Final Draft

APP STORE DATA MODEL

by

AKSHAT KARAMBE

Northeastern University
College of Engineering

Department of Mechanical & Industrial Engineering
334 Snell Engineering Center
360 Huntington Avenue
Boston, MA 02115

Table of Contents Cluster Overview

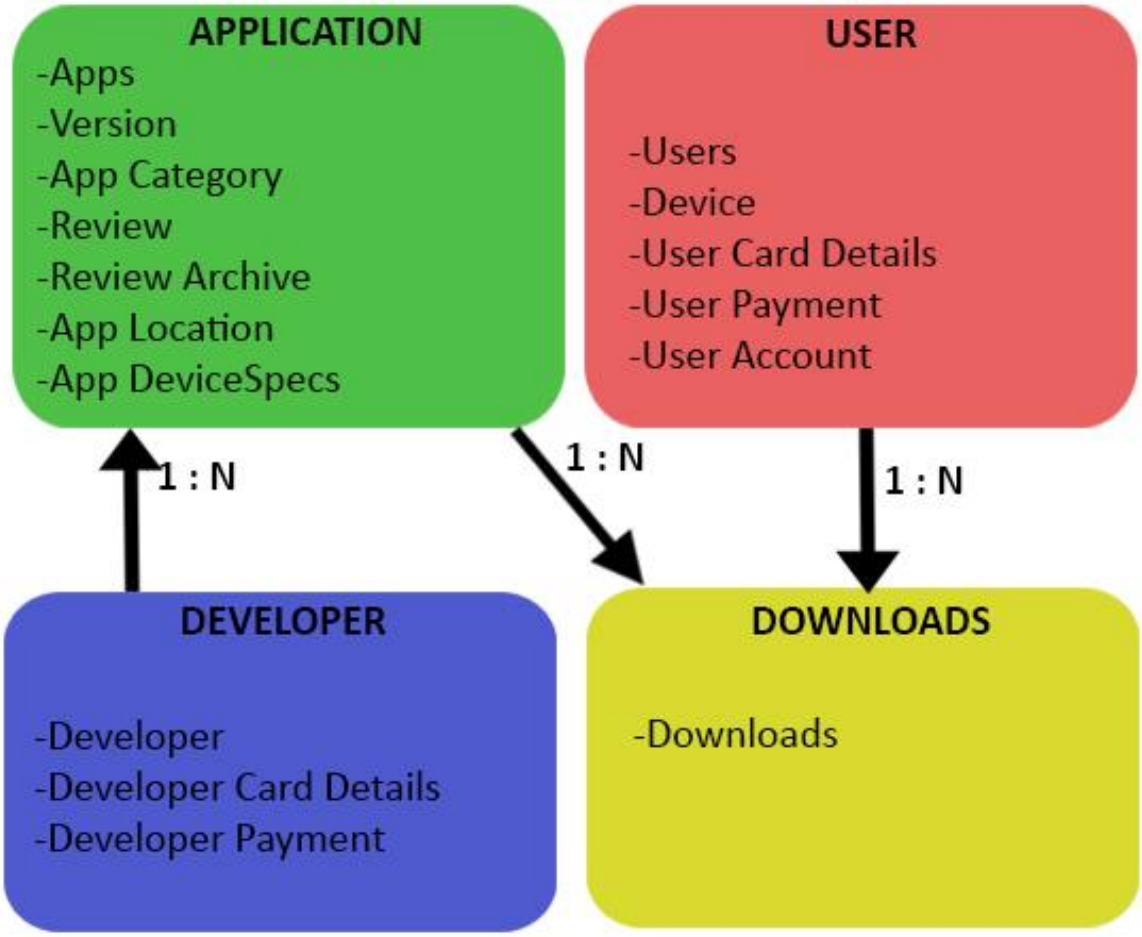
Introduction

The document is generated to provide an outline understanding of a database model of an App Store. This store consists data of all the applications listed in the store as well as the users and developer data.

The database model is created using Toad Data Modeler tool. This model consists of mainly four clusters i.e. Application Cluster, Developer Cluster, User Cluster, and Downloads Cluster, and it also defines the relationship amongst the different entities within the model. Using the Toad Data Modeler tool the DDL script is generated and with the help of that script, a database has been created in Microsoft SQL Server. After the formation of the database, it is populated with the sample data using DML scripts. Additionally, it provides a series of simple queries on the sample data in the database along with the generated output.

The document also gives a concise yet clear description of all the clusters, entities and attributes. It also provides information about the business rules and their scope in the database model. An example data is also provided in the document to get a clear view of the database model. Along with this, the Relationship Description table gives us the information about how the relationship is between the entities. And finally, there is a picture of Data Model showing all clusters, entities, attributes, and relationships

Cluster Overview



Revision History

Following are the revision comments for the document.

1.) Added 3 more tables in the Apps Clusters viz *Review_Archive*, *App_Location,* and *App_DeviceSpecs.*

2.) Added attributes like *InApp_Adv*, *InApp_Purchases, Contact*, and *Website* in Apps entity which gives us more information if there are any in-app advertisements, in-app purchases a user can do and contact information so that a user can visit the website of the application.

3.) Changed the attribute name in the App entity from *App_ageflag* to *App_AgeFlag* to have the consistency in attribute naming.

4.) Added one more entity named *User_Account* in the User Cluster. This will be mandatory to log in to the app store.

5.) Added attributes *Device_CompanyName*, *Device_Ram*, *Device_OS* in the Device entity, this will help us know if the device used by the user is compatible in downloading the app.

6.) Changed the attribute names in the User entity from *User_fname* to *User_FirstName*, *User_lname* to *User_LastName*, *User_gender* to *User_Gender*, *User_phone* to *User_Phone* to have the consistency in attribute naming.

7.) Changed the attribute name in the Developer entity from *Developer_id* to *Developer_Id*, *Developer_fname* to *Developer_FirstName, Developer_lname* to *Developer_LastName*, *Developer_phone* to *Developer_Phone, Developer_address* to *Developer_Address,* and *Developer_webaddress* to *Developer_WebAddress* to have the consistency in attribute naming.

8.) Changed the attribute name in the DeveloperCardDetails entity from *Zip* to *ZipCode* to have the consistency in attribute naming.

9.) Added attributes *DeveloperPayment_Type* and *DeveloperPayment_Amount* in the DeveloperPayment entity to get the transaction information details of the developer.

10.) Added attribute *UserPayment_Amount* in the UserPayment entity to get the transaction information details of the User.

11.) Earlier there was one to many relationship between Developer entity and Apps entity which I changed to many to many.

11.) Earlier there was one to many relationship between Developer entity and Apps entity which I changed to many to many.

12.) Added a non-identifying one to many relationship between Apps entity and Developer Payment entity.

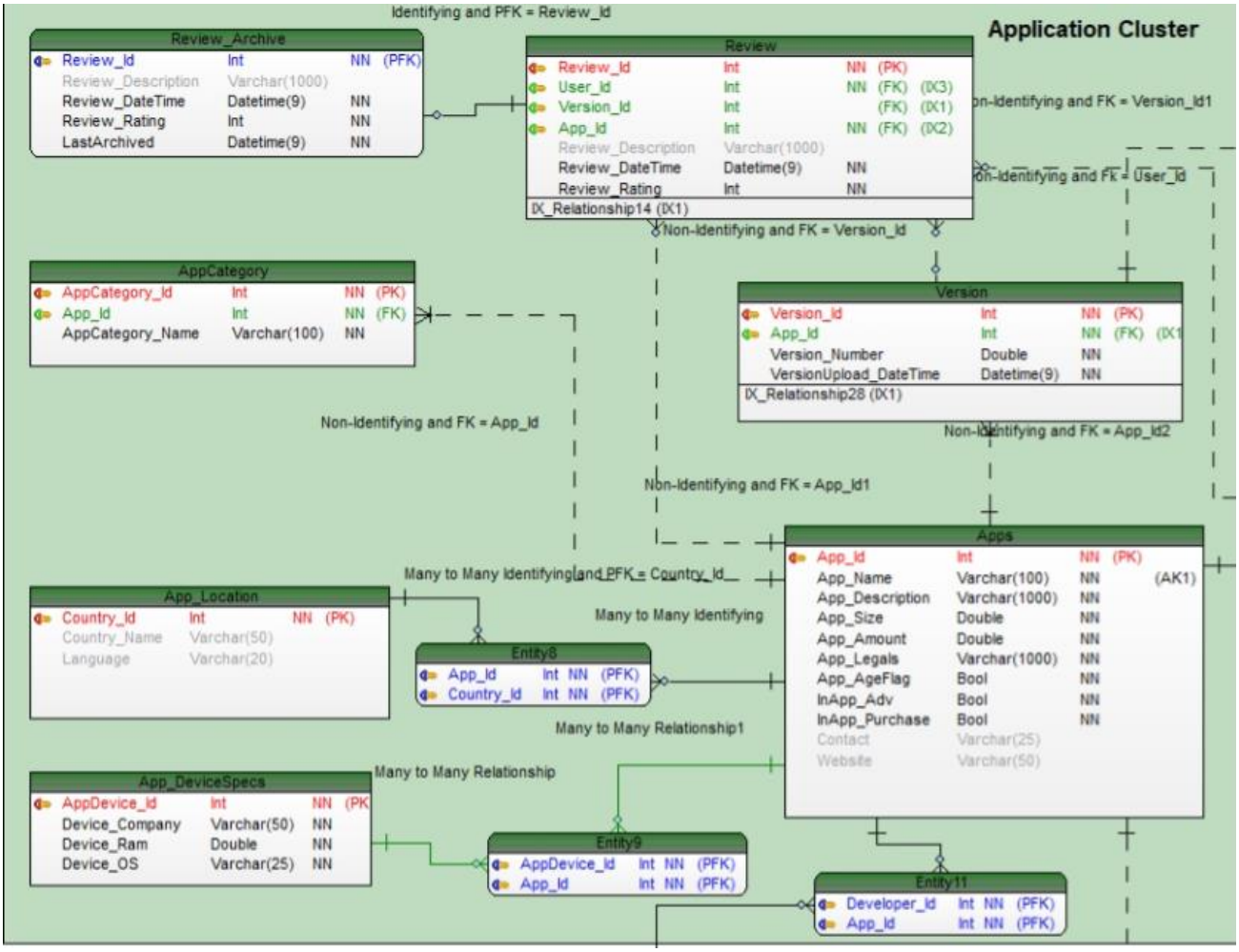
Cluster Description

1.) Cluster – Application

This cluster comprises of 7 entities viz. Apps, Version, AppCategory, Review, Review Archive, App Location, App DeviceSpecs.

This cluster would give us the idea about an application which is in the App Store. The application will be developed by a developer (person or company).

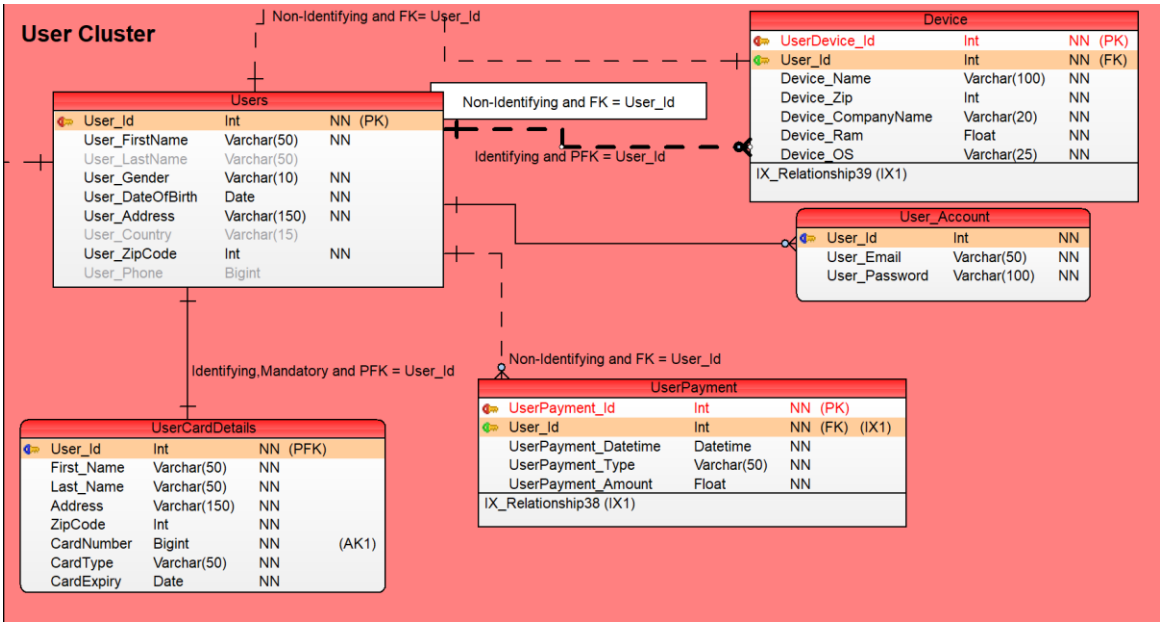
This cluster will give us the information of all the versions of the application, reviews given by the users, under which category the application falls and what are the minimum device specifications needed for the application to download.



