2(255)	Yes	-	-
			1 - 4	

```
CREATE TABLE Emergency_Contact(  
    EmployeeID INTEGER DEFAULT 10000001,  
    Name varchar(255) NOT NULL,  
    Relationship varchar(255) NOT NULL,  
    Contact_Phone varchar(255),  
    FOREIGN KEY (EmployeeID) REFERENCES EMPLOYEE(ID),  
    PRIMARY KEY(EmployeeID,Name));
```

## 19. Creating Table: Dependent

Column Name	Data Type	Nullable	Default	Primary Key
EMPLOYEEID	NUMBER	No	10000001	1
NAME	VARCHAR2(255)	No	-	2
RELATIONSHIP_WITH_EMPLOYEE	VARCHAR2(255)	No	-	-
DATE_OF_BIRTH	DATE	Yes	-	-
SEX	VARCHAR2(2)	No	-	-
SSN	NUMBER	No	-	-
1 - 6				

CREATE TABLE Dependent(

```

EmployeeID INTEGER DEFAULT 10000001,
Name varchar(255) NOT NULL,
Relationship_with_Employee varchar(255) NOT NULL,
Date_of_Birth DATE,
Sex varchar(2) NOT NULL CHECK (Sex IN ('M','F','NULL)),
SSN INTEGER NOT NULL,
FOREIGN KEY (EmployeeID) REFERENCES EMPLOYEE(ID),
PRIMARY KEY(EmployeeID,Name));

```

## 20. Creating Table: Creates

Column Name	Data Type	Nullable	Default	Primary Key
ARTIST_ID	NUMBER	No	-	1
PIECE_ID	NUMBER	No	-	2
<b>1 - 2</b>				

CREATE TABLE Creates(

```

Artist_ID INTEGER NOT NULL,
PIECE_ID INTEGER NOT NULL,
FOREIGN KEY (Artist_ID) REFERENCES ARTIST(ID),
FOREIGN KEY (PIECE_ID) REFERENCES PIECE(ID),
PRIMARY KEY(Artist_ID,PIECE_ID)
);
```

## 21. Creating Table: Is Overseen

Column Name	Data Type	Nullable	Default	Primary Key
ID_PIECE	NUMBER	No	1	1
EMP_ID	NUMBER	No	1000001	2
<b>1 - 2</b>				

CREATE TABLE isOverseen (

```

ID_Piece INT DEFAULT 1 NOT NULL,
Emp_ID INT DEFAULT 1000001 NOT NULL,
FOREIGN KEY (ID_Piece) REFERENCES PIECE(ID) ON DELETE CASCADE,
FOREIGN KEY (Emp_ID) REFERENCES EMPLOYEE(ID) ON DELETE CASCADE,
PRIMARY KEY(ID_Piece,Emp_ID)
);
```

## 22. Creating Table: Is Governed

Column Name	Data Type	Nullable	Default	Primary Key
SECTION_ID	NUMBER	No	1	1
EMPLOYEE_ID	NUMBER	No	1000001	2
<b>1 - 2</b>				

```
CREATE TABLE isGoverned (
    Section_ID INT DEFAULT 1 NOT NULL,
    Employee_ID INT DEFAULT 1000001 NOT NULL,
    FOREIGN KEY (Section_ID) REFERENCES SECTION(ID) ON DELETE CASCADE,
    FOREIGN KEY (Employee_ID) REFERENCES EMPLOYEE(ID) ON DELETE CASCADE,
    PRIMARY KEY(Section_ID,Employee_ID)
);
```

## 23. Creating Table: Is Managed

Column Name	Data Type	Nullable	Default	Primary Key
DEP_ID	NUMBER	No	1	1
EMP_ID	NUMBER	No	1000001	2
<b>1 - 2</b>				

```
CREATE TABLE isManaged (
    Dep_ID INT DEFAULT 1 NOT NULL,
    Emp_ID INT DEFAULT 1000001 NOT NULL,
    FOREIGN KEY (Dep_ID) REFERENCES DEPARTMENT(ID) ON DELETE CASCADE,
    FOREIGN KEY (Emp_ID) REFERENCES EMPLOYEE(ID) ON DELETE CASCADE,
    PRIMARY KEY(Dep_ID,Emp_ID)
);
```

## 24. Creating Table: Visits

Column Name	Data Type	Nullable	Default	Primary Key
VISITOR_ID	NUMBER	No	-	1
SECTIONID	NUMBER	No	-	2
TICKETID	NUMBER	No	-	-
TICKET_CATEGORY	VARCHAR2(255)	Yes	-	-
				1 - 4

```
CREATE TABLE visits(
    Visitor_ID INT NOT NULL,
    SectionID INT NOT NULL,
    TicketID INT NOT NULL,
    Ticket_Category varchar(255),
    FOREIGN KEY(Visitor_ID) REFERENCES VISITOR(ID) ON DELETE CASCADE,
    FOREIGN KEY(SectionID) REFERENCES SECTION(ID) ON DELETE CASCADE,
    PRIMARY KEY(Visitor_ID,SectionID)
);
```

## 25. Creating Table: Donates

Column Name	Data Type	Nullable	Default	Primary Key
SPONSOR_ID	NUMBER	No	-	1
DEPARTMENT_ID	NUMBER	No	-	2
AMOUNT	NUMBER	No	-	-
				1 - 3

```
CREATE TABLE donates(
    Sponsor_ID INT NOT NULL,
    Department_ID INT NOT NULL,
    Amount INT NOT NULL check(Amount>0),
    FOREIGN KEY(Sponsor_ID) REFERENCES Sponsor(ID) ON DELETE CASCADE,
    FOREIGN KEY(Department_ID) REFERENCES Department(ID) ON DELETE CASCADE,
    PRIMARY KEY(Sponsor_ID,Department_ID)
);
```

## 26. Creating Table: Sponsors

Column Name	Data Type	Nullable	Default	Primary Key
SPONSOR_ID	NUMBER	No	-	1
EVENT_ID	NUMBER	No	-	2
AMOUNT	NUMBER	No	-	-
				1 - 3

```
CREATE TABLE sponsors(
    Event_ID INT NOT NULL,
    Sponsor_ID INT NOT NULL,
    Amount INT NOT NULL check(Amount>0),
    FOREIGN KEY(Sponsor_ID) REFERENCES Sponsor(ID) ON DELETE CASCADE,
    FOREIGN KEY(Event_ID) REFERENCES Event(ID) ON DELETE CASCADE,
    PRIMARY KEY(Sponsor_ID,Event_ID));
```

## 27. Creating Table: Supervision

Column Name	Data Type	Nullable	Default	Primary Key
EMPLOYEE_ID	NUMBER	No	-	1
SUPERVISOR_ID	NUMBER	No	-	2
<a href="#">1 - 2</a>				

CREATE TABLE supervises(

    Employee\_ID INT NOT NULL,

    Supervisor\_ID INT NOT NULL,

    FOREIGN KEY(Employee\_ID) REFERENCES Employee(ID) ON DELETE CASCADE,

    FOREIGN KEY(Supervisor\_ID) REFERENCES Employee(ID) ON DELETE CASCADE,

    PRIMARY KEY(Employee\_ID,Supervisor\_ID)

);

## 28. Creating Table: Employee Nationality

Column Name	Data Type	Nullable	Default	Primary Key
EMPLOYEE_ID	NUMBER	No	-	1
NATIONALITY	VARCHAR2(100)	No	-	2
<a href="#">1 - 2</a>				

Create TABLE EMPNATIONALITY(

    EMPLOYEE\_ID INTEGER NOT NULL,

    NATIONALITY VARCHAR(100) NOT NULL,

    FOREIGN KEY (EMPLOYEE\_ID) REFERENCES EMPLOYEE(ID),

    PRIMARY KEY(EMPLOYEE\_ID,NATIONALITY)

);

## 29. Creating Table: Interests

Column Name	Data Type	Nullable	Default	Primary Key
VISITOR_ID	NUMBER	No	1	1
INTERESTS	VARCHAR2(250)	No	-	2
<b>1 - 2</b>				

CREATE TABLE Interests(

```

        Visitor_ID INTEGER DEFAULT 1,
        Interests varchar(250) NOT NULL,
        FOREIGN KEY (Visitor_ID) REFERENCES Visitor(ID),
        PRIMARY KEY(Visitor_ID,Interests)
    );

```

## 30. Creating Table: Comments

Column Name	Data Type	Nullable	Default	Primary Key
FEEDBACKENTRYID	NUMBER	No	1	1
COMMENTS	VARCHAR2(250)	No	-	2
<b>1 - 2</b>				

CREATE TABLE Comments(

```

        FeedbackEntryID INTEGER DEFAULT 1,
        Comments varchar(250) NOT NULL,
        FOREIGN KEY (FeedbackEntryID) REFERENCES FEEDBACKPROGRAM(ENTRYID),
        PRIMARY KEY(FeedbackEntryID,Comments)
    );

```

### 31. Creating Table: Tags

Column Name	Data Type	Nullable	Default	Primary Key
PIECE_ID	NUMBER	No	1	1
TAG	VARCHAR2(250)	No	-	2
				1 - 2

CREATE TABLE Tags(

```

    Piece_ID INTEGER DEFAULT 1,
    Tag varchar(250) NOT NULL,
    FOREIGN KEY (Piece_ID) REFERENCES PIECE(ID),
    PRIMARY KEY(Piece_ID,Tag)
);
```

### 32. Creating Table: Cultures

Column Name	Data Type	Nullable	Default	Primary Key
PIECE_ID	NUMBER	No	1	1
CULTURE	VARCHAR2(250)	No	-	2
				1 - 2

CREATE TABLE Cultures(

```

    Piece_ID INTEGER DEFAULT 1,
    Culture varchar(250) NOT NULL,
    FOREIGN KEY (Piece_ID) REFERENCES PIECE(ID),
    PRIMARY KEY(Piece_ID,Culture)
);
```

### 33. Creating Table: Department Extension

Column Name	Data Type	Nullable	Default	Primary Key
DEPT_ID	NUMBER	No	1	1
EXTENSION	NUMBER	No	1225	2
1 - 2				

```
CREATE TABLE Dept_Extension(  
    Dept_ID INTEGER DEFAULT 1,  
    Extension INTEGER DEFAULT 1225 NOT NULL,  
    FOREIGN KEY (Dept_ID) REFERENCES DEPARTMENT(ID),  
    PRIMARY KEY(Dept_ID,Extension)  
);
```

### 34. Creating Table: Provides

Column Name	Data Type	Nullable	Default	Primary Key
PROVISIONERID	NUMBER	No	-	1
MEDIATORID	NUMBER	No	-	2
PIECEID	NUMBER	No	1006	3
DATE_OF_PURCHASE	VARCHAR2(150)	No	-	-
MEDIATOR_FEES	NUMBER	No	-	-
TEMPORARY	VARCHAR2(15)	Yes	-	-
COUNTRY_OF_ORIGIN	VARCHAR2(20)	No	-	-
1 - 7				

CREATE TABLE provides(

```

    ProvisionerID INT NOT NULL,
    MediatorID INT NOT NULL,
    PieceID INT DEFAULT 1006 NOT NULL,
    Date_of_purchase varchar(150) not null,
    Mediator_Fees INT NOT NULL CHECK(Mediator_Fees>0),
    Temporary varchar(15),
    Country_Of_Origin varchar(20) NOT NULL,
    FOREIGN KEY(ProvisionerID) REFERENCES Provisioner(ID) ON DELETE CASCADE,
    FOREIGN KEY(MediatorID) REFERENCES Mediator(ID) ON DELETE CASCADE,
    FOREIGN KEY(PieceID) REFERENCES Piece(ID) ON DELETE CASCADE,
    PRIMARY KEY(ProvisionerID,MediatorID,PieceID)
);
```

### 35. Creating Table: Supplies

Column Name	Data Type	Nullable	Default	Primary Key
SUPPLIER_ID	NUMBER	No	-	1
EQUIPMENT_SR	NUMBER	No	-	2
DEPT_ID	NUMBER	No	-	3
				1 - 3

```
CREATE TABLE supplies(
    Supplier_ID INT NOT NULL,
    Equipment_SR INT NOT NULL,
    Dept_ID INT NOT NULL,
    FOREIGN KEY(Supplier_ID) REFERENCES Supplier(ID) ON DELETE CASCADE,
    FOREIGN KEY(Equipment_SR) REFERENCES Equipment(Serial_Number) ON DELETE
    CASCADE,
    FOREIGN KEY(Dept_ID) REFERENCES Department(ID) ON DELETE CASCADE,
    PRIMARY KEY(Supplier_ID,Equipment_SR,Dept_ID)
);
```

### 36. Creating Table: Is Hosted

Column Name	Data Type	Nullable	Default	Primary Key
EVENT_ID	NUMBER	No	-	1
PARTNER_ID	NUMBER	No	-	2
DEPARTMENT_ID	NUMBER	No	-	3
SECTION_ID	NUMBER	No	-	4
				1 - 4

```
CREATE TABLE ishosted(
    Event_ID INT NOT NULL,
    Partner_ID INT NOT NULL,
    Department_ID INT NOT NULL,
    Section_ID INT NOT NULL,
    FOREIGN KEY(Event_ID) REFERENCES Event(ID) ON DELETE CASCADE,
    FOREIGN KEY(Partner_ID) REFERENCES Partner(ID) ON DELETE CASCADE,
    FOREIGN KEY(Section_ID) REFERENCES Section(ID) ON DELETE CASCADE,
    FOREIGN KEY(Department_ID) REFERENCES Department(ID) ON DELETE CASCADE,
    PRIMARY KEY(Event_ID,Partner_ID,Department_ID,Section_ID)
);
```

## XII. Inserting Data Using SQL

### 01. Inserting Data into Piece

(Note that due to the limitation of space, each image is extended to view the full tuple)

(The following images do not show all tuples)

