|  |  |
| --- | --- |
| MLB THE SHOW DATABASE MANAGEMENT SYSTEM | Daniel Goetz and Joey Sabel  IST 659 M004 Spring 2020 |

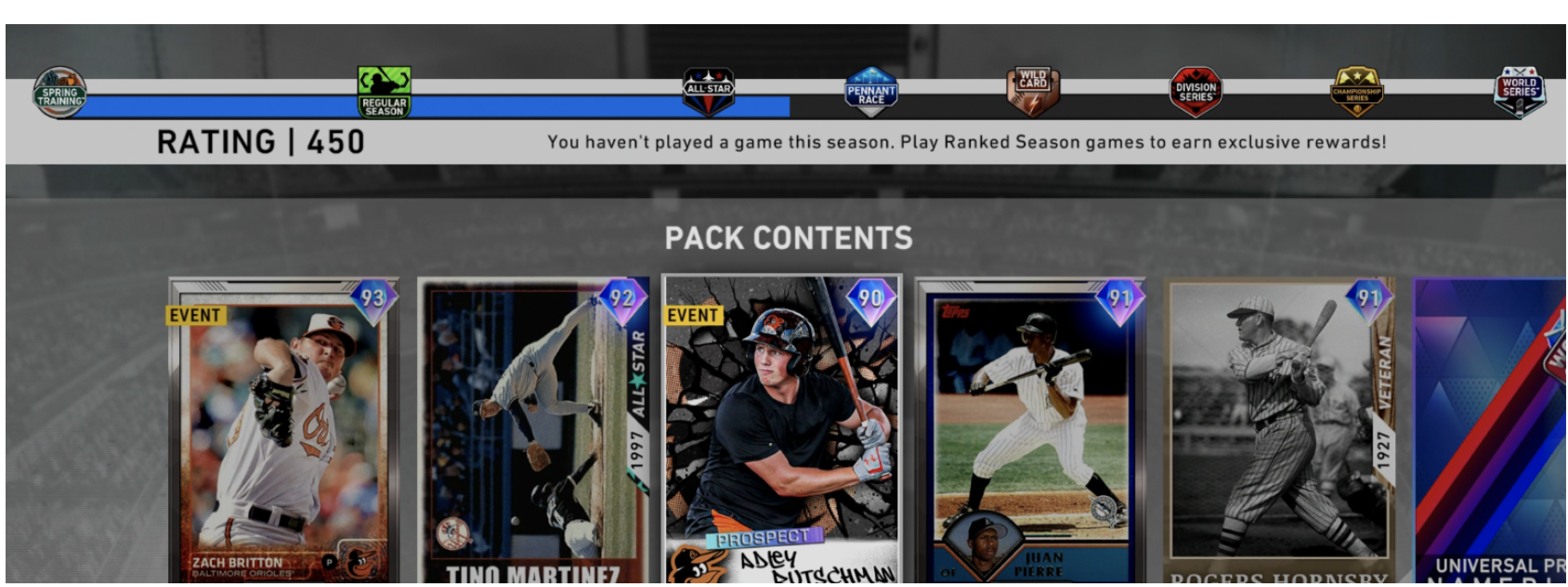
Content

Project Summary

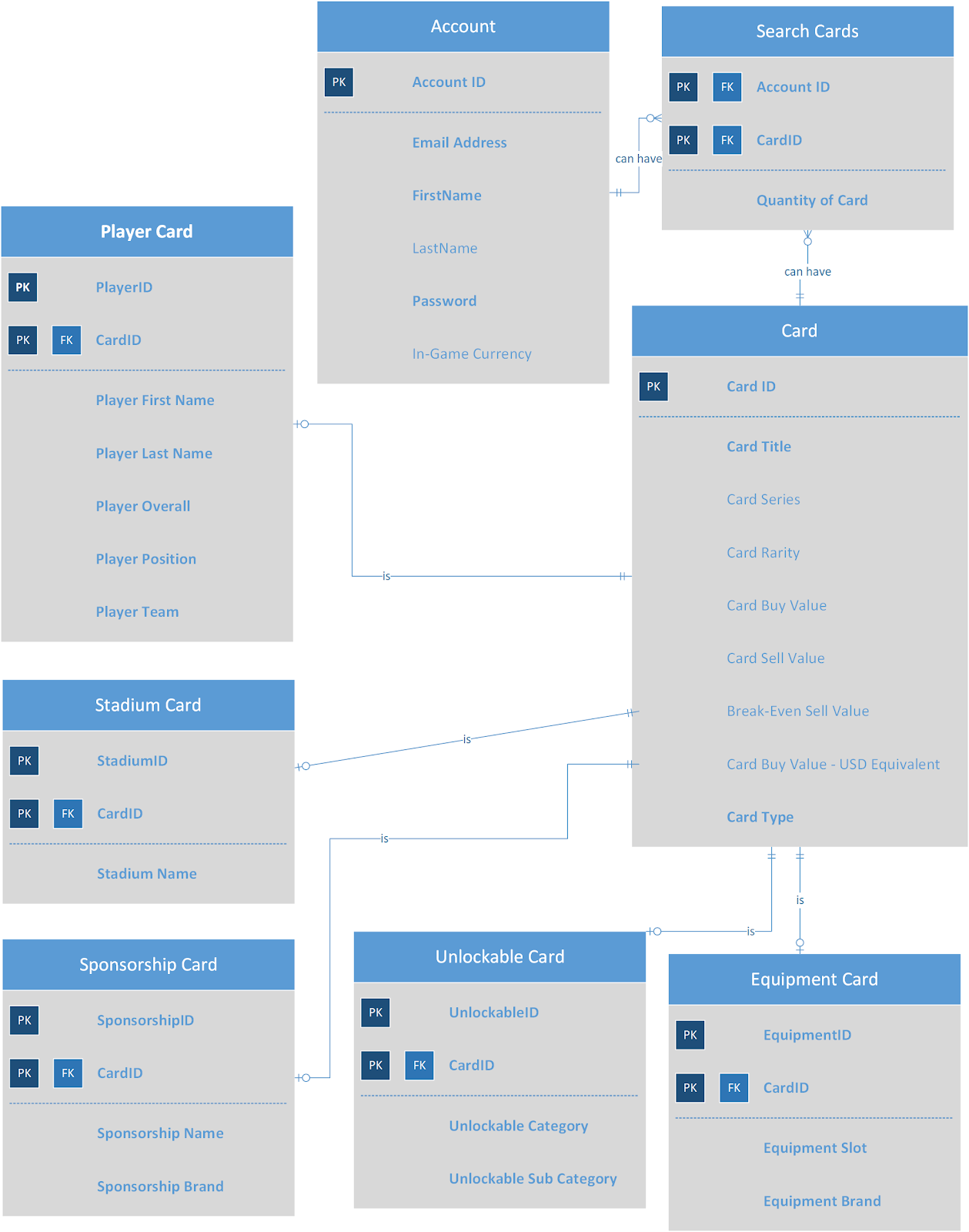
A recent trend in video games of all genres is a marketplace for items or cards in the game, which can be purchased with an in-game economy and/or money from the real world. Some of these games have an expansive number of items making it hard to keep track of which things are important and how much they cost in in-game currency. This might seem like an issue for few people, but a hardcore group of video game players would very much benefit from a database of all of the items or cards in the game they play. This happens everywhere from first-person shooters to sports video games. This proposal would allow users of the game we chose, MLB The Show, to have an exact, detailed database of all of the cards available on the Marketplace. It will allow players to query the names of cards they want, attributes of cards they are interested in, and the value in in-game currency, and the real-world equivalent in USD. This proposal is for an organization, website, or application that does not currently exist, but would, after this project’s completion and would have a working staff of two, in an entirely online location.