2.) Cluster – Users

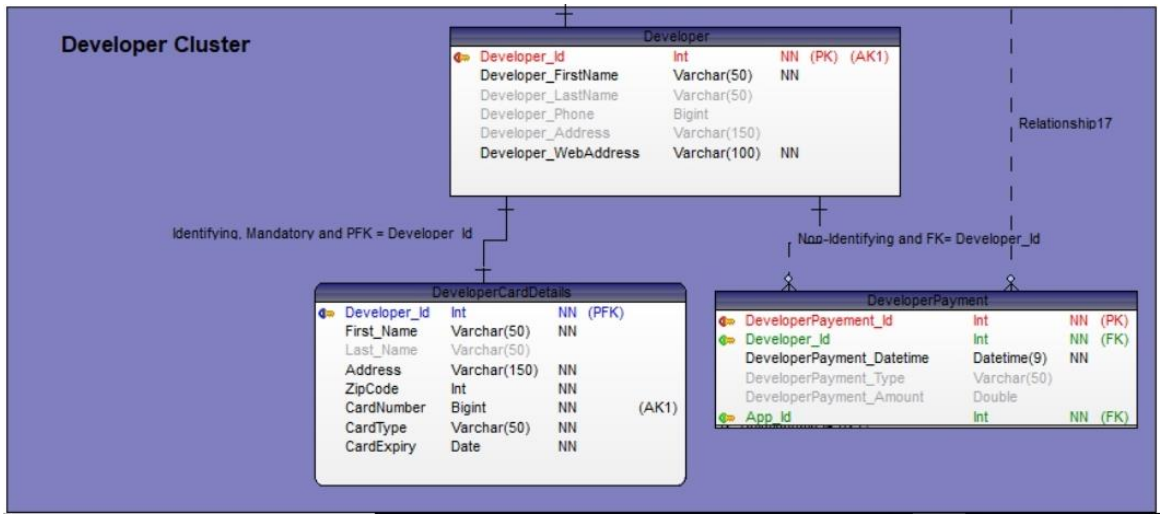
This cluster comprises of 5 entities viz. Users, Device, User Card Details, User Payment, and User Account.

This cluster would give the idea about the users which are using the applications. Also, it will give us the information of the device or devices used by the users to download the app along with the user card details and any payments done. Also, each user will have to make an account to download any application.



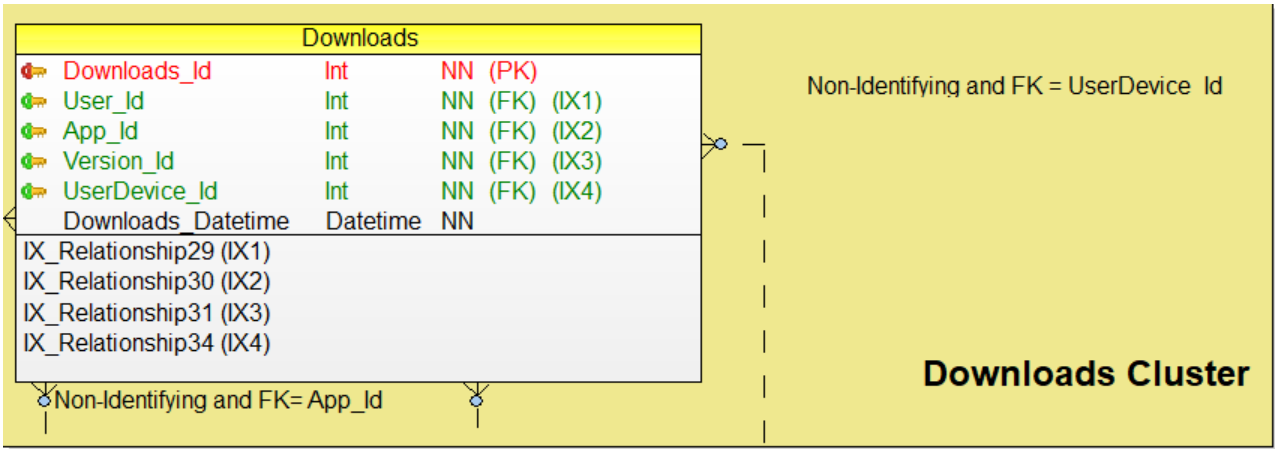
3.) Cluster – Developer

This cluster comprises of 3 entities viz. Developer, Developer Card Details, and Developer Payment. This cluster would give the details about the developer which are developing the applications. Also, it will give us the information of the developer card details and any payments done or received.



4.) Cluster – Downloads

This cluster comprises of only one entity named Downloads. From this cluster we can make a number of analysis viz number of downloads by each user on each device, number of downloads by versions, and a number of download per application. Entities like Apps, Users, Version, and UserDevice are joined with this table. We can also find the date and time of the downloaded application.



Entity Details and Business Rules

1.) App-Application

Entity Definition – This is what the app stores are made for. This is one of the most important entities in the App store database with lots of attribute under it. The purpose of this entity is to briefly describe what the application is. It will contain all the details relating to the particular app.

Business Rules (IS- In Scope || OS- Out of Scope)

APP-IS01: Each application will definitely have a Version.

APP-IS02: Each application will definitely have a Developer or an Author.

APP-IS03: Each application may or may not have a review and ratings.

APP-IS04: Each application may or may not get downloaded in a device by a user.

APP-IS05: Each application will have a unique name and a unique identification number.

APP-IS05: Each application will definitely be listed in any one of the categories.

APP-IS06: There is an indicator which will state if the application can show an advertisement or not.

APP-IS07: Applications are flagged if they have any restrictions like age.

APP-IS08: Each application will be tracked to check if it is free or paid.

APP-IS09: There is an indicator which will tell us that if the app has any in-app purchases for users.

APP-IS10: Each application will have its size mention in MB's (MegaBytes) only.

APP-OS01: Friends who downloaded app is out of scope

APP-OS02: Tracing application by the city is out of scope.

Business Name (Attribute Name)	Datatype	Key/Null	Definition	Example Data
Application Identifier (App_ID)	Int	PK NN	Uniquely Identifies the Application and also generates the sequence number assigned by the system	1001
Developer Identifier (Developer_Id)	Int	FK NN	This is the foreign key in this table. We can get Developers data through this	3001
Application Name (App_Name)	Varchar(100)	Unique NN	This is the application name which will be unique for every application	Facebook
Application Description (App_Description)	Varchar(1000)	NN	This attribute will describe the application	Free social networking website
Application Size (App_Size)	Float	NN	Size of the application. Will only be in MB's	34.5MB
Application Amount (App_Amount)	Float	NN	This will tell us the amount of the application in Dollars only	$0
Legal information about apps (App_Legals)	Varchar(1000)	NN	This will give us the information about the app agreements and the laws.	Terms and Conditions
Age Restriction (App_AgeFlag)	Bit	NN	This will tell us whether the app has any restrictions or not.	0
Inside App Purchases InApp_Purchases	Bit	NN	This will tell us if there are any in-app purchases a user can make	0
Inside Application Advertisements InApp_Adv	Bit	NN	This will tell us if the application has any advertisements	1
Contact Info Contact	Varchar(25)	Null	This will give us the contact information for the application. This can be a NULL field	
Web Address Website	Varchar(50)	Null	This will have the web address for the application if there is any. This can be a NULL Field.	https://www.facebook.com/

Sample Data:

App_Id	Developer_id	App_Name	App_Description	App_Size	App_Amount	App_Legals	App_AgeFlag	InApp_Adv	InApp_Purchase	Contact	Website
1001	3001	Facebook	Free social networking website	34.4	0	Term & Conditions	0	1	0		https://www.facebook.com/
1002	3002	Angry Bird	Angry Birds is a video game franchise created by Finnish company Rovio Entertainmen	12.5	10	Term & Conditions	0	1	1		https://www.angrybirds.com/
1003	3003	Quora	Quora is a question-and-answer site where questions are asked, answered	15	0	Term & Conditions	0	0	0		https://www.quora.com/topic/Websites
1004	3004	WhatsApp	WhatsApp Messenger is a cross-platform instant messaging application	10	0	Term & Conditions	0	0	0		https://www.whatsapp.com/
1005	3005	Clash of Clans	Clash of Clans is a freemium mobile strategy video game	100	30	Term & Conditions	1	1	1		https://clashofclans.com/

DML to Insert Sample Data:

```
SELECT * FROM Apps
```

100 %

Results | Messages

	App_Id	Developer_Id	App_Name	App_Description	App_Size	App_Amount	App_Legals	App_AgeFlag	InApp_Adv	InApp_Purchase	Contact	Website
1	1001	3001	Facebook	Free social networking website	34.4	0	Term & Conditions	0	1	0	NULL	https://www.facebook
2	1002	3002	Angry Bird	Angry Birds is a video game franchise created by Fi...	12.5	10	Term & Conditions	0	1	1		"https://www.angrybird
3	1003	3003	Quora	Quora is a question-and-answer site where questio...	15	0	Term & Conditions	0	0	0		https://www.quora.com
4	1004	3004	WhatsApp	WhatsApp Messenger is a cross-platform instant m...	10	0	Term & Conditions	0	0	0		https://www.whatsapp
5	1005	3005	Clash of Clans	Clash of Clans is a freemium mobile strategy video ...	100	30	Term & Conditions	1	1	1		https://clashofclans.co

2.) Ver-Version

Entity Definition – This is made so that we can get the data of the current or the previous versions of a certain application. This will be helpful in analyzing how well a certain version of the application is doing.

Business Rules (IS- In Scope || OS- Out of Scope)

VER-IS01: Each version will definitely have one application.

VER-IS02: Each version of a particular application may or may not get downloaded by a user.

VER-IS03: Each version may or may not get reviewed by the users.

VER-IS04: Version for each application must be maintained.

Business Name (Attribute Name)	Datatype	Key/Null	Definition	Example Data
Version Identifier (Version_Id)	Int	PK NN	Uniquely identifies each version for a certain application and also generates the sequence number assigned by the system	1201
Application Identifier (App_Id)	Int	NN	This is the foreign key by which we can fetch the apps data.	1001
Version Number (Version_Number)	Float	NN	This will give us the information of all the versions for a certain application	1.01
Version Upload Date and Time (VersionUpload_DateTime)	Datetime	NN	This will tell us when was the version uploaded by the developer.	2010-05-05 00:00:00

Sample Data:

Version_Id	App_Id	Version_Number	VersionUpload_DateTime
1201	1001	1.01	05/05/2010 0:00
1202	1001	1.02	06/06/2010 0:00
1203	1001	1.03	07/07/2010 0:00
1204	1002	1.01	05/05/2011 0:00
1205	1003	1.01	05/05/2012 0:00
1206	1003	1.02	06/06/2012 0:00
1207	1004	1.01	08/08/2010 0:00

1208	1004	2.01	11/11/2010 0:00
1209	1005	1.01	11/11/2015 0:00

DML to Insert Sample Data:

```
SELECT * FROM Version
```

100 %

Results | Messages

	Version_Id	App_Id	Version_Number	VersionUpload_DateTime
1	1201	1001	1.01	2010-05-05 00:00:00.000
2	1202	1001	1.02	2010-06-06 00:00:00.000
3	1203	1001	1.03	2010-07-07 00:00:00.000
4	1204	1002	1.01	2011-05-05 00:00:00.000
5	1205	1003	1.01	2012-05-05 00:00:00.000
6	1206	1003	1.02	2012-06-06 00:00:00.000
7	1207	1004	1.01	2010-08-08 00:00:00.000

Query executed successfully. | DESKTOP-F55KP3S\SQLDEV17 (1... | DESKTOP-F55KP3S\AXAT (65) | AppStoreVersion2 | 00:00:00 | 9 rows

3.) Rev-Review

Entity Definition –In this entity, the users can post their experience of using the certain application. The attributes of this entity will update regularly. The purpose of this entity is to allow the app users to give their opinions on the open forum which will be helpful for the new downloaders. It's very helpful when you want an unbiased opinion about a certain app.

Business Rules (IS- In Scope || OS- Out of Scope)

REV-IS01: For every review, there must be a user.

REV-IS02: There will be definitely one application for every review given by the user.

REV-IS03: There will be definitely one version for every review given by the user.

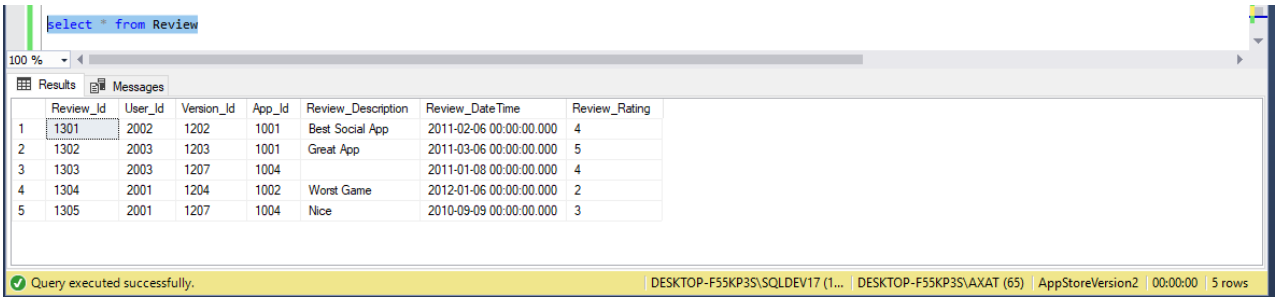
Business Name (Attribute Name)	Datatype	Key/Null	Definition	Example Data
Review Identifier (Review_Id)	Int	PK NN	Each review will have a unique identification and also generates the sequence number assigned by the system	1301
User Identifier (User_Id)	Int	FK NN	Foreign key from the User table to fetch the user information.	2002
Version Identifier (Version_Id)	Int	FK NN	Foreign key from the Version Table to fetch the Version information.	1202
Application Identifier (App_Id)	Int	FK NN	Foreign key from the Apps Table to fetch the Application information.	1001
User Comments (Review_Description)	Varchar(1000)	Null	Description of the review given by the user. One of the main reason this table is made for	best social app.

Review Post Date (Review_DateTime)	Datetime	NN	On what date the user has given the review.	2011-02-06 00:00:00
User Ratings (Review_Rating)	Int	NN	What rating the user wants to give the application	4

Sample Data:

Review_Id	User_Id	Version_Id	App_Id	Review_Description	Review_DateTime	Review_Rating
1301	2002	1202	1001	Best Social App	06/02/2011 0:00	4
1302	2003	1203	1001	Great App	06/03/2011 0:00	5
1303	2003	1207	1004		08/01/2011 0:00	4
1304	2001	1204	1002	Worst Game	06/01/2012 0:00	2
1305	2001	1207	1004	Nice	09/09/2010 0:00	3

DML to Insert Sample Data:



4.) RevArc-Review Archive

Entity Definition – This entity is used to archive the reviews given by the user. This entity will be updated once in every 6 months.

