Kyle Schryver
Final Project
OMIS 105

Proposal

Sapphire Entertainment (SE) is an entertainment company that designs, produces, and sells video games to a market in the United States. The company is based in San Francisco, California and has been around since 2016. SE has around 100 employees, has produced 5 unique games, and has about 10,000 active users on their games. Some games are required to be purchased before they can be played, while others offer a free-to-play model. The company has grown very quickly, but has suffered from poor information storage and accessibility. For purposes of organization and analysis, the company wishes to implement a database, which they believe will allow them to grow faster while maintaining good relations with customers and ensuring things stay in check.

A video game refers to one of the games developed and produced by SE. Games can also be in development stages or not for sale. Games can be purchased or downloaded by any number of customers, given that the game has been released. Games have the following attributes: Game ID, Release Date (Optional), Name. Games can be further sorted into 2 (and only 2) types: free-to-play (known as F2P) and pay-to-play (known as P2P). P2P games have an attribute of price. Games cannot be both F2P and P2P. Games must be one of the two types.

Customers may purchase any number of P2P video games. Customers may also download any number of F2P video games. Customer information includes:Customer ID, Name, IP Address. If a customer downloads a free game, they are given a free account with the following information: Free Account ID and Free Character Information (which is a composite attribute including Character Name, Character Type, and Character Creation Date). If a customer buys a game, they are given a premium account with the following information: Premium Account ID, Premium Status. Accounts are kept separate for different games and are used to track character information and premium status respectively. An account is bound to one customer and one game, but a game could have many accounts and a customer could have many accounts.

At SE all games that are published need a server to run. These servers have the following attributes and can host zero or one game: Server ID, Server Name, Bandwidth. Servers also require at least 1 full time administrator assigned to them, and administrators can work on any number of servers. “Servers” are considered to be a software architecture hosting a game. Servers also require hardware available at SE in order to function. The physical hardware required for servers will be known as “Supercomputers”. Supercomputers have a Supercomputer ID, Capacity, and LocationArea. Servers need one and exactly one supercomputer to run on, and one supercomputer can support at most one server. Servers cannot exist without a supercomputer, but a supercomputer can exist without a server. Supercomputers are assigned at least one hardware engineer, and hardware engineers can be assigned to many supercomputers.

There are three (and only 3) types of employees at SE: Administrators, Developers, and Hardware Engineers. An employee can only be one of the three subtypes, and must be one of the subtypes. Employees have the following attributes: Employee ID and Employee Name. Administrators also have a multivalued attribute known as permission number. Developers have a multivalued attribute of skills. Games can have zero to many developers and developers can work on one to many games. Developers are also managed by one, and only one administrator. Administrators can manage many Developers. Hardware Engineers are managed by another hardware engineer, and one hardware engineer can manage many others.

When developers work on Video Games, they track their check-in time on work and check-out time. They also note the feature of the game which they worked on. When hardware engineers work, they track their time spent on a supercomputer, which is used to indicate downtime for that supercomputer.

Data Dictionary

EMPLOYEES

Name	**Data Type**	**Valid Range of Values**	**Description**
Employee ID	bigint	>0	Unique identifier for employees
Employee Type	char(1)	p, h, or a	Determines the type of employee
Employee Name	nvarchar(100)	Any string	First and last name of employee

HARDWARE ENGINEER

Name	**Data Type**	**Valid Range of Values**	**Description**
Hemployee ID	bigint	>0	Unique identifier for hardware engineer employees

Hardware Manager ID	bigint	>0	Unique identifier to specify managers of hardware engineers

MAINTENANCE

Name	**Data Type**	**Valid Range of Values**	**Description**
Supercomputer ID	bigint	>0	Unique identifier and foreign key for supercomputers
Hemployee ID	bigint	>0	Foreign key to determine what engineer worked on the supercomputer
Time Spent	int	>=0	Specify the amount of time an engineer, in minutes worked on a supercomputer

SUPERCOMPUTER

Name	**Data Type**	**Valid Range of Values**	**Description**
Supercomputer ID	bigint	>0	Unique identifier for supercomputers
Capacity	int	>0	Specify the amount of capacity the supercomputer can handle, in TB
Location	nvarchar(100)	Any string	Specify where the supercomputer is located, uses a number based system tracked as a string