Video gaming is a growing industry that is worth many billions of dollars annually. According to Business Insider, the video game industry is worth about $120 billion and growing annually.  With that in mind, there are many hundreds of thousands to many millions of players of major games which are released on a regular schedule. We are trying to solve the issue of having a difficult to navigate market for many games, but starting with our selected video game, MLB The Show. Players of the game often express frustration with the clunky and hard to navigate controls of the in-game marketplace, citing typing with analog sticks worked with their thumbs and button inputs, and this project aims to take all of this frustration away. The current solution is primarily the marketplace that is just in the game as well as another hard to navigate, slow, and fairly unknown website that the creators of the game put out. This solution is not adequate because it lacks many economic functions that are basic to being competitive on the marketplace. For example, the game only displays Sell prices and Buy prices, like ask and bid prices. This would seem to be adequate, but in this game, there is a 10% tax for all cards when they are sold, so the seller receives less in-game currency than they were selling it for. For casual players, this is fine, but for the hardcore players who would use our database, they would aim to make a profit on ever single card they buy and sell. This process is known as “Flipping”, where they buy a card at a low price and sell it for a higher price. Our database would let players input the amount of in-game currency they want to spend on a card, and how much profit they can get back. It does not track any purchases for the player.

The business functions that will be considered are all of the economic functions that players would be interested in for their marketplace needs. Things such as current Buy price, current Sell price, total number of selected cards you can buy with your current in-game currency total and more would be at the top of our list for our proposed system. The functions that have been considered and are not necessarily going to be included would be for this database to expand to other games as well, like FIFA or the NBA 2k series. These games would be perfect for our audience as there would be a lot of crossover for users, but we are unsure whether there would be time to get this database operational in the time we have been given.

The users of our database would be players of the video game MLB The Show primarily, but there could also be research done on the marketplace of this game and the other considered games in an economic/academic setting, showing how people react to changes in market conditions. Everyone would have equal access to the data that we would provide. Nothing would be held behind any barriers because we would be repurposing public data found in the video game itself and allowing users to become more competitive in the marketplace. The data questions users have will be different depending on the strategies the users like and the behavior of each user. They all likely want to know the price of their favorite cards or cards they think will allow them to be the most profitable if they choose to “Flip” them. Users in an academic setting would likely want long-term data for how the market behaves over time.

Relational Data Model



Data Dictionary

1. Account
   1. Records information about the account that the users of the game.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entity Name: Account | Entry and Attributes | Field Type | Nullable | Foreign Key Constraints | Description |
| Primary Key | AccountID | CHAR(10) | NOT NULL |  | Unique ID for each account |
|  | AccountEmail | VARCHAR(30) | NOT NULL |  | Email address of user |
|  | AccountFirstName | VARCHAR(20) | NOT NULL |  | First name of user |
|  | AccountLastName | VARCHAR(20) | NULL |  | Last name of user |
|  | AccountPassword | VARCHAR(20) | NOT NULL |  | Password used for account |
|  | AccountCurrency | INT | NULL |  | Amount of virtual currency the account has in the MLB 20 The Show |

1. Card
   1. Shows information about the basic characteristics of each individual card in the database

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entity Name: Account | Entry and Attributes | Field Type | Nullable | Foreign Key Constraints | Description |
| Primary Key | CardID | CHAR(10) | NOT NULL |  | Unique ID for each card |
|  | CardTitle | VARCHAR(30) | NOT NULL |  | Name for each card |
|  | CardSeries | VARCHAR(20) | NULL |  | Series for each card (if applicable) |
|  | CardRarity | VARCHAR(20) | NULL |  | Rarity of each card (if applicable) |
|  | CardBuyValue | INt | NULL |  | Minimum value to buy card on market to guarantee immediate sale |
|  | CardSellValue | INT | NULL |  | Maximum value to sell card on market to guarantee immediate sale |
|  | BreakEvenSellValue | INT | NULL |  | Amount card would need to be sold for to break even based on current Buy Now value |
|  | USDDollarEQ | FLOAT | NULL |  | US dollar equivalent of Buy Now value based on 24,000 of in game currency per $20 |
|  | CardType | VARCHAR(12) | Not NULL |  | Type of card: Plyayer, sponsorship, unlockable, etc. |

1. Search Card
   1. A relationship table that connects Accounts and Cards. An account may contain one to many cards, while a card may be in only one account

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entity Name: Account | Entry and Attributes | Field Type | Nullable | Foreign Key Constraints | Description |
| Primary Key  Foreign Key 1 | AccountID | CHAR(10) | NOT NULL | Table  Account  (AccountID) | Unique ID for each account |
| Primary Key  Foreign Key 2 | CardID | CHAR(10) | NOT NULL | Table Card (CardID) | Name for each card |
|  | QuantityOfCard | INT | NULL |  | Number of each card (if applicable) |

1. Player Card
   1. Shows detailed characteristics of each player card, including position, team, and handedness

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entity Name: Account | Entry and Attributes | Field Type | Nullable | Foreign Key Constraints | Description |
| Primary Key | PlayerID | CHAR(10) | NOT NULL |  | Unique ID for each player card |
| Primary Key  Foreign Key | CardID | CHAR(10) | NOT NULL | Table Card (CardID) | Unique ID for each card |
|  | PlayerFirstName | VARCHAR(30) | NOT NULL |  | First Name of the Player |
|  | PlayerLastName | VARCHAR(30) | NOT NULL |  | Last Name of the Player |
|  | PlayerOverall | INT | NOT NULL |  | Player overall |
|  | PlayerPosition | VARCHAR(20) | NOT NULL |  | Position each player plays |
|  | PlayerTeam | VARCHAR(20) | NOT NULL |  | Team player plays on |

1. Stadium Card
   1. Shows more detailed characteristics of each stadium in the game

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entity Name: Account | Entry and Attributes | Field Type | Nullable | Foreign Key Constraints | Description |
| Primary Key  Foreign Key 1 | CardID | CHAR(10) | NOT NULL | Table  Card  (CardID) | Unique ID for each card |
| Primary Key | StadiumID | CHAR(10) | NOT NULL |  | UniqueID for each stadium |
|  | StadiumName | VARCHAR(30) | NOT NULL |  | The actual name for each stadium |

1. Unlockable Card
   1. Unlockables include name plates, badges, and icons. Table give categories of each unlockable.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entity Name: Account | Entry and Attributes | Field Type | Nullable | Foreign Key Constraints | Description |
| Primary Key | UnlockableID | CHAR(10) | NOT NULL |  | Unique ID for each unlockable |
| Primary Key  Foreign Key 1 | CardID | CHAR(10) | NOT NULL | Table  Card  (CardID) | Unique ID for each card |
|  | Unlockable Category | VARCHAR(20) | NOT NULL |  | Category for each card |
|  | Unlockable Sub Category | VARCHAR(20) | NOT NULL |  | Sub category for each card |