Business Rules (IS- In Scope || OS- Out of Scope)

REVARC-IS01: All the reviews will be stored here for the future reference.

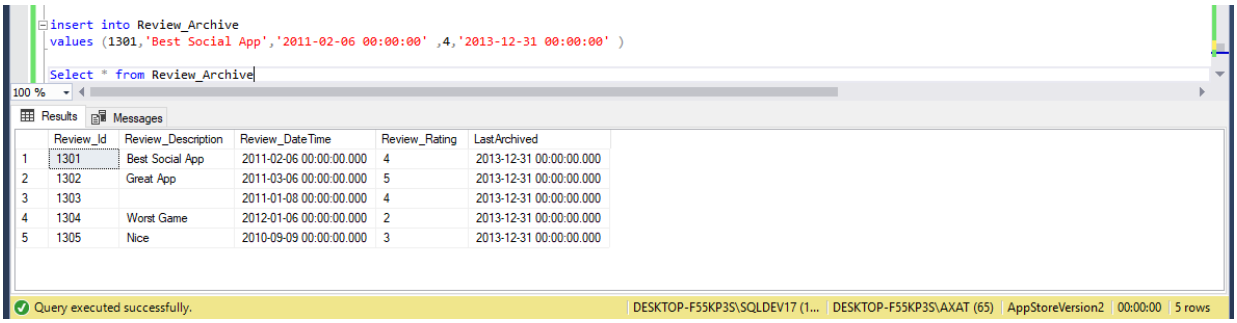
REVARC-IS02: This will be updated every 6 months

Business Name (Attribute Name)	Datatype	Key/Null	Definition	Example Data
Review Identifier (Review_Id)	Int	PFK NN	This is the PFK that is it's the primary key from the Review table as this table is dependent on Review Table	1301
User Comments (Review_Description)	Varchar(1000)	Null	Description of the review given by the user.	best social app.
Review Post Date (Review_DateTime)	Datetime	NN	On what date the user has given the review.	2011-02-06 00:00:00
User Ratings (Review_Rating)	Int	NN	What rating the user wants to give the application	4
Last Date of Archive (LastArchived)	Datetime	NN	This will tell us when was the data archived	2013-12-31 00:00:00

Sample Data:

Review_Id	Review_Description	Review_DateTime	Review_Rating	LastArchived
1301	Best Social App	06/02/2011 0:00	4	31/12/2013 0:00
1302	Great App	06/03/2011 0:00	5	31/12/2013 0:00
1303		08/01/2011 0:00	4	31/12/2013 0:00
1304	Worst Game	06/01/2012 0:00	2	31/12/2013 0:00
1305	Nice	09/09/2010 0:00	3	31/12/2013 0:00

DML to Insert Sample Data:



```
insert into Review_Archive
values (1301,'Best Social App','2011-02-06 00:00:00' ,4,'2013-12-31 00:00:00' )

Select * from Review_Archive
```

100 %

Results | Messages

	Review_Id	Review_Description	Review_DateTime	Review_Rating	LastArchived
1	1301	Best Social App	2011-02-06 00:00:00.000	4	2013-12-31 00:00:00.000
2	1302	Great App	2011-03-06 00:00:00.000	5	2013-12-31 00:00:00.000
3	1303		2011-01-08 00:00:00.000	4	2013-12-31 00:00:00.000
4	1304	Worst Game	2012-01-06 00:00:00.000	2	2013-12-31 00:00:00.000
5	1305	Nice	2010-09-09 00:00:00.000	3	2013-12-31 00:00:00.000

Query executed successfully. | DESKTOP-F55KP3S\SQLDEV17 (1... | DESKTOP-F55KP3S\AXAT (65) | AppStoreVersion2 | 00:00:00 | 5 rows

5.) AppCat- Application Category

Entity Definition –The purpose of this entity is to tell the users in which category the application has been listed ie. is it in the "Top 50" list or can be found in the "Games" category. This will make easy for the users to find a certain application.

Business Rules (IS- In Scope || OS- Out of Scope)

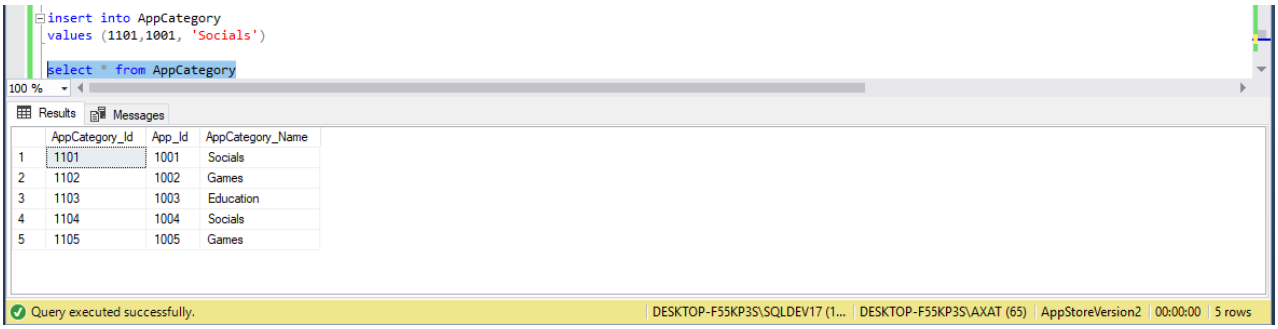
APPCAT-IS01: Every category may or may not have an application listed.

Business Name (Attribute Name)	Datatype	Key/Null	Definition	Example Data
Application category Identifier (AppCategory_Id)	Int	PK NN	Uniquely identifies each category for a certain application and also generates the sequence number assigned by the system	1101
Application Identifier (App_Id)	Int	NN	This is the foreign key by which we can fetch the apps data.	1001
Application category Name (AppCategory_Name)	VarChar(100)	NN	This will tell us the name of the category in which the application has been listed.	Socials

Sample Data:

AppCategory_Id	App_Id	AppCategory_Name
1101	1001	Socials
1102	1002	Games
1103	1003	Education
1104	1004	Socials
1105	1005	Games

DML to Insert Sample Data:



6.) AppLoc- Application Location

Entity Definition –The purpose of this entity is to tell in which country is the application available and in what language is the application available to download.

Business Rules (IS- In Scope || OS- Out of Scope)

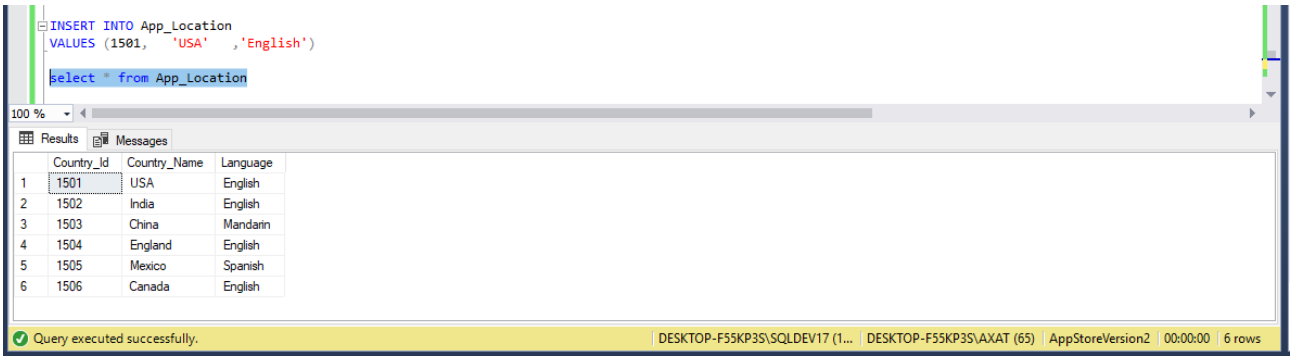
APPLOC-IS01: The location and language of each application are captured.

Business Name (Attribute Name)	Datatype	Key/Null	Definition	Example Data
Country Identification (Country_Id)	Int	PK NN	Uniquely identifies each country and also generates the sequence number assigned by the system	1501
Name of the Country (Country_Name)	Varchar(50)	Null	This will list the country name for in which the app has been released	USA
The language of the App Language	Varchar(20)	Null	This will list the language in which the app has been released	English

Sample Data:

Country_Id	Country_Name	Language
1501	USA	English
1502	India	English
1503	China	Mandarin
1504	England	English
1505	Mexico	Spanish
1506	Canada	English

DML to Insert Sample Data:



7.) AppDevSpec- Application Device Specifications

Entity Definition –This entity will tell us the minimum device specification required for the application to get downloaded in the device.

Business Rules (IS- In Scope || OS- Out of Scope)

APPDEVSPEC-IS01: Listing of the devices operating system required for the applications.

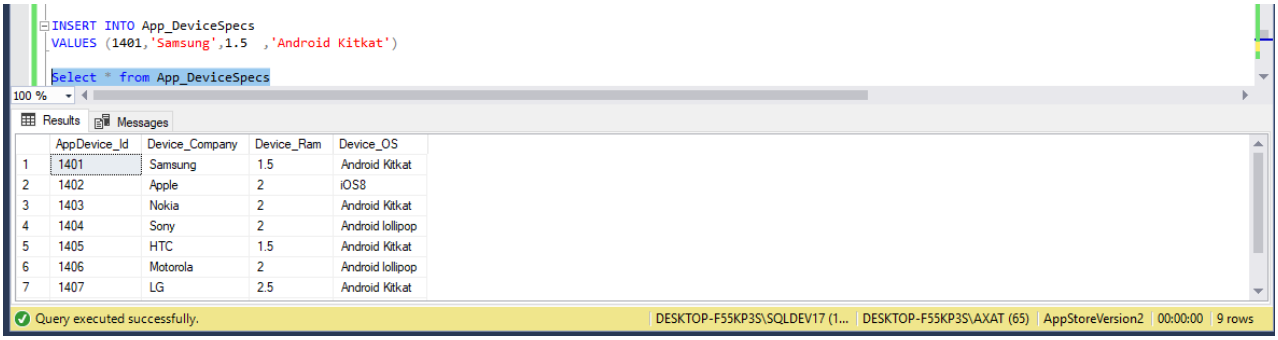
APPDEVSPEC-IS02: Listing of the devices ram required to run the applications.

Business Name (Attribute Name)	Datatype	Key/Null	Definition	Example Data
Application Device Identification (AppDevice_Id)	Int	PK NN	Uniquely identifies each device and also generates the sequence number assigned by the system	1401
Manufacturer of the Device (Device_Company)	Varchar(50)	NN	This will give us the company of the device	Samsung
Ram of the Device Device_Ram	Float	NN	This will give the minimum requirement of ram for the particular device	1.5
Operating System of the phone (Device_OS)	Varchar(25)	NN	This will give the minimum requirement of Operating System for the particular device	Android KitKat

Sample Data:

AppDevice_Id	Device_Company	Device_Ram	Device_OS
1401	Samsung	1.5	Android KitKat
1402	Apple	2	iOS8
1403	Nokia	2	Android KitKat
1404	Sony	2	Android Lollipop
1405	HTC	1.5	Android KitKat
1406	Motorola	2	Android Lollipop
1407	LG	2.5	Android KitKat
1408	OnePlus	1.8	Android Lollipop
1409	BlackBerry	2	Android Lollipop

DML to Insert Sample Data:



8.) Usr - Users

Entity Definition- This entity will have details of the users who will have an account in the App store. Every user will have a unique user id which will be very useful to maintain the records. We can store data of every individual in a very methodical order so that we can retrieve the data in the future.

Business Rules (IS- In Scope || OS- Out of Scope)

USR-IS01: For every user, a unique user id will be generated.

USR-IS02: Each user may or may not download applications

USR-IS03: Every user may or may not give a review for a particular application.

USR-IS04: Every user may or may not need to pay for the paid applications

USR-IS05: Every user may or may not have a single device or multiple devices on which he/she may download the application.

USR-IS06: Every user will have his/her card details to make the purchase if needed.

USR-IS07: Every user must have a user account to download an app.

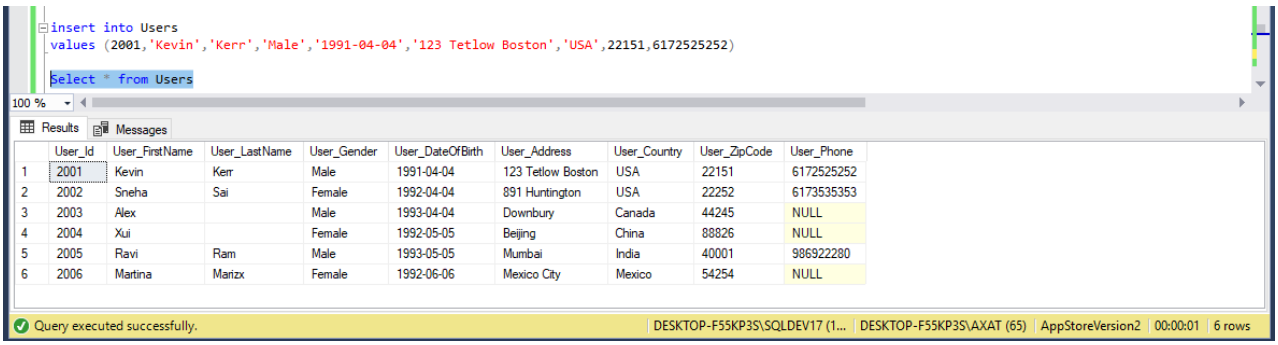
Business Name (Attribute Name)	Datatype	Key/Null	Definition	Example Data
User Identifier (User_ID)	Int	PK NN	Identifies each user uniquely and also generates the sequence number assigned by the system	2001
User First Name (User_FirstName)	Varchar(50)	NN	First Name of the User, helpful in sorting the data by name	Kevin
User last Name (User_LastName)	Varchar(50)		Family or the Last name of the User, helpful in sorting the data by name	Kerr
User Gender (User_Gender)	Varchar(10)	NN	Gender of the person, helpful in sorting the data by gender	Male
Date of Birth (User_DateOfBirth)	Date	NN	Gives us the date of birth of the user. Based on this we can flag application as appropriate or not appropriate	04/04/1991
User Address details (User_Address)	Varchar(150)	NN	Address of the user to sort the data by location	123 Tetlow Boston
Country of the User (User_Country)	Varchar(15)	Null	The country in which the user is using the application	USA
User zip code (User_ZipCode)	Int	NN	This will tell us the are of the user	02215

User Phone contact (User_Phone)	Bigint	Null	This will give us the contact of the user.	6172525252

Sample Data:

User_Id	User_FirstName	User_LastName	User_Gender	User_DateOfBirth	User_Address	User_Country	User_ZipCode	User_Phone
2001	Kevin	Kerr	Male	04/04/1991	123 Tetlow Boston	USA	22151	6172525252
2002	Sneha	Sai	Female	04/04/1992	891 Huntington	USA	22252	6173535353
2003	Alex		Male	04/04/1993	Downbury	Canada	44245	
2004	Xui		Female	05/05/1992	Beijing	China	88826	
2005	Ravi	Ram	Male	05/05/1993	Mumbai	India	40001	986922280
2006	Martina	Marizx	Female	06/06/1992	Mexico City	Mexico	54254	

DML to Insert Sample Data:



9.) UsrCd – User Card Details

Entity Definition- This entity will have credit or debit card details of the users who want to purchase the application or who want to do in-app purchases.