EDIT	ID	NAME	DATE_OF_LEAVE	DATE_OF_ENTRY	TYPE	DATE_OF_START	DATE_OF_END	INSPIRATION	CAN_BE SOLD	DATE_OF_AUCTION	SECTION_ID	DATE_OF_DISPLAY	V_ID
	210003	The Kiss	12/12/2022	11/19/2016	Painting	1907	1908	The Kiss is a vivid and sensual painting that was created as an outcry against the academic art of the 19th century. It was influenced by organic shapes, especially the wavy lines of plants and flowers.	True	12/14/2022	1	12/01/2020	6223
	210002	Guernica	-	05/25/2015	Painting	1937	1937	Picasso created Guernica in response to the bombing of Guernica on April 26, 1937, which took place in the Basque Country of	False	03/03/2014	1	06/02/2015	2122
	210011	First Microscope	01/02/2019	01/01/2019	Scientific	1585	1600	This is the first microscope that was ever invented	False	12/01/2020	9	12/31/2018	1122
	210009	Biggest Bee	01/27/2021	11/11/2018	Insect	1976	1977	Largest bee ever found on this planet	True	01/17/2021	4	12/12/2018	2928
	210010	Mommiest Cat	09/10/2020	05/05/2018	Stuffed animal	1999	2000	It is believed that this cat had the largest number of offsprings at once, 55	True	09/13/2020	3	01/10/2019	3222
	210008	Smallest Dog ever	09/10/2020	05/05/2018	Stuffed animal	1976	1977	Smallest dog ever found in the world	True	09/13/2020	3	01/03/2019	6222
	210007	Enviest Lion	-	04/09/2018	Stuffed animal	1981	1983	Lion attacked a small village called Blat and almost killed everyone until a small guy killed him	False	01/01/2018	3	07/07/2018	2322
	210015	DBA MASTER PIECE	-	04/05/2022	Artifacts	2022	2022	A beautiful item that the DBA of our museum found and decided to place it in our museum	False	01/01/2022	10	05/05/2022	2222
	210014	Prince Fakheredine scarf	-	11/10/2009	Textiles	1880	1884	The scarf of the Lebanese prince Fakheredine II	False	11/07/2009	8	12/01/2009	6222

INSERT INTO PIECE VALUES(210003, 'The Kiss', DATE '2022-12-12', DATE' 2016-11-19', 'Painting', '1907', '1908', 'The Kiss is a vivid and sensual painting that was created as an outcry against the academic art of the 19th century. It was influenced by organic shapes, especially the wavy lines of plants and flowers', 'True', DATE '2022-12-14', 1, DATE '2020-12-1', 6223);

INSERT INTO PIECE VALUES(210002, 'Guernica', NULL, DATE '2015-5-25', 'Painting', '1937', '1937', 'Picasso created Guernica in response to the bombing of Guernica on April 26, 1937, which took place in the Basque Country of northern Spain at the behest of the Spanish Nationalists. Guernica was destroyed by Nazi Germany and Fascist Italy.', 'False', DATE' 2014-3-3', 1, DATE' 2015-6-2', 2122);

INSERT INTO PIECE VALUES(210001, 'The Starry Night', NULL, DATE '2015-4-7', 'Painting', '1889', '1889', 'While seeking treatment for his mental problems for twelve months in 1889–1890 at the Saint-

Paul-de-Mausole institution in Saint-Rémy, southern France, the artist was inspired by the view from his window.', 'False', DATE '2014-4-14', 1, DATE '2014-6-4', 2222);

INSERT INTO PIECE VLAUES(210004, 'Venus of Willendorf', DATE '2018-12-12', DATE '2016-12-12', 'Sculpture', '2400 BCE', '2200 BCE', 'The majority of historians concur that they performed a ritualistic function and probably praised concepts associated with fertility, including as femininity, deities, and eroticism', 'True', DATE '2018-12-1', 2, DATE '2017-1-3', 1122);

INSERT INTO PIECE VALUES(210000, 'Mona Lisa', NULL, DATE '2014-3-3', 'Painting', '1503', '1504', 'The painting is a portrait of Madam Lisa Giocondo. She was wife of a wealthy Florentine', 'True', DATE '2014-2-2', 1, DATE '2014-4-4', 2122);

INSERT INTO PIECE VALUES(210005, 'the TerraCotta Army', DATE '2019-5-5', DATE '2017-2-4', 'Sculpture', '248 BCE', '208 BCE', 'Despite being created exclusively by the Chinese, they were Western culture inspired.', 'True', DATE '2019-5-15', 2, DATE '2016-1-9', 2356);

INSERT INTO PIECE VALUES(210006, 'David', DATE '2020-3-4', DATE '2017-2-26', 'Sculpture', '1501 ', '1504 ', 'While Michelangelo drew the physical shape of the statue on the principles of ancient Roman sculpture—specifically, representations of the hero Hercules, who has long been a symbol of the independent city of Florence—while drawing his inspiration for the David narrative from the bible.', 'True', DATE '2020-2-14', 2, DATE '2017-7-7', 2928);

INSERT INTO PIECE VALUES(210011, 'First Microscope', DATE '2019-1-2', DATE '2019-1-1', 'Scientific', '1585 ', '1600 ', 'This is the first microscope that was ever invented', 'False', DATE '2020-12-1', 9, DATE '2018-12-31', 1122);

INSERT INTO PIECE VALUES(210009, 'Biggesr Bee', DATE '2021-1-27', DATE '2018-11-11', 'Insect', '1976 ', '1977 ', 'Largest bee ever found on this planet', 'True', DATE '2021-1-17', 4, DATE '2018-12-12', 2928);

INSERT INTO PIECE VALUES(210010, 'Mommiest Cat', DATE '2020-9-10', DATE '2018-5-5', 'Stuffed animal', '1999 ', '2000 ', 'It is claimed that this cat had the largest number of offsprings at once, 55', 'True', DATE '2020-9-13', 3, DATE '2019-1-10', 3222);

INSERT INTO PIECE VALUES(210008, 'Smallest Dog ever', DATE '2020-9-10', DATE '2018-5-5', 'Stuffed animal', '1976 ', '1977 ', 'Smallest dog ever found in the world', 'True', DATE '2020-9-13', 3, DATE '2019-1-3', 6222);

INSERT INTO PIECE VALUES(210007, 'Enviest Lion', NULL, DATE '2018-4-9', 'Stuffed animal', '1981', '1983', 'Lion attacked a small village called Blat and almost killed everyone until a small guy killed him', 'False', DATE '2018-1-1', 3, DATE '2018-7-7', 2322);

INSERT INTO PIECE VALUES(210015, 'DBA MASTER PIECE', NULL, DATE '2022-4-5', 'Artifacts', '2022', '2022', 'A beautiful item that the DBA of our museum found and decided to place it in our museum', 'False', DATE '2022-1-1', 10, DATE '2022-5-5', 2222);

INSERT INTO PIECE VALUES(210014, 'Prince Fakheredine scarf', NULL, DATE '2009-11-10', 'Textiles', '1880', '1884', 'The scarf of the Lebanese prince Fakheredine II', 'False', DATE '2009-11-7', 8, DATE '2009-12-1', 6222);

INSERT INTO PIECE VALUES(210012, 'First written paper', NULL, DATE '2019-2-28', 'Historic', '179 BCE', '141 BCE', 'This is the first written paper ever in history', 'True', DATE '2019-2-27', 6, DATE '2019-4-5', 2928);

INSERT INTO PIECE VALUES(210013, 'First invented Computer', NULL, DATE '2011-3-5', 'Technology', '1936', '1938', 'Babbage created the Analytical Engine plans around the middle', 'True', DATE '2011-1-1', 7, DATE '2011-4-4', 1122);

## 02. Inserting Data into Artist

EDIT	ID	NAME	DATE_OF_DEATH	DATE_OF_BIRTH	BIOGRAPHY	NATIONALITY
	1006	Leonardo da Vinci	05/02/1519	04/15/1452	Italian poet, he was well known for the Monalisa painting	Italian
	1001	Vincent van Gogh	07/29/1890	03/30/1853	Dutch Post Impressionist painter. Known for the starry night painting	Dutch
	1002	Gustav Klimt	02/06/1918	07/14/1862	An Austrian Painter. He founded the Vienna Sezession school of painting.	Hungarian
	1045	Pablo Picasso	10/25/1881	04/08/1973	Painter, sculptor, printmaker, ceramicist, and theater designer from Spain	Spanish
	1230	Malek Al Adghal	-	12/04/1980	Brave and strong Lebanes man. Well known for the biggest bee	Lebanese
	1073	Malek Abboud	08/02/2022	08/03/1970	Brave and strong. Born in Syria and lived in lebanon	Syrian
	1040	Zacharias Janssen	-	-	A famous Dutch inventor and spectacle maker	Dutch
	1010	Charles Babbage	12/26/1791	04/15/1452	He is accredited for inventing the first mechanical computer in history	British
	1120	Mohammad Hamieh	09/01/1810	04/15/1452	Designer of Prince Fakherdine II scarf	Lebanese
	1011	Jad Al Akram	-	12/12/2000	The data base administrator of our system	Palestenian
	1070	Michael Angelo	02/18/1564	03/06/1475	Sculptor, painter, architect, and Italian poet. Known for the David Sculpture	Italian

row(s) 1 - 11 of 11

INSERT INTO ARTIST VALUES(1006,'Leonardo da Vinci',DATE '1519-5-2',DATE '1452-4-15','Italian poet, he was well known for the Monalisa painting','Italian');

INSERT INTO ARTIST VALUES(1001,'Vincent van Gogh',DATE '1890-7-29',DATE '1853-3-30','Dutch Post Impressionist painter. Known for the starry night painting','Dutch');

INSERT INTO ARTIST VALUES(1002,'Gustav Klimt',DATE '1918-2-6',DATE '1862-7-14','An Austrian Painter. He founded the Vienna Sezession school of painting.','Hungarian');

INSERT INTO ARTIST VALUES(1045,'Pablo Picasso',DATE '1881-10-25',DATE '1973-4-8','Painter, sculptor, printmaker, ceramicist, and theater designer from Spain','Spanish');

INSERT INTO ARTIST VALUES(1230,'Malek Al Adghal',NULL,DATE '1980-12-4','Brave and strong Lebanes man. Well known for the biggest bee','Lebanese');

INSERT INTO ARTIST VALUES(1073,'Malek Abboud',DATE '2022-8-2',DATE '1970-8-3','Brave and strong. Born in Syria and lived in lebanon','Syrian');

INSERT INTO ARTIST VALUES(1040,'Zacharias Janssen',NULL,NULL,'A famous Dutch inventor and spectacle maker','Dutch');

INSERT INTO ARTIST VALUES(1010,'Charles Babbage',DATE '1791-12-26',DATE '1452-4-15','He is accredited for inventing the first mechanical computer in history','British');

INSERT INTO ARTIST VALUES(1120,'Mohammad Hamieh',DATE '1810-9-1',DATE '1452-4-15','Designer of Prince Fakherdine II scarf','Lebanese');

INSERT INTO ARTIST VALUES(1011,'Jad Al Akram',NULL,DATE '2000-12-12','The data base administrator of our system','Palestenian');

INSERT INTO ARTIST VALUES(1070,'Michael Angelo',DATE '1564-2-18',DATE '1475-3-6','Sculptor, painter, architect, and Italian poet. Known for the David Sculpture','Italian');

### 03. Inserting Data into Section

EDIT	ID	NAME	LOCATION
	1	Paintings	Floor 1 room 1
	2	Sculptures	Floor 1 room 3
	3	Stuffed animals	Floor 2 room 4
	4	Insects	Floor 3 room 5
	5	Astronomy	Floor 4 room 2
	6	History	Floor 4 room 4
	7	Technology	Floor 4 room 3
	8	Textiles	Floor 1 room 6
	9	Natural Sciences	Floor 2 room 5
	10	Artifacts	Floor 5 room 5
row(s) 1 - 10 of 10			

```
INSERT INTO SECTION VALUES(1,'Paintings','Floor 1 room 1');
```

```
INSERT INTO SECTION VALUES(2,'Sculptures','Floor 1 room 3');
```

```
INSERT INTO SECTION VALUES(3,'Stuffed animals','Floor 2 room 4');
```

```
INSERT INTO SECTION VALUES(4,'Insects','Floor 3 room 5');
```

```
INSERT INTO SECTION VALUES(5,'Astronomy','Floor 4 room 2');
```

```
INSERT INTO SECTION VALUES(6,'History','Floor 4 room 4');
```

```
INSERT INTO SECTION VALUES(7,'Technology','Floor 4 room 3');
```

```
INSERT INTO SECTION VALUES(8,'Textiles','Floor 1 room 6');
```

```
INSERT INTO SECTION VALUES(9,'Natural Sciences','Floor 2 room 5');
```

```
INSERT INTO SECTION VALUES(10,'Artifacts','Floor 5 room 5');
```

#### 04. Inserting Data into Provisioner

EDIT	ID	NAME	PHONE	EMAIL
	15432	Brian Miller & Co	71766344	B.miller32@hotmail.com
	13452	Scott Harris	70987559	Scottharris@hotmail.com
	11223	Paint of Heart	01445667	poh@gmail.com
	10012	Michelle Garza	78452610	m.garza14@gmail.com
	12096	Woodcraft Store	03547813	wrcraftstore@gmail.com
	51343	Local Art Gallery	03547813	artgallery@gmail.com
	32451	Cassava Art	01476234	Cassavaart05@hotmail.com
	65421	Sculpture Gallery	03447987	Sculpturega12@gmail.com
	43241	Studio Art	01234725	Studioart32@gmail.com
	61239	The Noble Gallery	03583919	TNGallery@gmail.com

row(s) 1 - 10 of 10

```

INSERT INTO PROVISIONER VALUES(15432,'Brian Miller & Co', '71766344', 'B.miller32@hotmail.com');

INSERT INTO PROVISIONER VALUES(13452,'Scott Harris', '70987559','Scottharris@hotmail.com');

INSERT INTO PROVISIONER VALUES(11223,'Paint of Heart','01445667','poh@gmail.com ');

INSERT INTO PROVISIONER VALUES(10012,'Michelle Garza','78452610','m.garza14@gmail.com');

INSERT INTO PROVISIONER VALUES(12096,'Woodcraft Store','03547813','wrcraftstore@gmail.com');

INSERT INTO PROVISIONER VALUES(51343,'Local Art Gallery','03547813','artgallery@gmail.com');

INSERT INTO PROVISIONER VALUES(32451,'Cassava Art','01476234','Cassavaart05@hotmail.com');

INSERT INTO PROVISIONER VALUES(65421,'Sculpture Gallery','03447987','Sculpturega12@gmail.com');

INSERT INTO PROVISIONER VALUES(43241,'Studio Art','01234725','Studioart32@gmail.com');

INSERT INTO PROVISIONER VALUES(61239,'The Noble Gallery','03583919','TNGallery@gmail.com');

```

## 05. Inserting Data into Mediator

EDIT	ID	NAME	PHONE	EMAIL
	10001	Brian Young	70321654	brian.young@hotmail.com
	10002	Jennifer Garza	71983245	jennifer@gmail.com
	11005	Scott Miller	76468910	Scott05@hotmail.com
	11002	Amber Harris	70132564	Amharris02@hotmail.com
	10015	Michelle Miller	78856982	Mich.miller@hotmail.com
	11100	Brian Garza	70123655	bgarza@gmail.com
	10190	Carol Young	78334562	Carol03y@hotmail.com
	11180	Jennifer Smith	71224546	Jsmith12@gmail.com
	10009	Leslie Harrison	78544672	LesHarrison@gmail.com