1. Sponsorship Card
   1. Sponsorships include brands such as Adidas and Under Armour. Table shows a more detailed description of each sponsorship card.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entity Name: Account | Entry and Attributes | Field Type | Nullable | Foreign Key Constraints | Description |
| Primary Key  Foriegn Key 1 | CardID | CHAR(10) | NOT NULL | Table Card (Card ID) | Unique ID for each card |
| Primary Key | SponsorshipID | CHAR(10) | NOT NULL |  | Name for each card |
|  | Sponsorship | VARCHAR(30) | NOT NULL |  | Name of sponsorship |
|  | Sponsorship Brand | VARCHAR(20) | NOT NULL |  | BrandName for each card. |

1. Equipment Card
   1. Users can unlock equipment such as bats and gloves. This table gives a more detailed description of each equipment card.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entity Name: Account | Entry and Attributes | Field Type | Nullable | Foreign Key Constraints | Description |
| Primary Key  Foriegn Key 1 | CardID | CHAR(10) | NOT NULL | Table Card (Card ID) | Unique ID for each card |
| Primary Key | EquipmentID | CHAR(10)) | NOT NULL |  | Unique ID for each equipment card |
|  | EquipmentSlot | VARCHAR(20) | NOT NULL |  | Type of equipment (glove, shoes, etc.) |
|  | EquipmentBrand | VARCHAR(20) | NOT NULL |  | Brand of each equipment (Adidas,Mizuno) |

Other Business Rules

1.  Users can only search for one card at a time.

2.  All data comes directly from MLB The Show unless otherwise noted.

3.  Users can not directly make money directly off of our database.

4.  Users can not share our information with any other website or source.

5. We will protect user information in their accounts so that only they can access it.

6.  All user data will not be shared with any other website or source.

7. Each user must enter a UserID everytime they use our database.

8. All data provided will be as accurate as available.

9. We will take user feedback into consideration for improvements.

10. Users can not delete data from the database

Database System Infrastructure

We used the following tools to create and implement this project:

1. MS Visio: We used Visio to create the relational database model. We created entities and defined their attributes with the keys (primary & foreign). We built the relationships between entities.
2. SQL Server: We used SQL Served to create the tables and establish the relationship between the tables. We also populated the tables and used it to create views using SQL.
3. MS Access: We used Access to create forms and reports. We linked our tables that we originally created in SQL.
4. RStudio: Finally, we used RStudio to scape the MLB the show website to get all of our data needed to create the table. Used the rvest package to scrape data.

SQL Codes for Creating and Populating Tables

--dropping tables

drop table Account

drop table Cards

drop table searchCard

drop table Player

drop table Equipment

drop table Unlockables

drop table Sponsorships

drop table Stadium

--creating our tables

--create table Account

create table Account (

AccountID CHAR(10) NOT NULL,

AccountEmail VARCHAR(30) NOT NULL,

AccountFirstName VARCHAR(20) NOT NULL,

AccountLastName VARCHAR(20) NULL,

AccountPassword VARCHAR(20) NOT NULL,

AccountCurrency INT NULL,

constraint account\_pk primary key (AccountID)

)

--create table Cards

CREATE TABLE Cards(

CardID CHAR(10) NOT NULL PRIMARY KEY

,CardTitle VARCHAR(30) NOT NULL

,CardSeries VARCHAR(30)

,CardRarity VARCHAR(20) NOT NULL

,CardBuyValue INTEGER NOT NULL

,CardSellValue INTEGER NOT NULL

,BreakEvenSellValue INTEGER NOT NULL

,CardValueUSDEQ FLOAT NOT NULL

,CardType VARCHAR(12) NOT NULL

);

--create table searchCard

create table SearchCard(

AccountID CHAR(10) NOT NULL,

CardID INT Not NULL,

Quantity INT NULL,

constraint search\_pk1 primary key (AccountID,CardID),

constraint search\_fk1 foreign key (AccountID) references Account(AccountID),

constraint search\_fk2 foreign key (CardID) references Cards(CardID)

)

--create table Player

CREATE TABLE Player(

PlayerID CHAR(10) NOT NULL

,CardID CHAR(10) NOT NULL

,FirstName VARCHAR(10) NOT NULL

,LastName VARCHAR(50) NOT NULL

,Overall INTEGER NOT NULL

,Pos VARCHAR(20) NOT NULL

,Team VARCHAR(20) NOT NULL,

constraint player\_pk1 primary key (PlayerID,CardID),

constraint player\_fk1 foreign key (CardID) references Cards(CardID)

);

--create table Equipment

CREATE TABLE Equipment(

CardID CHAR(10) NOT NULL

,EquipmentID CHAR(10) NOT NULL

,Slot VARCHAR(18) NOT NULL

,Brand VARCHAR(18) NOT NULL

constraint equipment\_pk1 primary key (EquipmentID,CardID),

constraint equipment\_fk1 foreign key (CardID) references Cards(CardID)

);

--create table Unlockables

CREATE TABLE Unlockables(

CardID CHAR(10) NOT NULL

,UnlockableID CHAR(10) NOT NULL

,UnlockableCategory VARCHAR(40) NOT NULL

,UnlockableSubcategory VARCHAR(20) NOT NULL,

constraint unlock\_pk1 primary key (UnlockableID,CardID),

constraint unlock\_fk1 foreign key (CardID) references Cards(CardID)

);

--create table Sponsorships

CREATE TABLE Sponsorships(

CardID CHAR(10) NOT NULL

,SponsorshipID CHAR(10) NOT NULL

,Sponsorship VARCHAR(20) NOT NULL

,Brand VARCHAR(20) NOT NULL

constraint sponsor\_pk1 primary key (SponsorshipID,CardID),

constraint sponsor\_fk1 foreign key (CardID) references Cards(CardID)

);

--create table Stadium

CREATE TABLE Stadium(

CardID CHAR(10) NOT NULL

,StadiumID CHAR(10) NOT NULL

,Stadium VARCHAR(30) NOT NULL,

constraint stadium\_pk1 primary key (StadiumID,CardID),

constraint stadium\_fk1 foreign key (CardID) references Cards(CardID)

--populate tables

--populate table Account

insert into Account(AccountID,AccountEmail,AccountFirstName,AccountLastName,AccountPassword,AccountCurrency) Values (1,'gwilliams@gmail.com','Grant','Williams','password12','34383')

insert into Account(AccountID,AccountEmail,AccountFirstName,AccountLastName,AccountPassword,AccountCurrency) Values (2,'lbrown93@hotmail.com','Lucas','Brown','green123','58382')

insert into Account(AccountID,AccountEmail,AccountFirstName,AccountLastName,AccountPassword,AccountCurrency) Values (3,'tjordan365@gmail.com','Thomas','Jordan','sprinkle392','41113')

insert into Account(AccountID,AccountEmail,AccountFirstName,AccountLastName,AccountPassword,AccountCurrency) Values (4,'jgonzalez938@yahoo.com','James','Gonzalez','cookie123','104392')

insert into Account(AccountID,AccountEmail,AccountFirstName,AccountLastName,AccountPassword,AccountCurrency) Values (5,'tjefferson43@aol.com','Todd','Jefferson','rainbow','12938')

--populate table Card

Note: We inserted more cards than shown but this is just a sample.