Business Rules (IS- In Scope || OS- Out of Scope)

USRCD-IS01: Every user will definitely have their card details inserted to make purchases.

USRCD-IS02: Only credit cards and debit cards will be used to do transactions.

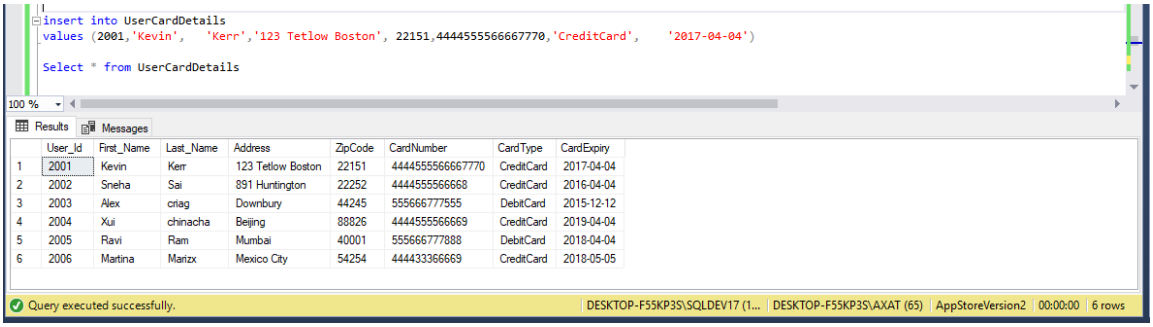
USRCD-OS01: Gift cards used to do the transaction is out of scope.

Business Name (Attribute Name)	Datatype	Key/Null	Definition	Example Data
User Identifier (User_Id)	Int	PFK NN	This is the PFK of this table from which we can get the user id as well as other information from the users table.	2001
Cardholder first name (First_name)	Varchar(50)	NN	The first name of the Cardholder	Kevin
Cardholder first name (Last_Name)	Varchar(50)	NN	The last name of the Cardholder	Kerr
Cardholder address (Address)	Varchar(150)	NN	Address of the user.	123 Tetlow Boston
Cardholder Zip code (ZipCode)	Int	NN	This will tell us the area cardholder is from.	22151
Card Number (CardNumber)	Bigint	NN	This will be the unique card number for each user.	4444555566667 777
Type of card used (CardType)	Varchar(50)	NN	This will tell us if the card used is Credit Card or Debit Card.	Credit Card
Expiry Date of the Card (card expiry)	Date	NN	This will tell us when the card is going to expire.	2017-04-04

Sample Data:

User_Id	First_name	Last_name	Address	ZipCode	CardNumber	CardType	CardExpiry
2001	Kevin	Kerr	123 Tetlow Boston	22151	4444555566667770	CreditCard	20170404
2002	Sneha	Sai	891 Huntington	22252	4444555566668	CreditCard	04/04/2016
2003	Alex	criag	Downbury	44245	555666777555	DebitCard	12/12/2015
2004	Xui	chinacha	Beijing	88826	4444555566669	CreditCard	04/04/2019
2005	Ravi	Ram	Mumbai	40001	555666777888	DebitCard	04/04/2018
2006	Martina	Marizx	Mexico City	54254	444433366669	CreditCard	05/05/2018

DML to Insert Sample Data:



```
insert into UserCardDetails
values (2001,'Kevin',   'Kerr','123 Tetlow Boston', 22151,4444555566667770,'CreditCard',    '2017-04-04')

Select * from UserCardDetails
```

100 %

Results | Messages

	User_Id	First_Name	Last_Name	Address	ZipCode	CardNumber	CardType	CardExpiry
1	2001	Kevin	Kerr	123 Tetlow Boston	22151	4444555566667770	CreditCard	2017-04-04
2	2002	Sneha	Sai	891 Huntington	22252	4444555566668	CreditCard	2016-04-04
3	2003	Alex	criag	Downbury	44245	555666777555	DebitCard	2015-12-12
4	2004	Xui	chinacha	Beijing	88826	4444555566669	CreditCard	2019-04-04
5	2005	Ravi	Ram	Mumbai	40001	555666777888	DebitCard	2018-04-04
6	2006	Martina	Marizx	Mexico City	54254	444433366669	CreditCard	2018-05-05

Query executed successfully. | DESKTOP-F55KP3S\SQLDEV17 (1... | DESKTOP-F55KP3S\AXAT (65) | AppStoreVersion2 | 00:00:00 | 6 rows

10.) UsrPtm – User Payment

Entity Definition- This entity will tell us for what the user has made the payment. This will also tell us the date and time of the payment done.

Business Rules (IS- In Scope || OS- Out of Scope)

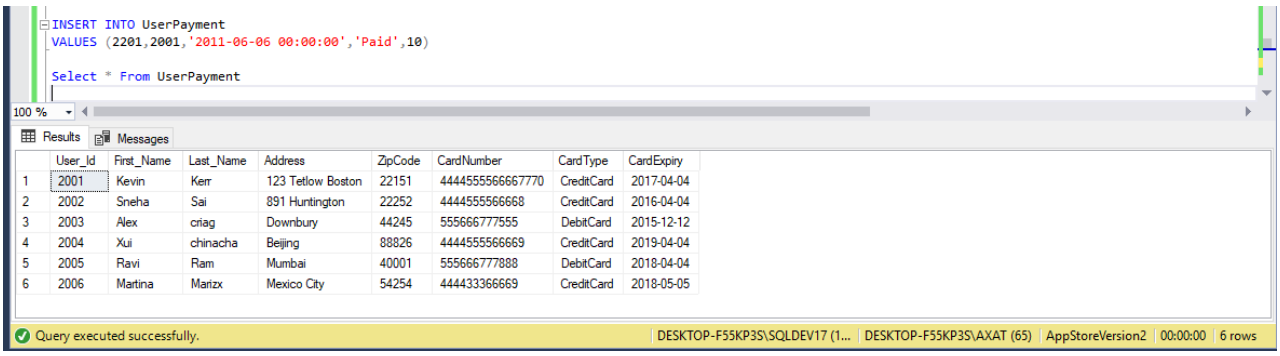
USRPTM-IS01: For every payment made there must be a user.

Business Name (Attribute Name)	Datatype	Key/Null	Definition	Example Data
Payment Identifier (UserPayment_Id)	Int	PK NN	This will identify each payment individually and also generates the sequence number assigned by the system. It will act as a transaction id.	2201
User Identifier (User_Id)	Int	NN	This is the FK of this table from which we can get the user id as well as other information.	2001
Date and time of the payment (UserPayment_Datetime)	Datetime	NN	At what time and on which date the user has made the payment	2011-06-06 00:00:00
The Type of Payment (UserPayment_Type)	Varchar(50)	NN	This will tell us if the user has paid or received the payment.	Paid
Amount of payment received/made (UserPayment_Amount)	Float	NN	This will give us the amount which was paid or received in dollars	$10

Sample Data:

UserPayment_Id	User_Id	UserPayment_Datetime	UserPayment_Type	UserPayment_Amount
2201	2001	06/06/2011 0:00	Paid	10
2202	2005	05/06/2011 0:00	Paid	10
2203	2006	06/06/2011 0:00	Paid	10
2204	2001	12/12/2015 0:00	Paid	30
2205	2005	12/12/2015 0:00	Paid	30

DML to Insert Sample Data:



```
INSERT INTO UserPayment
VALUES (2201,2001,'2011-06-06 00:00:00','Paid',10)

Select * From UserPayment
```

100 %

Results | Messages

	User_Id	First_Name	Last_Name	Address	ZipCode	CardNumber	CardType	CardExpiry
1	2001	Kevin	Kerr	123 Tetlow Boston	22151	4444555566667770	CreditCard	2017-04-04
2	2002	Sneha	Sai	891 Huntington	22252	4444555566668	CreditCard	2016-04-04
3	2003	Alex	criag	Downbury	44245	555666777555	DebitCard	2015-12-12
4	2004	Xui	chinacha	Beijing	88826	4444555566669	CreditCard	2019-04-04
5	2005	Ravi	Ram	Mumbai	40001	555666777888	DebitCard	2018-04-04
6	2006	Martina	Marizx	Mexico City	54254	444433366669	CreditCard	2018-05-05

Query executed successfully. | DESKTOP-F55KP3S\SQLDEV17 (1... | DESKTOP-F55KP3S\AXAT (65) | AppStoreVersion2 | 00:00:00 | 6 rows

11.) UsrAcc – User Account

Entity Definition- This entity will be storing the user's email and password. Each user must have an account in order to download the application.

Business Rules (IS- In Scope || OS- Out of Scope)

USRACC-IS01: The password must be encrypted before being inserted to maintain the integrity.

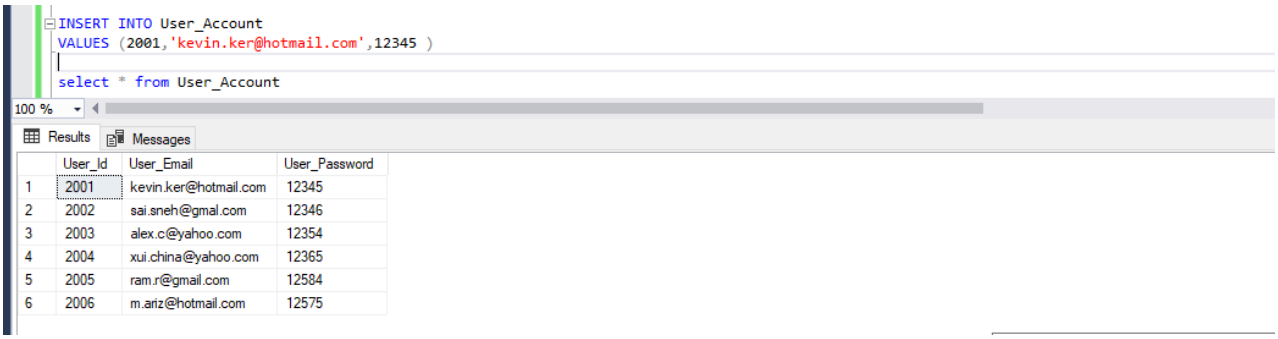
USRACC-IS02: Every user must have an account in order to download an app.

Business Name (Attribute Name)	Datatype	Key/Null	Definition	Example Data
User Identification (User_Id)	Int	PFK NN	This is the PFK of this table from which we can get the user id as well as other information	2001
Email Id of User (User_Email)	Varchar(50)	NN	This will be the login id of the user for the user account	kevin.ker@hotmail.com
Password of the User (User_Password)	Varchar(100)	NN	This will be needed to log in to the account	12345

Sample Data:

User_Id	User_Email	User_Password
2001	kevin.ker@hotmail.com	12345
2002	sai.sneh@gmal.com	12346
2003	alex.c@yahoo.com	12354
2004	xui.china@yahoo.com	12365
2005	ram.r@gmail.com	12584
2006	m.ariz@hotmail.com	12575

DML to Insert Sample Data:



12.) DV – Device

Entity Definition- This entity will tell us which device the user has been using to download the applications.

Business Rules (IS- In Scope || OS- Out of Scope)
DV-IS01: Each device will definitely have a user.

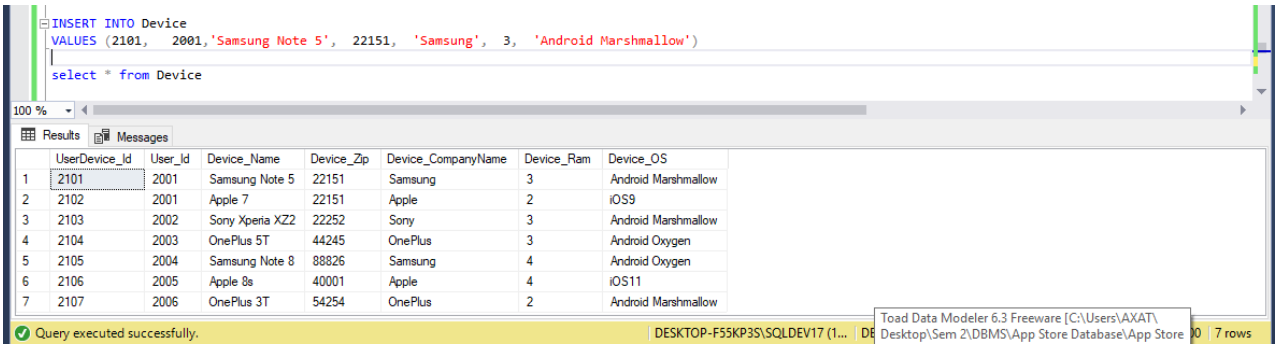
DV-IS02: Each device may or may not have a single or multiple application downloads.

Business Name (Attribute Name)	Datatype	Key/Null	Definition	Example Data
Device Identifier (Device_Id)	Int	PK NN	This will identify each device individually and also generates the sequence number assigned by the system	2101
User Identifier (User_Id)	Int	NN	This is the FK of this table from which we can get the user id as well as other information.	2001
Name of the device (Device_name)	Varchar(100)	NN	This will give us the name of the handset	Samsung Note 5
Area of the Device (Device_Zip)	Int	NN	This will give us the information where the device is been bought from	22151
Manufacturer of the Device (Device_CompanyName)	Varchar(50)	NN	This will give us the company name of the device	Samsung
Ram of the Device Device_Ram	Float	NN	This will give the minimum requirement of ram for the particular device	3
Operating System of the phone (Device_OS)	Varchar(25)	NN	This will give the minimum requirement of Operating System for the particular device	Android Marshmallow

Sample Data:

UserDevice_Id	User_Id	Device_name	Device_Zip	Device_CompanyName	Device_Ram	Device_OS
2101	2001	Samsung Note 5	22151	Samsung	3	Android Marshmallow
2102	2001	Apple 7	22151	Apple	2	iOS9
2103	2002	Sony Xperia XZ2	22252	Sony	3	Android Marshmallow
2104	2003	OnePlus 5T	44245	OnePlus	3	Android Oxygen
2105	2004	Samsung Note 8	88826	Samsung	4	Android Oxygen
2106	2005	Apple 8s	40001	Apple	4	iOS11
2107	2006	OnePlus 3T	54254	OnePlus	2	Android Marshmallow

DML to Insert Sample Data:



13.) DVP - Developer

Entity Definition – This provides the details of each author who developed the application for the app store. Gives knowledge about the authors/publisher who published certain apps.

Business Rules (IS- In Scope || OS- Out of Scope)

DVP-IS01: Every developer may or may not develop an application for the app store.

DVP-IS02: Every developer will have a unique identification number to track their uploads for future reference.

DVP-IS03: Every developer will pay or receive payment to and from the App Store.

DVP-IS04: Every developer will have its card details to make the purchase if needed.

Business Name (Attribute Name)	Datatype	Key/Null	Definition	Example Data
Developer Identifier (Developer_Id)	Int	PK Unique NN	Helps in identifying each author/developer uniquely and also generates the sequence number assigned by the system	3001
Developer First Name (Developer_FirstName)	Varchar(50)	NN	The first name of the author	Aeshna
Developer Last Name (Developer_LastName)	Varchar(50)	Null	The family name of the last name of the author	Kapoor
Developer First Name (Developer_Phone)	Bigint	NN	Phone number of the author	5872525252
Developer living Address (Developer_Address)	Varchar (150)	Null	This will tell us where the author resides or hails from	Boston
Developer Web Address (Developer_WebAddress)	Varchar (100)	NN	Website of the developer	http://www.aeshkapoor.com

Sample Data:

Developer_Id	Developer_F	Developer_LastNa me	Developer_Phone	Developer_Address	Developer_WebAddress
3001	Aeshna	Kapoor	58725252525	Boston	http://www.aeshkapoor.com
3002	Robert	Thomas		California	http://www.rbt.com
3003	Sachin	Jadhav		New Delhi	http://www.sjstudios.com
3004	Chun-An-Ch	Deiago	9595858585	Beijing	http://www.cac.com
3005	Nilesh	Nerkar	2252525252	Mumbai	http://nerkar.n.com
3006	Sthita	Joseph		Rochester	http://j.sthita.com

DML to Insert Sample Data:

```
INSERT INTO Developer
VALUES (3001,'Aeshna','Kapoor',58725252525,'Boston','http://www.aeshkapoor.com')

select * from Developer
```

100 %

Results | Messages

	Developer_Id	Developer_FirstName	Developer_LastName	Developer_Phone	Developer_Address	Developer_WebAddress
1	3001	Aeshna	Kapoor	58725252525	Boston	http://www.aeshkapoor.com
2	3002	Robert	Thomas	NULL	California	http://www.rbt.com
3	3003	Sachin	Jadhav	NULL	New Delhi	http://www.sjstudios.com
4	3004	Chun-An-Chou	Deiago	9595858585	Beijing	http://www.cac.com
5	3005	Nilesh	Nerkar	2252525252	Mumbai	http://nerkar.n.com
6	3006	Sthita	Joseph	NULL	Rochester	http://j.sthita.com

Query executed successfully. | DESKTOP-F55KP3S\SQLDEV17 (1... | DESKTOP-F55KP3S\AXAT (65) | AppStoreVersion2 | 00:00:00 | 6 rows

14.) DvpPtm – Developer Payment

Entity Definition- This entity will tell us for what the developer has made the payment or received the payment for. This will also tell us the date and time of the payment done.

Business Rules (IS- In Scope || OS- Out of Scope)
DVPPTM-IS01: For every payment made there must be a developer.

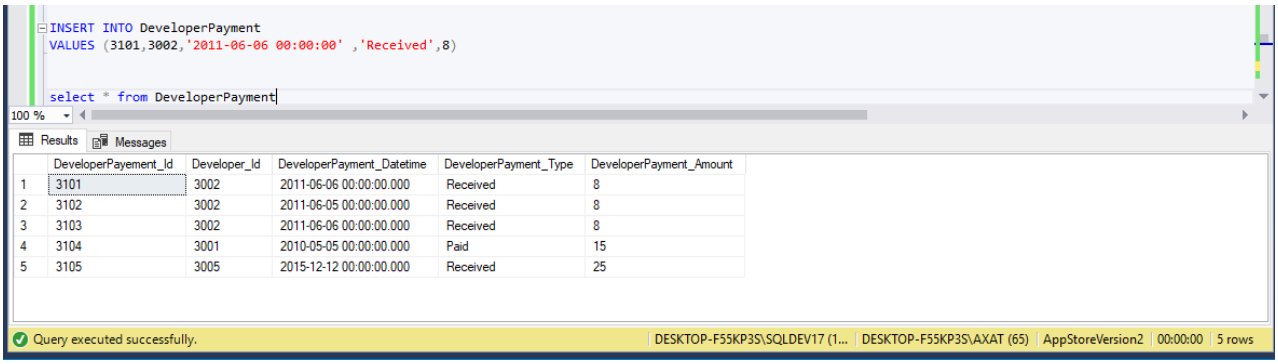
DVPPTM-IS02: Developer will pay for the app to be in the app store as well as receive money from the store.

Business Name (Attribute Name)	Datatype	Key/Null	Definition	Example Data
Developer Payment Identifier (DeveloperPayment_Id)	Int	PK NN	This will identify each payment individually and also generates the sequence number assigned by the system. It will act as a transaction id	3101
User Identifier (Developer_Id)	Int	NN	This is the FK of this table from which we can get the developer id as well as other information.	3002
Date and time of the payment (DeveloperPayment_Datetime)	Datetime	NN	At what time and on which date the developer has made or received the payment	2011-06-06 00:00:00
The Type of Payment (DeveloperPayment_Type)	Varchar(50)	Null	This will tell us about what the payment has been done.	Received
Amount Received/Paid (DeveloperPayment_Amount)	Float	Null	This is the amount received/paid to/by the developer in dollars	$8

Sample Data:

DeveloperPayement_Id	Developer_id	DeveloperPayment_Datetime	DeveloperPayment_Type	DeveloperPayment_Amount
3101	3002	06/06/2011 0:00	Received	8
3102	3002	05/06/2011 0:00	Received	8
3103	3002	06/06/2011 0:00	Received	8
3104	3001	05/05/2010 0:00	Paid	15
3105	3005	12/12/2015 0:00	Received	25

DML to Insert Sample Data:



15.) DvpCd – Developer Card Details

Entity Definition- This entity will have credit or debit card details of the developer who wants to purchase a place in the app store for its application or who want to put up and advertisement.

Business Rules (IS- In Scope || OS- Out of Scope)

DVPCD-IS01: Every developer will definitely have their card details inserted to make purchases.

DVPCD-IS02: The transaction will only be done by

Business Name (Attribute Name)	Datatype	Key/Null	Definition	Example Data
Developer Identifier (Developer_Id)	Int	PFK NN	This is the PFK of this table from which we can get the developer id as well as other information.	3001
Cardholder first name (First_Name)	Varchar(50)	NN	The first name of the Cardholder	Aeshna
Cardholder first name (Last_Name)	Varchar(50)	Null	The last name of the Cardholder	Kapoor
Cardholder address (Address)	Varchar(150)	NN	Address of the user.	Boston
Cardholder Zip code (ZipCode)	Int	NN	This will tell us where the cardholder is from.	22151
Card Number (CardNumber)	Bigint	NN	This will be the unique card number for each user.	1111222233334444
Type of card used (CardType)	Varchar(50)	NN	This will tell us if the card used is Credit Card or Debit Card.	Credit Card
Expiry Date of the Card (CardExpiry)	Date	NN	This will tell us when the card is going to expire.	01/01/2020

Sample Data:

Developer_id	First_name	Last_name	Address	ZipCode	CardNumber	CardType	CardExpiry
3001	Aeshna	Kapoor	Boston	22151	1.11122E+15	CreditCard	01/01/2020
3002	Robert	Thomas	California	4585255	2.22233E+15	DebitCard	04/04/2018
3003	Sachin	Jadhav	New Delhi	500414	3.33344E+15	DebitCard	04/04/2022
3004	Chun-An-Ch	Deiago	Beijing	5244566	3.33344E+15	CreditCard	01/09/2020
3005	Nilesh	Nerkar	Mumbai	400014	3.33322E+15	CreditCard	01/08/2020
3006	Sthita	Joseph	Rochester	78785	1.11156E+15	DebitCard	02/05/2022

DML to Insert Sample Data:

```sql
INSERT INTO DeveloperCardDetails
VALUES (3002,'Robert','Thomas','California',4585255,2222333344445550,'DebitCard','2018-04-04'),
(3003,  'Sachin',   'Jadhav',   'New Delhi',500414, 3333444455556660,   'DebitCard',    '2022-04-04'),
(3004,  'Chun-An-Chou', 'Deiago',   'Beijing',  5244566,    3333444422226660,   'CreditCard','2020-01-09'),
(3005,  'Nilesh',   'Nerkar',   'Mumbai',   400014, 3333222255556660,   'CreditCard',   '2020-01-08'),
(3006   ,'Sthita',  'Joseph',   'Rochester',    78785   ,1111555533334440   ,'DebitCard',   '2022-02-05')

select * from DeveloperCardDetails
```

100 %

Results | Messages

	Developer_Id	First_Name	Last_Name	Address	ZipCode	CardNumber	CardType	CardExpiry
1	3001	Aeshna	Kapoor	Boston	22151	1111222233334440	CreditCard	2020-01-01
2	3002	Robert	Thomas	California	4585255	2222333344445550	DebitCard	2018-04-04
3	3003	Sachin	Jadhav	New Delhi	500414	3333444455556660	DebitCard	2022-04-04
4	3004	Chun-An-Chou	Deiago	Beijing	5244566	3333444422226660	CreditCard	2020-01-09
5	3005	Nilesh	Nerkar	Mumbai	400014	3333222255556660	CreditCard	2020-01-08
6	3006	Sthita	Joseph	Rochester	78785	1111555533334440	DebitCard	2022-02-05

Query executed successfully. | DESKTOP-F55KP3S\SQLDEV17 (1... | DESKTOP-F55KP3S\AXAT (65) | AppStoreVersion2 | 00:00:00 | 6 rows

16.) Dwn-Downloads

Entity Definition – This will help us get the information about the downloads of the particular application. This will house the key information about which user downloaded which application and how many times an application has been downloaded on a particular device.

Business Rules (IS- In Scope || OS- Out of Scope)

DWN-IS01: Each download will definitely have one application.

DWN-IS02: Each download will definitely have one user.

DWN-IS03: If there is a download then it will definitely be on one of the devices.

DWN-IS04: If there is a download then it will definitely be of one of the versions.

