University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: October 19th, 2023

Group Number: 32

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Norman Vo	54398649	z5y8p	vo.hoang.quan2003@gmail.com
Danny Ngo	10112480	a3r8j	danhngo80@gmail.com
Kaz Kianoush	96885892	o3s5g	kazkianoush03@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

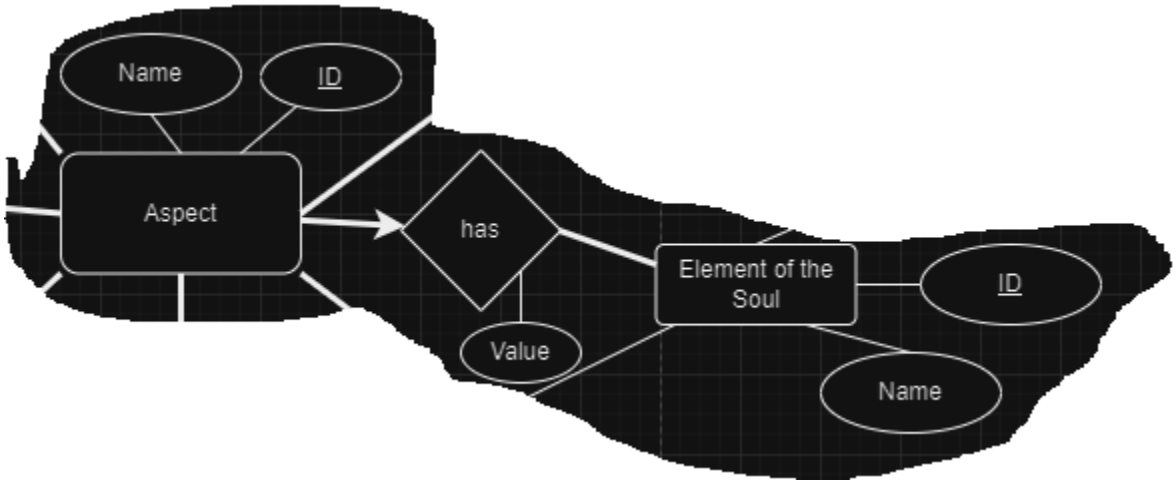
Summary

Our project is to present core information about items in the game "Book of Hours". This project acts as a centralized place where players can quickly look up data while playing the game. Since the game provides no tutorial to the player and most of the game mechanics are discovered through trial and error, we design this wikipedia for the player to look up the information quickly.

ER Diagram Changes

Since the first milestone, we have made the following changes to the ER Diagram:

- Removing ***aspects*** attributes from the entities in the ER diagram and creating ***aspects*** entity in the ER diagram. We learned that each entities' aspects are presented by its name and value and there are a total of 13 types of aspects in the game. Therefore, if a name of an aspect changes, we don't have to update the name of that aspect in other entities in the database.
  - ***Aspects*** entities have a total participation with every other entity on the ER Diagram.
  - We will present the aspects of other entities using the aspects' ID and presenting the ***aspect value*** in the relationship between the entities and the ***aspect table***.
  - The new change is shown in the following screenshot:



  -
- The following changes are made to update our understanding of the game between milestone 1 and milestone 2:
  - Changed the attribute "UniqueSource" from Numen to a one-to-one relationship "Unique Source" with book
  - Changed the relationship between Book and Memory to be many-to-one

The final ER Diagram is at the end of the document

Schema (primary key is underlined, candidate key is italicized, foreign key is bolded)

Since an entity's *name* is uniquely identifable across every entity in the game, the *name* for each entity is the ER diagram can be a candidate key.

ElementOfTheSoul(ID: char[20], *name*: char[20], **AssistantID**: char[20])

Aspect(ID: char[20], *name*: char[20], **BookID**: char[20], **SkillID**: char[20],**ItemID**: char[20],**PersonID**: char[20], **MemoryID**: char[20], **ElementOfTheSoulID**: char[20])

Memory(ID: char[20], *name*: char[40], Sources: char[100], isSound: boolean, isOmen:boolean, isPersistent: boolean, isWeather: boolean)

Numen(**ID**: char[20], ***bookID***: char[20])

Book(ID: char[20], *name*: char[80], Language: char[20], **AspectID**: char[20], **MemoryID**: char[20], **skillID**: char[20], **NumenID**: char[20], **elementOfTheSoulID**: char[20])