INSERT INTO Cards(CardID,CardTitle,CardSeries,CardRarity,CardBuyValue,CardSellValue,BreakEvenSellValue,CardValueUSDEQ,CardType) VALUES (11176,'Chad Wallach','Live','Common',74,12,13,'$0.06','Player');

INSERT INTO Cards(CardID,CardTitle,CardSeries,CardRarity,CardBuyValue,CardSellValue,BreakEvenSellValue,CardValueUSDEQ,CardType) VALUES (11216,'Mike Trout','Live','Diamond',323325,284000,315556,'$269.44','Player');

INSERT INTO Cards(CardID,CardTitle,CardSeries,CardRarity,CardBuyValue,CardSellValue,BreakEvenSellValue,CardValueUSDEQ,CardType) VALUES (11240,'Domingo Leyba','Live','Bronze',74,26,29,'$0.06','Player');

INSERT INTO Cards(CardID,CardTitle,CardSeries,CardRarity,CardBuyValue,CardSellValue,BreakEvenSellValue,CardValueUSDEQ,CardType) VALUES (11247,'Conner Menez','Live','Common',54,11,12,'$0.05','Player');

INSERT INTO Cards(CardID,CardTitle,CardSeries,CardRarity,CardBuyValue,CardSellValue,BreakEvenSellValue,CardValueUSDEQ,CardType) VALUES (11252,'Hanley Ramirez','Live','Bronze',160,45,50,'$0.13','Player');

INSERT INTO Cards(CardID,CardTitle,CardSeries,CardRarity,CardBuyValue,CardSellValue,BreakEvenSellValue,CardValueUSDEQ,CardType) VALUES (11271,'Deivy Grullon','Live','Bronze',78,37,41,'$0.07','Player');

INSERT INTO Cards(CardID,CardTitle,CardSeries,CardRarity,CardBuyValue,CardSellValue,BreakEvenSellValue,CardValueUSDEQ,CardType) VALUES (11272,'Garrett Whitlock','Live','Common',13,0,0,'$0.01','Player');

INSERT INTO Cards(CardID,CardTitle,CardSeries,CardRarity,CardBuyValue,CardSellValue,BreakEvenSellValue,CardValueUSDEQ,CardType) VALUES (11275,'Josh Taylor','Live','Silver',240,160,178,'$0.20','Player');

INSERT INTO Cards(CardID,CardTitle,CardSeries,CardRarity,CardBuyValue,CardSellValue,BreakEvenSellValue,CardValueUSDEQ,CardType) VALUES (11277,'Kevin Plawecki','Live','Bronze',81,39,43,'$0.07','Player');

INSERT INTO Cards(CardID,CardTitle,CardSeries,CardRarity,CardBuyValue,CardSellValue,BreakEvenSellValue,CardValueUSDEQ,CardType) VALUES (11287,'Chris Stratton','Live','Common',20,5,6,'$0.02','Player');

INSERT INTO Cards(CardID,CardTitle,CardSeries,CardRarity,CardBuyValue,CardSellValue,BreakEvenSellValue,CardValueUSDEQ,CardType) VALUES (11307,'William Contreras','Live','Common',33,6,7,'$0.03','Player');

--populate table Player

Note: We inserted more players than shown but this is just a sample.

INSERT INTO Player(PlayerID,CardID,FirstName,LastName,Overall,Pos,Team) VALUES (1,11216,'Mike','Trout',95,'CF','Angels');

INSERT INTO Player(PlayerID,CardID,FirstName,LastName,Overall,Pos,Team) VALUES (2,96979,'Miguel','Cabrera',94,'3B','Tigers');

INSERT INTO Player(PlayerID,CardID,FirstName,LastName,Overall,Pos,Team) VALUES (3,79922,'Vida','Blue',93,'SP','Athletics');

INSERT INTO Player(PlayerID,CardID,FirstName,LastName,Overall,Pos,Team) VALUES (4,1,'Roy','Oswalt',93,'SP','Astros');

INSERT INTO Player(PlayerID,CardID,FirstName,LastName,Overall,Pos,Team) VALUES (5,41929,'Joe','Carter',93,'RF','Blue Jays');

INSERT INTO Player(PlayerID,CardID,FirstName,LastName,Overall,Pos,Team) VALUES (6,61287,'Bob','Feller',93,'SP','Indians');

INSERT INTO Player(PlayerID,CardID,FirstName,LastName,Overall,Pos,Team) VALUES (7,61868,'Zack','Britton',93,'CP','Orioles');

INSERT INTO Player(PlayerID,CardID,FirstName,LastName,Overall,Pos,Team) VALUES (8,12577,'Eric','Gagne',92,'CP','Dodgers');

INSERT INTO Player(PlayerID,CardID,FirstName,LastName,Overall,Pos,Team) VALUES (9,54395,'Max','Scherzer',92,'SP','Nationals');

INSERT INTO Player(PlayerID,CardID,FirstName,LastName,Overall,Pos,Team) VALUES (10,86504,'Johnny','Bench',92,'C','Reds');

INSERT INTO Player(PlayerID,CardID,FirstName,LastName,Overall,Pos,Team) VALUES (11,77471,'Jacob','deGrom',92,'SP','Mets');

INSERT INTO Player(PlayerID,CardID,FirstName,LastName,Overall,Pos,Team) VALUES (12,92127,'John','Smoltz',92,'CP','Braves');

--populate table Equipment

Note: We inserted more equipment than shown but this is just a sample.

INSERT INTO Equipment(CardID,EquipmentID,Slot,Brand) VALUES (76481,1,'Cleats','Adidas');

INSERT INTO Equipment(CardID,EquipmentID,Slot,Brand) VALUES (56379,2,'Batting Gloves','Adidas');

INSERT INTO Equipment(CardID,EquipmentID,Slot,Brand) VALUES (41559,3,'Fielding Glove','Adidas');

INSERT INTO Equipment(CardID,EquipmentID,Slot,Brand) VALUES (52504,4,'Cleats','Adidas');

INSERT INTO Equipment(CardID,EquipmentID,Slot,Brand) VALUES (99322,5,'Cleats','Adidas');

INSERT INTO Equipment(CardID,EquipmentID,Slot,Brand) VALUES (87460,6,'Fielding Glove','All-Star');

INSERT INTO Equipment(CardID,EquipmentID,Slot,Brand) VALUES (69015,7,'Catcher Masks','All-Star');

INSERT INTO Equipment(CardID,EquipmentID,Slot,Brand) VALUES (47403,8,'Chest Protector','All-Star');

INSERT INTO Equipment(CardID,EquipmentID,Slot,Brand) VALUES (48195,9,'Leg Guards','All-Star');

INSERT INTO Equipment(CardID,EquipmentID,Slot,Brand) VALUES (59564,10,'Bat','Axe Bat');

INSERT INTO Equipment(CardID,EquipmentID,Slot,Brand) VALUES (17515,11,'Bat','Axe Bat');

INSERT INTO Equipment(CardID,EquipmentID,Slot,Brand) VALUES (34790,12,'Bat','Axe Bat');

INSERT INTO Equipment(CardID,EquipmentID,Slot,Brand) VALUES (60791,13,'Bat','Axe Bat');

INSERT INTO Equipment(CardID,EquipmentID,Slot,Brand) VALUES (54138,14,'Bat','Bat Skin');

--populate table Unlockables

Note: We inserted more unlockables than shown but this is just a sample.

