Game Information – Database Project
Group 8

Image Courtesy:
https://en.wikipedia.org/wiki/God_of_War_(2018_video_game)#/media/File:God_of_War_4_cover.jpg

Section 1: Overview

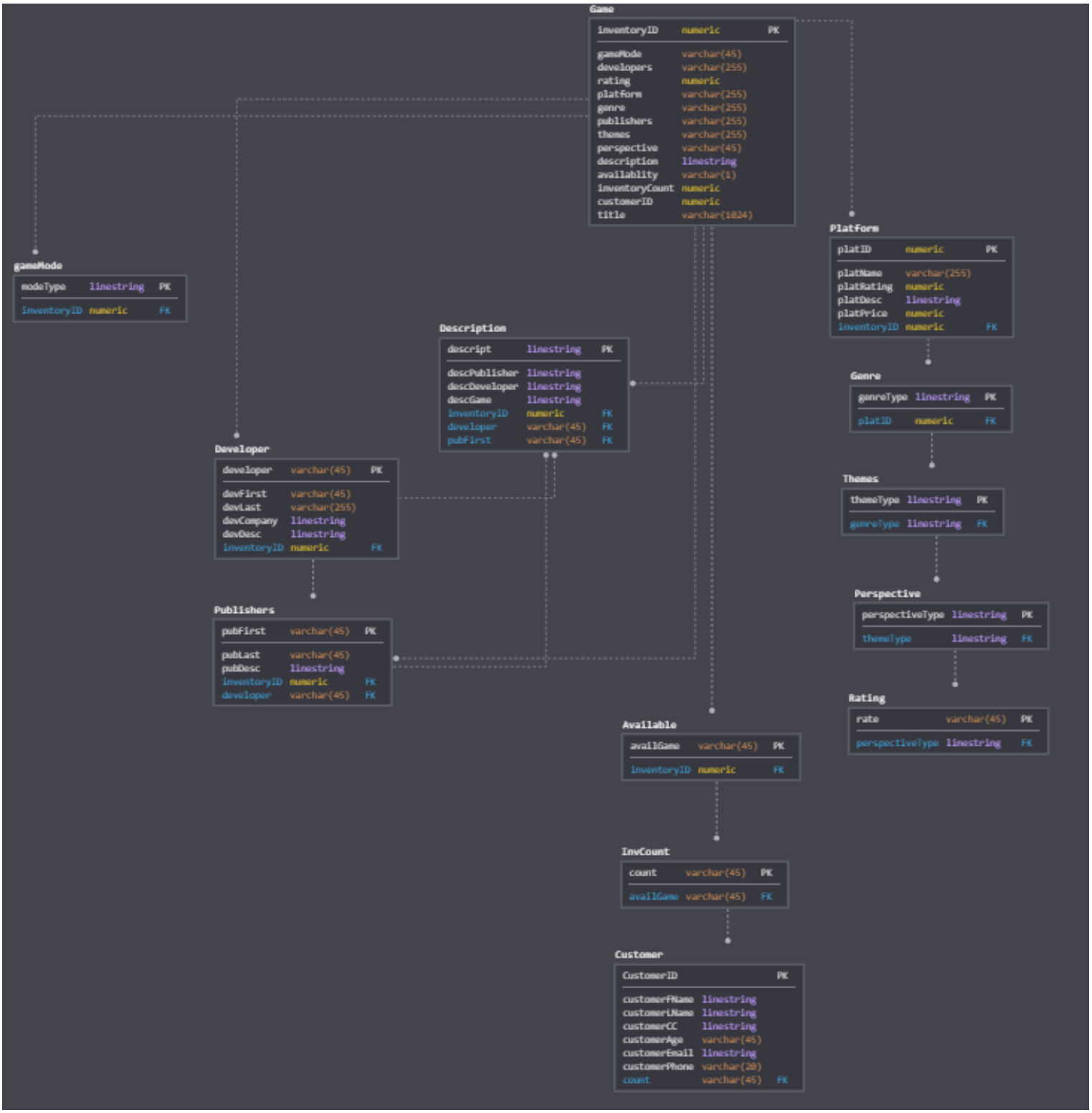
We have been tasked to create the back-end of a database; with it's intended use as informational only. The database's sole purpose is to provide the end user with as much information as possible that there is about the games available. Our vision was to create an expansive database which served the end user's needs, for example: finding out when a game was created, the release date of future games, information about certain sound tracks, the names of developers/publishers/creators of games, where the idea originated, and overall rating. These are just few ideas of what the database could've been. Considering that this is the prototype, we were able to develop key points of the Game Database. In the future it could be turned into a subscription based database, or even free like Wikipedia, derived solely on the donations of it's users. If the "paid subscription" is implemented, we could then create a front end website that allowed heavier loads of traffic and more expansive information to be created and stored.

Our functional specification for the utilities for such an application are as follows:

- We must store all key pieces of information about the game – Master Game Table.
- The Game must include child tables that allow for more descriptive information; if needed.
- The game must include a user rating system, that is solely based on the games available. If this is implemented, we must include critic reviews; To allow no-biased ratings
- The social aspect provided in the database is currently as registering your first and last name along with age and email (and credit card) in case you wanted to donate.
- We must keep COUNT of the available amount of games in our database that have been collected; Considering the database is still in BETA, we did not include a customer rental, and a table to record the amount of copies that have been rented out. But this can be implemented in the future.
- A user can currently query the database if they wanted to find out certain information about games. They can look at anything from the title of the game to the developer/publisher and soon to be tracks developed for the game.

Overall the database stores a variety of information in order to allow for such functionality. We have thought of various pieces of information to apply to the database to allow extensive information to be stored and queried when needed. Like we mentioned before, since the database is still in BETA, there is still more room to include more pieces of information.

Section 2: Comprehensive UML Diagram of Database



Section 3: Functional Dependencies

Our database follows the schema that was taught in class. However we still have a little redundancy; but this is because we want to allow the user to find more information than the basic items. In the future it can be built to reduce the amount of redundancy.

Section 4: Assertion

```
-- ****************** This has been created by Yousef
-- ************************************** `Game`
CREATE TABLE Game
(
 InventoryID     number(10,0) NOT NULL ,
 gameMode        varchar2(20) NOT NULL ,
 Developers      varchar2(20) NOT NULL ,
 Rating          number(10,0) NOT NULL ,
 Platform        varchar2(20) NOT NULL ,
 Genre           varchar2(20) NOT NULL ,
 Publishers      varchar2(20) NOT NULL ,
 Themes          varchar2(20) NOT NULL ,
 Perspective     varchar2(20) NOT NULL ,
 Description     varchar2(255) NOT NULL ,
 Availablity     varchar2(1) NOT NULL ,
 InventoryCount number(10,0) NOT NULL ,
 CustomerID      number(10,0) NOT NULL ,
 title           varchar2(20) NOT NULL,
PRIMARY KEY (inventoryID)
);

-- ************************************** `Platform`
CREATE TABLE Platform
(
 platID     number(10,0) NOT NULL ,
 platname    varchar2(25) NOT NULL ,
 platrating  number(10,0) NOT NULL ,
 platdesc    varchar2(25) NOT NULL,
 platprice   number(10,0) NOT NULL ,
 InventoryID number(10,0) NOT NULL ,
PRIMARY KEY (platID)
);



CREATE INDEX FK1 ON Platform (InventoryID);

ALTER TABLE Platform
ADD CONSTRAINT FK20
  FOREIGN KEY (inventoryID)
  REFERENCES Game (inventoryID);
```

```
-- ************************************** `gameMode`
CREATE TABLE gameMode
 (
  modeType varchar(45) NOT NULL ,
  InventoryID number(10,0) NOT NULL ,
  PRIMARY KEY (modeType)
 );

CREATE INDEX FK2 ON gameMode (inventoryID);

ALTER TABLE gameMode
ADD CONSTRAINT FK21
 FOREIGN KEY (inventoryID)
 REFERENCES Game (inventoryID);


-- ************************************** `Developer`

CREATE TABLE Developer
(
 developer    varchar2(45) NOT NULL ,
 devFirst     varchar2(10) NOT NULL ,
 devLast      varchar2(10) NOT NULL ,
 devCompany   varchar2(16) NOT NULL ,
 devDesc      varchar2(30) NOT NULL ,
 inventoryID  number(10) NOT NULL ,
PRIMARY KEY (developer)
);

CREATE INDEX FK3 ON Developer (inventoryID);

ALTER TABLE Developer
ADD CONSTRAINT FK22
 FOREIGN KEY (inventoryID)
 REFERENCES Game (inventoryID);


-- ************************************** `Available`
CREATE TABLE Available
(
 availGame   varchar2(1) NOT NULL ,
 inventoryID number(10) NOT NULL ,
PRIMARY KEY (availGame)
);

CREATE INDEX FK4 ON Available (inventoryID);

ALTER TABLE Available
ADD CONSTRAINT FK23
 FOREIGN KEY (inventoryID)
 REFERENCES Game (inventoryID);
```

```
-- ************************************** `Publishers`

CREATE TABLE Publishers
(
 pubFirst    varchar2(17) NOT NULL ,
 pubLast     varchar2(17) NOT NULL ,
 pubDesc      varchar2(30) NOT NULL ,
 inventoryID number(10) NOT NULL ,
 developer   varchar2(45) NOT NULL ,
PRIMARY KEY (pubFirst)
);

CREATE INDEX FK5 ON Publishers (inventoryID);

ALTER TABLE Publishers
ADD CONSTRAINT FK24
 FOREIGN KEY (inventoryID)
 REFERENCES Game (inventoryID);

ALTER TABLE Publishers
ADD CONSTRAINT FK25
 FOREIGN KEY (developer)
 REFERENCES Developer (developer);

-- ************************************** `InvCount`

CREATE TABLE InvCount
(
 count  varchar2(10) NOT NULL ,
 availGame varchar2(1) NOT NULL ,
PRIMARY KEY (count)
);

ALTER TABLE InvCount
ADD CONSTRAINT FK26
 FOREIGN KEY (availGame)
 REFERENCES Available (availGame);


-- ************************************** `Genre`

CREATE TABLE Genre
(
 genreType varchar2(11) NOT NULL ,
 platID    number(10) NOT NULL ,
PRIMARY KEY (genreType)
);

CREATE INDEX FK7 ON Genre (platID);

ALTER TABLE Genre
ADD CONSTRAINT FK27
 FOREIGN KEY (platID)
```

```
  REFERENCES Platform (platID);

   -- ************************************** `Themes`

   CREATE TABLE Themes
   (
    themeType varchar2(16) NOT NULL ,
    genreType varchar2(16) NOT NULL ,
   PRIMARY KEY (themeType)
   );

CREATE INDEX FK8 ON Themes (genreType);

ALTER TABLE Themes
ADD CONSTRAINT FK28
 FOREIGN KEY (genreType)
 REFERENCES Genre (genreType);

-- ************************************** `Description`
CREATE TABLE Description
(
 descript      varchar2(255) NOT NULL ,
 descPublisher  varchar2(30) NOT NULL ,
 descDeveloper  varchar2(30) NOT NULL ,
 descGame       varchar2(255) NOT NULL ,
 inventoryID   number(10) NOT NULL ,
 developer     varchar2(20) NOT NULL ,
 pubFirst      varchar2(10) NOT NULL ,
PRIMARY KEY (descript)
);

CREATE INDEX FK9 ON Description (developer);

ALTER TABLE Description
ADD CONSTRAINT FK29
  FOREIGN KEY (developer)
  REFERENCES Developer (developer);

ALTER TABLE Description
ADD CONSTRAINT FK30
  FOREIGN KEY (pubFirst)
  REFERENCES Publishers (pubFirst);

ALTER TABLE Description
ADD CONSTRAINT FK31
  FOREIGN KEY (inventoryID)
  REFERENCES Game (inventoryID);

-- ************************************** `Customer`

CREATE TABLE Customer
(
 CustomerID    varchar2(5) NOT NULL ,
```

```
 customerFName  varchar2(10) NOT NULL ,
 customerLName  varchar2(10) NOT NULL ,
 customerCC     varchar2(21) NOT NULL ,
 customerDOB   varchar2(13) NOT NULL ,
 customerEmail  varchar2(22) NOT NULL ,
 customerPhone varchar2(13) NOT NULL ,
 invCount      varchar2(8) NOT NULL ,
PRIMARY KEY (CustomerID)
);

CREATE INDEX FK10 ON Customer (invCount);

ALTER TABLE Customer
ADD CONSTRAINT FK32
 FOREIGN KEY (invCount)
 REFERENCES InvCount (count);

 -- ************************************ `Perspective`

 CREATE TABLE Perspective
 (
  perspectiveType varchar2(16) NOT NULL ,
  themeType varchar2(16) NOT NULL ,
 PRIMARY KEY (perspectiveType)
);

CREATE INDEX FK11 ON Perspective (themeType);

ALTER TABLE Perspective
ADD CONSTRAINT FK33
 FOREIGN  KEY (themeType)
 REFERENCES Themes (themeType);

 -- ************************************ `Rating`

 CREATE TABLE Rating
 (
  rate  varchar2(45) NOT NULL ,
  perspectiveType varchar(16) NOT NULL ,
 PRIMARY KEY (rate)
 );




 CREATE INDEX FK12 ON Rating (perspectiveType);

 ALTER TABLE Rating
 ADD CONSTRAINT FK34
  FOREIGN KEY (perspectiveType)
  REFERENCES Perspective (perspectiveType);
```

```
INSERT INTO Game
(InventoryID, gameMode, Developers, Rating, Platform, Genre, Publishers, Themes,
Perspective, Description, Availablity, InventoryCount, CustomerID, Title)
VALUES
(1, 'SP', 'Santa Monica STD', 96, 'PS4', 'Adventure', 'Sony ENT', 'Mythology', '3rd Person', 'Play as
Kratos in the new 2018 Game', 'T', 1, 0001, 'God of War');

INSERT INTO Game
(InventoryID, gameMode, Developers, Rating, Platform, Genre, Publishers, Themes,
Perspective, Description, Availablity, InventoryCount, CustomerID, Title)
VALUES
(2, 'SP', 'Capcom', 92, 'XB1', 'Hack/Slash', 'Capcom', 'Fantasy', '3rd Person', 'Play as three demon
hunters to save humanity from a demonic invasion', 'T', 1, 0002, 'Devil May Cry 5');

INSERT INTO PLATFORM
(PLATID, PLATNAME, PLATRATING, PLATDESC, PLATPRICE, INVENTORYID)
VALUES
(1, 'PS4', 96, 'DEVELOPED BY SONY', 269, 1);

INSERT INTO PLATFORM
(PLATID, PLATNAME, PLATRATING, PLATDESC, PLATPRICE, INVENTORYID)
VALUES
(2, 'PC', 100, 'PC Master Race', 1000, 1);

INSERT INTO PLATFORM
(PLATID, PLATNAME, PLATRATING, PLATDESC, PLATPRICE, INVENTORYID)
VALUES
(3, 'XB1', 90, 'DEVELOPED BY MICROSOFT', 499, 1);

INSERT INTO gameMode
(modeType, inventoryID)
VALUES
('SP', 1);

INSERT INTO Developer
(DEVELOPER, DEVFIRST, DEVLAST, DEVCOMPANY, DEVDESC, INVENTORYID)
VALUES
('READY AT DAWN', 'SCOTT', 'ROGERS', 'SANTA MONICA STD', 'VIDEO GAME DESIGNER', 1);

INSERT INTO Developer
(DEVELOPER, DEVFIRST, DEVLAST, DEVCOMPANY, DEVDESC, INVENTORYID)
VALUES
('Campcomi LTD', 'Hideaki', 'Itsuno', 'Capcom', 'Director', 2);

INSERT INTO AVAILABLE
(AVAILGAME, INVENTORYID)
VALUES
('T', 1);

INSERT INTO PUBLISHERS
(PUBFIRST, PUBLAST, PUBDESC, INVENTORYID, DEVELOPER)
VALUES
('Hirai', 'Kaz', 'CEO OF SONY ENT', 1, 'READY AT DAWN');
```

```
INSERT INTO PUBLISHERS
(PUBFIRST, PUBLAST, PUBDESC, INVENTORYID, DEVELOPER)
VALUES
('Kenzo', 'Tsujimoto', 'CEO OF CAPCOM LTD', 2, 'Capcom');

INSERT INTO INVCOUNT
(COUNT, AVAILGAME)
VALUES
(1, 'T');

INSERT INTO GENRE
(GENRETYPE, PLATID)
VALUES
('Adventure', 1 );

INSERT INTO GENRE
(GENRETYPE, PLATID)
VALUES
('Hack/Slash', 3);

INSERT INTO THEMES
(THEMETYPE, GENRETYPE)
VALUES
('Mythology', 'Adventure');


INSERT INTO THEMES
(THEMETYPE, GENRETYPE)
VALUES
('Action', 'Hack/Slash');

INSERT INTO DESCRIPTION
(DESCRIPT, DESCPUBLISHER, DESCDEVELOPER, DESCGAME, INVENTORYID, DEVELOPER,
PUBFIRST)
VALUES
('Table of Information', 'CEO OF SONY ENT', 'VIDEO GAME DESIGNER', 'Play as Kratos in the new
2018 Game', 1, 'READY AT DAWN', 'Hirai' );

INSERT INTO DESCRIPTION
(DESCRIPT, DESCPUBLISHER, DESCDEVELOPER, DESCGAME, INVENTORYID, DEVELOPER,
PUBFIRST)
VALUES
('Info', 'CEP OF CAPCOM', 'Game Director', 'Play as three demon hunters to save humanity from a
demonic invasion', 1, 'Capcom','Kenzo');

INSERT INTO Customer
(CUSTOMERID, CUSTOMERFNAME, CUSTOMERLNAME, CUSTOMERCC, CUSTOMERDOB,
CUSTOMEREMAIL,
CUSTOMERPHONE, INVCOUNT)
VALUES
(0001, 'Yousef', 'Jarrar', '4342-5620-8563-5541', '27', '005845836@csusb.edu', '909-537-7266', 1);
```

```
INSERT INTO Customer
(CUSTOMERID, CUSTOMERFNAME, CUSTOMERLNAME, CUSTOMERCC, CUSTOMERDOB,
CUSTOMEREMAIL,
CUSTOMERPHONE, INVCOUNT)
VALUES
(0002, 'Harry', 'Hacker', '1234-4567-8910-3942', '30', '002345678@csusb.edu', '909-321-4321', 1);

INSERT INTO Customer
(CUSTOMERID, CUSTOMERFNAME, CUSTOMERLNAME, CUSTOMERCC, CUSTOMERDOB,
CUSTOMEREMAIL,
CUSTOMERPHONE, INVCOUNT)
VALUES
(0003, 'Suarez', 'George', '9876-3214-5942-4521', '25', '006098556@csusb.edu', '909-352-1235', 1);

INSERT INTO PERSPECTIVE
(PERSPECTIVETYPE, THEMETYPE)
VALUES
('3rd Person', 'Mythology');

INSERT INTO RATING
(RATE, PERSPECTIVETYPE)
VALUES
(96, '3rd Person');

INSERT INTO RATING
(RATE, PERSPECTIVETYPE)
VALUES
(92, '3rd Person');



CREATE OR REPLACE TRIGGER print_game_info
BEFORE INSERT OR UPDATE ON Game
FOR EACH ROW
BEGIN
        dbms_output.put_line('Title: ' || :new.title);
        dbms_output.put_line('Rating: ' || :new.Rating);
        dbms_output.put_line('Platform: ' || :new.Platform);
        dbms_output.put_line('Genre: ' || :new.Genre);
        dbms_output.put_line('Description: ' || :new.Description);
        dbms_output.put_line('Availability: ' || :new.Availablity);
        dbms_output.put_line('InventoryCount: ' || :new.InventoryCount || chr(10));
END;
/

CREATE OR REPLACE TRIGGER inv_count_is_zero
BEFORE INSERT OR UPDATE  on Game
FOR EACH ROW
BEGIN
        IF :new.InventoryCount = 0 THEN :new.Availablity := 'F';
        ELSE :new.Availablity := 'T';
        END IF;
```

```
END;
/

CREATE OR REPLACE TRIGGER increase_inv_count
BEFORE INSERT on Game
FOR EACH ROW
BEGIN
        UPDATE Game g
        SET g.InventoryCount = g.InventoryCount + :new.InventoryCount
        WHERE g.CustomerID = :new.CustomerID;
END;
/

CREATE OR REPLACE TRIGGER deleted_customer
BEFORE DELETE on Customer
FOR EACH ROW
BEGIN
        dbms_output.put_line('Deleted Customer: ' || :old.CustomerFName);
END;
/



UPDATE Game
SET InventoryCount = InventoryCount + 1;


CREATE VIEW V1 AS
SELECT title, Description, Availablity, InventoryCount, CustomerID
FROM Game
WHERE InventoryID IS NOT NULL;



DELETE FROM Customer c
WHERE EXISTS (SELECT g.CustomerID
    FROM Game g
    WHERE c.CustomerID = g.CustomerID);
```