SERVER ADMIN

Name	Data Type	Valid Range of Values	Description
Server ID	bigint	>0	Foreign key and unique identifier for a server
Aemployee ID	bigint	>0	Foreign key and unqiue identifier for what administrator works on a server

SERVER

Name	Data Type	Valid Range of Values	Description
Server ID	bigint	>0	Unique Identifier for a server
Server name	nvarchar(100)	Any string	Specify the name of a server
Bandwidth	int	>0	Specify how much traffic the server can handle, in GB/s
Supercomputer ID	bigint	>0	Specify the foreign key for the associated supercomputer
Game ID	bigint	>0	Specify the associated game’s foreign key

ADMINISTRATOR

Name	Data Type	Valid Range of Values	Description
Aemployee ID	bigint	>0	Unique Identifier for administrator employees

ADMINISTRATOR PERMISSION NUMBER

Name	Data Type	Valid Range of Values	Description
Aemployee ID	bigint	>0	Unique Identifier for administrator employees
Permission Number	char(8)	>0	Specify a permission number that belongs to an administrator

DEVELOPER

Name	Data Type	Valid Range of Values	Description
Demployee ID	bigint	>0	Unique Identifier for a developer employee
Aemployee ID	bigint	>0	Unique Identifier for administrator employees as a foreign key

DEVELOPER SKILLS

Name	Data Type	Valid Range of Values	Description
Demployee ID	bigint	>0	Unique Identifier for a developer employee
Skill	nvarchar(100)	>0	Specify what skill a developer has

DEVELOPMENT

Name	Data Type	Valid Range of Values	Description
Demployee ID	bigint	>0	Unique Identifier for a developer employee
Game ID	bigint	>0	Unique Identifier and

			foreign key for the game being worked on
Check in time	datetime	Any datetime	Specify when the development started
Check out time	datetime	Any datetime	Specify when the development ended
Feature	nvarchar(100)	Any string	Specify what feature was worked on

VIDEO GAME

Name	**Data Type**	**Valid Range of Values**	**Description**
Game ID	bigint	>0	Unique Identifier for a game
Release date	datetime	Any datetime	Specify when a game was released
Game Type	char(1)	p or f	Specify whether the game is F2P or P2P
Video Game Name	nvarchar(100)	Any string	Specify the name of the video game

P2P

Name	**Data Type**	**Valid Range of Values**	**Description**
Pgame ID	bigint	>0	Unique identifier for P2P video games
Price	smallmoney	>0	Specify the price of the video game

F2P

Name	**Data Type**	**Valid Range of Values**	**Description**

Fgame ID	bigint	>0	Unique identifier for F2P video games

DOWNLOAD

Name	**Data Type**	**Valid Range of Values**	**Description**
Customer ID	bigint	>0	Specify the identifier for the customer who downloaded the game
Fgame ID	bigint	>0	Specify what game instance was downloaded

FREE ACCOUNT

Name	**Data Type**	**Valid Range of Values**	**Description**
Free Account ID	bigint	>0	Unique Identifier for a free account
Character Name	nvarchar(100)	Any string	Specify the name of the character
Character Type	nvarchar(100)	Any string	Specify the type of character
Character Creation Date	date	Any date	Specify the date when the character was created
Fgame ID	bigint	>0	Specify the associated game instance for the free account
Customer ID	bigint	>0	Specify the customer tied to the free account

CUSTOMER

Name	**Data Type**	**Valid Range of**	**Description**

		Values	
Customer ID	bigint	>0	Unique identifier for a specific customer
Customer Name	nvarchar(100)	Any string	Specify first and last name of a customer
Customer IP Address	varchar	>0	Specify the IP address of a customer

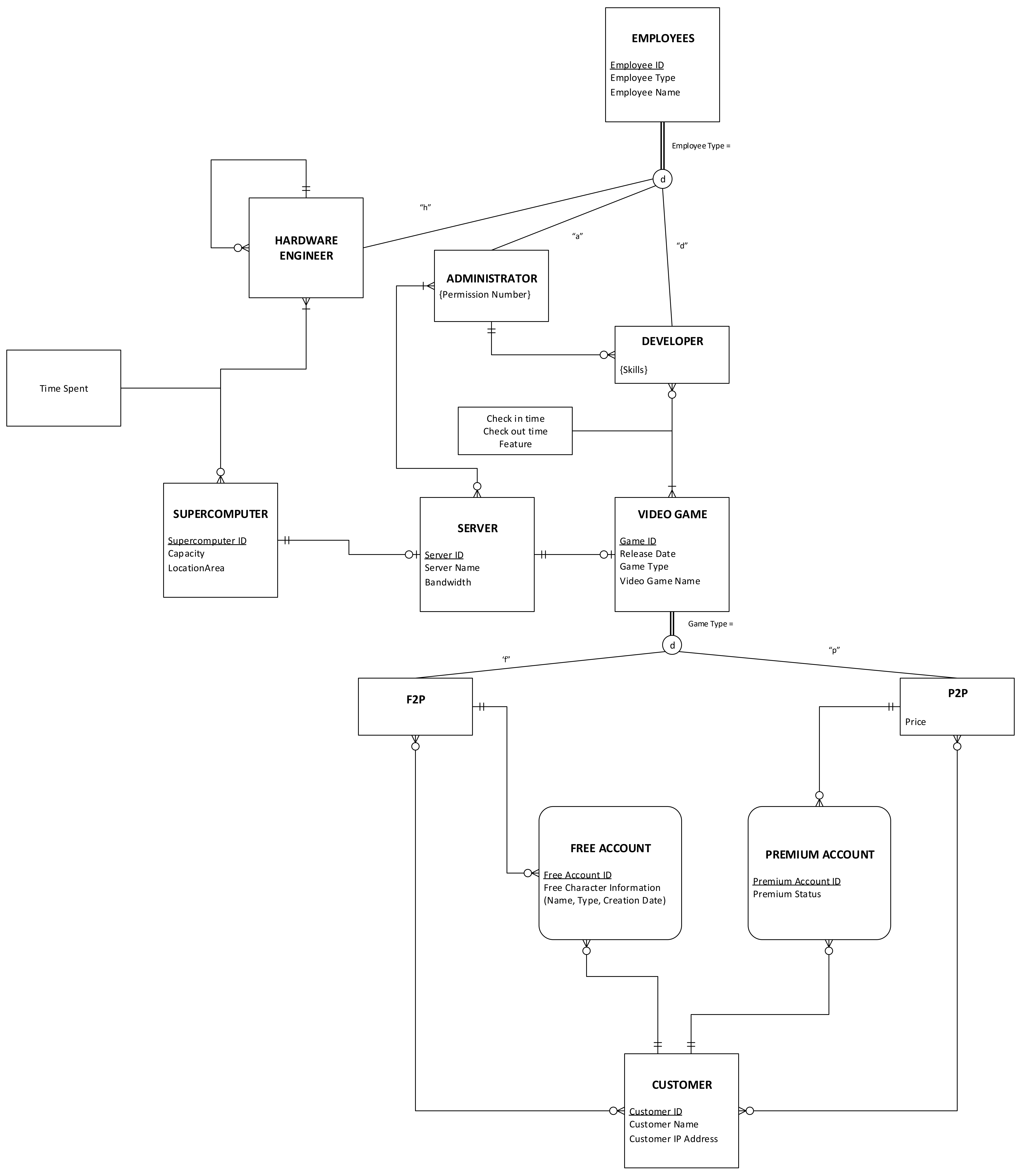
PURCHASE

Name	Data Type	Valid Range of Values	Description
Customer ID	bigint	>0	Unique identifier and foreign key for a customer
Pgame ID	bigint	>0	Unique identifier to specify what game instance the purchase is tied to

PREMIUM ACCOUNT

Name	Data Type	Valid Range of Values	Description
Premium Account ID	bigint	>0	Unique Identifier for a premium account instance
Premium Status	bit	True or false	Determines whether the account still has premium status, 1 for active, 0 for inactive
Customer ID	bigint	>0	Unique identifier and foreign key to specify the associated customer
Pgame ID	bigint	>0	Unique identifier and foreign key for the

			associated game instance


EMPLOYEES
Employee ID
Employee Type
Employee Name
Employee Type =
d
"h"
"a"
"d"
HARDWARE ENGINEER
ADMINISTRATOR
{Permission Number}
DEVELOPER
{Skills}
Time Spent
Check in time
Check out time
Feature
SUPERCOMPUTER
Supercomputer ID
Capacity
LocationArea
SERVER
Server ID
Server Name
Bandwidth
VIDEO GAME
Game ID
Release Date
Game Type
Video Game Name
Game Type =
d
"f"
"p"
F2P
P2P
Price
FREE ACCOUNT
Free Account ID
Free Character Information
(Name, Type, Creation Date)
PREMIUM ACCOUNT
Premium Account ID
Premium Status
CUSTOMER
Customer ID
Customer Name
Customer IP Address

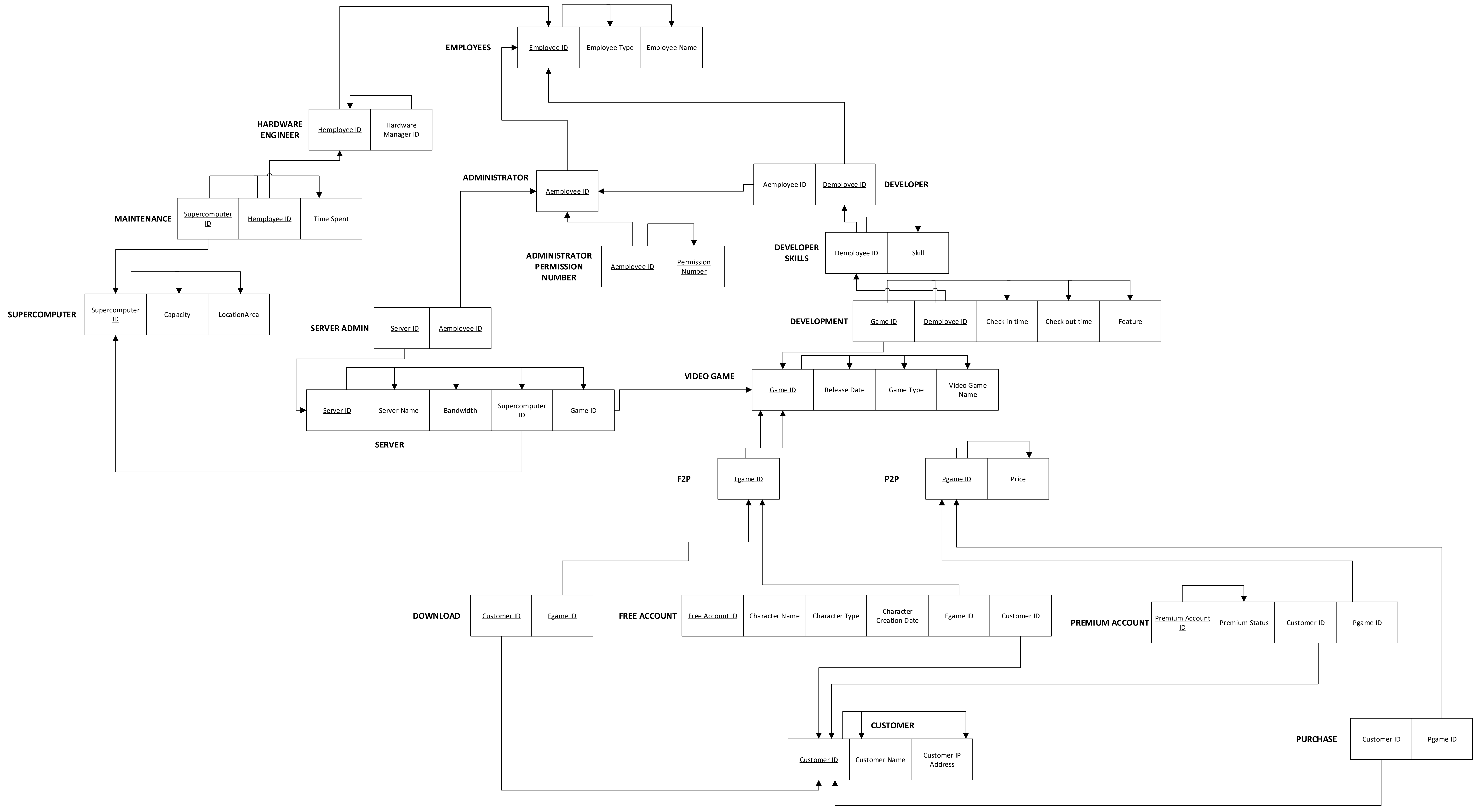

EMPLOYEES
Employee ID
Employee Type
Employee Name
HARDWARE ENGINEER
Hemployee ID
Hardware Manager ID
ADMINISTRATOR
Aemployee ID
DEVELOPER
Aemployee ID
Demployee ID
MAINTENANCE
Supercomputer ID
Hemployee ID
Time Spent
ADMINISTRATOR PERMISSION NUMBER
Aemployee ID
Permission Number
DEVELOPER SKILLS
Demployee ID
Skill
SUPERCOMPUTER
Supercomputer ID
Capacity
LocationArea
SERVER ADMIN
Server ID
Aemployee ID
DEVELOPMENT
Game ID
Demployee ID
Check in time
Check out time
Feature
VIDEO GAME
Game ID
Release Date
Game Type
Video Game Name
Server ID
Server Name
Bandwidth
Supercomputer ID
Game ID
SERVER
F2P
Fgame ID
P2P
Pgame ID
Price
DOWNLOAD
Customer ID
Fgame ID
FREE ACCOUNT
Free Account ID
Character Name
Character Type
Character Creation Date
Fgame ID
Customer ID
PREMIUM ACCOUNT
Premium Account ID
Premium Status
Customer ID
Pgame ID
CUSTOMER
Customer ID
Customer Name
Customer IP Address
PURCHASE
Customer ID
Pgame ID

Table Queries

```
-- Create Employee Table
CREATE TABLE Employee_T
             (EmployeeID          BIGINT     NOT NULL,
                       EmployeeType        CHAR(1)    NOT NULL,
                 EmployeeName              NVARCHAR(100),
CONSTRAINT Employee_PK PRIMARY KEY (EmployeeID));

-- Create Hardware Engineer Table
CREATE TABLE HardwareEng_T
             (HemployeeID         BIGINT     NOT NULL,
                       HmanagerID          BIGINT     NOT NULL,
CONSTRAINT HardwareEng_PK PRIMARY KEY (HemployeeID));

-- Create Supercomputer Table
CREATE TABLE Supercomputer_T
             (SupercomputerID     BIGINT     NOT NULL,
                       Capacity                   INT,
                 LocationArea              NVARCHAR(100),
CONSTRAINT Supercomputer_PK PRIMARY KEY (SupercomputerID));

-- Create MaintenanceTable
CREATE TABLE Maintenance_T
             (SupercomputerID     BIGINT     NOT NULL,
                       HemployeeID         BIGINT     NOT NULL,
                 TimeSpent                 INT,
CONSTRAINT Maintenance_PK PRIMARY KEY (SupercomputerID, HemployeeID),
CONSTRAINT Maintenance_FK1 FOREIGN KEY (HemployeeID) REFERENCES
HardwareEng_T(HemployeeID),
CONSTRAINT Maintenance_FK2 FOREIGN KEY (SupercomputerID) REFERENCES
Supercomputer_T(SupercomputerID));

-- Create Administrator Table
CREATE TABLE Administrator_T
             (AemployeeID        BIGINT     NOT NULL,
CONSTRAINT Administrator_PK PRIMARY KEY (AemployeeID));

-- Create Server Admin Table
CREATE TABLE ServerAdmin_T
             (ServerID            BIGINT     NOT NULL,
                       AemployeeID         BIGINT     NOT NULL,
CONSTRAINT ServerAdmin_FK1 FOREIGN KEY (AemployeeID) REFERENCES
Administrator_T(AemployeeID),
CONSTRAINT ServerAdmin_PK PRIMARY KEY (ServerID, AemployeeID));

-- Create Video Game Table
CREATE TABLE VideoGame_T
             (GameID             BIGINT     NOT NULL,
                       ReleaseDate        DATETIME,
                       GameType          CHAR(1)    NOT NULL,
                       VideoGameName     NVARCHAR(100),
CONSTRAINT VideoGame_PK PRIMARY KEY (GameID));

-- Create Server Table
```

```sql
CREATE TABLE Server_T
             (ServerID           BIGINT     NOT NULL,
                       ServerName          NVARCHAR(100),
                 Bandwidth              INT,
                       SupercomputerID    BIGINT,
                       GameID             BIGINT,
CONSTRAINT Server_FK1 FOREIGN KEY (SuperComputerID) REFERENCES
Supercomputer_T(SupercomputerID),
CONSTRAINT Server_FK2 FOREIGN KEY (GameID) REFERENCES VideoGame_T(GameID),
CONSTRAINT Server_PK PRIMARY KEY (ServerID));

-- Create Administrator Permission Number Table
CREATE TABLE AdministratorPermission_T
             (AemployeeID        BIGINT     NOT NULL,
                       PermissionNumber   CHAR(8)    NOT NULL,
CONSTRAINT AdministratorPermission_FK1 FOREIGN KEY (AemployeeID) REFERENCES
Administrator_T(AemployeeID),
CONSTRAINT AdministratorPermission_PK PRIMARY KEY (AemployeeID, PermissionNumber));

-- Create Developer Table
CREATE TABLE Developer_T
             (DemployeeID        BIGINT     NOT NULL,
                       AemployeeID         BIGINT               NOT NULL,
CONSTRAINT Developer_FK1 FOREIGN KEY (AemployeeID) REFERENCES
Administrator_T(AemployeeID),
CONSTRAINT Developer_PK PRIMARY KEY (DemployeeID));

-- Create Developer Skills Table
CREATE TABLE DeveloperSkills_T
             (DemployeeID        BIGINT     NOT NULL,
                       Skill                            NVARCHAR(100) NOT NULL,
CONSTRAINT DeveloperSkills_FK1 FOREIGN KEY (DemployeeID) REFERENCES
Developer_T(DemployeeID),
CONSTRAINT DeveloperSkills_PK PRIMARY KEY (DemployeeID, Skill));

-- Create Development Table
CREATE TABLE Development_T
             (DemployeeID        BIGINT     NOT NULL,
                       GameID                  BIGINT     NOT NULL,
                       Checkintime        DATETIME,
                       Checkouttime       DATETIME,
                       Feature            NVARCHAR(100),
CONSTRAINT Development_FK1 FOREIGN KEY (DemployeeID) REFERENCES Developer_T(DemployeeID),
CONSTRAINT DeveloperSkills_FK2 FOREIGN KEY (GameID) REFERENCES VideoGame_T(GameID),
CONSTRAINT Development_PK PRIMARY KEY (DemployeeID, GameID));

-- Create Customer Table
CREATE TABLE Customer_T
             (CustomerID         BIGINT     NOT NULL,
                       CustomerName              NVARCHAR(100),
                       CustomerIPAddress  VARCHAR,
CONSTRAINT Customer_PK PRIMARY KEY (CustomerID));

-- Create P2P Table
CREATE TABLE P2P_T
             (PgameID            BIGINT     NOT NULL,
                       Price                            SMALLMONEY NOT NULL,
CONSTRAINT P2P_FK1 FOREIGN KEY (PgameID) REFERENCES VideoGame_T(GameID),
CONSTRAINT P2P_PK PRIMARY KEY (PgameID));

-- Create F2P Table
CREATE TABLE F2P_T
             (FgameID            BIGINT     NOT NULL,
CONSTRAINT F2P_FK1 FOREIGN KEY (FgameID) REFERENCES VideoGame_T(GameID),
CONSTRAINT F2P_PK PRIMARY KEY (FgameID));

-- Create Download Table
CREATE TABLE Download_T
             (FGameID            BIGINT     NOT NULL,
                       CustomerID         BIGINT,
CONSTRAINT Download_FK1 FOREIGN KEY (FgameID) REFERENCES F2P_T(FgameID),
CONSTRAINT Download_FK2 FOREIGN KEY (CustomerID) REFERENCES Customer_T(CustomerID),
CONSTRAINT Download_PK PRIMARY KEY (FgameID, CustomerID));

-- Create Free Account Table
CREATE TABLE FreeAccount_T
             (FreeAccountID      BIGINT     NOT NULL,
                       CharacterName             NVARCHAR(100),
                       CharacterType      NVARCHAR(100),
                       CharacterCreationDate DATE DEFAULT GETDATE(),
                       FgameID            BIGINT,
                       CustomerID         BIGINT,
CONSTRAINT Freeaccount_FK1 FOREIGN KEY (FgameID) REFERENCES F2P_T(FgameID),
CONSTRAINT Freeaccount_FK2 FOREIGN KEY (CustomerID) REFERENCES Customer_T(CustomerID),
CONSTRAINT FreeAccount_PK PRIMARY KEY (FreeAccountID));

-- Create Purchase Table
CREATE TABLE Purchase_T
             (CustomerID          BIGINT     NOT NULL,
                       PgameID                 BIGINT,
CONSTRAINT Purchase_FK1 FOREIGN KEY (CustomerID) REFERENCES Customer_T(CustomerID),
CONSTRAINT Purchase_FK2 FOREIGN KEY (PgameID) REFERENCES P2P_T(PgameID),
CONSTRAINT Purchase_PK PRIMARY KEY (CustomerID, PgameID));

-- Create Premium Account Table
CREATE TABLE PremiumAccount_T
             (PremiumAccountID    BIGINT     NOT NULL,
                       PremiumStatus             BIT,
                       CustomerID         BIGINT,
                       PgameID            BIGINT,
CONSTRAINT PremiumAccount_FK1 FOREIGN KEY (CustomerID) REFERENCES Customer_T(CustomerID),
CONSTRAINT PremiumAccount_FK2 FOREIGN KEY (PgameID) REFERENCES P2P_T(PgameID),
CONSTRAINT PremiumAccount_PK PRIMARY KEY (PremiumAccountID));

ALTER TABLE HardwareEng_T ADD CONSTRAINT HardwareEng_FK1 FOREIGN KEY (HemployeeID)
REFERENCES Employee_T(EmployeeID);
ALTER TABLE Developer_T ADD CONSTRAINT Developer_FK2 FOREIGN KEY (DemployeeID) REFERENCES
Employee_T(EmployeeID);
ALTER TABLE Administrator_T ADD CONSTRAINT Administrator_FK1 FOREIGN KEY (AemployeeID)
REFERENCES Employee_T(EmployeeID);
ALTER TABLE ServerAdmin_T ADD CONSTRAINT ServerAdmin_FK2 FOREIGN KEY (ServerID)
REFERENCES Server_T(ServerID);
```