INSERT INTO Unlockables(CardID,UnlockableID,UnlockableCategory,UnlockableSubcategory) VALUES (84908,1,'Spring','Icon');

INSERT INTO Unlockables(CardID,UnlockableID,UnlockableCategory,UnlockableSubcategory) VALUES (64809,2,'The Spear LH1','Icon');

INSERT INTO Unlockables(CardID,UnlockableID,UnlockableCategory,UnlockableSubcategory) VALUES (28462,3,'The Spear RH1','Icon');

INSERT INTO Unlockables(CardID,UnlockableID,UnlockableCategory,UnlockableSubcategory) VALUES (57611,4,'The Spear LH2','Icon');

INSERT INTO Unlockables(CardID,UnlockableID,UnlockableCategory,UnlockableSubcategory) VALUES (15141,5,'The Spear RH2','Icon');

INSERT INTO Unlockables(CardID,UnlockableID,UnlockableCategory,UnlockableSubcategory) VALUES (87081,6,'Cartoon Stumble Run','Icon');

INSERT INTO Unlockables(CardID,UnlockableID,UnlockableCategory,UnlockableSubcategory) VALUES (27633,7,'Sombrero Icon','Icon');

INSERT INTO Unlockables(CardID,UnlockableID,UnlockableCategory,UnlockableSubcategory) VALUES (54497,8,'Clumsy Ice Cream Cone Icon','Icon');

--populate table Sponsorships

Note: We inserted more sponsorships than shown but this is just a sample.

INSERT INTO Sponsorships(CardID,SponsorshipID,Sponsorship,Brand) VALUES (12376,1,'Diamond Sponsorship','Old Hickory');

INSERT INTO Sponsorships(CardID,SponsorshipID,Sponsorship,Brand) VALUES (24407,2,'Diamond Sponsorship','Adidas');

INSERT INTO Sponsorships(CardID,SponsorshipID,Sponsorship,Brand) VALUES (39396,3,'Diamond Sponsorship','Nike');

INSERT INTO Sponsorships(CardID,SponsorshipID,Sponsorship,Brand) VALUES (96047,4,'Diamond Sponsorship','Louisville Slugger');

INSERT INTO Sponsorships(CardID,SponsorshipID,Sponsorship,Brand) VALUES (12398,5,'Diamond Sponsorship','Marucci');

INSERT INTO Sponsorships(CardID,SponsorshipID,Sponsorship,Brand) VALUES (92047,6,'Diamond Sponsorship','Sam Bat');

INSERT INTO Sponsorships(CardID,SponsorshipID,Sponsorship,Brand) VALUES (99395,7,'Diamond Sponsorship','Under Armour');

INSERT INTO Sponsorships(CardID,SponsorshipID,Sponsorship,Brand) VALUES (47112,8,'Diamond Sponsorship','Wilson');

INSERT INTO Sponsorships(CardID,SponsorshipID,Sponsorship,Brand) VALUES (59913,9,'Diamond Sponsorship','Jordan');

INSERT INTO Sponsorships(CardID,SponsorshipID,Sponsorship,Brand) VALUES (49354,10,'Gold Sponsorship','Lizard Skins');

INSERT INTO Sponsorships(CardID,SponsorshipID,Sponsorship,Brand) VALUES (45628,11,'Gold Sponsorship','Adidas');

--populate table Stadium

Note: We inserted more stadiums than shown but this is just a sample.

INSERT INTO Stadium(CardID,StadiumID,Stadium) VALUES (87396,1,'Old Atlanta');

INSERT INTO Stadium(CardID,StadiumID,Stadium) VALUES (19085,2,'Shibe Park');

INSERT INTO Stadium(CardID,StadiumID,Stadium) VALUES (65194,3,'Crosley Field');

INSERT INTO Stadium(CardID,StadiumID,Stadium) VALUES (71068,4,'Forbes Field');

INSERT INTO Stadium(CardID,StadiumID,Stadium) VALUES (92397,5,'Sportsman''s Park');

INSERT INTO Stadium(CardID,StadiumID,Stadium) VALUES (88469,6,'Hubert H. Humphrey Metrodome');

INSERT INTO Stadium(CardID,StadiumID,Stadium) VALUES (43463,7,'Old Miami');

INSERT INTO Stadium(CardID,StadiumID,Stadium) VALUES (59142,8,'Griffith Stadium');

INSERT INTO Stadium(CardID,StadiumID,Stadium) VALUES (30448,9,'Yankee Stadium (1976-2008)');

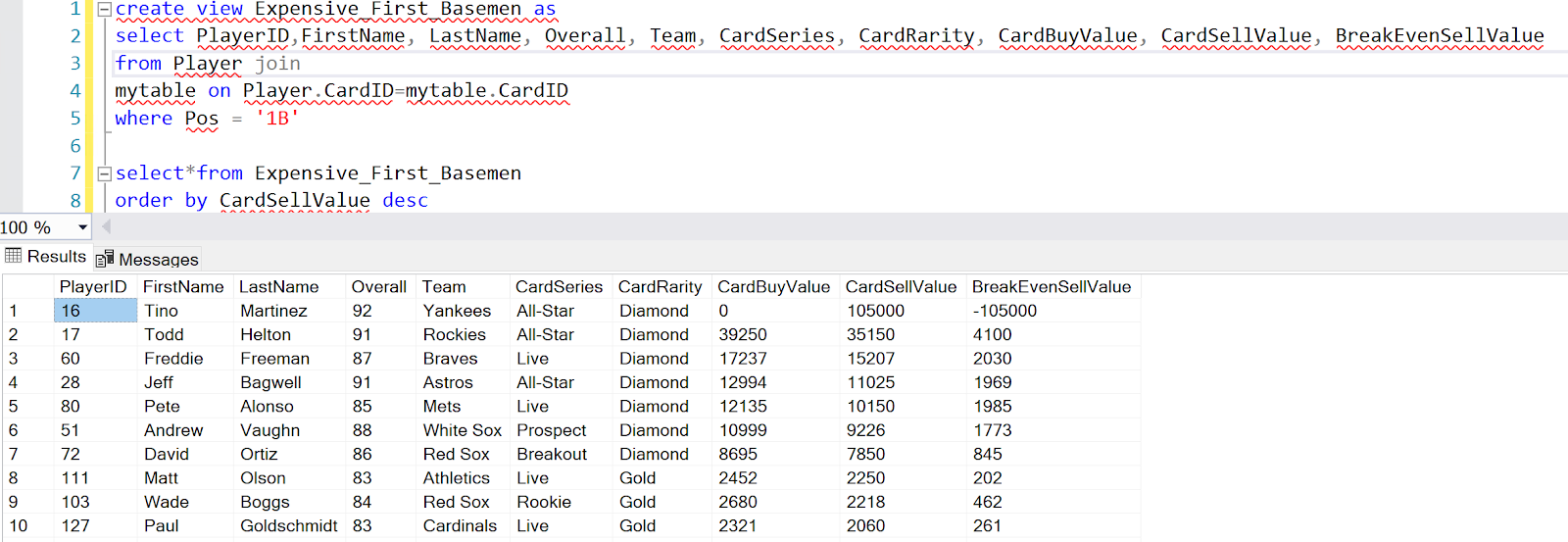
INSERT INTO Stadium(CardID,StadiumID,Stadium) VALUES (81436,10,'Shea Stadium');

