

# **MindRiteSQLTite**

## **Phandroid Database Report**

By

Cameron Beemster, Even Etienne, Andrew Gavrilă, Andy Tavoşo and Mark  
Whitaker

## Relationship Matrix:

Relationship Matrix				
	Users	Phones	Features	Threads
Users				Post/ReplyTo
Phones			Represent	
Features		Represented_by		
Threads	Posted_by/Replied_by			

## Business Rules:

Users can post zero to many threads.

Threads can only be posted by one user.

Users can reply to zero to many threads.

Threads can be replied to by zero to many users.

Threads can only be about one phone.

Threads cannot stand alone, so it lives as a weak entity.

Phones can have many threads about them.

Phones can only have one set of features.

A set of features can represent only one phone.

The Users table must have a unique User\_ID.

Thread\_URL must be unique under Threads.

Thread\_URL in the Reply\_to table do not have to be unique however RIP does need to be unique.

Feature\_Value under Represent can contain multiple values.

Thread content can have null values because some people post the title with no description

## **Indexes**

A unique key was created on Features and ReplyTo create a nonclustered index. Artificial FID on Features and RID on ReplyTo had a ordinal sequence to the distinct values within the column.

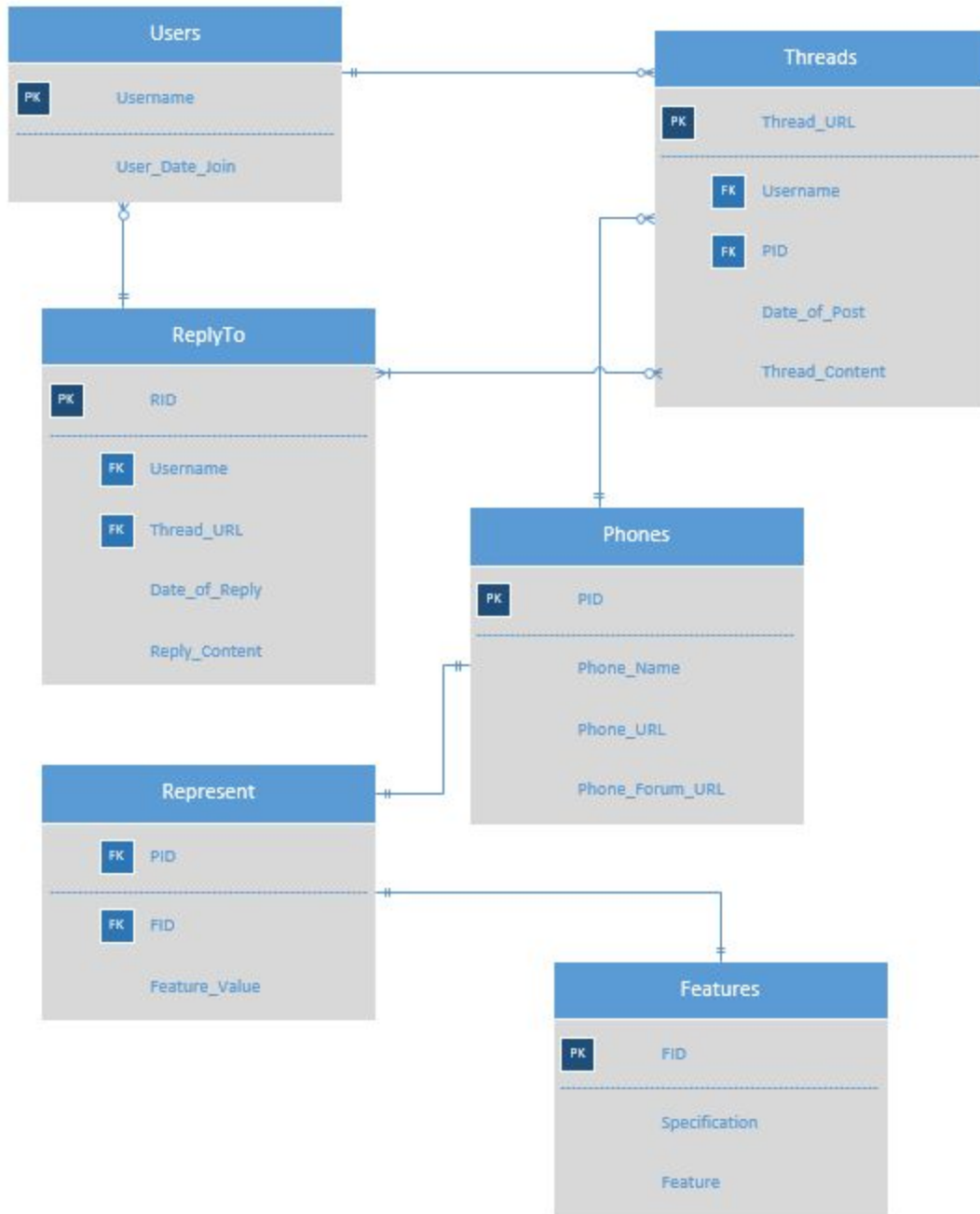
Primary keys were put into entities to ensure integrity when the order of

