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
```

```
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.