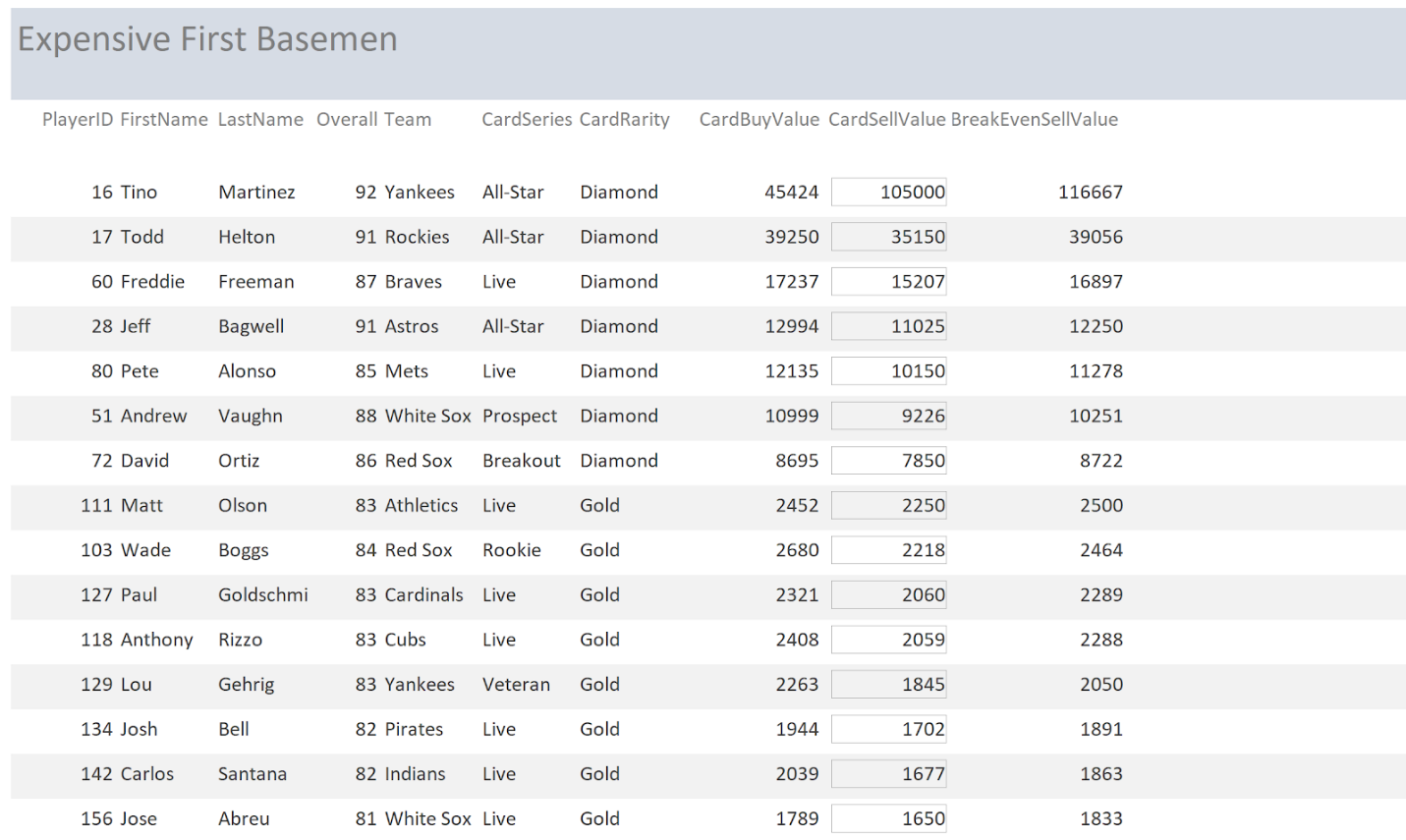
Major Data Questions

Following are some major data questions that our users would ask.

1.Who are the most expensive first basemen?

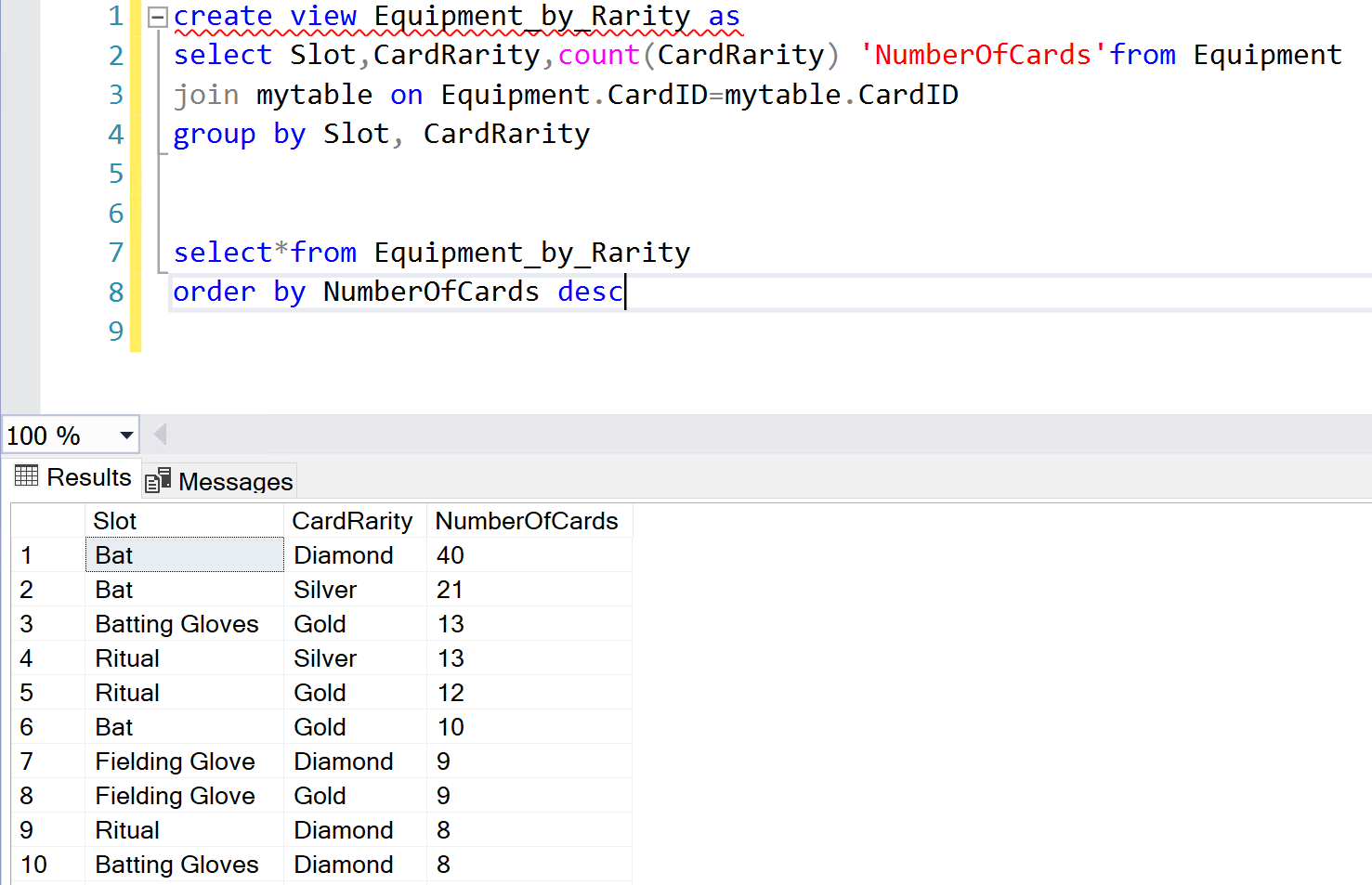
First Baseman is a premium position in MLB the Show. Users may want to see what first baseman will give them the best value if they were to acquire them in a pack or some other method. This view is useful for these individuals.