to pt used to create a non-clustered index on unique.

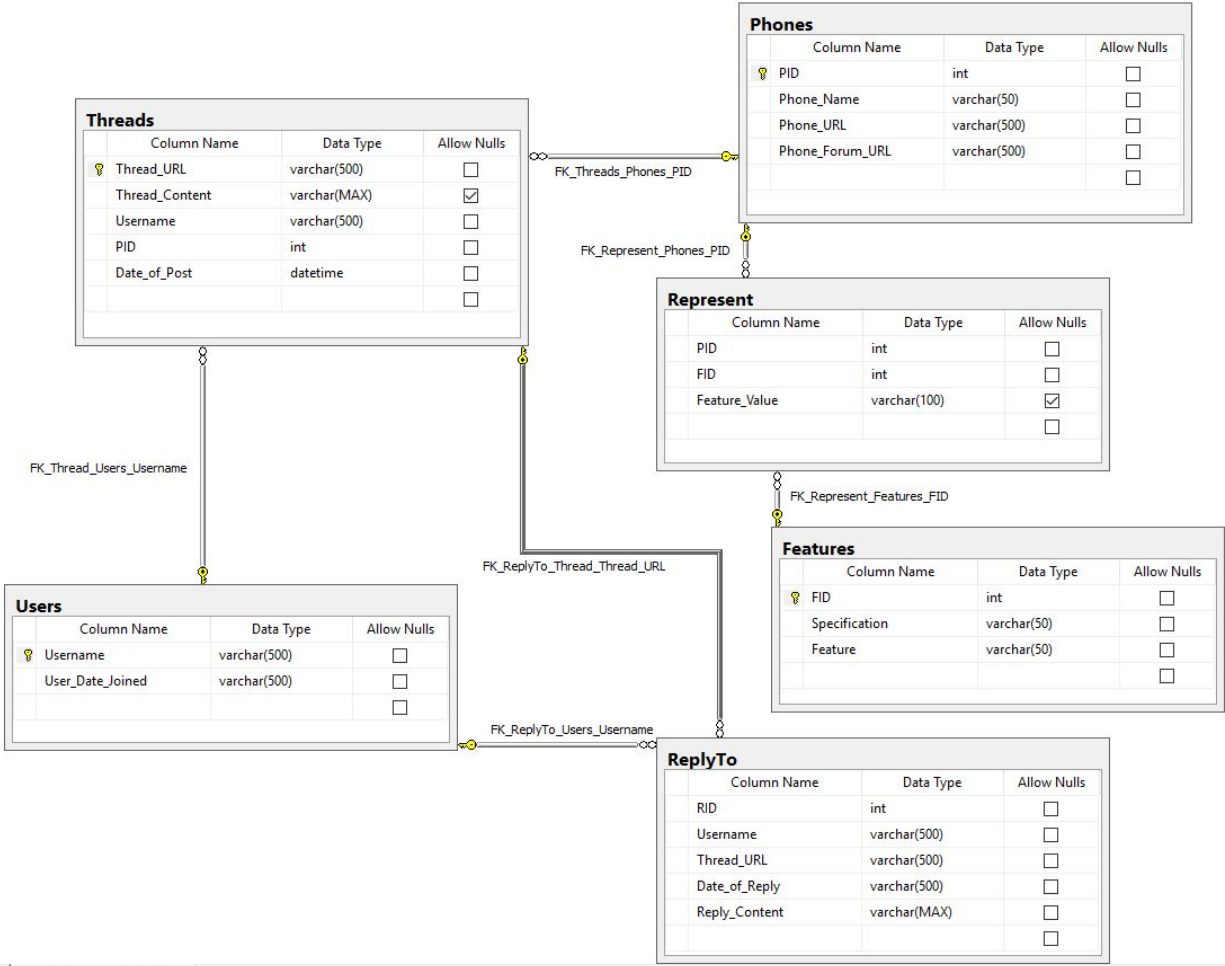
Clustered index on Primary key

Features unique index, phones PID\_PK, threads PK Thread\_URL, Replyto Unique index RID,

## Crows Foot:



ERD:



## Code:

```
CREATE TABLE Features
(
    FID INT IDENTITY NOT NULL
    ,Specification VARCHAR(50) NOT NULL
    ,Feature VARCHAR(50) NOT NULL
)
GO
```

```
CREATE UNIQUE INDEX UI_Features_FID
ON Features (FID)
GO
```

```
INSERT INTO Features
SELECT DISTINCT Specification
           ,Feature
FROM Group_2_MindRiteSQLTite..Smartphones_Data
```

```
CREATE TABLE Phones
(
    PID INT IDENTITY NOT NULL
    ,Phone_Name VARCHAR(50) NOT NULL
    ,Phone_URL VARCHAR(500) NOT NULL
    ,Phone_Forum_URL VARCHAR(500) NOT NULL
)
GO
```

```
ALTER TABLE Phones
ADD CONSTRAINT PK_Phones_PID PRIMARY KEY (PID)
```

```
INSERT INTO Phones
SELECT DISTINCT Smartphones_Data.Phone_Name
           ,Smartphones_Data.Phone_URL
           ,Smartphones_Data.Phone_Forum_URL
FROM Group_2_MindRiteSQLTite..Smartphones_Data
```

```
SELECT DISTINCT Smartphones_Data.Specification
```

```

        ,Smartphones_Data.Feature
        ,Features.FID
FROM
    Smartphones_Data
    INNER JOIN
    Features
    ON Smartphones_Data.Feature = Features.Feature

SELECT DISTINCT Group_2_MindRiteSQLTite..Smartphones_Data.Phone_Name
        ,Smartphones_Data.Phone_URL
        ,Smartphones_Data.Phone_Forum_URL
        ,Phones.PID
FROM
    Smartphones_Data
    INNER JOIN
    Phones
    ON Smartphones_Data.Phone_Name = Phones.Phone_Name

CREATE TABLE Represent
(
    PID INT NOT NULL
    ,FID INT NOT NULL
    ,Feature_Value VARCHAR(100) NULL
)
GO

INSERT INTO Represent (PID, FID, Feature_Value)
SELECT DISTINCT Phones.PID
        ,Features.FID
        ,Smartphones_Data.Feature_Value
FROM
    Phones
    INNER JOIN
    Smartphones_Data
    ON Smartphones_Data.Phone_Name = Phones.Phone_Name
    INNER JOIN
    Features
    ON Smartphones_Data.Feature = Features.Feature