Views

```
-- Views
CREATE VIEW GameDistribution AS
SELECT GameType, COUNT(GameID) as NumofGames
FROM VideoGame_T
WHERE ReleaseDate >= '2016-02-01'
GROUP BY GameType

CREATE VIEW SupercomputerDowntime AS
SELECT SupercomputerID, SUM(TimeSpent) as Downtime
FROM Maintenance_T
GROUP BY SupercomputerID

CREATE VIEW ActiveAccounts AS
SELECT COUNT(FreeAccountID) as NewFreeAccounts, (SELECT avg(case when PremiumStatus = 1
then 100 else 0 end) FROM PremiumAccount_T)  as PctPremiumActive
FROM FreeAccount_T
WHERE CharacterCreationDate >= '2016-06-01'
```

Business Justifications

View 1: Game Distribution

This first view shows the user at the company, the current number of free-to-play and pay-to-play games that have been released by the company. This is mainly for higher-level management to keep an eye on how many games they release in each sector in order to maintain balance at the company. If this data was used in combination with financial data at the company it could serve to project which games are more profitable and what the company should invest more developers into. The query specifically targets games that have been released after February of 2016 to keep the data current.

View 2: Supercomputer Downtime

The second view is designed for hardware engineers, administrators, and management. This view shows the cumulated downtime for the supercomputers, and therefore the servers and games at the company. The query groups the sum of all time spent working on the supercomputers together to calculate the aggregated downtime. This metric is useful for identifying which supercomputers may be getting old and which need to be replaced. It can also be tracked overtime to see if downtime is increasing or decreasing in the long run.

View 3: Active Accounts

The final view is another useful metric for management to compare metrics for both free-to-play and pay-to-play games. On the free-to-play side, the query shows the number of new accounts that have been created since June of 2016, to represent how much the accounts have been growing over the period since then. This helps track growth in free to play games. On the other side, there is a nested query to include information about pay-to-play games which shows the percentage of users with a premium status active. This considers all of the premium accounts and averages the active and inactive ones to provide a percentage showing how many accounts have the premium status. This is useful for management to track how popular their premium statuses are on the games and if the users are moving towards free to play or pay to play games.