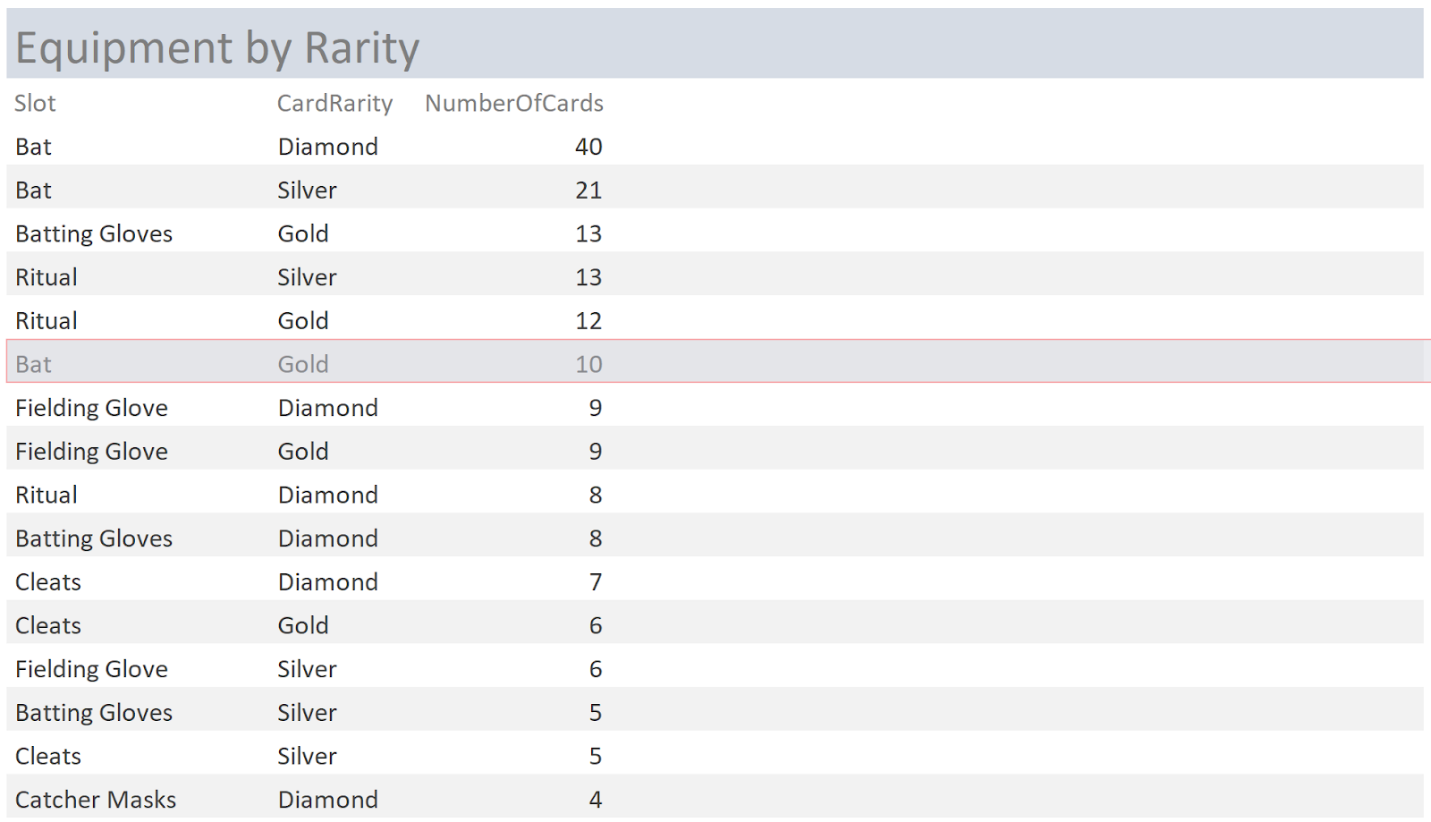




2. How many equipment items are there by rarity and card slot?

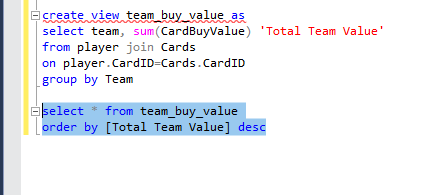
Equipment can be very valuable to players of MLB the Show. Therefore, users may want to see the type of equipment by rarity to see what piece of equipment they want to purchase. For example, Diamond batting gloves may be more expensive because there is only 8 of them as opposed to there being 40 diamond bats.





3. Which team has the most expensive cards?

Users might want to see how much their favorite team is worth on the MLB the show marketplace. This view in SQL gives them the opportunity to do just that. An interesting note is that free agents are actually worth more than some teams.

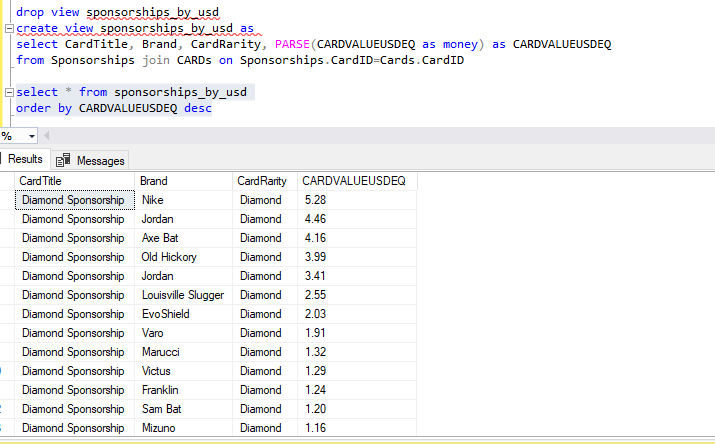




.