Workstation(ID: char[20], Room: char[20], *Name*: char[40], Evolution: char[20], Slots: char[80], **SkillID**: char[20])

Skill(ID: char[20], *Name*: char[40], SpecialAttribute: char[20], **WorkStationID**: char[20], **BookID**: char[20])

Visitor(**ID**: char[20], IsNumaOnly: boolean, **LanguageID**: char[20])

Language(**ID**: char[20], NotCraftable: boolean)

Item(ID: char[20], *Name*: char[40])
Item(ID: char[20], **WorkstationID:** char[20], **skillID**: char[20], *Name*: char[40])

Assistant(**ID**: char[20], Speciality:char[20], Cost: integer, Location: char[20], **ItemID**: char[20], **ElementOfTheSoulID**: char[20])

People(ID: char[20], *Name*: char[20])

WorkshopAspectRequirement(ID: char[20], **AspectID**: char[20], **WorkstationID**: char[20])

Functional Dependencies

Memory:
memory.ID -> memory.name, memory.sources, memory.isSound, memory.isOmen, memory.isPersistent, memory.isWeather

Numen:
numen.ID -> book.ID, memory.ID (numen is a subset of memory)

Book:
book.ID -> book.name, book.language
book.ID -> aspect.ID, memory.ID, ElementOfTheSoul.ID

Element Of the Soul:
elementOfTheSoul.ID -> elementOfTheSoul.name
elementOfTheSoul.ID -> assistant.ID

Aspect:
aspect.ID -> aspect.name
aspect.ID -> memory.ID

Workstation:
workstation.ID -> workstation.name, workstation.evolution, workstation.slots, workstation.room
workstation.ID, skill.ID -> item.ID
workstation.ID -> skill.ID

Skill:
skill.ID -> skill.name, skill.specialAttribute
skill.ID -> workstation.ID, book.ID

Language:
Skill.ID -> language.ID (language is a subset of skill)
language.ID -> language.name, language.specialAttribute, language.notCraftable

People:
people.ID -> people.name
people.ID -> aspect.ID

Assistant:
assistant.ID -> Assistant.name, assistant.specialty, assistant.cost, assistant.location

assistant.ID -> elementOfTheSoul.ID, item.ID, people.ID (assistant is a subset of people)

Visitor:
visitor.ID -> visitor.name, visitor.isNumaOnly
visitor.ID -> language.ID, people.ID (visitor is a subset of people)

Item:
item.ID -> aspect.ID, assistant.ID

WorkshopAspectRequirement (abbreviated to **WAR** throughout the document**):**
WAR.ID -> AspectID, WorkstationID

Normalization

Note about our work in this section: the attributes of an entity that are determined by the primary key of an entity are omitted for readability i.e workstation.ID -> workstation.room. These attributes will go in the first table that has the primary key, so in this case workstation.room will be in the table where workstation.ID is located

1/ ElementOfTheSoul(ID: char[20], Name: char[20], AssistantID: char[20])
elementOfTheSoul.ID -> elementOfTheSoul.name
elementOfTheSoul.ID -> assistant.ID
Closures:

elementOfTheSoul.ID+ = {elementOfTheSoul.ID, elementOfTheSoul.name, assistant.ID}
assistant.ID+ = {assistant.ID, elementOfTheSoul.ID, elementOfTheSoul.name}
The candidate key here are EOTS.ID or assistant.ID

=> Since each of EOTS and assistant is the superkey, all FDs in the ElementOfTheSoul set satisfies BCNF.

2/ Aspect(ID: char[20], name: char[20], BookID: char[20], SkillID: char[20], ItemID: char[20], PeopleID: char[20], MemoryID: char[20], ElementOfTheSoulID: char[20])
aspect.ID -> memory.ID
skill.ID -> book.ID
book.ID -> aspect.ID, memory.ID, ElementOfTheSoul.ID
people.ID -> aspect.ID
elementOfTheSoul.ID -> people.ID
item.ID -> aspect.ID
Closures:

aspect.ID+ = {aspect.ID, memory.ID, book.ID, ElementOfTheSoul.ID, people.ID}
memory.ID+ = {memory.ID}
skill.ID+ = {skill.ID, book.ID, aspect.ID, memory.ID, ElementOfTheSoul.ID, people.ID}
book.ID+ = {book.ID, aspect.ID, memory.ID, ElementOfTheSoul.ID, people.ID}

people.ID+= {people.ID, aspect.ID, memory.ID, book.ID, ElementOfTheSoul.ID, people.ID}
item.ID+ = {item.ID, aspect.ID, memory.ID, book.ID, ElementOfTheSoul.ID, people.ID}
=> No single one candidate key that allows us to know every other key in this set, performing finding minimal keys

Finding all Minimal Keys

Left	Middle	Right
item.ID skill.ID	book.ID aspect.ID memory.ID ElementOfTheSoul.ID people.ID	memory.ID

The superkeys for this set is **{item.ID and skill.ID}**.