```

```

/* 5/1 added code */

```

```
CREATE TABLE Users
(
    Username VARCHAR(500) NOT NULL
    ,User_Date_Joined VARCHAR(500) NOT NULL
)
GO
```

```
INSERT INTO Users
SELECT DISTINCT Username
    ,User_Date_Joined
FROM Group_2_MindRiteSQLTite..Smartphones_Data
```

```
INSERT INTO Users
SELECT DISTINCT First_Replier
    ,'1900-01-01'
FROM Group_2_MindRiteSQLTite..Smartphones_Data
WHERE First_Replier NOT IN
(
    SELECT Username
    FROM Users
)
```

```
CREATE TABLE Threads
(
    Thread_URL VARCHAR(500) NOT NULL
    ,Thread_Content VARCHAR(MAX) NULL
    ,Username VARCHAR(500) NOT NULL
    ,PID INT NOT NULL
    ,Date_of_Post DATETIME NOT NULL
)
```

```
ALTER TABLE Threads
ADD CONSTRAINT PK_Threads_Thread_URL PRIMARY KEY (Thread_URL)
```

```
INSERT INTO Threads (Thread_URL, Thread_Content, Username, PID, Date_of_Post)
SELECT DISTINCT Smartphones_Data.Thread_URL
    , Smartphones_Data.Thread_Content
    , Smartphones_Data.Username
    , Phones.PID
    , Smartphones_Data.Date_of_Post
FROM
    Group_2_MindRiteSQLTite..Smartphones_Data
```



INNER JOIN

Phones

ON Smartphones\_Data.Phone\_Name = Phones.Phone\_Name

CREATE TABLE ReplyTo

```
(
    RID INT IDENTITY NOT NULL
    ,Username VARCHAR(500) NOT NULL
    ,Thread_URL VARCHAR(500) NOT NULL
    ,Date_of_First_Reply VARCHAR(500) NOT NULL
    ,Reply_Content VARCHAR(MAX) NOT NULL
)
```

GO

CREATE UNIQUE INDEX UI\_ReplyTo\_RID

ON ReplyTo (RID)

GO

ALTER TABLE ReplyTo

ADD CONSTRAINT FK\_ReplyTo\_Threads\_Thread\_URL FOREIGN KEY (Thread\_URL)  
REFERENCES Threads (Thread\_URL)

GO

INSERT INTO ReplyTo

```
SELECT DISTINCT Users.Username
                ,Threads.Thread_URL
                ,Smartphones_Data.Date_of_First_Reply
                ,Smartphones_Data.Reply_Content
```

FROM

```
    Users
  INNER JOIN
    Smartphones_Data
    ON Smartphones_Data.First_Replier = Users.Username
  INNER JOIN
    Threads
    ON Smartphones_Data.Thread_Content = Threads.Thread_Content
```

ON Smartphones\_Data.Thread\_Content = Threads.Thread\_Content

*/\* Foreign Key \*/*

```
ALTER TABLE Threads
ADD CONSTRAINT FK_Threads_Users_Username FOREIGN KEY (Username)
REFERENCES Users (Username)
GO
```

```
ALTER TABLE Represent
ADD CONSTRAINT FK_Represent_Features_FID FOREIGN KEY (FID)
REFERENCES Features (FID)
GO
```

```
ALTER TABLE Represent
ADD CONSTRAINT FK_Represent_Phones_PID FOREIGN KEY (PID)
REFERENCES Phones (PID)
```

```
ALTER TABLE ReplyTo
ADD CONSTRAINT FK_ReplyTo_Users_Username FOREIGN KEY (Username)
REFERENCES Users (Username)
GO
```

```
ALTER TABLE Threads
ADD CONSTRAINT FK_Threads_Phones_PID FOREIGN KEY (PID)
REFERENCES Phones (PID)
```

## Business Questions:

1. What phones are the Top 10 most talked about phones in the 2013 holiday season?

### Why this question was chosen?

This business question will help Phandroid and Android with marketing promotions. With the knowledge of which phones were the most popular in the holiday season, Phandroid and Android can properly prepare for next season's holidays and have accurate inventory and customer support on the specific phones most in demand. The most talked about phones can either be praise for the phones or it can be customers with ambiguous thoughts about the phones. The company will need to look into why the phones are being posted about the most and what features they can add or subtract to make the conversation about phones go smoothly.

Code:

```
SELECT Top 10 COUNT(Distinct Thread_URL) AS 'Discussion Count'
, Phone_Name
FROM
    Phones
    Inner Join
    Threads
on Phones.PID = Threads.PID
WHERE Convert(Date, Date_of_Post) between '2013-11-29' and '2013-12-31'
GROUP BY Phone_Name
ORDER BY Count(Distinct Thread_URL) DESC
Go
```

Results:

	Discussion Count	Phone_Name
1	264	Samsung Galaxy S3
2	225	Samsung Galaxy S4
3	174	Samsung Galaxy Note 3
4	165	Nexus 5
5	78	LG G2
6	70	Samsung Galaxy Note 2
7	52	Motorola Moto X
8	40	Motorola DROID Maxx
9	34	Nexus 4
10	27	Motorola Moto G

2. What are the 3 most common months in which new Users joined Phandroid.com?

**Why this question was chosen:**

This question was chosen to help the company understand which months were the most popular for users to join the site. By gaining knowledge from this business question, the company can focus their IT support and advertisements on these months to drive customer engagement and prevent site crashes. By knowing when users join the most, Phandroid can be well prepared and stocked, ready to help any newcomers with questions they might have. Increasing customer support and availability, it will give Phandroid a good company image and increase traffic to the site.

Code:

```
select top 3 User_Date_Joined
,Count(Distinct Username) as 'User Count'
from Users
where User_Date_Joined between '2012-01-01' and '2012-12-31'
group by User_Date_Joined
order by 2 DESC
go
```

```
select top 3 User_Date_Joined
,Count(Distinct Username) as 'User Count'
from Users
where User_Date_Joined between '2011-01-01' and '2011-12-31'
group by User_Date_Joined
order by 2 DESC
go
```

```
select top 3 User_Date_Joined
,Count(Distinct Username) as 'User Count'
from Users
where User_Date_Joined between '2013-01-01' and '2013-12-31'
group by User_Date_Joined
order by 2 DESC
go
```

Results:

	user_date_joined	User Count
1	2011-07-01	1472
2	2011-06-01	1432
3	2011-08-01	1375

	user_date_joined	User Count
1	2012-07-01	1237
2	2012-01-01	1125
3	2012-06-01	1115

	user_date_joined	User Count
1	2013-01-01	766
2	2013-05-01	596
3	2013-03-01	561

3. Identify top 25 usernames who reply to the most threads?

**Why this question was chosen:**

This question helps the company understand which users are the most active and helpful on the forums. Forums are heavily community based and having a good user base is critical to the success of forums and customer attrition. By finding out the most active users, the company can give special privileges to incentivize those for the hard work they have been contributing. When other users see the special privileges given to the top users, it will incentivize the users to contribute more to gain these special privileges.

Code:

```
= Select top 25 Users.Username  
      ,Count(Date_of_Reply) as 'Reply Count'  
From  
      Users  
      inner join  
      ReplyTo  
      on Users.Username = ReplyTo.Username  
Group By Users.Username  
Order By 2 DESC  
go
```

Results:

	Username	Reply Count
1	EarlyMon	7447
2	SUroot	6704
3	drexappeal	6458
4	Guest	5406
5	funkylogik	5277
6	rushmore	4041
7	El Presidente	3757
8	Thom	3211
9	sean76	2885
10	ironass	2848
11	<s>AndreaCristiano</s>	2789
12	ScandaLeX	2569
13	Rxpert83	2465
14	dan330	2108
15	Rukbat	2079
16	marctronixx	2017
17	Slug	1977
18	JunBringer	1785
19	BigCiX	1784
20	OverByter	1626
21	Mr. Ed	1620
22	johnlgalt	1588
23	Szadzik	1580
24	Frisco	1550
25	matttye	1530