INSERT INTO MEDIATOR VALUES(10001,'Brian Young',70321654,'brian.young@hotmail.com');

INSERT INTO MEDIATOR VALUES(10002,'Jennifer Garza',71983245,'jennifer@gmail.com');

INSERT INTO MEDIATOR VALUES(11005,'Scott Miller',76468910,'Scott05@hotmail.com');

INSERT INTO MEDIATOR VALUES(11002,'Amber Harris',70132564,'Amharris02@hotmail.com');

INSERT INTO MEDIATOR VALUES(10015,'Michelle Miller',78856982,'Mich.miller@hotmail.com');

INSERT INTO MEDIATOR VALUES(11100,'Brian Garza',70123655,'bgarza@gmail.com');

INSERT INTO MEDIATOR VALUES(10006,'Lisa Taylor', '76567988','Lisa.taylor@gmail.com');

INSERT INTO MEDIATOR VALUES(10190,'Carol Young',78334562,'Carol03y@hotmail.com');

INSERT INTO MEDIATOR VALUES(11180,'Jennifer Smith',71224546,'Jsmith12@gmail.com');

INSERT INTO MEDIATOR VALUES(10009,'Leslie Harrison',78544672,'LesHarrison@gmail.com');

## 06. Inserting Data into Supplier

EDIT	ID	PHONE	EMAIL	NAME
	240000	Jalal H. Zeinnedin	+961 76-815 217	JalalZeinnedin3@gmail.com
	240001	Rania H. Masri	+961 76-236 742	RaniaHMasri@gmail.com
	240002	Mira K. Rhayyem	+961 76 891 432	MiraRhayyem@gmail.com
	240003	Jadullah N. Abdullah	+961 3 444 232	jadullahAbdullah99@gmail.com
	240004	Sirena H. Hamieh	+961 70 421 673	SirenaHHamieh@gmail.com
	240005	Rein H. Monther	+961 27 830 899	reinMothEr01@gmail.com
	240006	Safi L. Bahri	+961 27 830 309	SafiBahri02@gmail.com
	240007	Jamil J. Jaber	+961 81 934 671	jamilJJaber@gmail.com
	240008	Lana H. Halabi	+961 81 233 356	Lana2Halabi@gmail.com
	240009	Mansour H. Shahin	+961 78 400 004	MansourShahin77@gmail.com

row(s) 1 - 10 of 10

INSERT INTO SUPPLIER VALUES(240000,'Jalal H. Zeinnedin','+961 76-815 217','JalalZeinnedin3@gmail.com');

INSERT INTO SUPPLIER VALUES(240001,'Rania H. Masri','+961 76-236 742','RaniaHMasri@gmail.com');

INSERT INTO SUPPLIER VALUES(240002,'Mira K. Rhayyem','+961 76 891 432','MiraRhayyem@gmail.com');

INSERT INTO SUPPLIER VALUES(240003,'Jadullah N. Abdullah','+961 3 444 232','jadullahAbdullah99@gmail.com');

INSERT INTO SUPPLIER VALUES(240004,'Sirena H. Hamieh','+961 70 421 673','SirenaHHamieh@gmail.com');

INSERT INTO SUPPLIER VALUES(240005,'Rein H. Monther','+961 27 830 899','reinMothEr01@gmail.com');

INSERT INTO SUPPLIER VALUES(240006,'Safi L. Bahri','+961 27 830 309','SafiBahri02@gmail.com');

INSERT INTO SUPPLIER VALUES(240007,'Jamil J. Jaber','+961 81 934 671','jamilJJaber@gmail.com');

INSERT INTO SUPPLIER VALUES(240008,'Lana H. Halabi','+961 81 233 356','Lana2Halabi@gmail.com');

INSERT INTO SUPPLIER VALUES(240009,'Mansour H. Shahin','+961 78 400 004','MansourShahin77@gmail.com');

## 07. Inserting Data into Equipment

EDIT	SERIAL_NUMBER	NAME	CATEGORY
	1234560	Stage	Instruments
	1234576	AC	Electronic
	1234657	Lights	Instruments
	1243567	Frames	Instruments
	1324567	Van	Vehicles
	1432567	Display cases	Instruments
	1456732	Display pillars	Instruments
	1765432	Laptop	Electronic
	1223456	Stairs	Instruments
	1122334	Electric Generators	Mechanical

```
INSERT INTO Equipment VALUES(1234560,'Stage','Instruments');
```

```
INSERT INTO Equipment VALUES(1234576,'AC','Electronic');
```

```
INSERT INTO Equipment VALUES(1234657,'Lights','Instruments');
```

```
INSERT INTO Equipment VALUES(1243567,'Frames','Instruments');
```

```
INSERT INTO Equipment VALUES(1324567,'Van','Vehicles');
```

```
INSERT INTO Equipment VALUES(1432567,'Display cases','Instruments');
```

```
INSERT INTO Equipment VALUES(1456732,'Display pillars','Instruments');
```

```
INSERT INTO Equipment VALUES(1765432,'Laptop','Electronic');
```

```
INSERT INTO Equipment VALUES(1223456,'Stairs','Instruments');
```

```
INSERT INTO Equipment VALUES(1122334,'Electric Generators','Mechanical');
```

## 08. Inserting Data into Partner

EDIT	ID	NAME	TYPE	CLASS	PERCENTAGE_SHARE
	200003	Jad H. AlHajj	Education	Liable	13
	200001	Hussam M. Karam	Technology	Not Liable	.1
	200002	Ibrahim M. Ramadan	Technology	Liable	11
	200004	Mira H. Halabi	Technology	Liable	10
	200005	Nour N. Nasser	Education	Limited Liability	3
	200006	Alaa W. Ikkawi	Education	Partially Liable	7
	200007	Alaa I. Deeb	Education	Limited Liability	4
	200008	Wael K. Arakji	Technology	Partially Liable	9
	200009	Sahar N. Abbas	Education	Decision taker	28
	200010	Ali H. Haidar	Education	Liable	15

INSERT INTO Partner VALUES(200001,'Hussam M. Karam','Technology','Not Liable',0.1);

INSERT INTO Partner VALUES(200002,'Ibrahim M. Ramadan','Technology','Liable',11);

INSERT INTO Partner VALUES(200003,'Jad H. AlHajj','Education','Liable',13);

INSERT INTO Partner VALUES(200004,'Mira H. Halabi','Technology','Liable',10);

INSERT INTO Partner VALUES(200005,'Nour N. Nasser','Education','Limited Liability',3);

INSERT INTO Partner VALUES(200006,'Alaa W. Ikkawi','Education','Partially Liable',7);

INSERT INTO Partner VALUES(200007,'Alaa I. Deeb','Education','Limited Liability',4);

INSERT INTO Partner VALUES(200008,'Wael K. Arakji','Technology','Partially Liable',9);

INSERT INTO Partner VALUES(200009,'Sahar N. Abbas','Education','Decision taker',28);

INSERT INTO Partner VALUES(200010,'Ali H. Haidar','Education','Liable',15);

## 09. Inserting Data into Sponsor

EDIT	ID	NAME
	230000	Ali H. Allouch
	230001	Elias J. Aoun
	230002	Mira J. Karam
	230003	Leo L. Chhade
	230004	Sara H. Fares
	230005	Lina I. Ibrahim
	230006	Reem H. Sweid
	230007	Safi M. Shdeed
	230008	Ayman W. Hussein
	230009	Nadine H. Mousally
row(s) 1 - 10 of 10		

INSERT INTO SPONSOR VALUES(230000,'Ali H. Allouch');

INSERT INTO SPONSOR VALUES(230001,'Elias J. Aoun');

INSERT INTO SPONSOR VALUES(230002,'Mira J. Karam');

INSERT INTO SPONSOR VALUES(230003,'Leo L. Chhade');

INSERT INTO SPONSOR VALUES(230004,'Sara H. Fares');

INSERT INTO SPONSOR VALUES(230005,'Lina I. Ibrahim');

INSERT INTO SPONSOR VALUES(230006,'Reem H. Sweid');

INSERT INTO SPONSOR VALUES(230007,'Safi M. Shdeed');

INSERT INTO SPONSOR VALUES(230008,'Ayman W. Hussein');

INSERT INTO SPONSOR VALUES(230009,'Nadine H. Mousally');

## 10. Inserting Data into Event

EDIT	ID	NAME	TYPE	DESCRIPTION	DATE_AND_TIME
	220000	Marjeyouns Event	Tour	Tour around the museum including all sections	022-1-25 9 am: 12 pm
	220001	Blatt Event	Examination	Examining sculptures	2022-2-25 8 am: 12 pm
	220002	Hamra Event	VIP party	DJ, music and beverages	2022-3-26 9 am: 12 pm
	220003	Raoushe Event	VIP party	DJ, music and beverages	2022-1-25 8 pm:11:59 am
	220004	Mrayje Event	Examination	Examining PAintings	2021-12-24 9 am: 12 pm
	220005	Napoli Event	School trip	Visiting all sections, and having lunch	2020-5-21 8 am: 3 pm
	220006	Zaghreb Event	School trip	Visiting all sections, and having lunch	2021-11-11 9 am: 3 pm
	220007	LAU Event	School trip	Visiting all sections, and having lunch	2021-11-1 9 am: 3 pm
	220008	AUB Event	School trip	Visiting all sections, and having lunch	2021-11-16 9 am: 3 pm
	220009	LU Event	Examination	Examining PAintings and sculptures	2021-11-11 9 am: 6 pm

INSERT INTO EVENT VALUES(220000 , ' Marjeyouns Event','Tour','Tour around the museum including all sections','022-1-25 9 am: 12 pm');

INSERT INTO EVENT VALUES(220001,'Blatt Event', 'Examination', 'Examining sculptures','2022-2-25 8 am: 12 pm');

INSERT INTO EVENT VALUES(220002, 'Hamra Event', 'VIP party','DJ, music and beverages','2022-3-26 9 am: 12 pm');

INSERT INTO EVENT VALUES(220003,'Raoushe Event', 'VIP party','DJ, music and beverages','2022-1-25 8 pm:11:59 am');

INSERT INTO EVENT VALUES(220004 , 'Mrayje Event' , 'Examination' , 'Examining PAintings','2021-12-24 9 am: 12 pm');

INSERT INTO EVENT VALUES(220005, 'Napoli Event' , 'School trip' , 'Visiting all sections, and having lunch','2020-5-21 8 am: 3 pm');

INSERT INTO EVENT VALUES(220006, 'Zaghreb Event' , 'School trip',' Visiting all sections, and having lunch','2021-11-11 9 am: 3 pm');

INSERT INTO EVENT VALUES(220007 , 'LAU Event' , 'School trip',' Visiting all sections, and having lunch','2021-11-1 9 am: 3 pm');

INSERT INTO EVENT VALUES(220008 , 'AUB Event' , 'School trip',' Visiting all sections, and having lunch','2021-11-16 9 am: 3 pm');

INSERT INTO EVENT VALUES(220009 , 'LU Event' , 'Examination' , 'Examining PAintings and sculptures ','2021-11-11 9 am: 6 pm');

## 11. Inserting Data into Department

EDIT	ID	NAME	LOCATION	FAX	EMAIL	EMP_ID
	1007	Collection Management	Trablos	961 1 867798	Collection.Department1001@gmail.com	2019181200
	1006	Education	Floor 9	961 1 867698	Education.Department1006@gmail.com	2019180200
	1005	Facilities	Kraytem	961 1 867598	Facilities.Department1005@ageless.museum	2022081200
	1004	Security	Ground Floor	961 1 867498	Security.Department1004@ageless.museum	2022080200
	1003	Development	Floor 8	961 1 867398	Development.Department1003@ageless.museum	2015050515
	1002	Marketing	Floor 7	961 1 867298	Marketing.Department1002@ageless.museum	2022010101
	1001	Curatorial	Jounieh	961 1 867198	Curatorial.Department1001@ageless.museum	2020030202

row(s) 1 - 7 of 7

INSERT INTO DEPARTMENT VALUES(1007,'Collection Management' , 'Trablos', '961 1 867798' ,  
 'Collection.Department1001@gmail.com', 2019181200 );

INSERT INTO DEPARTMENT VALUES(1006 , 'Education', 'Floor 9' , '961 1 867698' ,  
 'Education.Department1006@gmail.com', 2019180200);

INSERT INTO DEPARTMENT VALUES(1005 , 'Facilities', 'Kraytem', '961 1  
 867598', 'Facilities.Department1005@ageless.museum', 2022081200);

INSERT INTO DEPARTMENT VALUES(1004,'Security','Ground Floor', '961 1 867498',  
 'Security.Department1004@ageless.museum', 2022080200);

INSERT INTO DEPARTMENT VALUES(1003,'Development', 'Floor 8', '961 1  
 867398', 'Development.Department1003@ageless.museum', 2015050515);

INSERT INTO DEPARTMENT VALUES(1002,'Marketing' , 'Floor 7', '961 1 867298',  
 'Marketing.Department1002@ageless.museum', 2022010101);

INSERT INTO DEPARTMENT VALUES(1001,'Curatorial', 'Jounieh' , '961 1  
 867198', 'Curatorial.Department1001@ageless.museum', 2020030202);

## 12. Inserting Data into Employee

EDIT	ID	FIRST_NAME	MIDDLE_NAME	LAST_NAME	SEX	SSN	DATE_OF_BIRTH	DATE_OF_START	DATE_OF_TERMINATION	ROLE	COUNTRY	STREET	BUILDING
	2022060100	Mahmoud	S.	Joumaa	M	123456	06/01/2003	06/01/2022	-	Guide	Lebanon	Beirut	Lotus
	2019020500	Hussein	K.	Bakri	M	335353	08/15/1990	02/05/2019	-	Manager	Lebanon	Hamra	Tulip
	2020030200	Hussein	A.	Ahmad	M	2456812	03/05/2003	03/02/2020	-	Sales	Lebanon	Hamra	Rose
	2020030202	Sara	F	Faskredine	F	8528529	04/15/2003	03/02/2020	-	IT	Lebanon	Hamra	Lily
	2015050512	Caren	B.	Hamieh	F	235468	06/07/1990	05/05/2015	-	Sales	Lebanon	Saida	Grass
	2023060100	Bassam	K.	Hamieh	M	1231231	06/07/1990	05/05/2015	-	Guide	Lebanon	Tripoli	Rose
	2022010101	Diana	A.	Prince	F	5679802	07/08/2000	01/01/2022	-	Statistics	Lebanon	Beirut	Lotus
	2015050515	Ahmad	A.	Almaza	M	1231456	06/07/1990	05/05/2015	-	IT	Lebanon	Tripoli	Rose
	2018080200	Ahmad	B.	Almaza	M	1231456	07/08/2000	08/02/2018	08/02/2021	Sales	Lebanon	Tripoli	Thorn
	2022080200	Ahmad	A.	Noura	M	1231406	07/08/2003	08/02/2022	-	Sales	Lebanon	Tripoli	Thorn
	2022080300	Anjnad	A.	Noura	M	1331406	07/08/1980	08/02/2032	-	Sales	Lebanon	Tripoli	Thorn
	2022081200	Jana	A.	Noura	F	1331406	07/08/1974	08/02/2012	-	IT	Lebanon	Beirut	Lili
	2022081100	John	A.	Noura	M	1331406	07/08/1974	08/02/2012	-	IT	Lebanon	Beirut	Lili
	2019080200	Ahmad	B.	Alrashed	M	1331456	06/08/2000	08/02/2019	08/22/2019	Sales	Lebanon	Beirut	Thorn
	2019180200	Alaa	B.	Alrashed	F	1337456	11/08/2000	08/02/2019	-	Sales	Lebanon	Beirut	Thorn