Decompose into BCNF item.ID -> aspect.ID
R1(itemID, aspectID), R2 (itemID, skillID, bookID, memoryID, ElementOfTheSoulID, peopleID)
Decomposing R2 using skill.ID -> book.ID
R3 (itemID, skillID, bookID), R4 (skillID, memoryID, ElementOfTheSoulID, peopleID)
Decomposing R4 using skill.ID -> memory.ID (implicit FD from skill.ID -> book.ID, book.ID -> memory.ID)
R5(skill.ID, memory.ID), R6(skill.ID, ElementOfTheSoul.ID, people.ID).
Decomposing R6 using skill.ID -> ElementOfTheSoul.ID, people.ID (implicit FDs from skill.ID -> book.ID, book.ID -> ElementOfTheSoul.ID, and ElementOfTheSoul.ID->people.ID)
R7 (skill.ID, ElementOfTheSoul.ID), R8 (skill.ID, people.ID)

Final Set: R1(itemID, aspectID), R3 (itemID, skill.ID, bookID), R5(skill.ID, memory.ID), R7 (skill.ID, ElementOfTheSoul.ID), R8 (skill.ID, people.ID)

Identify PKs and FKs for each of these sets

- R1: PK is itemID
- R3: PK is (itemID, skillID),
    - FK is itemID between R1's itemID and R3's itemID
    - FK is bookID between R3's bookID and Book (bookID)
- R5: PK is skillID.
    - FK is skillID between R5's skillID to R3's skillID
    - FK is memoryID between R5's memoryID to Memory (memoryID)
- R7: PK is skillID.
    - FK is skillID between R7's skillID and R5's skillID
    - FK is ElementOfTheSouldID between R7 and ElementOfTheSoul(ElementOfTheSouldID)
- R8: PK is skillID.
    - FK is skillID between R8's skillID and R7's skillID
    - FK is peopleID between R8 and People (peopleID)

3/ Memory(ID: char[20], name: char[40], Sources: char[100], isSound: boolean, isOmen:boolean, isPersistent: boolean, isWeather: boolean)

memory.ID -> memory.name, memory.sources, memory.isSound, memory.isOmen, memory.isPersistent, memory.isWeather

Closures:

memory.ID+ = {memory.name, memory.sources, memory.isSound, memory.isOmen, memory.isPersistent, memory.isWeather}
memoryID is a candidate key

=> Since memoryID is a superkey, all FDs in the Memory set satisfies BCNF.

4/ Workstation(ID: char[20], Room: char[20], Name: char[40], Evolution: char[20], Slots: char[80], SkillID: char[20])

workstation.ID -> skill.ID
skill.ID -> workstation.ID

Closures:

skill.ID+ = {skill.ID, workstation.ID}
workstation.ID+ = {workstation.ID, skill.ID}
The memoryID and bookID are candidate keys

=> Since each of memoryID and bookID are the superkey, all FDs in the Memory set satisfies BCNF.

5/ Skill(ID: char[20], Name: char[40], SpecialAttribute: char[20], WorkStationID: char[20], BookID: char[20])

skill.ID -> workstation.ID, book.ID
workstation.ID -> skill.ID

Closures:

skill.ID+ = {skill.ID, workstation.ID, book.ID}
workstation.ID+ = {workstation.ID, skill.ID, book.ID}
book.ID+ = {bookID}
=> SkillID and workstationID are both candidate keys

=> Since each of SkillID and workstationID are the superkey, all FDs in the Skill set satisfies BCNF.

6/ Assistant(ID: char[20], Speciality:char[20], Cost: integer, Location: char[20], ItemID: char[20], ElementOfTheSoulID: char[20])

assistant.ID -> elementOfTheSoul.ID, item.ID
elementOfTheSoul.ID -> assistant.ID
item.ID -> assistant.ID

Closures:

Assistant.ID+ = {assistant.ID, elementOfTheSoul.ID, item.ID}
ElementOfTheSoul.ID+ = {ElementOfTheSoul.ID, assistant.ID, item.ID}
item.ID+ = {item.ID, assistant.ID, elementOfTheSoul.ID}
=> Each Assistant.ID, EOTS.ID, and itemID can act as candidate keys

=> Since each of Assistant.ID, EOTS.ID, and itemID are the superkey, all FDs in the Skill set satisfies BCNF.

7/ WorkshopAspectRequirement(ID: char[20], AspectID: char[20], WorkstationID: char[20])
WAR.ID -> AspectID, WorkstationID
Closures:

WAR.ID+ = {WAR.ID, AspectID, WorkstationID}
WAR.ID is the candidate key

=> Since WAR.ID is the super key, all FDs in the WorkshopAspectRequirement set satisfies BCNF
8/ Visitor(ID: char[20], IsNumaOnly: boolean, LanguageID: char[20])
visitor.ID -> language.ID
Closures:

Visitor.ID+ = {Visitor.ID, language.ID}
VisitorID is the candidate key
=> Since VisitorID is the super key, all FDs in the Visitor set satisfies BCNF

SQL statements

Note about insertion: For IDs, go for 2 alphabet letters followed by 3 digits. This will be our formal naming convention in milestone 3. The name was shortened to ID on the ER diagram and previous section of this report is for readability.

Memory

```
CREATE TABLE Memory(
    memoryID CHAR(20) PRIMARY KEY,
    memoryName CHAR(40) NOT NULL,
    memorySources CHAR(100),
    memoryIsSound BIT(1), // BIT is a data type used to represent boolean in mySQL
    memoryIsOmen BIT(1),
    memoryIsPersistent BIT(1),
    memoryIsWeather BIT(1),
    UNIQUE (memoryName)
)
```

Note: The digits for ID is the row that memory was in the spreadsheet - 1.

```
INSERT INTO Memory(memoryID, memoryName, memorySources, memoryIsSound,
memoryIsOmen, memoryIsPersistent, memoryIsWeather)
VALUES (“ME001”, “Memory: Taste”, “Considering sustenance and beverages”, 0, 0, 0, 0)

INSERT INTO Memory(memoryID, memoryName, memorySources, memoryIsSound,
memoryIsOmen, memoryIsPersistent, memoryIsWeather)
VALUES (“ME002”, “Memory: Sound”, “Considering the Hush House Key”, 0, 0, 0, 0)
```

INSERT INTO Memory(memoryID, memoryName, memorySources, memoryIsSound, memoryIsOmen, memoryIsPersistent, memoryIsWeather)
VALUES (“ME007”, “Memory: Foresight”, “Edge or Forge books”, 0, 0, 0, 0)

INSERT INTO Memory(memoryID, memoryName, memorySources, memoryIsSound, memoryIsOmen, memoryIsPersistent, memoryIsWeather)
VALUES (“ME023”, “Beguiling Melody”, “Grail books or Grail 5 craft”, 1, 0, 0, 0)

INSERT INTO Memory(memoryID, memoryName, memorySources, memoryIsSound, memoryIsOmen, memoryIsPersistent, memoryIsWeather)
VALUES (“ME042”, “Earthquake Name”, “Scale 15 crafts”, 0, 1, 1, 0)

Numen

```
CREATE TABLE Numen(
        numenID CHAR(20) PRIMARY KEY,
        bookID CHAR(20) NOT NULL,
        FOREIGN KEY (numenID) REFERENCES Memory(memoryID)
                                ON DELETE CASCADE
                                ON UPDATE CASCADE,
        FOREIGN KEY (bookID) REFERENCES Book(bookID)
                                ON DELETE CASCADE
                                ON UPDATE CASCADE
)
```

INSERT INTO Memory(memoryID, memoryName, memorySources, memoryIsSound, memoryIsOmen, memoryIsPersistent, memoryIsWeather)
VALUES (“ME043”, “Numen: Loopholes”, “Serpent-Root”, 0, 0, 1, 0)
INSERT INTO Numen(numenID, bookID)
VALUES (“ME043”, “BK196”)

INSERT INTO Memory(memoryID, memoryName, memorySources, memoryIsSound, memoryIsOmen, memoryIsPersistent, memoryIsWeather)
VALUES (“ME044”, “Numen: Inescapable Confinement”, “Towards A Fundamental Aesthetic”, 0, 0, 1, 0)
INSERT INTO Numen(numenID, bookID)
VALUES (“ME044”, “BK105”)

INSERT INTO Memory(memoryID, memoryName, memorySources, memoryIsSound, memoryIsOmen, memoryIsPersistent, memoryIsWeather)
VALUES (“ME055”, “Numen: Three Rules”, “The Three And The Three (St Chiavi Manuscript)”, 0, 0, 1, 0)
INSERT INTO Numen(numenID, bookID)
VALUES (“ME055”, “BK165”)

INSERT INTO Memory(memoryID, memoryName, memorySources, memoryIsSound, memoryIsOmen, memoryIsPersistent, memoryIsWeather)
VALUES (“ME050”, “Numen: The Bells of Ys”, “Amiranis Beteli”, 0, 0, 1, 0)
INSERT INTO Numen(numenID, bookID)
VALUES (“ME050”, “BK039”)

INSERT INTO Memory(memoryID, memoryName, memorySources, memoryIsSound, memoryIsOmen, memoryIsPersistent, memoryIsWeather)
VALUES (“ME047”, “Numen: That Old Lost Music”, “The Turquoise Hand”, 0, 0, 1, 0)
INSERT INTO Numen(numenID, bookID)
VALUES (“ME047”, “BK206”)

Book

```
CREATE TABLE Book(
        bookID CHAR(20),
        bookName CHAR(80),
        language CHAR(20),
        FOREIGN KEY aspectID REFERENCES Aspect(aspectID)
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY memoryID REFERENCES Memory(memoryID)
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY skillID REFERENCES Skill(skillID)
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY elementOfTheSoulID REFERENCES
        ElementOfTheSoul(elementOfTheSoulID)
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY numenID REFERENCES Numen(numenID)
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        PRIMARY KEY (bookID),
        UNIQUE (bookName))
```

INSERT into Book(‘BK001’, ‘The Iron Book ’,FUCINE ’,’EDGE 8 ’, ‘Foresight’ ,’Disciplines of the Hammer’, NULL, NULL)

INSERT into Book(‘BK002’, ’The Ascendant’,’HYKSOS’,’EDGE 10’, ‘Contradiction’,’Edicts Martial’,NULL, NULL)

INSERT into Book('BK003','The Wound-Wounds', 'KILLASIMI', 'EDGE 18','Regret','Disciplines of the Scars',NULL, NULL)

INSERT into Book('BK004','De Horis Book 2', 'LATIN','EDGE 4','Foresight','Disciplines of the Hammer',NULL,NULL)

INSERT into Book('BK005', 'De Horis Book 3','LATIN','EDGE 4','Contradiction','Disciplines of the Scars',NULL,NULL)

Element Of the Soul

```
CREATE TABLE ElementOfTheSoul (
        elementOfTheSoulID CHAR(20),
        elementOfTheSoulName CHAR(20),
        assistantID CHAR(20),
        FOREIGN KEY assistantID REFERENCES Assistant(assistantID)
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        PRIMARY KEY (elementOfTheSoulID))
```

```
INSERT into ElementOfTheSoul('ES001','Chor',NULL)
INSERT into ElementOfTheSoul'ES002','Fet',NULL)
INSERT into ElementOfTheSoul('ES003','Shapt',NULL)
INSERT into ElementOfTheSoul('ES004','Trist'',NULL)
INSERT into ElementOfTheSoul('ES005','Wist',NULL)
```

Aspect

```
CREATE TABLE Aspect(
        aspectID CHAR(20),
        aspectName CHAR(20),
        itemID CHAR(20),
        FOREIGN KEY itemID REFERENCE Item(itemID)
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        UNIQUE (aspectName),
        PRIMARY KEY (aspectID))
```

```
INSERT into Aspect('AS001','MOON',NULL)
INSERT into Aspect('AS002','LANTERN',NULL)
INSERT into Aspect('AS003','FORGE',NULL')
INSERT into Aspect('AS004','SCALE',NULL)
INSERT into Aspect('AS005','WINTER',NULL)
```

Workstation

```
CREATE TABLE Workstation (workstationID CHAR(20),
workstationName CHAR(40),
room CHAR(20),
evolution CHAR(20),
slots CHAR(80),
PRIMARY KEY (workstationID),
UNIQUE (skillID))

INSERT into Workstation('WS001', 'Altar of the Knot', 'Our Lady Beneath', 'Bosk', 'SK001')
INSERT into Workstation('WS002', 'Shrine: The Chancel', 'Chancel of the Abbey Church',
'Horomachistry', 'SK002')
INSERT into Workstation('WS003', 'Practice Garden Dummy', 'Practic Garden', 'Illumination',
'SK003')
INSERT into Workstation('WS004', 'Altar: Ascite', 'Chapel Ascite', 'Nyctodromy', 'SK004')
INSERT into Workstation('WS005', 'Mirrors', 'Hall of Division', 'Nyctodromy', 'SK005')
```

WorkshopAspectRequirement

```
CREATE TABLE WorkshopAspectRequirement(WorkshopAspectRequirementID CHAR(20),
PRIMARY KEY WorkshopAspectRequirementID,
FOREIGN KEY AspectID REFERENCES Aspect(aspectID) ON DELETE SET DEFAULT ON UPDATE
CASCADE,
FOREIGN KEY WorkstationID REFERENCES Workstation(workstationID) ON DELETE SET
DEFAULT ON UPDATE CASCADE)

INSERT into WorkshopAspectRequirement ('WA001',  'AS001', 'WS001')
INSERT into WorkshopAspectRequirement ('WA002',  'AS002', 'WS002')
INSERT into WorkshopAspectRequirement ('WA003',  'AS003', 'WS003')
INSERT into WorkshopAspectRequirement ('WA004',  'AS004', 'WS004')
INSERT into WorkshopAspectRequirement ('WA005',  'AS005', 'WS005')
```

Skill

```
CREATE TABLE SkillItem (
        skillID CHAR(20),
        skillName CHAR(40),
        specialAttribute CHAR(20),
        PRIMARY KEY skillID,
        UNIQUE skillName,
        FOREIGN KEY (aspectID) REFERENCES Aspect(aspectID)
                ON DELETE CASCADE
                ON UPDATE CASCADE)

INSERT into SkillItem('SK001', 'Disciplines of the Hammer', null)
```

```
INSERT into SkillItem('SK002', 'Disciplines of the Scars', null)
INSERT into SkillItem('SK003', 'Ragged Crossroads', null)
INSERT into SkillItem('SK004', 'Sharps', null)
INSERT into SkillItem('SK005', 'Sickle & Eclipse', "CORRUPTION")

CREATE TABLE ItemSkillBook (
        itemID CHAR (20),
        skillID CHAR(20),
        bookID CHAR (20),
        PRIMARY KEY (itemID, skillID),
        FOREIGN KEY (itemID) REFERENCES SkillItem(itemID)
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY (bookID) REFERENCES Book(bookID)
        ON DELETE CASCADE
        ON UPDATE CASCADE)

INSERT into ItemSkillBook ('IT001', 'SK001', 'BK005')
INSERT into ItemSkillBook ('IT002', 'SK002', 'BK003')
INSERT into ItemSkillBook ('IT003', 'SK003', 'BK001')
INSERT into ItemSkillBook ('IT006', 'SK004', 'BK004')
INSERT into ItemSkillBook ('IT021', 'SK005', 'BK002')

CREATE TABLE SkillMemory(
        skillID CHAR(20),
        PRIMARY KEY skillID,
        FOREIGN KEY (skillID) REFERENCES ItemSkillBook(skillID)
                ON DELETE CASCADE
                ON UPDATE CASCADE
        FOREIGN KEY (memoryID) REFERENCES Memory(memoryID)
                 ON DELETE CASCADE
                 ON UPDATE CASCADE)

INSERT into SkillMemory('SK001','ME001')
INSERT into SkillMemory('SK002','ME002')
INSERT into SkillMemory('SK003','ME007')
```

```
INSERT into SkillMemory('SK004','ME023')
INSERT into SkillMemory('SK005','ME042')

CREATE TABLE SkillEOTS (
        skillID CHAR(20),
        PRIMARY KEY skillID,
        FOREIGN KEY (skillID) REFERENCES SkillMemory(skillID)
                ON DELETE CASCADE
                ON UPDATE CASCADE
        FOREIGN KEY (elementOfTheSoulID) REFERENCES
ElementOfTheSoul(elementOfTheSoulID)
                ON DELETE CASCADE
                ON UPDATE CASCADE)

INSERT into SkillEOTS ('SK001',''ES001')
INSERT into SkillEOTS ('SK002',''ES002')
INSERT into SkillEOTS ('SK003','ES003')
INSERT into SkillEOTS ('SK004',''ES004')
INSERT into SkillEOTS ('SK005',''ES005')


CREATE TABLE SkillPeople (
        skillID CHAR(20),
        PRIMARY KEY skillID,
        FOREIGN KEY (skillID) REFERENCES SkillEOTS (skillID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
        FOREIGN KEY (peopleID) REFERENCES People(peopleID)
        ON DELETE CASCADE
        ON UPDATE CASCADE)

INSERT into SkillPeople ('SK001', null, "PE001")
INSERT into SkillPeople ('SK002', "ES001", null)
INSERT into SkillPeople ('SK003', "ES002", "PE002")
INSERT into SkillPeople ('SK004', null, null)
INSERT into SkillPeople ('SK005', "ES004", null)
```

Language

```
CREATE TABLE Language(
        languageID CHAR(20),
        notCraftable Bit(1),
        PRIMARY KEY languageID,
        FOREIGN KEY languageID REFERENCES Skill(skillID)
                ON DELETE CASCADE
```

```
                ON UPDATE CASCADE
)

INSERT into Language(“SK007”, 0)
INSERT into Language(“SK012”, 0)
INSERT into Language(“SK017”, 1)
INSERT into Language(“SK039”, 0)
INSERT into Language(“SK058”, 1)
```

People

```
CREATE TABLE People(
        peopleID CHAR(20],
        peopleName CHAR(20],
        PRIMARY KEY (peopleID),
        UNIQUE(peopleName)
)

INSERT into People(“PE001”, “Mr Peter Agdistis”)
INSERT into People(“PE002”, “Dr Ibn al-Adim”)
INSERT into People(“PE003”, “Lt Arthur Thomas Moore (Ret.)”)
INSERT into People(“PE029”, “Traveling Musician”)
INSERT into People(“PE030”, “Fugitive”)
```

Assistant

```
CREATE TABLE Assistant(
        assistantID CHAR(20) PRIMARY KEY,
        assistantSpecialty CHAR(20),
        assistantCost INTEGER,
        assistantLocation CHAR(20),
        itemID CHAR(20),
        elementOfTheSoulID CHAR(20),
        FOREIGN KEY assistantID REFERENCES People(peopleID)
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY itemID REFERENCES Item(itemID)
                ON DELETE SET NULL
                ON UPDATE CASCADE,
        FOREIGN KEY elementOfTheSoulID REFERENCES
ElementOfTheSoul(elementOfTheSoulID) ON DELETE SET NULL
                                        ON UPDATE CASCADE

)

INSERT into Assistant(“PE031”, “Metal”, 12, “Smithy”, null, null)
```

INSERT into Assistant(“PE032”, “Fabric and Fibre”, 12, “Kille House”, null, null)
INSERT into Assistant(“PE033”, “Wood”, 12, “Kille House”, null, null)
INSERT into Assistant(“PE039”, null, 24, “Sweet Bones”, null, null)
INSERT into Assistant(“PE040”, null, 24, “Sweet Bones”, null, null)

Visitor

```
CREATE TABLE Visitor(
        visitorID CHAR(20) PRIMARY KEY,
        visitorIsNumaOnly BIT(1),
        languageID CHAR(20),
        FOREIGN KEY visitorID REFERENCES People(peopleID)
                        ON DELETE CASCADE
                        ON UPDATE CASCADE,
        FOREIGN KEY languageID REFERENCES Language(languageID)
                        ON DELETE CASCADE
                        ON UPDATE CASCADE,
        UNIQUE (languageID)
)
```

INSERT into Visitor (“PE001”, 0, “SK017”)
INSERT into Visitor (“PE009”, 0, “SK012”)
INSERT into Visitor (“PE013”, 0, “SK039”)
INSERT into Visitor (“PE022”, 1, “SK007”)
INSERT into Visitor (“PE024”, 1, “SK058”)

Item

```
CREATE TABLE Item(
        itemID CHAR(20),
        itemName CHAR (20),
        UNIQUE itemName,
        PRIMARY KEY (itemID))
```

INSERT into Item (‘IT001’, ‘Torgue's Cleansing’)
INSERT into Item (‘IT006’, ‘Rosehip Jam’)
INSERT into Item (‘IT007’, ‘Refulgin’)
INSERT into Item (‘IT018’, ‘Catwink’)
INSERT into Item (‘IT020’, ‘Rubywise Ruin’)