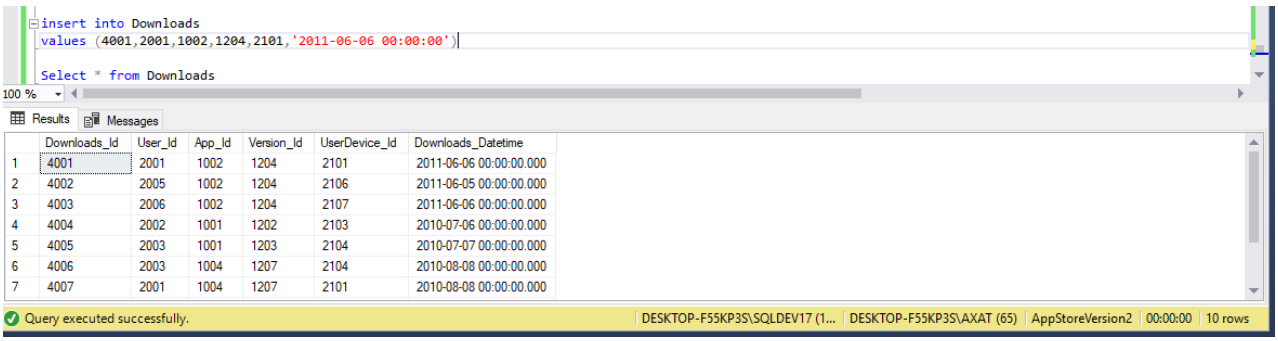
Business Name (Attribute Name)	Datatype	Key/Null	Definition	Example Data
Download Identifier (Download_Id)	Int	PK Unique Null	Uniquely identifies each download of each application and also generates the sequence number assigned by the system	4001
Users Identification (User_Id)	Int	FK NN	Foreign key from the user's entity to fetch the users information	2001
Application Identification (App_Id)	Int	FK NN	Foreign key from the Apps entity to fetch the data about the application.	1002
Version Identifier (Version_Id)	Int	FK NN	Foreign key from the version entity to fetch the version information	1204
Device Identifier (UserDevice_Id)	Int	FK NN	Foreign key from the Device entity to fetch the device information	2101

Download Date & Time (Downloads_Datetime)	Datetime	NN	Date and time when each download has taken place	2011-06-06 00:00:00

Sample Data:

Downloads_Id	User_Id	App_Id	Version_Id	UserDevice_Id	Downloads_Datetime
4001	2001	1002	1204	2101	06/06/2011 0:00
4002	2005	1002	1204	2106	05/06/2011 0:00
4003	2006	1002	1204	2107	06/06/2011 0:00
4004	2002	1001	1202	2103	06/07/2010 0:00
4005	2003	1001	1203	2104	07/07/2010 0:00
4006	2003	1004	1207	2104	08/08/2010 0:00
4007	2001	1004	1207	2101	08/08/2010 0:00
4008	2005	1004	1208	2106	11/11/2010 0:00
4009	2005	1005	1209	2106	12/12/2015 0:00
4010	2001	1005	1209	2101	12/12/2015 0:00

DML to Insert Sample Data:



Relationship Description

Table 1	Table 2	Relationship Type	Identifying / Non-Identifying	Description
Apps	Version	One to many and Mandatory one, Mandatory many,	Non-Identifying	Every App will definitely have at least one version and for every version, there must be an application
App	Downloads	One to many and Mandatory one, Not Mandatory many	Non-Identifying	Each app may or may not get downloaded but for every download, there must be an app
App	Reviews	One to many and Mandatory one, Not Mandatory many	Non-Identifying	Each app may or may not get reviewed by a user but for every review, there must be an app
App	AppCategory	One to many and Mandatory one, Mandatory many	Non-Identifying	Each app will be listed in at least one category and for every category, there will be an app
App	App_Location	One to many and Mandatory one, Mandatory many	Identifying	Each app will be released for certain countries or locations hence each app will be released for at least one country and a country will at least have one app
App	App_DeviceSpecs	One to many and Mandatory one, Mandatory many	Identifying	For each app, there will be minimum specification requirements.
App	DeveloperPayment	One to many and Mandatory one, Not Mandatory many	Non-Identifying	For each payment there must be an app for which the developer has been paid
Version	Downloads	One to many and Mandatory one, Not Mandatory many	Non-Identifying	Each version of a certain app may or may not get downloaded by a user but if there is a download then It must of for a certain version of a particular app

Version	Reviews	One to many and Not Mandatory one, Not Mandatory many	Non-Identifying	Each version of a certain app may or may not get reviewed by a user and it is not necessary to have a review for each version.
Review	Review_Archive	One to One and Mandatory one, Not Mandatory One	Identifying	The reviews will be achieved once every 6 months. This table exists because of the Review Table
Developer	App	Many to Many and Mandatory	Identifying	Each developer may or may not develop an app but if an app is developed then it must have a developer or multiple developers.
Developer	DeveloperCardDetails	One to One and Mandatory for both	Identifying	Each developer must have its card details to pay for any kind of advertisements or apps.
Developer	DeveloperPayment	One to Many and Mandatory one, Not Mandatory many	Non-Identifying	Each developer may or may not get paid or may or may not make a payment but if there is a transaction then there must be a developer.
Users	Reviews	One to Many and Mandatory one, Not Mandatory many	Non-Identifying	A user may or may not give a review. The user may give multiple reviews but if there is a review then it must be given by a user
Users	Downloads	One to Many and Mandatory one, Not Mandatory many	Non-Identifying	A user may or may not download one app or multiple apps. But if there is a download made then it must have a user associated with it
Users	Device	One to Many and Mandatory one, Not Mandatory many	Non-Identifying	A user may or may not have one or multiple devices on which it can download the app.

				But if a download is made on a device then it will definitely have a user
Users	User_Account	One to Many and Mandatory One, Not Mandatory Many	Identifying	For a user to make any download it should have an account. It may have one or multiple accounts. But if there is an account then it will be a user.
Users	UserPayment	One to Many and Mandatory one, Not Mandatory many	Non-Identifying	Each User may or may not make a payment but if there is a transaction then there must be a user.
Users	UserCardDetails	One to One and Mandatory for both	Identifying	Each user must have its card details to pay for any kind of advertisements or apps.
Device	Downloads	One to Many and Mandatory one, Not Mandatory many	Non-Identifying	For every device, there may or may not have one or multiple downloads but for every application downloaded there must be a device on which it gets downloaded

Basic Query on Sample Data & Output:

In this query, we can find out versionwise reviews given to each app.

```
SELECT Apps.App_Name, Review.Review_Description, Version.Version_Number
FROM Apps
JOIN Version ON Apps.App_Id = Version.App_Id
JOIN Review ON Version.Version_Id = Review.Version_Id
```

100 %

Results | Messages

	App_Name	Review_Description	Version_Number
1	Facebook	Best Social App	1.02
2	Facebook	Great App	1.03
3	WhatsApp		1.01
4	Angry Bird	Worst Game	1.01
5	WhatsApp	Nice	1.01

Query executed successfully. | DESKTOP-F55KP3S\SQLDEV17 (1... | DESKTOP-F55KP3S\AXAT (54) | AppStoreVersion2 | 00:00:00 | 5 rows

In this query, we can see which applications are stored in the 'Socials' category

```
SELECT  Apps.App_Id, Apps.App_Name, AppCategory.AppCategory_Name

FROM Apps

JOIN AppCategory ON Apps.App_Id = AppCategory.App_Id

WHERE AppCategory_Name = 'Socials';
```

100 %

Results | Messages

	App_Id	App_Name	AppCategory_Name
1	1001	Facebook	Socials
2	1004	WhatsApp	Socials

Query executed successfully. | DESKTOP-F55KP3S\SQLDEV17 (1... | DESKTOP-F55KP3S\AXAT (54) | AppStoreVersion2 | 00:00:00 | 2 rows

From this query, we can find out what devices the female users are using.

```
SELECT Users.User_FirstName,Users.User_Gender, Device.Device_Name

FROM Users JOIN Device ON Users.User_Id = Device.User_Id

WHERE Users.User_Gender = 'Female'
```

100 %

Results | Messages

	User_FirstName	User_Gender	Device_Name
1	Sneha	Female	Sony Xperia XZ2
2	Xui	Female	Samsung Note 8
3	Martina	Female	OnePlus 3T

Query executed successfully. | DESKTOP-F55KP3S\SQLDEV17 (1... | DESKTOP-F55KP3S\AXAT (54) | AppStoreVersion2 | 00:00:00 | 3 rows

From this query, we come to know who downloaded which application along with the version number

```
SELECT Users.User_FirstName , Apps.App_Name, Downloads.Downloads_Datetime, Version.Version_Number

FROM Users

JOIN Downloads ON Users.User_Id = Downloads.User_Id

JOIN Apps ON Downloads.App_Id = Apps.App_Id

JOIN Version ON Apps.App_Id = Version.App_Id
```

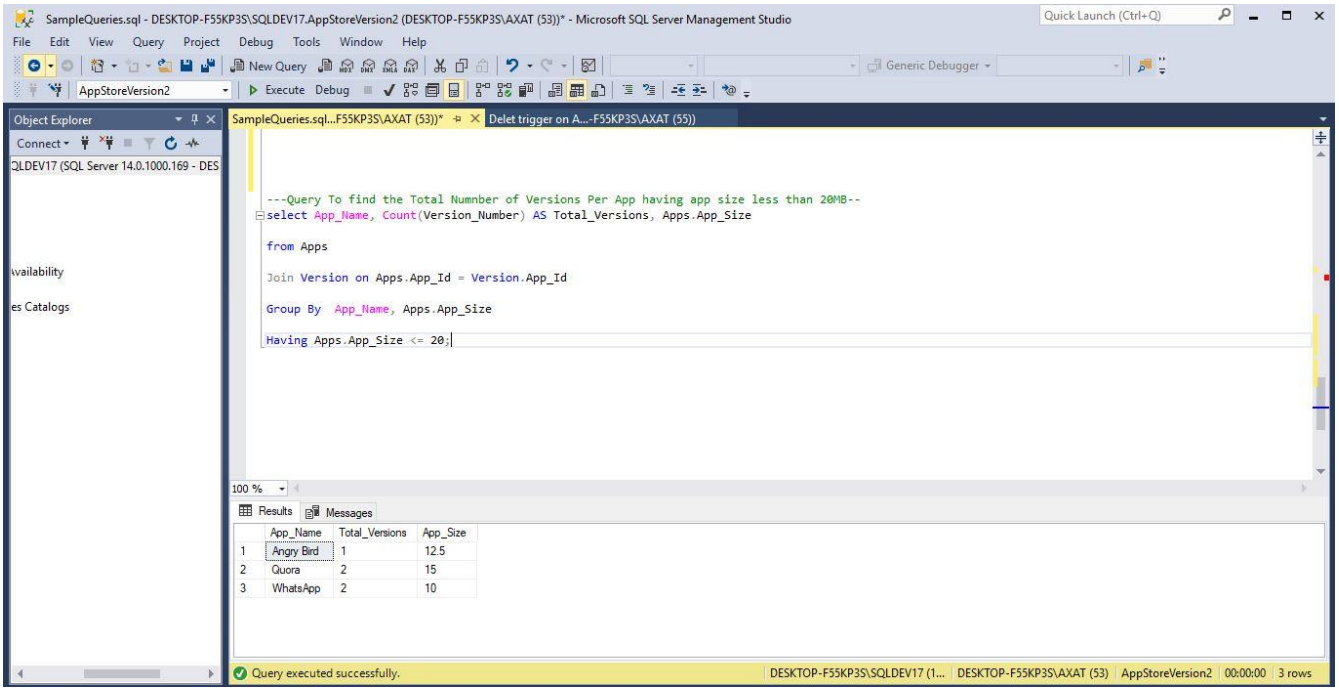
100 %

Results | Messages

	User_FirstName	App_Name	Downloads_Datetime	Version_Number
1	Kevin	Angry Bird	2011-06-06 00:00:00.000	1.01
2	Ravi	Angry Bird	2011-06-05 00:00:00.000	1.01
3	Martina	Angry Bird	2011-06-06 00:00:00.000	1.01
4	Sneha	Facebook	2010-07-06 00:00:00.000	1.01
5	Sneha	Facebook	2010-07-06 00:00:00.000	1.02
6	Sneha	Facebook	2010-07-06 00:00:00.000	1.03
7	Alex	Facebook	2010-07-07 00:00:00.000	1.01
8	Alex	Facebook	2010-07-07 00:00:00.000	1.02
9	Alex	Facebook	2010-07-07 00:00:00.000	1.03
10	Alex	WhatsApp	2010-08-08 00:00:00.000	1.01
11	Alex	WhatsApp	2010-08-08 00:00:00.000	2.01
12	Kevin	WhatsApp	2010-08-08 00:00:00.000	1.01
13	Kevin	WhatsApp	2010-08-08 00:00:00.000	2.01
14	Ravi	WhatsApp	2010-11-11 00:00:00.000	1.01
15	Ravi	WhatsApp	2010-11-11 00:00:00.000	2.01
16	Ravi	Clash of ...	2015-12-12 00:00:00.000	1.01
17	Kevin	Clash of ...	2015-12-12 00:00:00.000	1.01

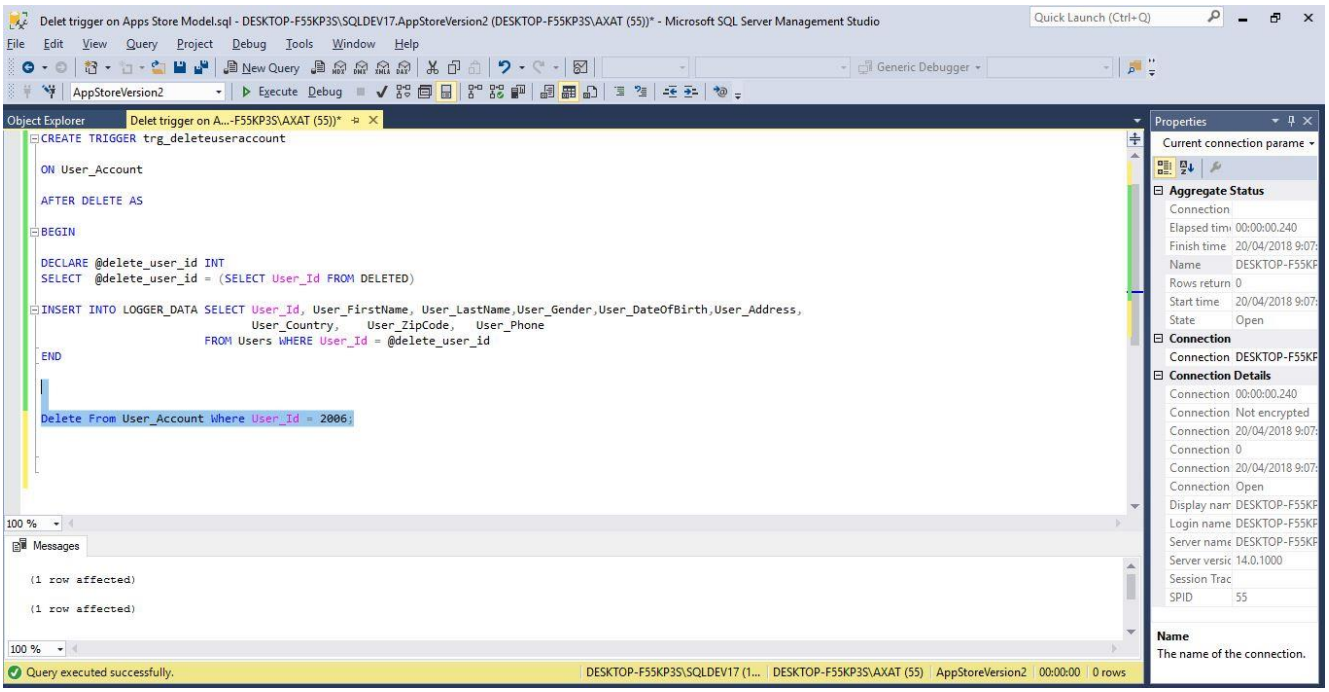
Query executed successfully. | DESKTOP-F55KP3S\SQLDEV17 (1... | DESKTOP-F55KP3S\AXAT (54) | AppStoreVersion2 | 00:00:00 | 17 rows