INSERT INTO EMPLOYEE VALUES(2022060100, 11011101110100, 'Mahmoud', 'S.', 'Joumaa', 'M', 0123456, DATE '2003-06-01', DATE '2022-06-01', NULL, 'Guide', 'Lebanon', 'Beirut', 'Lotus', 'msj@ageless.museum', 8375);

INSERT INTO EMPLOYEE VALUES(2019020500, 11011101010100, 'Hussein', 'K.', 'Bakri', 'M', 0335353, DATE '1990-08-15', DATE '2019-02-05', NULL, 'Manager', 'Lebanon', 'Hamra', 'Tulip', 'hkb@ageless.museum', 6025);

INSERT INTO EMPLOYEE VALUES(2020030200, 11011101010000, 'Hussein', 'A.', 'Ahmad', 'M', 2456812, DATE '2003-03-05', DATE '2020-03-02', NULL, 'Sales', 'Lebanon', 'Hamra', 'Rose', 'haa@ageless.museum', 4051);

INSERT INTO EMPLOYEE VALUES(2020030200, 11011101010000, 'Hussein', 'A.', 'Ahmad', 'M', 2456812, DATE '2003-03-05', DATE '2020-03-02', NULL, 'Sales', 'Lebanon', 'Hamra', 'Rose', 'haa@ageless.museum', 4051);

INSERT INTO EMPLOYEE VALUES(2020030202, 01011101010011, 'Sara', 'F.', 'Faskredine', 'F', 8528529, DATE '2003-04-15', DATE '2020-03-02', NULL, 'IT', 'Lebanon', 'Hamra', 'Lily', 'sff@ageless.meuseum', 3232);

INSERT INTO EMPLOYEE VALUES(2015050512, 01011101110000, 'Caren', 'B.', 'Hamieh', 'F', 0235468, DATE '1990-06-07', DATE '2015-05-05', NULL, 'Sales', 'Lebanon', 'Saida', 'Grass', 'cbh@ageless.meuseum', 2051);

INSERT INTO EMPLOYEE VALUES(2023060100, 10011100110100, 'Bassam', 'K.', 'Hamieh', 'M', 1231231, DATE '1990-06-07', DATE '2015-05-05', NULL, 'Guide', 'Lebanon', 'Tripoli', 'Rose', 'bkh@ageless.meuseum', 3375);

INSERT INTO EMPLOYEE VALUES(2022010101, 10011101110101, 'Diana', 'A.', 'Prince', 'F', 5679802, DATE '2000-07-08', DATE '2022-01-01', NULL, 'Statistics', 'Lebanon', 'Beirut', 'Lotus', 'dap@ageless.meuseum', NULL);

INSERT INTO EMPLOYEE VALUES(2015050515, 11000101110101, 'Ahmad', 'A.', 'Almaza', 'M', 1231456, DATE '1990-06-07', DATE '2015-05-05', NULL, 'IT', 'Lebanon', 'Tripoli', 'Rose', 'aaa@ageless.meuseum', 1232);

INSERT INTO EMPLOYEE VALUES(2018080200, 10011101110100, 'Ahmad', 'B.', 'Almaza', 'M',

1231456,DATE '2000-07-08',DATE '2018-08-02',DATE '2021-08-02', 'Sales', 'Lebanon', 'Tripoli', 'Thorn', 'aba@ageless.meuseum', 9091);

INSERT INTO EMPLOYEE VALUES(2022080200, 11011101010101,'Ahmad', 'A.', 'Noura', 'M', 1231406,DATE '2003-07-08',DATE '2022-08-02',NULL, 'Sales', 'Lebanon', 'Tripoli', 'Thorn', 'aha@ageless.meuseum', 9091);

INSERT INTO EMPLOYEE VALUES(2022080300, 11001101010100,'Amjad', 'A.', 'Noura', 'M', 1331406,DATE '1980-07-08',DATE '2032-08-02',NULL, 'Sales', 'Lebanon', 'Tripoli', 'Thorn', 'ama@ageless.meuseum', 9091);

INSERT INTO EMPLOYEE VALUES(2022081200, 11011100010110,'Jana', 'A.', 'Noura', 'F', 1331406,DATE '1974-07-08',DATE '2012-08-02',NULL, 'IT', 'Lebanon', 'Beirut', 'Lili', 'ama@ageless.meuseum', 2012);

INSERT INTO EMPLOYEE VALUES(2022081100, 00011001010100, 'John ', 'A.', 'Noura', 'M', 1331406,DATE '1974-07-08',DATE '2012-08-02',NULL, 'IT', 'Lebanon', 'Beirut', 'Lili', 'ama@ageless.meuseum', 2012);

INSERT INTO EMPLOYEE VALUES(2019080200, 10011001010100  
, 'Ahmad', 'B.', 'Alrashed', 'M', 1331456, DATE '2000-06-08', DATE '2019-08-02', DATE '2019-08-22', 'Sales', 'Lebanon', 'Beirut', 'Thorn', 'abar@ageless.meuseum', 9091);

INSERT INTO EMPLOYEE VALUES(2019180200, 00111001010100,'Alaa', 'B.', 'Alrashed', 'F', 1337456, DATE '2000-11-08', DATE '2019-08-02', NULL, 'Sales', 'Lebanon', 'Beirut', 'Thorn', 'ALA@ageless.meuseum', 9091);

INSERT INTO EMPLOYEE VALUES(2019151000, 01111001010100,'Reem', 'J.', 'Almonzer', 'F', 6137456, DATE '1990-11-08', DATE '2018-08-02', NULL, 'IT', 'Lebanon', 'Beirut', 'Lili', 'ream@ageless.meuseum', 1105);

INSERT INTO EMPLOYEE VALUES(2019151200, 00011011010100,'Rose', 'B.', 'Almonzer', 'F', 9137456, DATE '1990-11-08', DATE '2018-08-02', NULL, 'IT', 'Lebanon', 'Beirut', 'Lili', 'ram@ageless.meuseum', 1105);

INSERT INTO EMPLOYEE VALUES(2019181200, 10011101011100,'Kinda', 'B.', 'Alrashed', 'F', 1137456, DATE '1990-11-08', DATE '2015-08-02', NULL, 'Sales', 'Lebanon', 'Beirut', 'Lili', 'Kind@ageless.meuseum', 1105);

### 13. Inserting Data into Feedback Program

EDIT	ENTRYID	SATISFACTION_RATING	VISITOR_ID
	11111	7	1122
	11112	4	6232
	11113	9	2882
	11114	9	3222
	11115	1	6222
	11116	8	2322
	11117	5	2122
	11118	8	2928
	11119	6	2356
	11110	9	2222

```
INSERT INTO FeedbackProgram VALUES(11111,7,1122);
```

```
INSERT INTO FeedbackProgram VALUES(11112,4,6223);
```

```
INSERT INTO FeedbackProgram VALUES(11113,9,2882);
```

```
INSERT INTO FeedbackProgram VALUES(11114,9,3222);
```

```
INSERT INTO FeedbackProgram VALUES(11115,1,6222);
```

```
INSERT INTO FeedbackProgram VALUES(11116,8,2322);
```

```
INSERT INTO FeedbackProgram VALUES(11117,5,2122);
```

```
INSERT INTO FeedbackProgram VALUES(11118,8,2928);
```

```
INSERT INTO FeedbackProgram VALUES(11119,6,2356);
```

```
INSERT INTO FeedbackProgram VALUES(11110,9,2222);
```

## 14. Inserting Data into Visitor

EDIT	ID	FIRST_NAME	MIDDLE_INITIAL	LAST_NAME	GIFT_POINTS	PHONE_NUMBER	EMAIL
	2222	Hussein	A.	Ammar	33	76815217	hsn@gmail.com
	2322	Ali	A.	Ammar	23	76816217	aliaa@gmail.com
	6222	Hussein	A.	Hammoud	41	76345217	HsnHamm@gmail.com
	3222	Kinda	A.	Deeb	0	76352879	kindadeeb@gmail.com
	2122	Kosay	A.	Mahmood	33	7687617	kossay@gmail.com
	2928	Bader	A.	Shantaf	133	76916817	baderr@gmail.com
	6232	Sahar	B.	Abbas	553	76415257	saharAA@gmail.com
	2882	Hussein	A.	Alhelou	33	76111217	hsnhelou@gmail.com
	1122	George	A.	Ammar	63	76115417	george@gmail.com
	2356	JAd	A.	Hajj	12	81236781	jado@gmail.com

row(s) 1 - 10 of 10

INSERT INTO VISITOR VALUES(2222,'Hussein','A.','Ammar',33,'76815217','hsn@gmail.com');

INSERT INTO VISITOR VALUES(2322,'Ali','A.','Ammar',23,'76816217','aliaa@gmail.com');

INSERT INTO VISITOR VALUES(6222,'Hussein','A.','Hammoud',41,'76345217','HsnHamm@gmail.com');

INSERT INTO VISITOR VALUES(3222,'Kinda','A.','Deeb',0,'76352879','kindadeeb@gmail.com');

INSERT INTO VISITOR VALUES(2122,'Kosay','A.','Mahmood',33,'7687617','kossay@gmail.com');

INSERT INTO VISITOR VALUES(2928,'Bader','A.','Shantaf',133,'76916817','baderr@gmail.com');

INSERT INTO VISITOR VALUES(6232,'Sahar','B.','Abbas',553,'76415257','saharAA@gmail.com');

INSERT INTO VISITOR VALUES(2882,'Hussein','A.','Alhelou',33,'76111217','hsnhelou@gmail.com');

INSERT INTO VISITOR VALUES(1122,'George','A.','Ammar',63,'76115417','george@gmail.com');

INSERT INTO VISITOR VALUES(2356,'JAd','A.','Hajj',12,'81236781','jado@gmail.com');

## 15. Inserting Data into Reward

EDIT	EMPLOYEEID	TYPE	AMOUNT	TIME	DATE_IMPLEMENTED	DETAILS
	2019020500	Bonus	150	-	02/12/2022	Working Extra hours
	2020030200	Vacation	-	20	04/10/2022	Loyalty for working for the museum for a long time
	2022010101	Bonus	200	-	08/09/2022	Great work maintaining the museum's pieces safety and cleanliness.
	2022080200	Bonus	100	-	06/07/2022	Helped in raising a good profile for the museum

row(s) 1 - 4 of 4

INSERT INTO REWARD VALUES(2019020500,'Bonus',150,NULL, DATE '2022-02-12','Working Extra hours');

INSERT INTO REWARD VALUES(2020030200,'Vacation',NULL,20, DATE '2022-04-10','Loyalty for working for the museum for a long time');

INSERT INTO REWARD VALUES(2022010101,'Bonus',200,NULL, DATE '2022-08-09','Great work maintaining the museum's pieces safety and cleanliness.');

INSERT INTO REWARD VALUES(2022080200,'Bonus',100,NULL, DATE '2022-06-07','Helped in raising a good profile for the museum');

## 16. Inserting Data into Medical Record

EDIT	EMPLOYEEID	DATE_INITIALIZED	BLOOD_TYPE	HEIGHT	WEIGHT	BLOOD_PRESSURE
	2022060100	10/11/2009	O+	175	75	110/70
	2019020500	04/10/2010	A-	183	84	105/85
	2020030202	06/08/2006	AB+	168	58	107/80
	2015050512	06/08/2007	O-	170	82	110/85
	2023060100	07/04/2012	B+	178	75	115/70
	2022010101	05/04/2007	B-	165	57	120/90
	2015050515	12/07/2012	O+	160	60	110/80
	2018080200	09/02/2010	A-	168	62	112/84
	2022080200	10/11/2008	A+	174	63	117/83
	2022080300	04/12/2010	O+	185	87	110/85

INSERT INTO MEDICAL\_RECORD VALUES(2022060100,DATE '2009-10-11','O+',175,75,'110/70');

INSERT INTO MEDICAL\_RECORD VALUES(2019020500,DATE '2010-04-10','A-',183,84,'105/85');

INSERT INTO MEDICAL\_RECORD VALUES(2020030202,DATE '2006-06-08','AB+',168,58,'107/80');

INSERT INTO MEDICAL\_RECORD VALUES(2015050512,DATE '2007-06-08','O-',170,82,'110/85');

INSERT INTO MEDICAL\_RECORD VALUES(2023060100,DATE '2012-07-04','B+',178,75,'115/70');

INSERT INTO MEDICAL\_RECORD VALUES(2022010101,DATE '2007-05-04','B-',165,57,'120/90');

INSERT INTO MEDICAL\_RECORD VALUES(2015050515,DATE '2012-12-07','O+',160,60,'110/80');

INSERT INTO MEDICAL\_RECORD VALUES(2018080200,DATE '2010-09-02','A-',168,62,'112/84');

INSERT INTO MEDICAL\_RECORD VALUES(2022080200,DATE '2008-10-11','A+',174,63,'117/83');

INSERT INTO MEDICAL\_RECORD VALUES(2022080300,DATE '2010-04-12','O+',185,87,'110/85');

## 17. Inserting Data into Police Record

EDIT	EMPLOYEEID	DATE_ISSUED	ARRESTS	CONVICTIONS	CRIMINAL_PROCEEDINGS
	2022060100	11/11/2019	False	False	False
	2019020500	12/10/2020	False	True	False
	2020030200	04/09/2018	True	False	False
	2020030202	05/04/2017	False	False	False
	2015050512	02/03/2020	True	False	True
	2015050512	07/06/2021	True	True	True
	2015050515	02/05/2020	False	False	False
	2018080200	12/10/2019	False	False	True
	2022010101	08/15/2020	True	False	False

INSERT INTO POLICE\_RECORD VALUES(2022060100, DATE '2019-11-11','False','False','False');

INSERT INTO POLICE\_RECORD VALUES(2019020500, DATE '2020-12-10','False','True','False');

INSERT INTO POLICE\_RECORD VALUES(2020030200, DATE '2019-03-8','False','True','True');

INSERT INTO POLICE\_RECORD VALUES(2020030202, DATE '2017-05-04','False','False','False');

INSERT INTO POLICE\_RECORD VALUES(2015050512, DATE '2020-02-03','True','False','True');

INSERT INTO POLICE\_RECORD VALUES(2015050512, DATE '2021-07-06','True','True','False');

INSERT INTO POLICE\_RECORD VALUES(2015050515, DATE '2020-02-05','False','False','False');

INSERT INTO POLICE\_RECORD VALUES(2018080200, DATE '2019-12-10','False','False','True');

INSERT INTO POLICE\_RECORD VALUES(2022010101, DATE '2020-08-15','True','False','False');

## 18. Inserting Data into Emergency Contact

EDIT	EMPLOYEEID	NAME	RELATIONSHIP	CONTACT_PHONE
	2020030200	John Cole	Uncle	70321864
	2020030200	Antony Ahmad	Father	78347961
	2020030200	Erna Ahmad	Mother	78342321
	2020030200	Lesley Ahmad	Sister	70423482
	2020030200	George Fakhredine	Brother	71473892
	2020030200	Geoffrey Fakhredine	Uncle	78645897
	2020030202	Lance Carter	Friend	70926739
	2020030202	Emory Arias	Friend	76329100
	2019020500	Amanda English	Aunt	78961238
	2019020500	Starla Bakri	Sister	76324901

```

INSERT INTO Emergency_Contact VALUES(2019020500,'Starla Bakri','Sister','76324901');

INSERT INTO Emergency_Contact VALUES(2020030200,'John Cole','Uncle','70321864');

INSERT INTO Emergency_Contact VALUES(2020030200,'Antony Ahmad','Father','78347961');

INSERT INTO Emergency_Contact VALUES(2020030200,'Erna Ahmad','Mother','78342321');

INSERT INTO Emergency_Contact VALUES(2020030200,'Lesley Ahmad','Sister','70423482');

INSERT INTO Emergency_Contact VALUES(2020030200,'George Fakhredine','Brother','71473892');

INSERT INTO Emergency_Contact VALUES(2020030200,'Geoffrey Fakhredine','Uncle','78645897');

INSERT INTO Emergency_Contact VALUES(2020030202,'Lance Carter','Friend','70926739');

INSERT INTO Emergency_Contact VALUES(2020030202,'Emory Arias','Friend','76329100');

INSERT INTO Emergency_Contact VALUES(2019020500,'Amanda English','Aunt','78961238');

```

## 19. Inserting Data into Dependent

EDIT	EMPLOYEEID	NAME	RELATIONSHIP_WITH_EMPLOYEE	DATE_OF_BIRTH	SEX	SSN
	2019020500	Lionel Bakri	Child	06/10/2009	M	456987321
	2019020500	Kaye Bakri	Child	05/09/2010	F	564328912
	2020030200	Mark Ahmad	Parent	04/08/1972	M	543897232
	2020030200	Ericka Ahmad	Parent	01/04/1975	F	324612873
	2015050515	Lina Almaza	Child	07/05/2008	F	623781823
	2015050515	Fern Almaza	Child	03/06/2012	M	873112092

row(s) 1 - 6 of 6

INSERT INTO Dependent VALUES(2019020500,'Lionel Bakri','Child', DATE '2009-06-10','M',456987321);

INSERT INTO Dependent VALUES(2019020500,'Kaye Bakri','Child', DATE '2010-05-09','F',564328912);

INSERT INTO Dependent VALUES(2020030200,'Mark Ahmad','Parent', DATE '1972-04-08','M',543897232);

INSERT INTO Dependent VALUES(2020030200,'Ericka Ahmad','Parent', DATE '1975-01-04','F',324612873);

INSERT INTO Dependent VALUES(2015050515,'Lina Almaza','Child', DATE '2008-07-05','F',623781823);

INSERT INTO Dependent VALUES(2015050515,'Fern Almaza','Child',DATE '2012-03-06','M',873112092);

## 20. Inserting Data into Creates

EDIT	ARTIST_ID	PIECE_ID
	1001	210000
	1001	210001
	1002	210002
	1002	210003
	1002	210005
	1010	210013
	1011	210015
	1045	210004
	1070	210006
	1073	210008
	1073	210009
	1120	210014
	1230	210007
row(s) 1 - 13 of 13		

```

INSERT INTO Creates VALUES(1001,210000);
INSERT INTO Creates VALUES(1001,210001);
INSERT INTO Creates VALUES(1002,210002);
INSERT INTO Creates VALUES(1002,210003);
INSERT INTO Creates VALUES(1045,210004);
INSERT INTO Creates VALUES(1002,210005);
INSERT INTO Creates VALUES(1070,210006);
INSERT INTO Creates VALUES(1230,210008);
INSERT INTO Creates VALUES(1230,210007);
INSERT INTO Creates VALUES(1073,210009);
INSERT INTO Creates VALUES(1010,210013);
INSERT INTO Creates VALUES(1120,210014);
INSERT INTO Creates VALUES(1011,210015);

```

## 21. Inserting Data into Is Overseen

EDIT	ID_PIECE	EMP_ID
	210000	2019020500
	210000	2022060100
	210001	2015050512
	210002	2022060100
	210003	2019020500
	210004	2015050512
	210005	2022060100
	210006	2019020500
	210007	2022060100
	210008	2015050512
	210009	2019020500
	210010	2015050512
	210011	2015050512

```
INSERT INTO isOverseen VALUES(210000,2022060100);
INSERT INTO isOverseen VALUES(210000,2019020500);
INSERT INTO isOverseen VALUES(210001,2015050512);
INSERT INTO isOverseen VALUES(210002,2022060100);
INSERT INTO isOverseen VALUES(210003,2019020500);
INSERT INTO isOverseen VALUES(210004,2015050512);
INSERT INTO isOverseen VALUES(210005,2022060100);
INSERT INTO isOverseen VALUES(210006,2019020500);
INSERT INTO isOverseen VALUES(210008,2015050512);
INSERT INTO isOverseen VALUES(210007,2022060100);
INSERT INTO isOverseen VALUES(210009,2019020500);
INSERT INTO isOverseen VALUES(210010,2015050512);
INSERT INTO isOverseen VALUES(210011,2015050512);
```

## 22. Inserting Data into Is Governed

EDIT	SECTION_ID	EMPLOYEE_ID
	1	2019020500
	2	2015050512
	3	2022060100
	4	2019020500
	5	2022060100
	6	2019020500
	7	2015050512
	8	2022060100
	9	2019020500
	10	2022060100

row(s) 1 - 10 of 10

```
INSERT INTO isGoverned VALUES(1,2019020500);
```

```
INSERT INTO isGoverned VALUES(2,2015050512);
```

```
INSERT INTO isGoverned VALUES(3,2022060100);
```

```
INSERT INTO isGoverned VALUES(4,2019020500);
```

```
INSERT INTO isGoverned VALUES(5,2022060100);
```

```
INSERT INTO isGoverned VALUES(6,2019020500);
```

```
INSERT INTO isGoverned VALUES(7,2015050512);
```

```
INSERT INTO isGoverned VALUES(8,2022060100);
```

```
INSERT INTO isGoverned VALUES(9,2019020500);
```

```
INSERT INTO isGoverned VALUES(10,2022060100);
```

### 23. Inserting Data into Is Managed

EDIT	DEP_ID	EMP_ID
	1001	2019020500
	1002	2019020500
	1003	2019020500
	1004	2019020500
	1005	2019020500
	1006	2019020500
	1007	2019020500

```
INSERT INTO isManaged VALUES(1001,2019020500);
```

```
INSERT INTO isManaged VALUES(1002,2019020500);
```

```
INSERT INTO isManaged VALUES(1003,2019020500);
```

```
INSERT INTO isManaged VALUES(1004,2019020500);
```

```
INSERT INTO isManaged VALUES(1005,2019020500);
```

```
INSERT INTO isManaged VALUES(1006,2019020500);
```

```
INSERT INTO isManaged VALUES(1007,2019020500);
```

## 24. Inserting Data into Visits

EDIT	VISITOR_ID	SECTIONID	TICKETID	TICKET_CATEGORY
	2222	10	1001	Adult
	2322	9	1002	Adult
	6222	8	1003	Adult
	3222	7	1023	Child
	2122	6	1032	Child
	2928	5	1043	Adult
	6232	4	1009	Child
	2882	3	1102	Adult
	1122	2	1004	Adult
	2356	1	1005	Child

row(s) 1 - 10 of 10

```
INSERT INTO visits VALUES(2222,10,1001,'Adult');
```

```
INSERT INTO visits VALUES(2322,9,1002,'Adult');
```

```
INSERT INTO visits VALUES(6222,8,1003,'Adult');
```

```
INSERT INTO visits VALUES(3222,7,1023,'Child');
```

```
INSERT INTO visits VALUES(2122,6,1032,'Child');
```

```
INSERT INTO visits VALUES(2928,5,1043,'Adult');
```

```
INSERT INTO visits VALUES(6232,4,1009,'Child');
```

```
INSERT INTO visits VALUES(2882,3,1102,'Adult');
```

```
INSERT INTO visits VALUES(1122,2,1004, 'Adult');
```

```
INSERT INTO visits VALUES(2356,1,1005,'Child');
```

## 25. Inserting Data into Donates

EDIT	SPONSOR_ID	DEPARTMENT_ID	AMOUNT
	230000	1001	900
	230001	1002	550
	230002	1003	2000
	230003	1004	350
	230004	1005	200
	230005	1006	40000
	230006	1007	2500
	230000	1004	200

```
INSERT INTO donates VALUES(230000,1001,900);
```

```
INSERT INTO donates VALUES(230001,1002,550);
```

```
INSERT INTO donates VALUES(230002,1003,2000);
```

```
INSERT INTO donates VALUES(230003,1004,350);
```

```
INSERT INTO donates VALUES(230004,1005,200);
```

```
INSERT INTO donates VALUES(230005,1006,40000);
```

```
INSERT INTO donates VALUES(230006,1007,2500);
```

```
INSERT INTO donates VALUES(230000,1004,200);
```

## 26. Inserting Data into Sponsors

EDIT	EVENT_ID	SPONSOR_ID	AMOUNT
	220000	230000	1100
	220001	230001	550
	220002	230002	2010
	220003	230003	3500
	220004	230004	2000
	220005	230005	3000
	220006	230006	250
	220007	230007	200
	220008	230008	810
	220009	230009	400
	220008	230009	500

row(s) 1 - 11 of 11

INSERT INTO sponsors VALUES(220000,230000,1100);

INSERT INTO sponsors VALUES(220001,230001,550);

INSERT INTO sponsors VALUES(220002,230002,2010);

INSERT INTO sponsors VALUES(220003,230003,3500);

INSERT INTO sponsors VALUES(220004,230004,2000);

INSERT INTO sponsors VALUES(220005,230005,3000);

INSERT INTO sponsors VALUES(220006,230006,250);

INSERT INTO sponsors VALUES(220007,230007,200);

INSERT INTO sponsors VALUES(220008,230008,810);

INSERT INTO sponsors VALUES(220009,230009,400);

INSERT INTO sponsors VALUES(220008,230009,500);

## 27. Inserting Data into Supervises

EDIT	EMPLOYEE_ID	SUPERVISOR_ID
	2015050512	2019020500
	2015050512	2022010101
	2019020500	2019020500
row(s) 1 - 3 of 3		

```
INSERT INTO supervises VALUES(2019020500,2019020500);
```

```
INSERT INTO supervises VALUES(2015050512,2019020500);
```

```
INSERT INTO supervises VALUES(2015050512, 2022010101);
```

## 28. Inserting Data into Employee Nationality

EDIT	EMPLOYEE_ID	NATIONALITY
	2015050512	Britain
	2015050512	Lebanese
	2015050515	Lebanese
	2018080200	Lebanese
	2019020500	Lebanese
	2020030200	Lebanese
	2020030202	Lebanese
	2022010101	British
	2022010101	Lebanese
	2022060100	Lebanese
row(s) 1 - 10 of 10		

INSERT INTO EMPNATIONALITY VALUES(2022060100,'Lebanese');

INSERT INTO EMPNATIONALITY VALUES(2019020500,'Lebanese');

INSERT INTO EMPNATIONALITY VALUES(2020030200,'Lebanese');

INSERT INTO EMPNATIONALITY VALUES(2020030202,'Lebanese');

INSERT INTO EMPNATIONALITY VALUES(2015050512,'Lebanese');

INSERT INTO EMPNATIONALITY VALUES(2015050512,'Britain');

INSERT INTO EMPNATIONALITY VALUES(2015050515,'Lebanese');

INSERT INTO EMPNATIONALITY VALUES(2018080200,'Lebanese');

INSERT INTO EMPNATIONALITY VALUES(2022010101,'Lebanese');

INSERT INTO EMPNATIONALITY VALUES(2022010101,'British');

## 29. Inserting Data into Interests

EDIT	VISITOR_ID	INTERESTS
	1122	Geography
	2122	Paintings
	2222	Historical
	2322	Paintings
	2356	Paintings
	2882	Animals
	2928	Sculptures
	3222	Modern
	6222	Sculptures
	6232	Textiles
row(s) 1 - 10 of 10		

```
INSERT INTO Interests VALUES(2222,'Historical');
```

```
INSERT INTO Interests VALUES(2322,'Paintings');
```

```
INSERT INTO Interests VALUES(6222,'Sculptures');
```

```
INSERT INTO Interests VALUES(3222,'Modern');
```

```
INSERT INTO Interests VALUES(2122,'Paintings');
```

```
INSERT INTO Interests VALUES(2928,'Sculptures');
```

```
INSERT INTO Interests VALUES(6232,'Textiles');
```

```
INSERT INTO Interests VALUES(2882,'Animals');
```

```
INSERT INTO Interests VALUES(1122,'Geography');
```

```
INSERT INTO Interests VALUES(2356,'Paintings');
```

### 30. Inserting Data into Comments

EDIT	FEEDBACKENTRYID	COMMENTS
	11110	Loved visiting the science part of the museum
	11111	Enjoyed in the most part the unique sculptures
	11112	The museum is big and hard to manage
	11113	Gained a lot of valuable information
	11114	The art pieces were beautiful
	11115	Appreciate the way the employees help
	11116	Enjoyed seeing the sculptures
	11117	Loved seeing the mummified animals
	11118	The ancient textiles were very interesting
	11119	Gained a lot information that reinforced my education

INSERT INTO Comments VALUES(11111,'Enjoyed in the most part the unique sculptures');

INSERT INTO Comments VALUES(11112,'The museum is big and hard to manage');

INSERT INTO Comments VALUES(11113,'Gained a lot of valuable information');

INSERT INTO Comments VALUES(11114,'The art pieces were beautiful');

INSERT INTO Comments VALUES(11115,'Appreciate the way the employees help');

INSERT INTO Comments VALUES(11116,'Enjoyed seeing the sculptures');

INSERT INTO Comments VALUES(11117,'Loved seeing the mummified animals');

INSERT INTO Comments VALUES(11118,'The ancient textiles were very interesting');

INSERT INTO Comments VALUES(11119,'Gained a lot information that reinforced my education');

INSERT INTO Comments VALUES(11110,'Loved visiting the science part of the museum');

### 31. Inserting Data into Tags

EDIT	PIECE_ID	TAG
	210000	Leonardo
	210000	Mona Lisa
	210001	Starry
	210002	Picasso
	210003	Kiss
	210004	David
	210005	Enviest Lion
	210006	Smallest dog
	210007	Biggest bee
	210008	mommiest
	210009	microscope
	210010	Babbage
	210011	prince
	210011	scarf

```

INSERT INTO TAGS VALUES(210000,'Mona Lisa');
INSERT INTO TAGS VALUES(210000,'Leonardo');
INSERT INTO TAGS VALUES(210001,'Starry');
INSERT INTO TAGS VALUES(210002,'Picasso');
INSERT INTO TAGS VALUES(210003,'Kiss');
INSERT INTO TAGS VALUES(210004,'David');
INSERT INTO TAGS VALUES(210005,'Enviest Lion');
INSERT INTO TAGS VALUES(210006,'Smallest dog');
INSERT INTO TAGS VALUES(210007,'Biggest bee');
INSERT INTO TAGS VALUES(210008,'mommiest');
INSERT INTO TAGS VALUES(210009,'microscope');
INSERT INTO TAGS VALUES(210010,'Babbage');
INSERT INTO TAGS VALUES(210011,'scarf');
INSERT INTO TAGS VALUES(210011,'prince');

```

## 32. Inserting Data into Cultures

EDIT	PIECE_ID	CULTURE
	210000	Italian
	210001	French
	210002	Spanish
	210003	Austrian
	210004	Italian
	210005	Lebanese
	210005	Syrian
	210006	Lebanese
	210007	Lebanese
	210008	Lebanese
	210009	Belgian
	210010	British
	210011	Lebanese
	210011	Royal families culture

```

INSERT INTO Cultures VALUES(210000,'Italian');
INSERT INTO Cultures VALUES(210001,'French');
INSERT INTO Cultures VALUES(210002,'Spanish');
INSERT INTO Cultures VALUES(210003,'Austrian');
INSERT INTO Cultures VALUES(210004,'Italian');
INSERT INTO Cultures VALUES(210005,'Lebanese');
INSERT INTO Cultures VALUES(210005,'Syrian');
INSERT INTO Cultures VALUES(210006,'Lebanese');
INSERT INTO Cultures VALUES(210007,'Lebanese');
INSERT INTO Cultures VALUES(210008,'Lebanese');
INSERT INTO Cultures VALUES(210009,'Belgian');
INSERT INTO Cultures VALUES(210010,'British');
INSERT INTO Cultures VALUES(210011,'Lebanese');
INSERT INTO Cultures VALUES(210011,'Royal families culture');

```

### 33. Inserting Data into Extension

EDIT	DEPT_ID	EXTENSION
	1001	1225
	1001	1226
	1002	1231
	1002	1233
	1003	1227
	1004	1221
	1004	1229
	1005	1555
	1006	1224
	1007	1223
	1007	1240

```
INSERT INTO Dept_Extension VALUES(1001,1225);
```

```
INSERT INTO Dept_Extension VALUES(1001,1226);
```

```
INSERT INTO Dept_Extension VALUES(1003,1227);
```

```
INSERT INTO Dept_Extension VALUES(1004,1229);
```

```
INSERT INTO Dept_Extension VALUES(1004,1221);
```

```
INSERT INTO Dept_Extension VALUES(1006,1224);
```

```
INSERT INTO Dept_Extension VALUES(1007,1223);
```

```
INSERT INTO Dept_Extension VALUES(1007,1240);
```

```
INSERT INTO Dept_Extension VALUES(1002,1231);
```

```
INSERT INTO Dept_Extension VALUES(1002,1233);
```

```
INSERT INTO Dept_Extension VALUES(1005,1555);
```

### 34. Inserting Data into Provides

EDIT	PROVISIONERID	MEDIATORID	PIECEID	DATE_OF_PURCHASE	MEDIATOR_FEES	PRICE	TEMPORARY	COUNTRY_OF_ORIGIN
	15432	10001	210000	11/11/2021	50	500	-	France
	13452	10002	210001	10/9/2021	30	550	-	Italy
	11223	11005	210002	04/03/2019	25	600	04/03/2020	Spain
	10012	11002	210003	10/02/2018	40	500	-	Britain
	12096	10015	210004	09/12/2020	45	650	-	Italy
	51343	11100	210005	30/06/2021	30	700	30/06/2022	Spain
	32451	10190	210006	13/07/2021	55	350	15/08/2022	Japan
	65421	10190	210007	14/09/2019	60	540	-	China
	43241	11180	210008	18/10/2020	100	750	-	France
	61239	10009	210009	26/05/2019	250	400	26/05/2021	Australia
	15432	11100	210010	30/06/2020	120	450	-	France
	13452	10190	210011	26/04/2019	255	600	-	Italy
	11223	11180	210012	22/03/2021	50	350	22/03/2022	China
	51343	10001	210013	20/08/2019	75	455	-	Japan
	32451	10002	210014	12/09/2020	100	250	12/09/2022	Britain

row(s) 1 - 15 of 16

```

INSERT INTO provides VALUES(15432,10001,210000,'11/11/2021',50,500,NULL,'France');

INSERT INTO provides VALUES(13452,10002,210001,'10/9/2021',30,550,NULL,'Italy');

INSERT INTO provides VALUES(11223,11005,210002,'04/03/2019',25,600,'04/03/2020','Spain');

INSERT INTO provides VALUES(10012,11002,210003,'10/02/2018',40,500,NULL,'Britain');

INSERT INTO provides VALUES(12096,10015,210004,'09/12/2020',45,650,NULL,'Italy');

INSERT INTO provides VALUES(51343,11100,210005,'30/06/2021',30,700,'30/06/2022','Spain');

INSERT INTO provides VALUES(32451,10190,210006,'13/07/2021',55,350,'15/08/2022','Japan');

INSERT INTO provides VALUES(65421,10190,210007,'14/09/2019',60,540,NULL,'China');

INSERT INTO provides VALUES(43241,11180,210008,'18/10/2020',100,750,NULL,'France');

INSERT INTO provides VALUES(61239,10009,210009,'26/05/2019',250,400,'26/05/2021','Australia');

INSERT INTO provides VALUES(15432,11100,210010,'30/06/2020',120,450,NULL,'France');

INSERT INTO provides VALUES(13452,10190,210011,'26/04/2019',255,600,NULL,'Italy');

INSERT INTO provides VALUES(11223,11180,210012,'22/03/2021',50,350,'22/03/2022','China');

INSERT INTO provides VALUES(51343,10001,210013,'20/08/2019',75,455,NULL,'Japan');

INSERT INTO provides VALUES(32451,10002,210014,'12/09/2020',100,250,'12/09/2022','Britain');

INSERT INTO provides VALUES(32451,11180,210015,'04/02/2018',75,545,NULL,'Spain');

```

### 35. Inserting Data into Supplies

EDIT	SUPPLIER_ID	EQUIPMENT_SR	DEPT_ID
	240000	1234560	1001
	240001	1234576	1002
	240002	1234657	1003
	240003	1243567	1004
	240004	1324567	1005
	240005	1432567	1006
	240006	1456732	1007
	240007	1223456	1002
	240007	1765432	1001
	240008	1122334	1006
	240008	1223456	1001
	240008	1432567	1005
	240009	1234576	1004
row(s) 1 - 13 of 13			

```

INSERT INTO supplies VALUES(240000,1234560,1001);
INSERT INTO supplies VALUES(240001,1234576,1002);
INSERT INTO supplies VALUES(240002,1234657,1003);
INSERT INTO supplies VALUES(240003,1243567,1004);
INSERT INTO supplies VALUES(240004,1324567,1005);
INSERT INTO supplies VALUES(240005,1432567,1006);
INSERT INTO supplies VALUES(240006,1456732,1007);
INSERT INTO supplies VALUES(240007,1765432,1001);
INSERT INTO supplies VALUES(240007,1223456,1002);
INSERT INTO supplies VALUES(240008,1122334,1006);
INSERT INTO supplies VALUES(240008,1223456,1001);
INSERT INTO supplies VALUES(240008,1432567,1005);
INSERT INTO supplies VALUES(240009,1234576,1004);

```

### 36. Inserting Data into Is Hosted

EDIT	EVENT_ID	PARTNER_ID	DEPARTMENT_ID	SECTION_ID
	220000	200001	1001	1
	220001	200002	1002	2
	220002	200003	1003	3
	220003	200004	1004	4
	220004	200005	1005	5
	220005	200006	1006	6
	220006	200007	1007	7
	220007	200001	1005	1
	220007	200008	1001	8
	220007	200009	1004	9
	220008	200001	1002	7
	220008	200010	1003	10
	220009	200003	1004	4
	220009	200009	1003	3

```

INSERT INTO isHosted VALUES(220000,200001,1001,1);
INSERT INTO isHosted VALUES(220001,200002,1002,2);
INSERT INTO isHosted VALUES(220002,200003,1003,3);
INSERT INTO isHosted VALUES(220003,200004,1004,4);
INSERT INTO isHosted VALUES(220004,200005,1005,5);
INSERT INTO isHosted VALUES(220005,200006,1006,6);
INSERT INTO isHosted VALUES(220006,200007,1007,7);
INSERT INTO isHosted VALUES(220007,200008,1001,8);
INSERT INTO isHosted VALUES(220007,200009,1004,9);
INSERT INTO isHosted VALUES(220008,200010,1003,10);
INSERT INTO isHosted VALUES(220008,200001,1002,7);
INSERT INTO isHosted VALUES(220009,200009,1003,3);
INSERT INTO isHosted VALUES(220009,200003,1004,4);
INSERT INTO isHosted VALUES(220007,200001,1005,1);

```

## XIII. Querying the Database

### 01. Education and Animals

As per the request of a teacher in charge of a school trip today, retrieve the names of the artists of all the pieces in the Stuffed Animals section.

NAME	BIOGRAPHY	NATIONALITY
Malek Abboud	Brave and strong. Born in Syria and lived in Lebanon	Syrian
Malek Al Adghal	Brave and strong Lebanese man. Well known for the biggest bee	Lebanese

```
SELECT NAME,Biography,Nationality  
FROM ARTIST  
WHERE ID IN  
(SELECT ARTIST_ID FROM CREATES  
WHERE PIECE_ID IN  
(SELECT ID  
FROM PIECE  
WHERE Section_ID IN(SELECT Section.ID  
FROM SECTION  
WHERE Section.NAME='Stuffed animals')))
```

## 02. An Important Impression

A very important painting called “Impression” is being shipped in to the Paintings section today. Get the names of all employees who either oversee a piece in that section or govern the Paintings section.

FIRST_NAME	MIDDLE_NAME	LAST_NAME	ROLE
Mahmoud	S.	Joumaa	Guide
Hussein	K.	Bakri	Manager
Caren	B.	Hamieh	Sales

Select First\_Name,Middle\_Name, Last,Name, Role

From Employee where ID IN( Select Employee\_ID

From ISGOVERNED

Where Section\_ID IN(Select ID from Section where Name = ‘Paintings’)OR ID in(Select Emp\_ID

From ISOVERSEEN WHERE ID\_PIECE IN (Select ID from PIECE

where Section\_ID IN(Select ID from Section where Name = ‘Paintings’)))

### 03. An Average Raise

After receiving more funds from a new sponsor, the museum's directory board has decided to provide a raise to all employees who earn less than the average salary. The raise is computed as follows: Increase an employee's salary by  $x\%$  where  $x$  is the number of dependents an employee has.

Before querying:

NAME	LAST_NAME	SEX	SSN	DATE_OF_BIRTH	DATE_OF_START	DATE_OF_TERMINATION	ROLE	COUNTRY	STREET	BUILDING	EMAIL	DEPARTMENTAL_EXTENSION	IMAGE	SALARY
Joumaa	M	123456	06/01/2003	06/01/2022	-		Guide	Lebanon	Beirut	Lotus	msj@ageless.museum	8375	1101101010100	16000
Bakri	M	335353	08/15/1990	02/05/2019	-		Manager	Lebanon	Hamra	Tulip	hkb@ageless.museum	6025	1101101010100	24000
Ahmad	M	2456812	03/05/2003	03/02/2020	-		Sales	Lebanon	Hamra	Rose	haa@ageless.museum	4051	1101101010000	12800
Faskredine	F	8528529	04/15/2003	03/02/2020	-		IT	Lebanon	Hamra	Lily	sff@ageless.museum	3232	1011101010011	123455
Hamieh	F	235468	06/07/1990	05/05/2015	-		Sales	Lebanon	Saida	Grass	cbh@ageless.museum	2051	1011101100000	10000
Hamieh	M	1231231	06/07/1990	05/05/2015	-		Guide	Lebanon	Tripoli	Rose	bkh@ageless.museum	3375	1001100110100	19200
Prince	F	5679802	07/08/2000	01/01/2022	-		Statistics	Lebanon	Beirut	Lotus	dap@ageless.museum	-	1001101110101	19200
Almaza	M	1231456	06/07/1990	05/05/2015	-		IT	Lebanon	Tripoli	Rose	aaa@ageless.museum	1232	11000101110101	16000
Almaza	M	1231458	07/08/2000	08/02/2018	08/02/2021		Sales	Lebanon	Tripoli	Thorn	aba@ageless.museum	9091	1001101110100	19200
Noura	M	1231458	07/08/2003	08/02/2022	-		Sales	Lebanon	Tripoli	Thorn	aha@ageless.museum	9091	1101101010101	14400
Noura	M	1331406	07/08/1980	08/02/2032	-		Sales	Lebanon	Tripoli	Thorn	ama@ageless.museum	9091	11001101010100	12800
Noura	F	1331406	07/08/1974	08/02/2012	-		IT	Lebanon	Beirut	Lili	ama@ageless.museum	2012	1101100010110	16000
Noura	M	1331406	07/08/1974	08/02/2012	-		IT	Lebanon	Beirut	Lili	ama@ageless.museum	2012	11001010100	2080
Alrashed	M	1331458	06/08/2000	08/02/2019	08/22/2019		Sales	Lebanon	Beirut	Thorn	abar@ageless.museum	9091	10011001010100	20800
Alrashed	F	1337456	11/08/2000	08/02/2019	-		Sales	Lebanon	Beirut	Thorn	ALA@ageless.museum	9091	111001010100	123455

row(s) 1 - 15 of 18

NAME	LAST_NAME	SEX	SSN	DATE_OF_BIRTH	DATE_OF_START	DATE_OF_TERMINATION	ROLE	COUNTRY	STREET	BUILDING	EMAIL	DEPARTMENTAL_EXTENSION	IMAGE	SALARY
Almonzer	F	6137458	11/08/1990	08/02/2018	-		IT	Lebanon	Beirut	Lili	ream@ageless.museum	1105	111001010100	123455
Almonzer	F	9137456	11/08/1990	08/02/2018	-		IT	Lebanon	Beirut	Lili	ram@ageless.museum	1105	11011010100	123455
Alrashed	F	1137456	11/08/1990	08/02/2015	-		Sales	Lebanon	Beirut	Lili	Kind@ageless.museum	1105	1001101011100	123455

(4) row(s) 16 - 18 of 18

After querying:

NAME	LAST_NAME	SEX	SSN	DATE_OF_BIRTH	DATE_OF_START	DATE_OF_TERMINATION	ROLE	COUNTRY	STREET	BUILDING	EMAIL	DEPARTMENTAL_EXTENSION	IMAGE	SALARY
Joumaa	M	123456	06/01/2003	06/01/2022	-		Guide	Lebanon	Beirut	Lotus	msj@ageless.museum	8375	1101101110100	25600
Bakri	M	335353	08/15/1990	02/05/2019	-		Manager	Lebanon	Hamra	Tulip	hkb@ageless.museum	6025	1101101010100	38400
Ahmad	M	2456812	03/05/2003	03/02/2020	-		Sales	Lebanon	Hamra	Rose	haa@ageless.museum	4051	1101101010000	20480
Faskredine	F	8528529	04/15/2003	03/02/2020	-		IT	Lebanon	Hamra	Lily	sff@ageless.museum	3232	1011101010011	123455
Hamieh	F	235468	06/07/1990	05/05/2015	-		Sales	Lebanon	Saida	Grass	cbh@ageless.museum	2051	1011101100000	16000
Hamieh	M	1231231	06/07/1990	05/05/2015	-		Guide	Lebanon	Tripoli	Rose	bkh@ageless.museum	3375	1001100110100	30720
Prince	F	5679802	07/08/2000	01/01/2022	-		Statistics	Lebanon	Beirut	Lotus	dap@ageless.museum	-	1001101110101	30720
Almaza	M	1231456	06/07/1990	05/05/2015	-		IT	Lebanon	Tripoli	Rose	aaa@ageless.museum	1232	11000101110101	25600
Almaza	M	1231458	07/08/2000	08/02/2018	08/02/2021		Sales	Lebanon	Tripoli	Thorn	aba@ageless.museum	9091	1001101110100	30720
Noura	M	1231406	07/08/2003	08/02/2022	-		Sales	Lebanon	Tripoli	Thorn	aha@ageless.museum	9091	1101101010101	23040
Noura	M	1331406	07/08/1980	08/02/2032	-		Sales	Lebanon	Tripoli	Thorn	ama@ageless.museum	9091	110011010100	20480
Noura	F	1331406	07/08/1974	08/02/2012	-		IT	Lebanon	Beirut	Lili	ama@ageless.museum	2012	1101100010110	25600
Noura	M	1331406	07/08/1974	08/02/2012	-		IT	Lebanon	Beirut	Lili	ama@ageless.museum	2012	11001010100	3328
Alrashed	M	1331456	06/08/2000	08/02/2019	08/22/2019		Sales	Lebanon	Beirut	Thorn	abar@ageless.museum	9091	10011001010100	33280
Alrashed	F	1337456	11/08/2000	08/02/2019	-		Sales	Lebanon	Beirut	Thorn	ALA@ageless.museum	9091	110001010100	123455

NAME	LAST_NAME	SEX	SSN	DATE_OF_BIRTH	DATE_OF_START	DATE_OF_TERMINATION	ROLE	COUNTRY	STREET	BUILDING	EMAIL	DEPARTMENTAL_EXTENSION	IMAGE	SALARY
Almonzer	F	6137458	11/08/1990	08/02/2018	-		IT	Lebanon	Beirut	Lili	ream@ageless.museum	1105	111001010100	123455
Almonzer	F	9137456	11/08/1990	08/02/2018	-		IT	Lebanon	Beirut	Lili	ram@ageless.museum	1105	11011010100	123455
Alrashed	F	1137456	11/08/1990	08/02/2015	-		Sales	Lebanon	Beirut	Lili	Kind@ageless.museum	1105	1001101011100	123455

(4) row(s) 16 - 18 of 18

Update Employee

Set Salary = Salary \* (1 + 0.1 \* (

Select Count(\*)

From Employee, Dependent

Where Employee.ID = Dependent.EmployeeID))

Where ID IN (Select ID From Employee Where Salary < (Select avg(Salary) From Employee))

#### 04. Bonus Promotion

The museum has received a donation after its success hosting an event. For managing such a successful event, employees Mahmoud S. Joumaa and Bassam K. Hamieh are now promoted to managers of Education and Security departments respectively. They do not change roles as they will still be taking on these responsibilities. Their salary witnesses a 1.75 increase.

Before querying:

EDIT	DEP_ID	EMP_ID
	1001	2019020500
	1002	2019020500
	1003	2019020500
	1004	2019020500
	1005	2019020500
	1006	2019020500
	1007	2019020500
row(s) 1 - 7 of 7		

After querying:

EDIT	DEP_ID	EMP_ID
	1001	2019020500
	1002	2019020500
	1003	2019020500
	1004	2023060100
	1005	2019020500
	1006	2022060100
	1007	2019020500
row(s) 1 - 7 of 7		

Update isManaged

Set EMP\_ID =

CASE

WHEN DEP\_ID = (Select ID

From Department

Where Name = 'Education')

THEN (Select ID

From Employee

Where (First\_Name = 'Mahmoud' and Middle\_Name = 'S.' and Last\_Name = 'Joumaa'))

WHEN DEP\_ID = (Select ID

From Department

Where Name = 'Security')

THEN (Select ID From Employee

Where (First\_Name = 'Bassam' and Middle\_Name = 'K.' and Last\_Name = 'Hamieh'))

ELSE (Select ID FROM Employee where (Role = 'Manager'))

END

## 05. The Artist with Most Paintings

For statistical inquiries, the sales and marketing departments have requested to view the Name and Biography of the artist with the most number of paintings in the museum.

NAME	BIOGRAPHY
Gustav Klimt	An Austrian Painter. He founded the Vienna Sezession school of painting.

Select Name, Biography

From Artist

Where ID IN

(Select Artist\_ID

From Creates

GROUP BY Artist\_ID

HAVING Count (\*) IN

(Select MAX(pCount)

From (Select Count(\*) pCount From Creates GROUP BY Artist\_ID)))

## 06. Coordination Meeting

At the end of the current quarter, the employees who work in sections with satisfaction rating less than 5 are to be called for a meeting. Retreive the Names of these employees.

FIRST_NAME	MIDDLE_NAME	LAST_NAME
Mahmoud	S.	Joumaa
Hussein	K.	Bakri

Select First\_Name, Middle\_Name, Last\_Name

From Employee

WHERE ID IN (Select ID From Employee where ID IN

(Select Employee\_ID From ISGOVERNED Where Section\_ID IN

(SELECT SECTIONID FROM VISITS WHERE VISITOR\_ID IN

(SELECT Visitor\_ID From FEEDBACKPROGRAM Where Satisfaction\_Rating < 5))) OR ID IN

(Select EMP\_ID From ISOVERSEEN WHERE ID\_PIECE IN

(Select ID From Piece where Section\_ID IN

(SELECT SECTIONID FROM VISITS WHERE VISITOR\_ID IN

(SELECT Visitor\_ID From FEEDBACKPROGRAM Where Satisfaction\_Rating < 5))))

## 07. Prison Break

New policies and regulations have been adopted by the mayor of this city which entail that all employees with at least one previous offense must now receive 87.5% of their current salaries.

Before querying:

NAME	LAST_NAME	SEX	SSN	DATE_OF_BIRTH	DATE_OF_START	DATE_OF_TERMINATION	ROLE	COUNTRY	STREET	BUILDING	EMAIL	DEPARTMENTAL_EXTENSION	IMAGE	SALARY
Joumaa	M	123456	06/01/2003	06/01/2022	-		Guide	Lebanon	Beirut	Lotus	msj@ageless.museum	8375	11011101110100	25600
Bakri	M	335353	08/15/1990	02/05/2019	-		Manager	Lebanon	Hamra	Tulip	hbk@ageless.museum	6025	11011101010100	33600
Ahmad	M	2456812	03/05/2003	03/02/2020	-		Sales	Lebanon	Hamra	Rose	haa@ageless.museum	4051	11011101010000	17920
Faskredine	F	8528529	04/15/2003	03/02/2020	-		IT	Lebanon	Hamra	Lily	sff@ageless.museum	3232	1011101010011	123455
Hamieh	F	235466	06/07/1990	05/05/2015	-		Sales	Lebanon	Saida	Grass	cb@ageless.museum	2051	1011101100000	14000
Hamieh	M	1231231	06/07/1990	05/05/2015	-		Guide	Lebanon	Tripoli	Rose	bkk@ageless.museum	3375	100111001110100	30720
Prince	F	5679802	07/08/2000	01/01/2022	-		Statistics	Lebanon	Beirut	Lotus	dap@ageless.museum	-	100111001110101	28880
Almaza	M	1231456	06/07/1990	05/05/2015	-		IT	Lebanon	Tripoli	Rose	aaa@ageless.museum	1232	11000101110101	25600
Almaza	M	1231456	07/08/2000	08/02/2018	08/02/2021		Sales	Lebanon	Tripoli	Thom	abb@ageless.museum	9091	10011101110100	26880
Noura	M	1231406	07/08/2003	08/02/2022	-		Sales	Lebanon	Tripoli	Thom	ana@ageless.museum	9091	1101101010101	23040
Noura	M	1331406	07/08/1990	08/02/2002	-		Sales	Lebanon	Tripoli	Thom	ama@ageless.museum	9091	11001101010100	20480
Noura	F	1331406	07/08/1974	08/02/2012	-		IT	Lebanon	Beirut	Lili	ama@ageless.museum	2012	11011100010110	25600
Noura	M	1331406	07/08/1974	08/02/2012	-		IT	Lebanon	Beirut	Lili	ama@ageless.museum	2012	1100101010100	3328
Airashed	M	1331456	06/08/2000	08/02/2019	08/22/2019		Sales	Lebanon	Beirut	Thom	abar@ageless.museum	9091	10011001010100	33280
Airashed	F	1337456	11/08/2000	08/02/2019	-		Sales	Lebanon	Beirut	Thom	ALA@ageless.museum	9091	111001010100	123455

row(s) 1 - 15 of 18 >

After quering:

NAME	LAST_NAME	SEX	SSN	DATE_OF_BIRTH	DATE_OF_START	DATE_OF_TERMINATION	ROLE	COUNTRY	STREET	BUILDING	EMAIL	DEPARTMENTAL_EXTENSION	IMAGE	SALARY
Almonzer	F	6137456	11/08/1990	08/02/2018	-		IT	Lebanon	Beirut	Lili	ream@ageless.museum	1105	111001010100	123455
Almonzer	F	9137456	11/08/1990	08/02/2018	-		IT	Lebanon	Beirut	Lili	ram@ageless.museum	1105	11011010100	123455
Airashed	F	1137456	11/08/1990	08/02/2015	-		Sales	Lebanon	Beirut	Lili	Kind@ageless.museum	1105	10011101011100	123455

< row(s) 16 - 18 of 18 >

Update Employee

Set Salary = Salary \* 0.875

Where ID in (Select ID

From Police\_Record, Employee

Where Employee.ID = EmployeeID and (Arrests = 'True' or Convictions = 'True' or Criminal\_Proceedings = 'True'))

## 08. Grand Theft Sneak

A theft has occurred in the museum. The police have already identified that the mediator and provisioner involved are “Brian Young” and “Brian Miller & Co” respectively. However, they require help identifying all the employees who could have helped these two parties sneak the art piece outside of the museum.

FIRST_NAME	MIDDLE_NAME	LAST_NAME
Hussein	K.	Bakri
Mahmoud	S.	Joumaa

Select First\_Name, Middle\_Name, Last\_Name

From EMPLOYEE

Where ID IN (

Select EMP\_ID

From ISOVERSEEN

WHERE ID\_PIECE IN (

Select PieceID

From Provides

Where MEDIATORID = (Select ID From Mediator Where Name = 'Brian Young') and ProvisionerID = (Select ID From Provisioner Where Name = 'Brian Miller & Co'))

## XIV. Normalization Up to Boyce-Codd Normal Form

All relations must undergo the normalization process after being created to assure satisfactory standards. The normalization process ranges from the first normal form to Boyce-Codd Normal Form (more commonly known as Normal Form 3.5). If a relation does not pass any of these normal forms, it is decomposed (normalized) into several relations.

### 01. First Normal Form

This normal form disallows multivalued attributes, composite attributes, or nested relations. All values of each attribute must be single and atomic to pass the first Normal Form. It is important to note that all relations pass the first Normal Form (1NF) naturally as the design of the relational model disallows, by definition, multivalued attributes, composite attributes, and nested relations.

### 02. Second Normal Form

A relation is said to be in second Normal Form (2NF) if it satisfies the conditions of the first normal form and every non-prime attribute of that relation is fully functionally dependent on the primary key.

### 03. Third Normal Form

For a relation to pass the third Normal Form (3NF), it must pass the second normal form such that no non-prime attribute of the relation is transitively dependent on the primary key.

### 04. Boyce-Codd Normal Form

The Boyce-Codd Normal Form (3.5NF) augments the third normal form such that every determinant in all functional dependencies is either a candidate key or a primary key.

## Normalizing Each Relation:

### 01. Piece

ID	Name	Date of leave	Date of entry	Type	Date of start	Date of end	Inspirati on	Can be Sold	Date of auction	Section_ ID (FK)	Date of Display	V_ID
----	------	---------------	---------------	------	---------------	-------------	--------------	-------------	-----------------	------------------	-----------------	------

**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

**2NF:** This relation passes 2NF because every non-prime attribute is fully functionally dependent by the primary key (ID).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (ID).

**3.5NF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

### 02. Artist

ID	Name	Date of Death	Date of Birth	Biography	Nationality
----	------	---------------	---------------	-----------	-------------

**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

**2NF:** This relation passes 2NF because every non-prime attribute is fully functionally dependent by the primary key (ID).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (ID).

**3.5NF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

### 03. Section

<u>ID</u>	Name	Location
-----------	------	----------

**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

**2NF:** This relation passes 2NF because every non-prime attribute is fully functionally dependent by the primary key (ID).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (ID).

**3.5NF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

### 04. Provisioner

<u>ID</u>	Name	Phone	Email
-----------	------	-------	-------

**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

**2NF:** This relation passes 2NF because every non-prime attribute is fully functionally dependent by the primary key (ID).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (ID).

**3.5NF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

## 05. Mediator

<u>ID</u>	Name	Phone	Email
-----------	------	-------	-------

**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

**2NF:** This relation passes 2NF because every non-prime attribute is fully functionally dependent by the primary key (ID).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (ID).

**3.5NF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

## 06. Supplier

<u>ID</u>	Phone	Email	Name
-----------	-------	-------	------

**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

**2NF:** This relation passes 2NF because every non-prime attribute is fully functionally dependent by the primary key (ID).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (ID).

**3.5NF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

## 07. Equipment

Serial Number	Name	Category
---------------	------	----------

**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

**2NF:** This relation passes 2NF because every non-prime attribute is fully functionally dependent by the primary key (Serial Number).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (Serial Number).

**3.5NF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

## 08. Partner

ID	Name	Type	Percentage Share
----	------	------	------------------

**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

**2NF:** This relation passes 2NF because every non-prime attribute is fully functionally dependent by the primary key (ID).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (ID).

**3.5NF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

09. Sponsor

ID	Name
----	------

**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

**2NF:** This relation passes 2NF because every non-prime attribute is fully functionally dependent by the primary key (ID).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (ID).

**3.5NF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

10. Event

ID	Name	Type	Description	Date & Time
----	------	------	-------------	-------------

**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

**2NF:** This relation passes 2NF because every non-prime attribute is fully functionally dependent by the primary key (ID).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (ID).

**3.5NF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

## 11. Department

ID	Name	Location	Fax	Email
----	------	----------	-----	-------

**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

**2NF:** This relation passes 2NF because every non-prime attribute is fully functionally dependent by the primary key (ID).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (ID).

**3.5NF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

## 12. Employee

ID	Image	First_Name	Middle_Name	Last_Name	Sex	SSN
Date_of_Birth		Date_of_Start			Date_of_Termination	
Country	Street	Building	Email		Departmental_Extension	

**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

**2NF:** This relation passes 2NF because every non-prime attribute is fully functionally dependent by the primary key (ID).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (ID).

**3.5NF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

### 13. Feedback Program

EntryID	Satisfaction Rating	Visitor_ID (FK)
---------	---------------------	-----------------

**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

**2NF:** This relation passes 2NF because every non-prime attribute is fully functionally dependent by the primary key (EntryID).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (EntryID).

**3.5NF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

### 14. Visitor

ID	FirstName	MiddleInitial	LastName	Gift_Points	Phone_Number	Email
----	-----------	---------------	----------	-------------	--------------	-------

**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

**2NF:** This relation passes 2NF because every non-prime attribute is fully functionally dependent by the primary key (ID).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (ID).

**3.5NF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

15. Reward

<u>EmployeeID</u> (FK)	Type	Count	Amount	Date implemented	Details

**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

**2NF:** This relation passes 2NF because every non-prime attribute is fully functionally dependent by the primary key (EmployeeID, Type, Count).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (EmployeeID, Type, Count).

**3.5NF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

16. Medical Record

<u>EmployeeID</u> (FK)	<u>Date Initialized</u>	Blood Type	Height	Weight	Blood pressure

**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

**2NF:** This relation passes 2NF because every non-prime attribute is fully functionally dependent by the primary key (EmployeeID, Date Initialized).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (EmployeeID, Date Initialized).

**3.5NF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

17. Police Record

<u>EmployeeID</u> (FK)	<u>Date Issued</u>	Arrests	Convictions	Criminal Proceedings
------------------------	--------------------	---------	-------------	----------------------

**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

**2NF:** This relation passes 2NF because every non-prime attribute is fully functionally dependent by the primary key (EmployeeID, Date Issued).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (EmployeeID, Date Issued).

**3.5NF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

18. Emergency Contact

<u>EmployeeID</u> (FK)	<u>Name</u>	Relationship	Contact Phone
------------------------	-------------	--------------	---------------

**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

**2NF:** This relation passes 2NF because every non-prime attribute is fully functionally dependent by the primary key (EmployeeID, Name).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (EmployeeID, Name).

**3.5NF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

19. Dependent

<u>EmployeeID</u> (FK)	<u>Name</u>	Relationship with Employee	Date of Birth	Sex	SSN

1NF: This relation passes 1NF because the values of each attribute are all single and atomic.

2NF: This relation passes 2NF because every non-prime attribute is fully functionally dependent by the primary key (EmployeeID, Name).

3NF: This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (EmployeeID, Name).

3.5NF: This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

20. creates

<u>Artist_ID</u> (FK)	<u>Piece_ID</u> (FK)
-----------------------	----------------------

This relation passes all normal forms (1NF, 2NF, 3NF, and 3.5NF) because all of its attributes compose the primary key (Artist\_ID, Piece\_ID).

21. is overseen

<u>ID_Piece</u> (FK)	<u>Emp_ID</u> (FK)
----------------------	--------------------

This relation passes all normal forms (1NF, 2NF, 3NF, and 3.5NF) because all of its attributes compose the primary key (ID\_Piece, Emp\_ID).

22. is governed

<u>Section_ID (FK)</u>	<u>Employee_ID (FK)</u>
------------------------	-------------------------

This relation passes all normal forms (**1NF**, **2NF**, **3NF**, and **3.5NF**) because all of its attributes compose the primary key (Artist\_ID, Piece\_ID).

23. is managed

<u>Dep_ID (FK)</u>	<u>Emp_ID (FK)</u>
--------------------	--------------------

This relation passes all normal forms (**1NF**, **2NF**, **3NF**, and **3.5NF**) because all of its attributes compose the primary key (Artist\_ID, Piece\_ID).

24. visits

<u>Visitor_ID (FK)</u>	<u>Section_ID (FK)</u>	Ticket ID	Ticket Category
------------------------	------------------------	-----------	-----------------

**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

**2NF:** This relation does not pass 2NF because Ticket Category depends only on part of the primary key (Visitor\_ID), so it must undergo 2NF normalization.

<u>Visitor_ID (FK)</u>	<u>Section_ID (FK)</u>	Ticket ID
<i>visits (1)</i>		

<u>Visitor_ID (FK)</u>	Ticket Category
<i>visits (2)</i>	

**3NF:** Each of these relations pass 3NF because no non-prime attribute is transitively dependent on the primary key (Visitor\_ID, Section\_ID) or (Visitor\_ID) in relations visits (1) and visits (2) respectively.

**3.5NF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

25. donates

<u>Sponsor_ID (FK)</u>	<u>Department_ID</u>	Amount in (\$)
------------------------	----------------------	----------------

**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

**2NF:** This relation passes 2NF because every non-prime attribute is fully functionally dependent by the primary key (Sponsor\_ID, Department\_ID).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (Sponsor\_ID, Department\_ID).

**3.5NF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

26. sponsors

Event_ID (FK)	Sponsor_ID (FK)	Amount in (\$)
---------------	-----------------	----------------

**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

**2NF:** This relation passes 2NF because every non-prime attribute is fully functionally dependent by the primary key (Event\_ID, Sponsor\_ID).

**3NF:** This relation passes 3NF because no non-prime attribute is transitively dependent on the primary key (Event\_ID, Sponsor\_ID).

**3.5NF:** This relation passes BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

27. supervision

Supervisor_ID	Supervisee_ID
---------------	---------------

This relation passes all normal forms (**1NF**, **2NF**, **3NF**, and **3.5NF**) because all of its attributes compose the primary key (Supervisor\_ID, Supervisee\_ID).

28. Employee Nationality

Employee_ID	Nationality
-------------	-------------

This relation passes all normal forms (**1NF**, **2NF**, **3NF**, and **3.5NF**) because all of its attributes compose the primary key (Employee\_ID, Nationality).

## 29. Interests

<u>Visitor_ID</u> (FK from visitor)	<u>Interests</u>
-------------------------------------	------------------

This relation passes all normal forms (**1NF**, **2NF**, **3NF**, and **3.5NF**) because all of its attributes compose the primary key (Visitor\_ID, Interests).

## 30. Comments

<u>EntryID</u> (FK from feedback program)	<u>Comment</u>
---	----------------

This relation passes all normal forms (**1NF**, **2NF**, **3NF**, and **3.5NF**) because all of its attributes compose the primary key (Entry\_ID, Comment).

## 31. Tags

<u>Piece_ID</u> (FK)	<u>Piece_tags</u>
----------------------	-------------------

This relation passes all normal forms (**1NF**, **2NF**, **3NF**, and **3.5NF**) because all of its attributes compose the primary key (Piece\_ID, Piece\_tags).

## 32. Culture

<u>Piece_ID</u> (FK)	<u>Piece_Culture</u>
----------------------	----------------------

This relation passes all normal forms (**1NF**, **2NF**, **3NF**, and **3.5NF**) because all of its attributes compose the primary key (Piece\_ID, Piece\_Culture).

### 33. Extension

<u>Dept_ID (FK)</u>	<u>Dept_Extension</u>
---------------------	-----------------------

This relation passes all normal forms (**1NF**, **2NF**, **3NF**, and **3.5NF**) because all of its attributes compose the primary key (Dept\_ID, Dept\_Extension).

### 34. provides

ProvisionerID (FK from provisioner)	MediatorID (FK from mediator)	PieceID (FK from piece)	Date of purchase	Price	Mediator Fees	Temporary	Country of origin
			↓	↑	↑	↑	↑

**1NF:** This relation passes 1NF because the values of each attribute are all single and atomic.

**2NF:** This relation does not pass 2NF because Mediator Fees depend on (MediatorID, PieceID) while price and country of origin depend on (PieceID).

ProvisionerID (FK from provisioner)	MediatorID (FK from mediator)	PieceID (FK from piece)	Temporary	Price
			↓	↓

*provides (1)*

MediatorID (FK from mediator)	PieceID (FK from piece)	Mediator Fees

*provides (2)*

PieceID (FK from piece)	Price	Country of origin

*provides (3)*

**3NF**: These three resulting relations pass 3NF because no non-prime attribute is transitively dependent on the primary key.

**3.5NF**: These three resulting relations pass BCNF because every determinant in all functional dependencies is either a candidate key or a primary key.

### 35. supplies

<u>Supplier_ID (FK)</u>	<u>Equipment_SR (FK)</u>	<u>Dept_ID (FK)</u>
-------------------------	--------------------------	---------------------

This relation passes all normal forms (**1NF**, **2NF**, **3NF**, and **3.5NF**) because all of its attributes compose the primary key (Supplier\_ID, Equipment\_SR, Dept\_ID).

### 36. is hosted

<u>Event_ID (FK)</u>	<u>Partner_ID (FK)</u>	<u>Department_ID (FK)</u>	<u>Section_ID (FK)</u>
----------------------	------------------------	---------------------------	------------------------

This relation passes all normal forms (**1NF**, **2NF**, **3NF**, and **3.5NF**) because all of its attributes compose the primary key (Event\_ID, Partner\_ID, Department\_ID, Section\_ID).

## XV. Conclusion

In conclusion, the purpose of this database design is to keep track of the different entities that are important for the museum like the employee, piece, department, and artist entities. We also tried to make sure to include all the necessary attributes that define and give the characteristics of the entities that we included in our design. We drew an ER-diagram which follows the Peter-Chen notation using Microsoft Visio as the drawing tool. There are a variety of relations that describe how entities interact with each other. For example: an artist creates a piece, a piece is displayed in a section, and a section is governed by a department.

During the second phase of the project, we have mapped out our established logical design into a logical implementation of the relational model following the seven-step algorithm as described in the Fundamentals of Database Systems book. Some data was then filled to provide a sample state of the database.

As for the third phase of the project, we used SQL to implement our database on Oracle DBMS Express Edition (11g). We have fully documented the processes of creating relations, inserting data into the database, and querying the database to leverage its functionality in the museum workspace.

The fourth, and final, phase of this project deals with the normalization, or decomposition, process of relations so that all relations pass up to the Boyce-Codd Normal Form (3.5NF).

The database has been designed and implemented with a variety of end-users in mind. Parametric users can track who visited which sections at what time. Sophisticated users can leverage SQL and its aggregate functions to collect any needed statistics. The DBA can retrieve any required attribute of all entities at any time and manipulate the database to maintain its up-to-date state.

This project has been a great learning opportunity for all three of us. We were able to practically apply the concepts we've learned to design the conceptual ER diagram, map the ER into various relations, query the database using SQL on Oracle, and normalize every relation to pass BCNF.

We were also able to amend and adjust each phase of the project according to our instructor's, Dr. Bakri's, feedback which helped shape this finalized version of the project.

## XVI. References

El Masri, R. Navathe, S. B. (2016). *Fundamentals of database systems* (7<sup>th</sup> edition). Pearson

CDC (How to Calculate BMI)

[https://www.cdc.gov/nccdphp/dnpao/growthcharts/training/bmiage/page5\\_1.html#:~:text=With%20the%20metric%20system%2C%20the,by%2010%2C000%2C%20can%20be%20used.](https://www.cdc.gov/nccdphp/dnpao/growthcharts/training/bmiage/page5_1.html#:~:text=With%20the%20metric%20system%2C%20the,by%2010%2C000%2C%20can%20be%20used.)

Museum [Definition]

[https://www.oxfordlearnersdictionaries.com/definition/american\\_english/museum#:~:text=museum,-noun,modern%20art%20a%20science%20museum](https://www.oxfordlearnersdictionaries.com/definition/american_english/museum#:~:text=museum,-noun,modern%20art%20a%20science%20museum)

## XVII. Instructor Feedback

Dear Dr. Bakri, we highly appreciate the time and effort you put into reviewing our finalized version of the project. Thank you for your consistent guidance and the opportunity to work on such a great project!