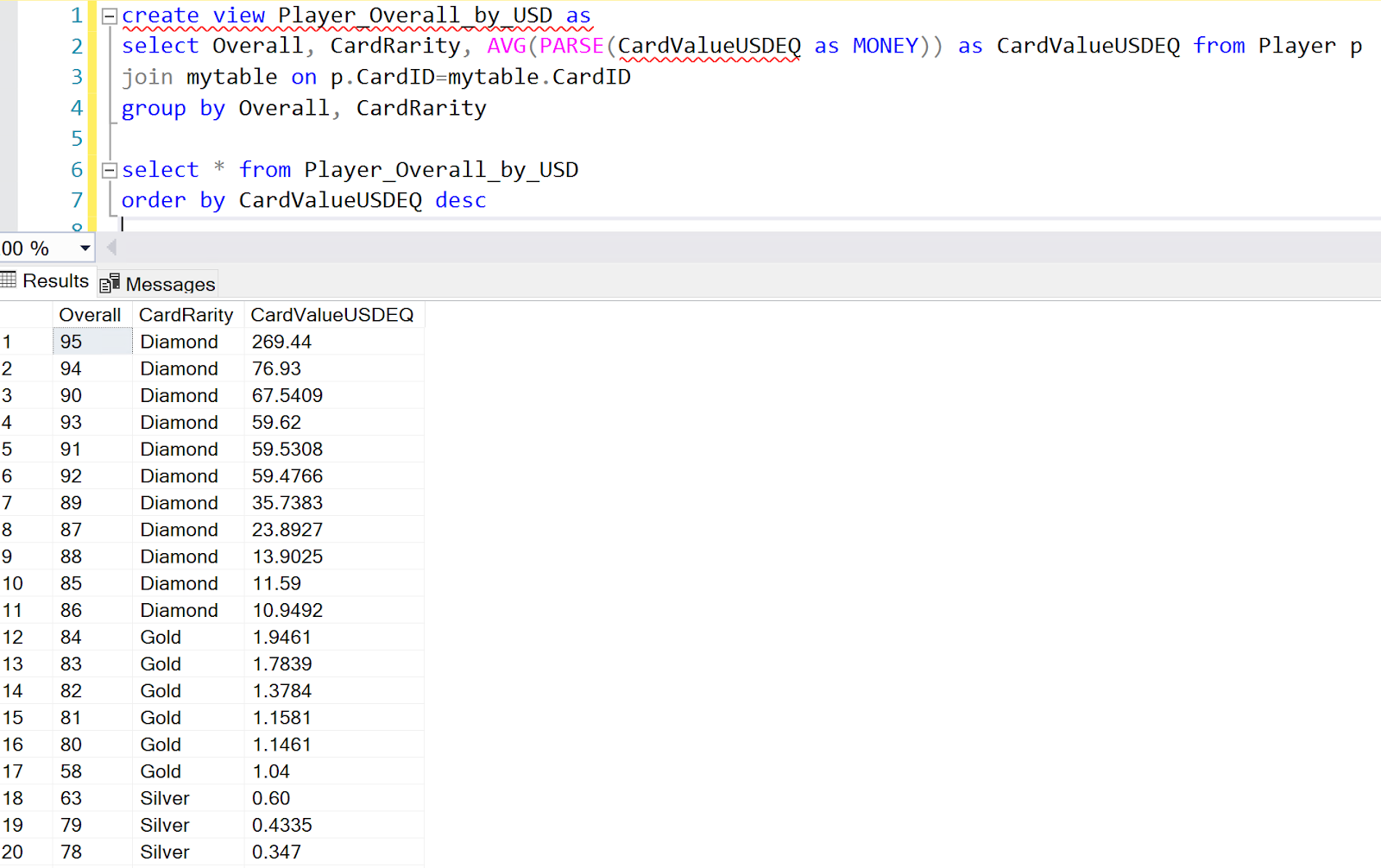
4. What are the most expensive sponsorships by USD?

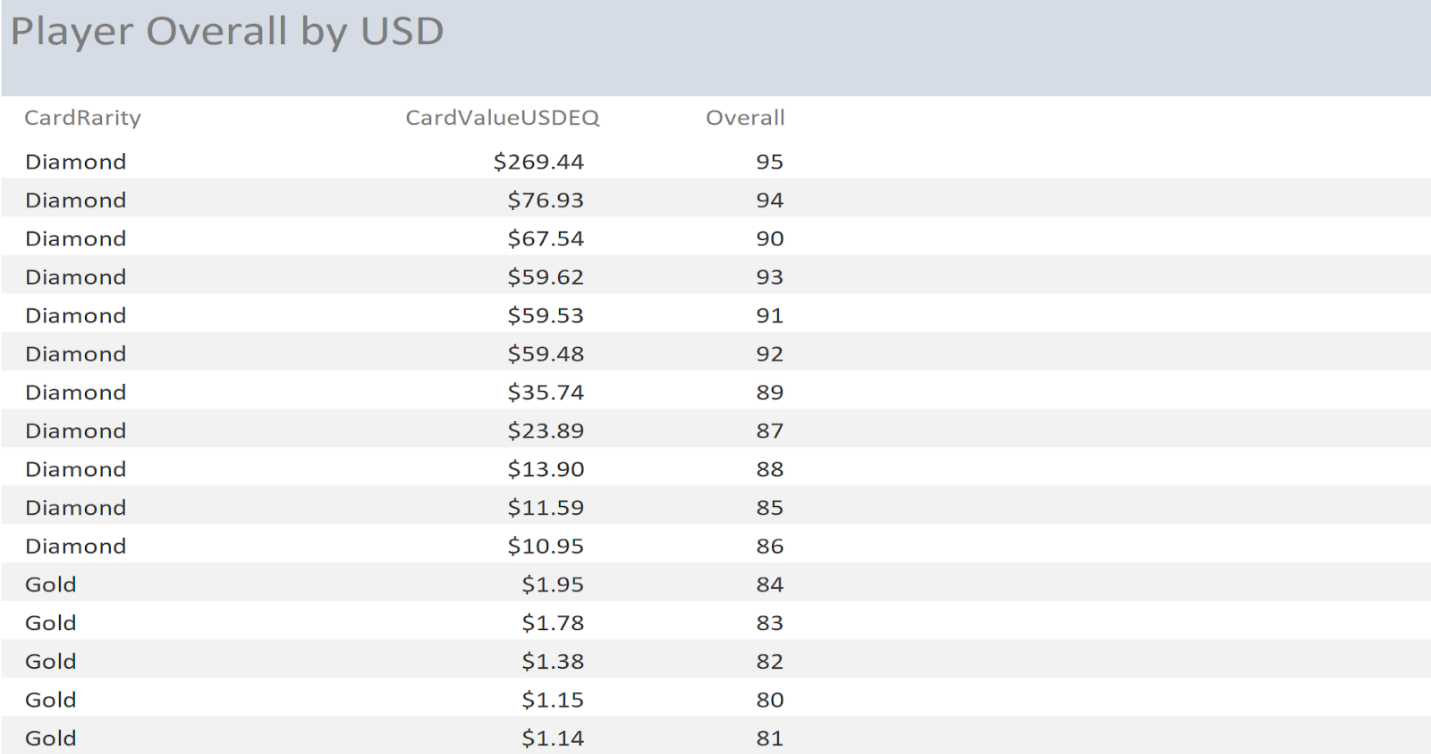
Sponsorships are important in MLB the show because they allow you to get rewards in cash for completing certain tasks. This view shows the most expensive ones (in USD)



5.What is the most expensive player overall by average US Dollar equivalent?

This view shows how much each player overall is worth. For example, if a user wanted to add an impact 91 overall to their team, it would cost an average of $59.53 in USD.





Interface Implementation - Forms and Reports

Login Page:



