

CPSC 304 Project Cover Page

Milestone #:2

Date: September 27, 2023

Group Number: 111

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Neil Li	22681720	a0y6f	neillyx26@gmail.com
Kevin Liu	94200474	z4f4p	kevinliu2002@yahoo.ca
Dane Urban	67510214	g1v5g	z5363301@ad.unsw.edu.au

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 email 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 team intends on creating a deck building database for a card game for our project, in which players can look up decks they can build, and find the cards necessary for that deck. They can also see the listings for those cards and where they are sold, as well as who are the artists that created that card.

ER diagram:

Here is the changed ER diagram:

After thoughtful discussion, it came to our realisation that art is not a weak entity as many art pieces online can often be assigned unique IDs and do not need their artists to identify them.

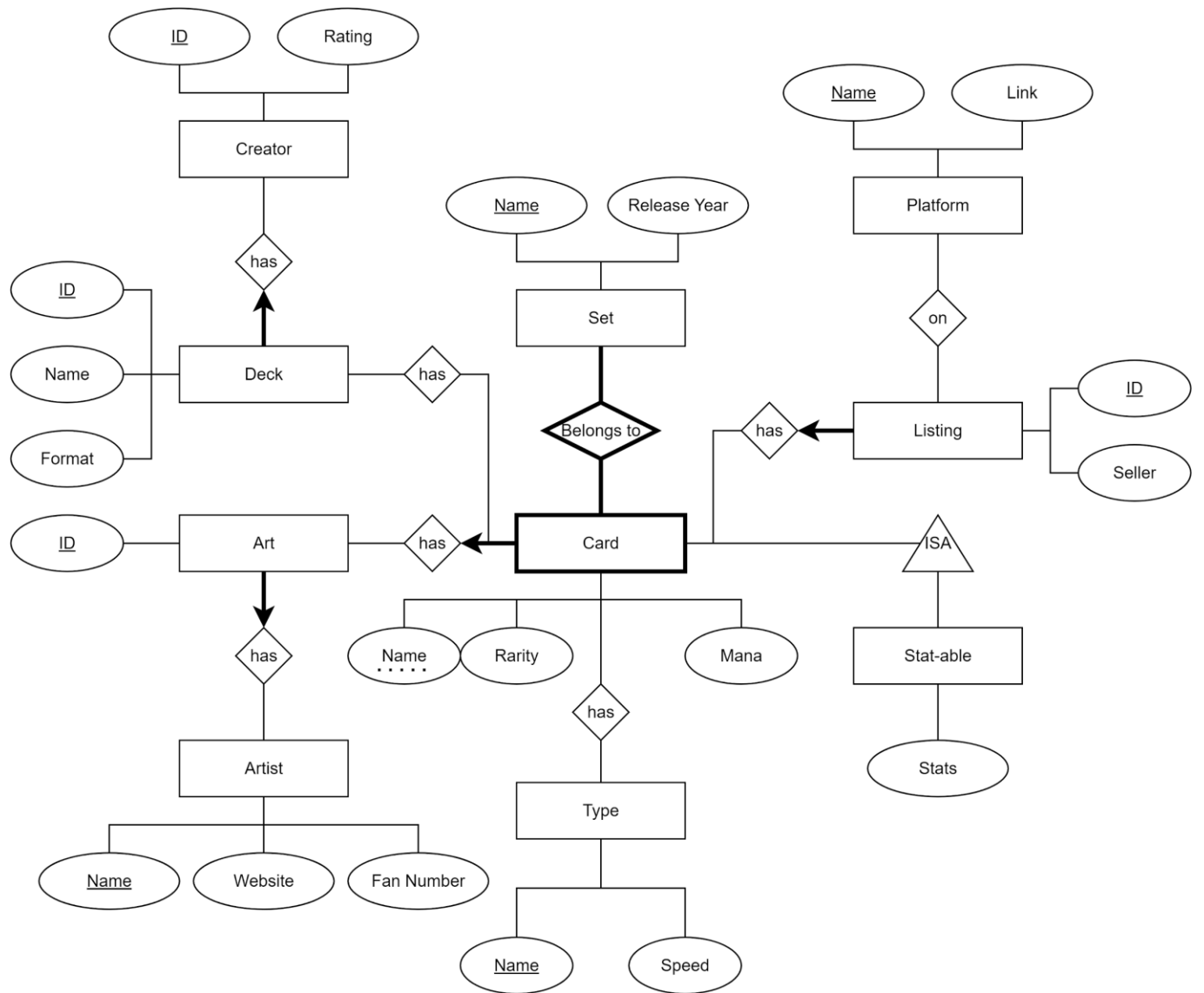
Listing is not a weak entity either, as a listing can still be identified without the card that is being sold.

Type is now its own entity rather than an attribute to a card now that we realize it as the type of the card also carries information about the speed at which the card could be played at, as such it needs to be its own entity to represent that fully.

Card is actually a weak entity as different versions of a card with the same name can be printed in different sets, and those could sometimes have different rarities and arts.

ISA Constraint: stat-able is disjoint and partial, as cards either have stats in the format of life/health, or they don't.

Decks no longer have a participation constraint as it is possible for incomplete decks to exist and for placeholder decks to exist.



Relational Schema (Bold = FK, Underline = PK):

Platform (platformName: char[64], platformLink: char[256])

- Candidate Key: {platformName}
- Renamed from simply Name and Link to platformName and platformLink offer better clarity

Listings_On_Platform (**platformName**: char[64], listingID: integer)

- Candidate Key: {platformName, listingID}
- Renamed from simply Name and ID to platformName and listingID offer better clarity

Card_Listing (listingID: integer, seller: char[32], cardName: char[64] NOT NULL, setName: char[32] NOT NULL)

- Candidate Key: {listingID}
- Renamed from ID to offer better clarity, renamed from name to cardName and setName to distinguish between two names

Drew_Art (artID: integer, **artistName**: char[32] NOT NULL)

- Candidate Key: {artID}
- Renamed from simply ID and Name to artID and artistName offer better clarity

Artist (artistName: char[32], artistSite: char[256], fanNumber: integer)

- Candidate Key: {artistName}
- Renamed from simply Name and Website to artistName and artistSite offer better clarity, and removed space in fanNumber

Card_With_Art (cardName: char[64], **setName**: char[32], **artID**: integer, rarity: char[1], mana: integer, stats: char[7])

- Candidate Key: {cardName, setName}
- Renamed from ID to artID for better clarity, renamed from name to cardName and setName to distinguish between two names

Set (setName: char[32], releaseYear: integer)

- Candidate Key: {setName}
- Renamed to setName to offer better clarity and to avoid future collisions.

Types_of_Card (**typeName**: char[8], **cardName**: char[64], **setName**: char[32])

- Candidate Key: {typeName}
- renamed from name to cardName, setName, and typeName to distinguish between three names

Type (typeName: char[8], playSpeed: char[1])

- Candidate Key: {typeName}
- Renamed from name to typeName to offer better clarity and to avoid future collisions

Creator (creatorID: char[64], creatorRating: double)

- Candidate Key: {creatorID}
- Renamed from ID and Rating to creatorID and creatorRating to offer better clarity

Deck_From_Creator (deckID: integer, deckName: char[64], playFormat: char[4], **creatorID**: char[64] NOT NULL)

- Candidate Key: {deckID}
- Renamed from Name to deckName to offer better clarity, renamed from ID to creatorID and deckID to distinguish the two names.

Card_In_Deck (**deckID**: integer, **cardName**: char[64], **setName**: char[32])

- Candidate Key: {deckID}
- Renamed from ID to deckID for better clarity, renamed from name to cardName and setName to distinguish between two names

Functional Dependencies:

Platform:

platformName -> platformLink

Listings_On_Platform:

listingID -> seller

listingID -> format

listingID -> cardName

listingID -> setName

Drew_Art:

artID -> artistName

Artist:

artistName -> artistSite

artistName -> fanNumber

Card_With_Art:

cardName, setName -> artID

cardName, setName -> rarity

cardName, setName -> typeName

cardName -> mana

cardName -> stats

artID -> cardName (Not key FD)

Set:

setName -> releaseYear

Type:

typeName -> playSpeed

Creator:

creatorID -> creatorRating

Deck_From_Creator:

deckID -> deckName

deckID -> playFormat

deckID -> creatorID

Normalization:

Our team will choose to normalize into BCNF form.

The only thing that breaks BCNF is in the table Card_With_Art, as artID is one the left hand side of an FD but is not a superkey.

As such we perform a reduction on artID -> cardName, and that gets us:

Card_With_Art (cardName: char[64], artID: integer)

Components_Of_Card(setName: char[32], **artID**: integer, rarity: char[1], mana: integer, stats: char[7])

And now these two are correctly in BCNF form.

For all other references to cardName as the key, it will now have artID as the key.

SQL DDL Creation Statements:

We know that oracle does not support on update cascade, but we will keep it as a conceptual reference.

```
CREATE TABLE Platform(  
    platformName: char[64],  
    platformLink: char[256],  
    PRIMARY KEY (platformName),  
)
```



```
CREATE TABLE Listings_On_Platforms(  
platformName: char[64],  
listingID: integer,  
PRIMARY KEY (platformName, listingID),  
FOREIGN KEY (platformName) REFERENCES Platform  
ON DELETE SET NULL  
ON UPDATE CASCADE,  
FOREIGN KEY (listingID) REFERENCES Card_Listing  
ON DELETE CASCADE  
ON UPDATE CASCADE,  
)
```

```
CREATE TABLE Card_Listing(  
listingID: integer,  
Seller: char[32],  
artID: integer NOT NULL,  
setName: char[32] NOT NULL,  
PRIMARY KEY (listingID),  
FOREIGN KEY (artID) REFERENCES Card_With_Art  
ON DELETE CASCADE  
ON UPDATE CASCADE,  
FOREIGN KEY (setName)  
ON DELETE CASCADE  
ON UPDATE CASCADE,  
)
```

```
CREATE TABLE Drew_Art(  
    artID: integer,  
    artistName: char[32],  
    PRIMARY KEY (artID),  
    FOREIGN KEY (artistName) REFERENCES Artist  
    ON DELETE CASCADE  
    ON UPDATE CASCADE,  
)
```

```
CREATE TABLE Artist(  
    artistName: char[32],  
    artistSite: char[256],  
    fanNumber: integer,  
    PRIMARY KEYS(artistName)  
)
```

```
CREATE TABLE Card_With_Art(  
    cardName: char[64],  
    artID: integer NOT NULL,  
    PRIMARY KEY(artID),  
    FOREIGN KEY(artID) REFERENCES Drew_Art  
    ON DELETE CASCADE  
    ON UPDATE CASCADE,  
)
```

```
CREATE TABLE Components_Of_Card(  
  setName: char[32],  
  artID: integer NOT NULL,  
  rarity: char[1] NOT NULL,  
  mana: integer,  
  Stats: char[7],  
  PRIMARY KEY(setName),  
  FOREIGN KEY(setName) REFERENCES Set  
  ON DELETE CASCADE  
  ON UPDATE CASCADE,  
  FOREIGN KEY(artID) REFERENCES Card_With_Art  
  ON DELETE CASCADE  
  ON UPDATE CASCADE,  
)
```

```
CREATE TABLE Set(  
  setName: char[32],  
  releaseYear: integer,  
  PRIMARY KEY(setName),  
)
```

```
CREATE TABLE Types_of_Card(  
    artID: integer,  
    setName: char[32],  
    typeName: char[8],  
    PRIMARY KEY(artID, setName, typeName),  
    FOREIGN KEY(typeName) REFERENCES Type  
    ON DELETE CASCADE  
    ON UPDATE CASCADE,  
    FOREIGN KEY(artID) REFERENCES Card_With_Art  
    ON DELETE CASCADE  
    ON UPDATE CASCADE,  
    FOREIGN KEY(setName) REFERENCES Components_Of_Card  
    ON DELETE CASCADE  
    ON UPDATE CASCADE,  
)
```

```
CREATE TABLE Type(  
    typeName: char[8],  
    playSpeed: char[1],  
    PRIMARY KEY(typeName),  
)
```

```
CREATE TABLE Creator(  
creatorID: char[64],  
creatorRating: double,  
PRIMARY KEY(creatorID),  
)
```

```
CREATE TABLE Deck_From_Creator(  
deckID: integer,  
deckName: char[64],  
playFormat: char[4],  
creatorID: char[64],  
PRIMARY KEY(deckID),  
FOREIGN KEY(creatorID) REFERENCES Creator  
ON DELETE SET DEFAULT,  
ON UPDATE CASCADE,  
)
```

```
CREATE TABLE Card_In_Deck(  
deckID: integer,  
artID: integer,  
setName: char[32],  
PRIMARY KEY(deckID, artID, setName),  
FOREIGN KEY(deckID) REFERENCES Deck_From_Creator,  
FOREIGN KEY(artID) REFERENCES Card_With_Art,  
FOREIGN KEY(setName) REFERENCES Components_Of_Card,  
)
```

INSERT Statements:

Platform:

1. INSERT
 INTO Platform(platformName, platformLink)
 VALUES ('MTGGoldfish', '<https://www.mtggoldfish.com/>')
2. INSERT
 INTO Platform(platformName, platformLink)
 VALUES ('401 Games', '<https://store.401games.ca/>')
3. INSERT
 INTO Platform(platformName, platformLink)
 VALUES ('TCGplayer', '<https://www.tcgplayer.com/>')
4. INSERT
 INTO Platform(platformName, platformLink)
 VALUES ('Card Kingdom', '<https://www.cardkingdom.com/>')
5. INSERT
 INTO Platform(platformName, platformLink)
 VALUES ('Face to Face Games', '<https://www.facetofacegames.com/>')

Listings_On_Platform:

1. INSERT
 INTO Listings_On_Platform(platformName, listingID)
 VALUES ('MTGGoldfish', 211965)
2. INSERT
 INTO Listings_On_Platform(platformName, listingID)
 VALUES ('401 Games', 689028)

3. INSERT

INTO Listings_On_Platform(platformName, listingID)

VALUES ('TCGPlayer', 171228)

4. INSERT

INTO Listings_On_Platform(platformName, listingID)

VALUES ('Card Kingdom', 567164)

5. INSERT

INTO Listings_On_Platform(platformName, listingID)

VALUES ('Face to Face Games', 896019)

Card_Listing:

1. INSERT

INTO Card_Listing(listingID, Seller, artID, setName)

VALUES (211965, 'Pedram', 872019, 'March of the Machine')

2. INSERT

INTO Card_Listing(listingID, Seller, artID, setName)

VALUES (689028, 'Alkiviadis', 550603, 'Commander 2016')

3. INSERT

INTO Card_Listing(listingID, Seller, artID, setName)

VALUES (171228, 'Sham'a', 921559, 'Dominaria Remastered')

4. INSERT

INTO Card_Listing(listingID, Seller, artID, setName)

VALUES (567164, 'Hel', 687437, 'Eventide')

5. INSERT

INTO Card_Listing(listingID, Seller, artID, setName)

VALUES (896019, 'Ultán', 621525, 'Wilds of Eldraine')

Drew_Art:

1. INSERT
 INTO Drew_Art(artID, artistName)
 VALUES (872019, 'Kekai Kotaki')
2. INSERT
 INTO Drew_Art(artID, artistName)
 VALUES (550603, 'Ryan Alexander Lee')
3. INSERT
 INTO Drew_Art(artID, artistName)
 VALUES (921559, 'Jim Nelson')
4. INSERT
 INTO Drew_Art(artID, artistName)
 VALUES (687437, 'Dave Allsop')
5. INSERT
 INTO Drew_Art(artID, artistName)
 VALUES (621525, 'Vincent Christiaens')

Artist:

1. INSERT
 INTO Artist(artistName, artistSite, fanNumber)
 VALUES ('Kekai Kotaki', '<http://www.kekaiart.com/>', 7670)
2. INSERT
 INTO Artist(artistName, artistSite, fanNumber)
 VALUES ('Ryan Alexander Lee', '<https://www.ryanleeart.com/>', 91656)
3. INSERT
 INTO Artist(artistName, artistSite, fanNumber)
 VALUES ('Jim Nelson', '<https://jimnelsonart.com/>', 926)

4. INSERT

INTO Artist(artistName, artistSite, fanNumber)

VALUES ('Dave Allsop', '<https://www.artstation.com/daveallsop>', 4507)

5. INSERT

INTO Artist(artistName, artistSite, fanNumber)

VALUES ('Vincent Christiaens', '<https://vincentchristiaens.pb.online/>', 5810)

Card_With_Art:

1. INSERT

INTO Card_With_Art(cardName, artID)

VALUES ('Invasion of Kamigawa', 872019)

2. INSERT

INTO Card_With_Art(cardName, artID)

VALUES ('Commander's Sphere', 550603)

3. INSERT

INTO Card_With_Art(cardName, artID)

VALUES ('Obsessive Search', 921559)

4. INSERT

INTO Card_With_Art(cardName, artID)

VALUES ('Slippery Bogle', 687437)

5. INSERT

INTO Card_With_Art(cardName, artID)

VALUES ('Plunge into Winter', 621525)

Components_Of_Card:

1. INSERT

INTO Components_Of_Card(setName, artID, rarity, mana, stats)

VALUES ('March of the Machine', 872019, 'U', 4, NULL)

2. INSERT

INTO Components_Of_Card(setName, artID, rarity, mana, stats)

VALUES ('Commander 2016', 550603, 'C', 3, NULL)

3. INSERT

INTO Components_Of_Card(setName, artID, rarity, mana, stats)

VALUES ('Dominaria Remastered', 921559, 'C', 1, NULL)

4. INSERT

INTO Components_Of_Card(setName, artID, rarity, mana, stats)

VALUES ('Eventide', 687437, 'C', 1, 1/1)

5. INSERT

INTO Components_Of_Card(setName, artID, rarity, mana, stats)

VALUES ('Wilds of Eldraine', 621525, 'C', 2, NULL)

Set:

1. INSERT

INTO Set(setName, releaseYear)

VALUES ('March of the Machine', 2023)

2. INSERT

INTO Set(setName, releaseYear)

VALUES ('Commander 2016', 2016)

3. INSERT

INTO Set(setName, releaseYear)

VALUES ('Domaria Remastered', 2023)

4. INSERT

INTO Set(setName, releaseYear)

VALUES ('Eventide', 2008)

5. INSERT

INTO Set(setName, releaseYear)

VALUES ('Wilds of Eldraine', 2023)

Types_Of_Card:

1. INSERT

INTO Types_Of_Card(artID, setName, typeName)

VALUES (872019, 'March of the Machine', 'B')

2. INSERT

INTO Types_Of_Card(artID, setName, typeName)

VALUES (550603, 'Commander 2016', 'A')

3. INSERT

INTO Types_Of_Card(artID, setName, typeName)

VALUES (921559, 'Domaria Remastered', 'S')

4. INSERT

INTO Types_Of_Card(artID, setName, typeName)

VALUES (687437, 'Eventide', 'C')

5. INSERT

INTO Types_Of_Card(artID, setName, typeName)

VALUES (621525, 'Wilds of Eldraine', 'I')

Type:

1. INSERT

INTO Type(typeName, playSpeed)

VALUES ('B', 'S')

2. INSERT

INTO Type(typeName, playSpeed)

VALUES ('A', 'S')

3. INSERT

INTO Type(typeName, playSpeed)

VALUES ('S', 'S')

4. INSERT

INTO Type(typeName, playSpeed)

VALUES ('C', 'S')

5. INSERT

INTO Type(typeName, playSpeed)

VALUES ('I', 'I')

Creator:

1. INSERT

INTO Creator(creatorID, creatorRating)

VALUES ('Darth_Zharlner', 1.99)

2. INSERT

INTO Creator(creatorID, creatorRating)

VALUES ('Hatsu', 2.45)

3. INSERT

INTO Creator(creatorID, creatorRating)

VALUES ('Pariah462', 3.20)

4. INSERT

INTO Creator(creatorID, creatorRating)

VALUES ('sumdumname', 4.90)

```
5. INSERT
    INTO      Creator(creatorID, creatorRating)
    VALUES   ('Anonymous', 2.78)
```

Deck_From_Creator:

```
1. INSERT
    INTO      Deck_From_Creator(deckID, deckName, playFormat, creatorID)
    VALUES   (296753, 'Ninja Tribal', 'STAN', 'Darth_Zharlner')

2. INSERT
    INTO      Deck_From_Creator(deckID, deckName, playFormat, creatorID)
    VALUES   (99275, 'Trap Cards', 'COMM', 'Hatsu')

3. INSERT
    INTO      Deck_From_Creator(deckID, deckName, playFormat, creatorID)
    VALUES   (977177, 'Madness Discard', 'PAUP', 'Pariah462')

4. INSERT
    INTO      Deck_From_Creator(deckID, deckName, playFormat, creatorID)
    VALUES   (864695, 'Keyword Soup', 'COMM', 'sumdumname')

5. INSERT
    INTO      Deck_From_Creator(deckID, deckName, playFormat, creatorID)
    VALUES   (261092, 'Modern Tap Down', 'MDRN', 'Anonymous')
```

Card_In_Deck:

```
1. INSERT
    INTO      Card_In_Deck(deckID, artID, setName)
    VALUES   (296753, 872019, 'March of the Machine')

2. INSERT
    INTO      Card_In_Deck(deckID, artID, setName)
    VALUES   (99275, 550603, 'Commander 2016')
```

3. INSERT

INTO Card_In_Deck(deckID, artID, setName)

VALUES (977177, 921559, 'Dominaria Remastered')

4. INSERT

INTO Card_In_Deck(deckID, artID, setName)

VALUES (864695, 687437, 'Eventide')

5. INSERT

INTO Card_In_Deck(deckID, artID, setName)

VALUES (261092, 621525, 'Wilds of Eldraine')