Query to find out the number of versions per application

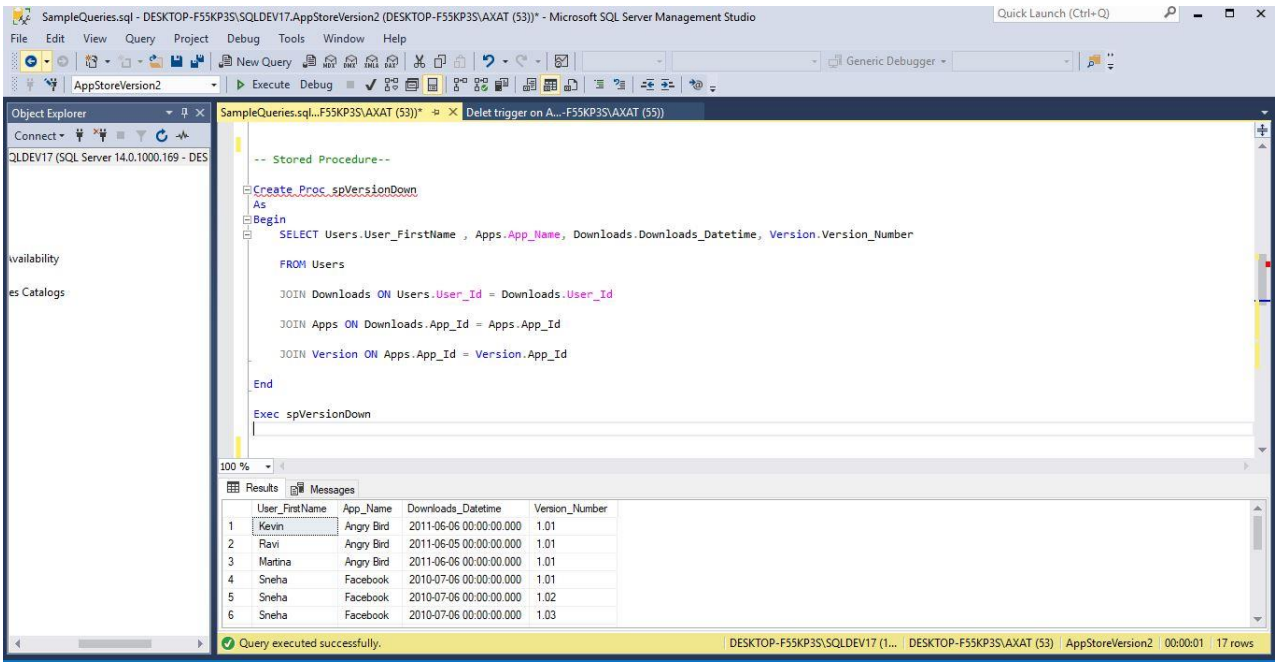


Triggers & Stored Procedures:

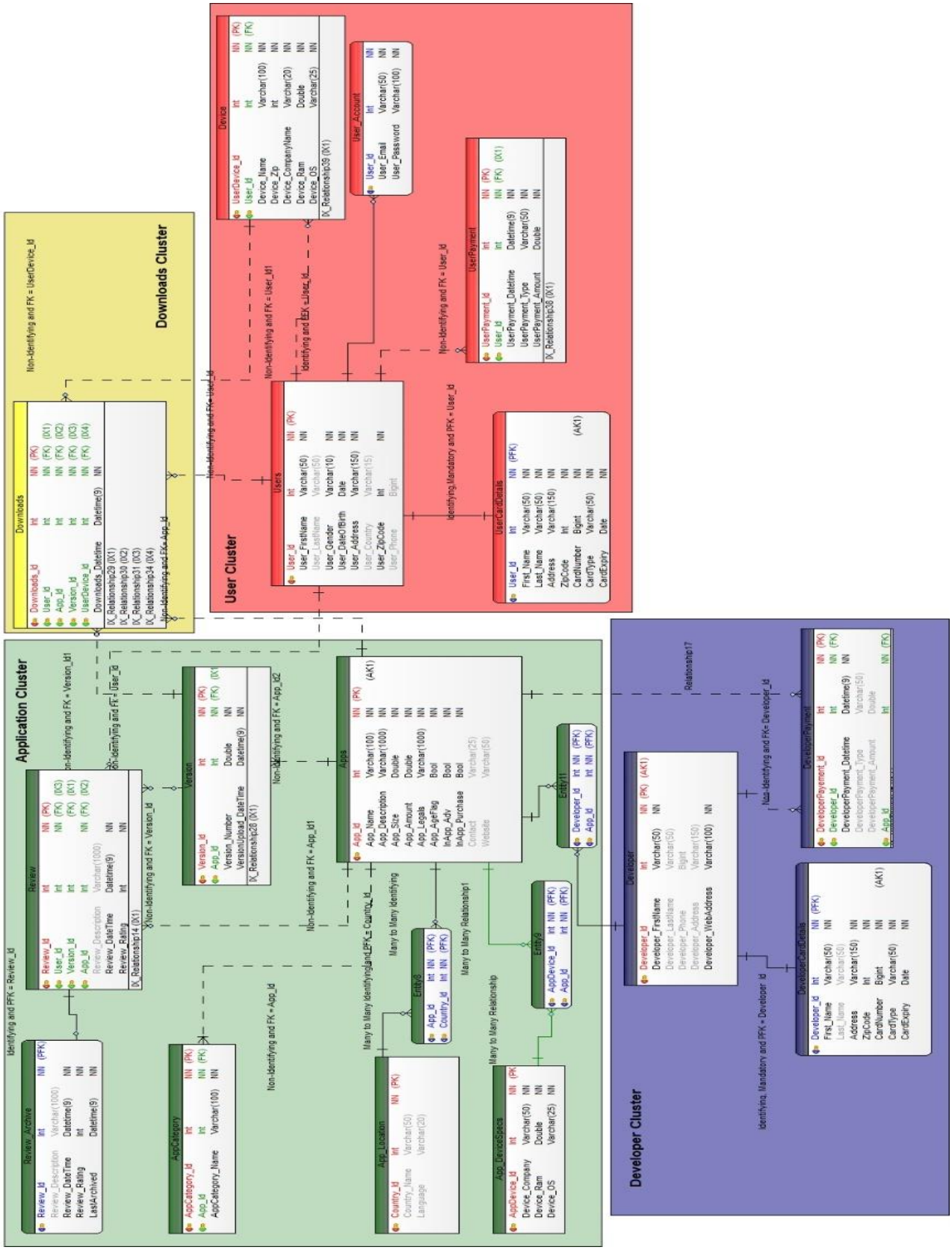
I have written a trigger which will save the user data into another table whenever an account is being deleted.



Stored Procedure:

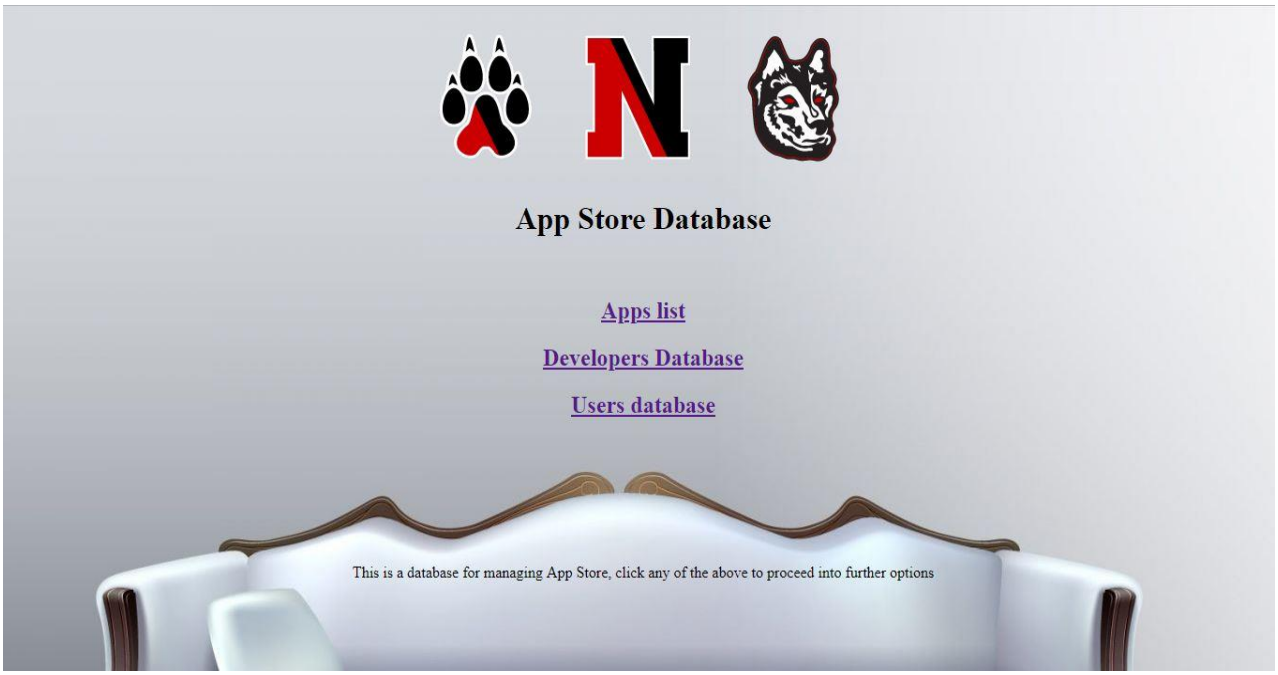


Data Model

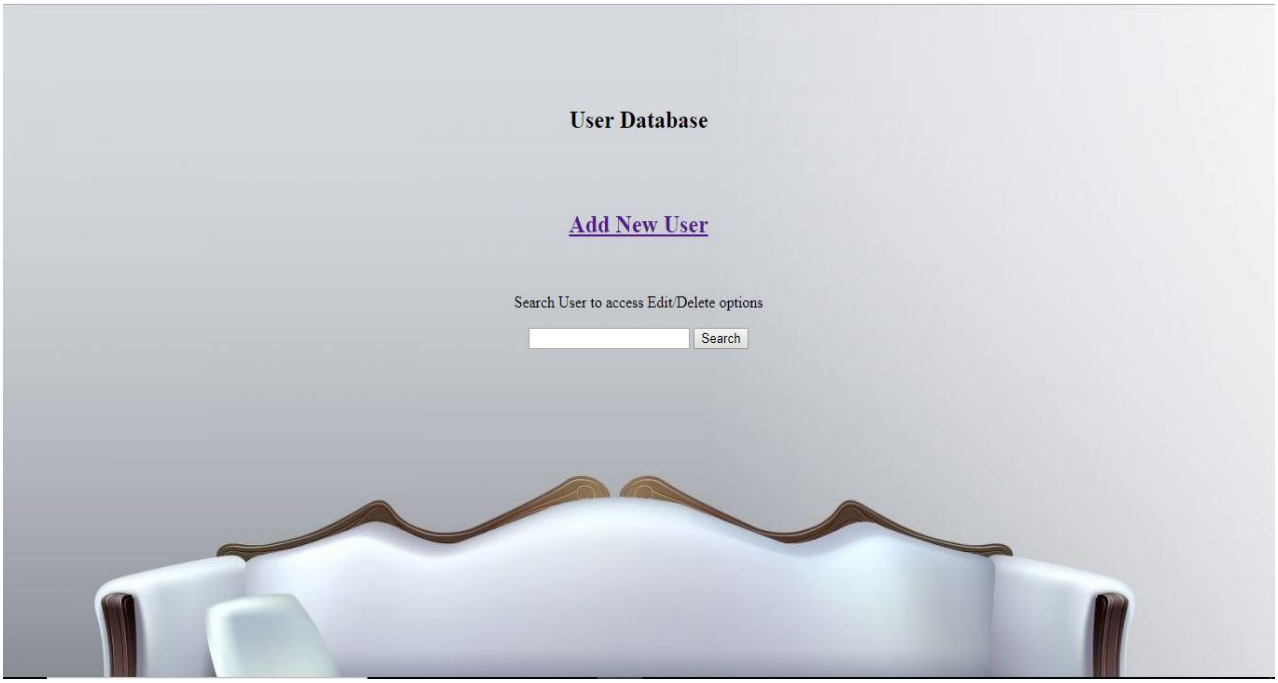

Downloads Cluster
Downloads
Downloads_Id int NN (PK)
User_Id int NN (FK) (IX1)
App_Id int NN (FK) (IX2)
Version_Id int NN (FK) (IX3)
UserDevice_Id int NN (FK) (IX4)
Downloads_Datetime Datetime(9) NN
IX_Relationship29 (IX1)
IX_Relationship30 (IX2)
IX_Relationship31 (IX3)
IX_Relationship34 (IX4)
Non-Identifying and FK= UserDevice_Id
Non-Identifying and FK= User_Id
User Cluster
Users
User_Id Int NN (PK)
User_Firstname Varchar(50) NN
User_Lastname Varchar(50)
User_Gender Varchar(10) NN
User_Date0fBirth Date NN
User_Address Varchar(150) NN
User_Country Varchar(15)
User_ZipCode Int NN
User_Phone Bigint
Non-Identifying and FK = User_Id1
Identifying and FK = User_Id
Device
UserDevice_Id Int NN (PK)
User_Id Int NN (FK)
Device_Name Varchar(100) NN
Device_Zip Int NN
Device_CompanyName Varchar(20) NN
Device_Ram Double NN
Device_OS Varchar(25) NN
IX_Relationship39 (IX1)
User_Account
User_Id Int NN
User_Email Varchar(50) NN
User_Password Varchar(100) NN
Non-Identifying and FK = User_Id
UserPayment
UserPayment_Id Int NN (PK)
User_Id Int NN (FK) (IX1)
UserPayment_Datetime Datetime(9) NN
UserPayment_Type Varchar(50) NN
UserPayment_Amount Double NN
IX_Relationship38 (IX1)
Identifying,Mandatory and PFK = User_Id
UserCardDetails
User_Id Int NN (PFK)
First_Name Varchar(50) NN
Last_Name Varchar(50) NN
Address Varchar(150) NN
ZipCode Int NN
CardNumber Bigint NN (AK1)
CardType Varchar(50) NN
CardExpiry Date NN
Application Cluster
Review_Archive
Review_Id Int NN (PFK)
Review_Description Varchar(1000)
Review_DateTime Datetime(9) NN
Review_Rating Int NN
LastArchived Datetime(9) NN
Identifying and PFK = Review_Id
Review
Review_Id Int NN (PK)
User_Id Int NN (FK) (IX3)
Version_Id Int (FK) (IX1)
App_Id Int NN (FK) (IX2)
Review_Description Varchar(1000)
Review_DateTime Datetime(9) NN
Review_Rating Int NN
IX_Relationship14 (IX1)
Non-Identifying and FK = Version_Id
Non-Identifying and FK = Version_Id1
Non-Identifying and FK = User_Id
Version
Version_Id Int NN (PK)
App_Id Int NN (FK) (IX1)
Version_Number Double NN
VersionUpload_DateTime Datetime(9) NN
IX_Relationship28 (IX1)
Non-Identifying and FK = App_Id2
Non-Identifying and FK = App_Id1
Apps
App_Id Int NN (PK)
App_Name Varchar(100) NN (AK1)
App_Description Varchar(1000) NN
App_Size Double NN
App_Amount Double NN
App_Legals Varchar(1000) NN
App_AgeFlag Bool NN
InApp_Adv Bool NN
InApp_Purchase Bool NN
Contact Varchar(25)
Website Varchar(50)
Entity11
Developer_Id Int NN (PFK)
App_Id Int NN (PFK)
Entity9
AppDevice_Id Int NN (PFK)
App_Id Int NN (PFK)
Entity8
App_Id Int NN (PFK)
Country_Id Int NN (PFK)
Many to Many Identifying
Many to Many Identifying and PFK = Country_Id
Many to Many Relationship1
Many to Many Relationship
AppCategory
AppCategory_Id Int NN (PK)
App_Id Int NN (FK)
AppCategory_Name Varchar(100) NN
Non-Identifying and FK = App_Id
App_Location
Country_Id Int NN (PK)
Country_Name Varchar(50)
Language Varchar(20)
App_DeviceSpecs
AppDevice_Id Int NN (PK)
Device_Company Varchar(50) NN
Device_Ram Double NN
Device_OS Varchar(25) NN
Developer Cluster
Developer
Developer_Id Int NN (PK) (AK1)
Developer_FirstName Varchar(50) NN
Developer_LastName Varchar(50)
Developer_Phone Bigint
Developer_Address Varchar(150)
Developer_Web-Address Varchar(100) NN
Relationship17
Non-Identifying and FK= Developer_Id
DeveloperPayment
DeveloperPayement_Id Int NN (PK)
Developer_Id Int NN (FK)
DeveloperPayment_Datetime Datetime(9) NN
DeveloperPayment_Type Varchar(50)
DeveloperPayment_Amount Double
App_Id Int NN (FK)
DeveloperCardDetails
Developer_Id Int NN (PFK)
First_Name Varchar(50) NN
Last_Name Varchar(50)
Address Varchar(150) NN
ZipCode Int NN
CardNumber Bigint NN (AK1)
CardType Varchar(50) NN
CardExpiry Date NN
Identifying, Mandatory and PFK = Developer_Id

PHP: (Front End)

This the home page

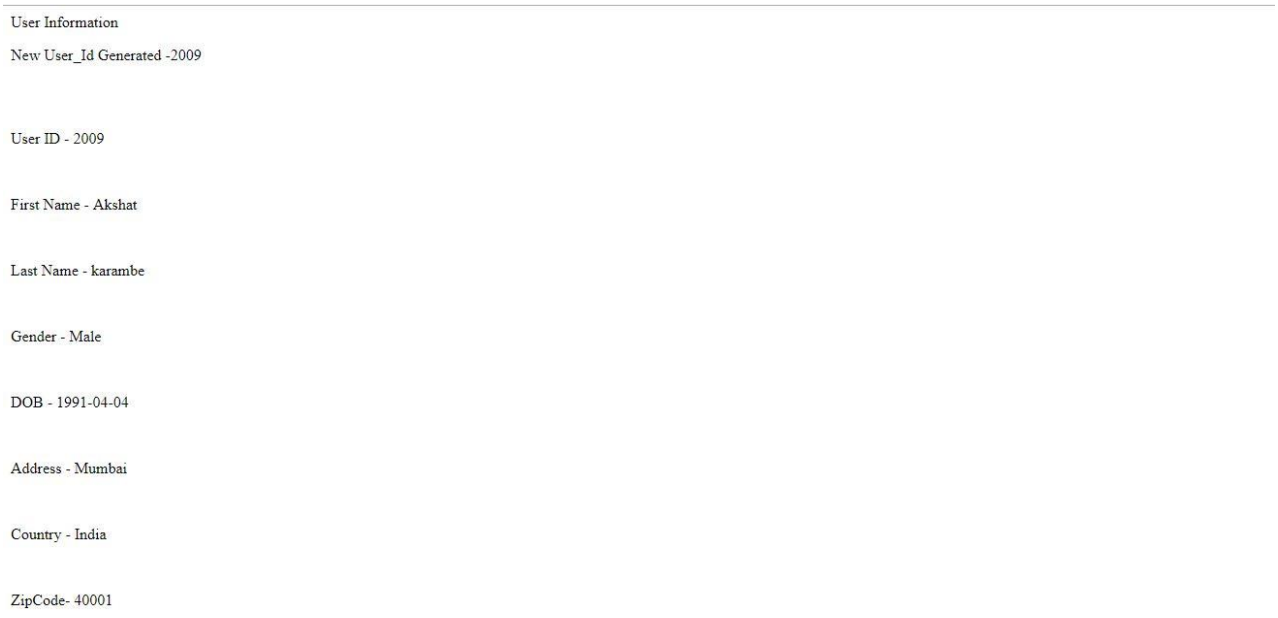


After clicking the User Database button

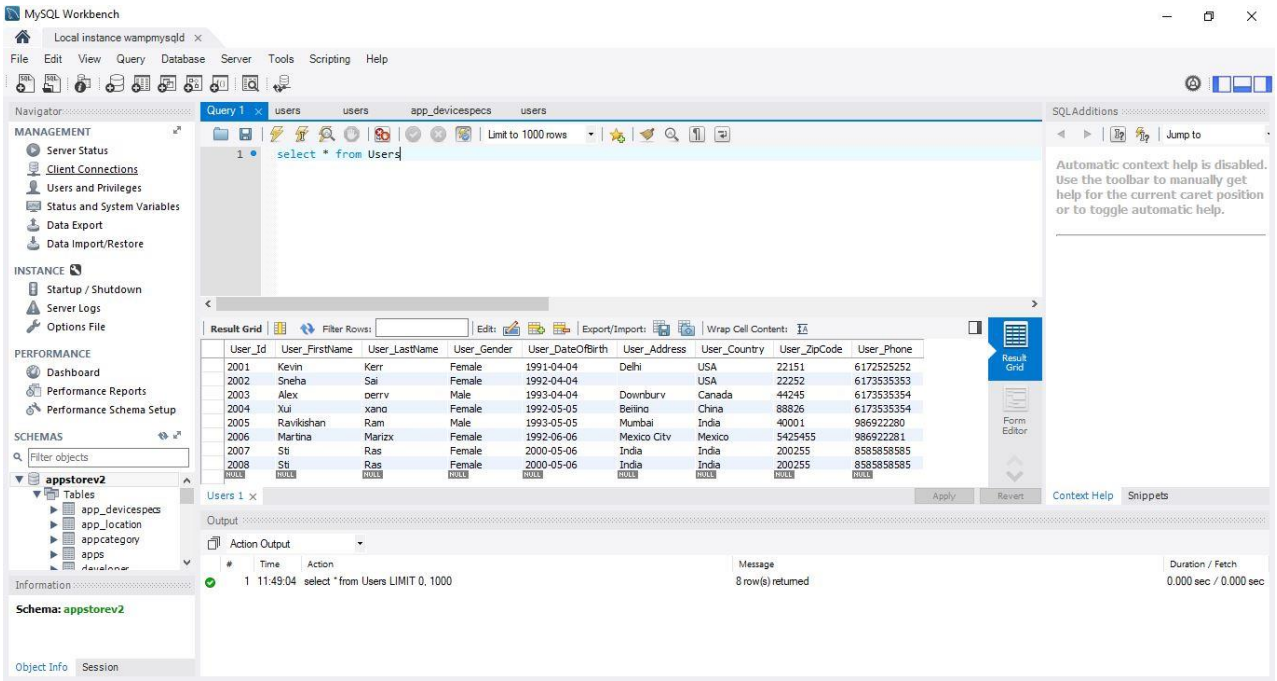


After clicking the Add New User button

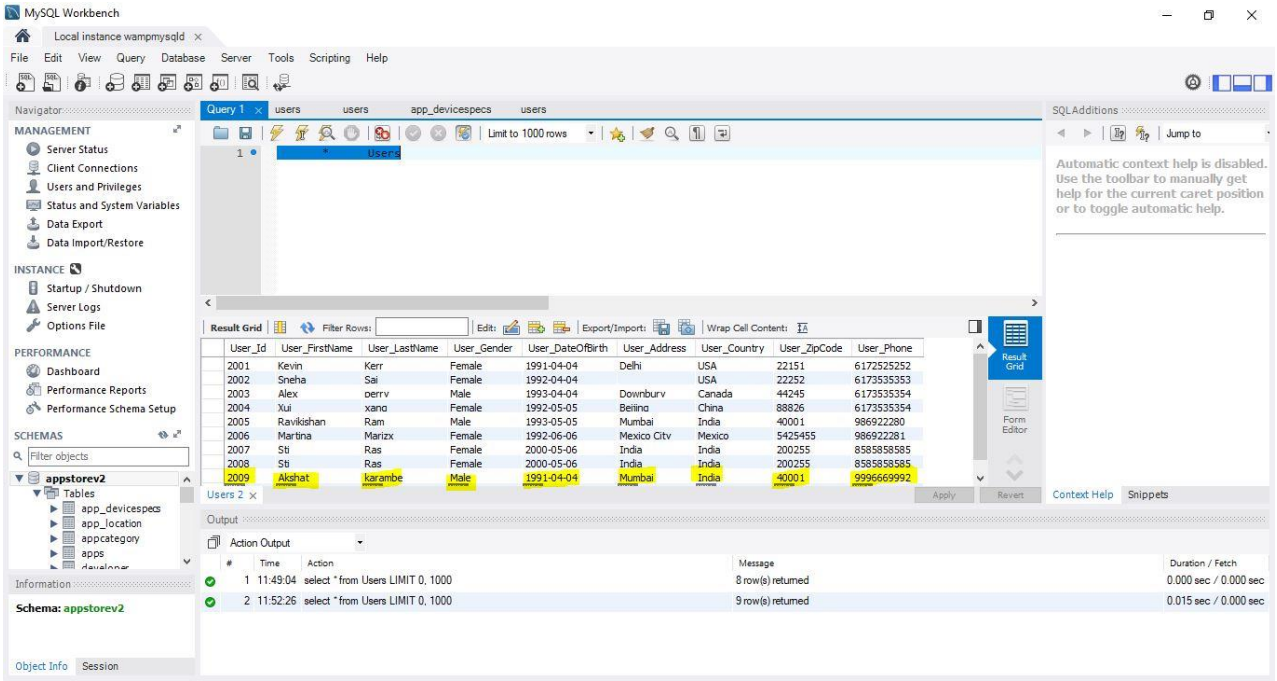
Front end page showing the user is added and the User Id is Auto Incremented to 2009



Last entry before adding the user in the back end



Back end page showing the user is added and the User Id is Auto Incremented to 2009



After Clicking on the Search Button

Search Results

User ID	First Name	Last Name	Gender	DOB	Address	Country	Zipcode	Contact	Action
2001	Kevin	Kerr	Female	1991-04-04	Delhi	USA	22151	6172525252	Edit Delete
2002	Sneha	Sai	Female	1992-04-04		USA	22252	6173535353	Edit Delete
2003	Alex	perry	Male	1993-04-04	Downbury	Canada	44245	6173535354	Edit Delete
2004	Xui	xang	Female	1992-05-05	Beijing	China	88826	6173535354	Edit Delete
2005	Ravikishan	Ram	Male	1993-05-05	Mumbai	India	40001	986922280	Edit Delete
2006	Martina	Marizx	Female	1992-06-06	Mexico City	Mexico	5425455	986922281	Edit Delete
2007	Sti	Ras	Female	2000-05-06	India	India	200255	8585858585	Edit Delete
2008	Sti	Ras	Female	2000-05-06	India	India	200255	8585858585	Edit Delete
2009	Akshat	karambe	Male	1991-04-04	Mumbai	India	40001	9996669992	Edit Delete

After clicking on the edit button, we can see that the apart from the User Id everything can be edited

Fill in New User's Details

User ID:	2009
First Name	Akshat
Last Name	karambe
Gender	Male
DOB	1991-04-04
Address	Mumbai
Country	India
Zipcode	40001
Contact	9996669992

Update Student Reset

Front end page showing that the users address and ZipCode has been updated

User ID - 2009

First Name - Akshat

Last Name - karambe

Gender - Male

DOB - 1991-04-04

Address - NewYork

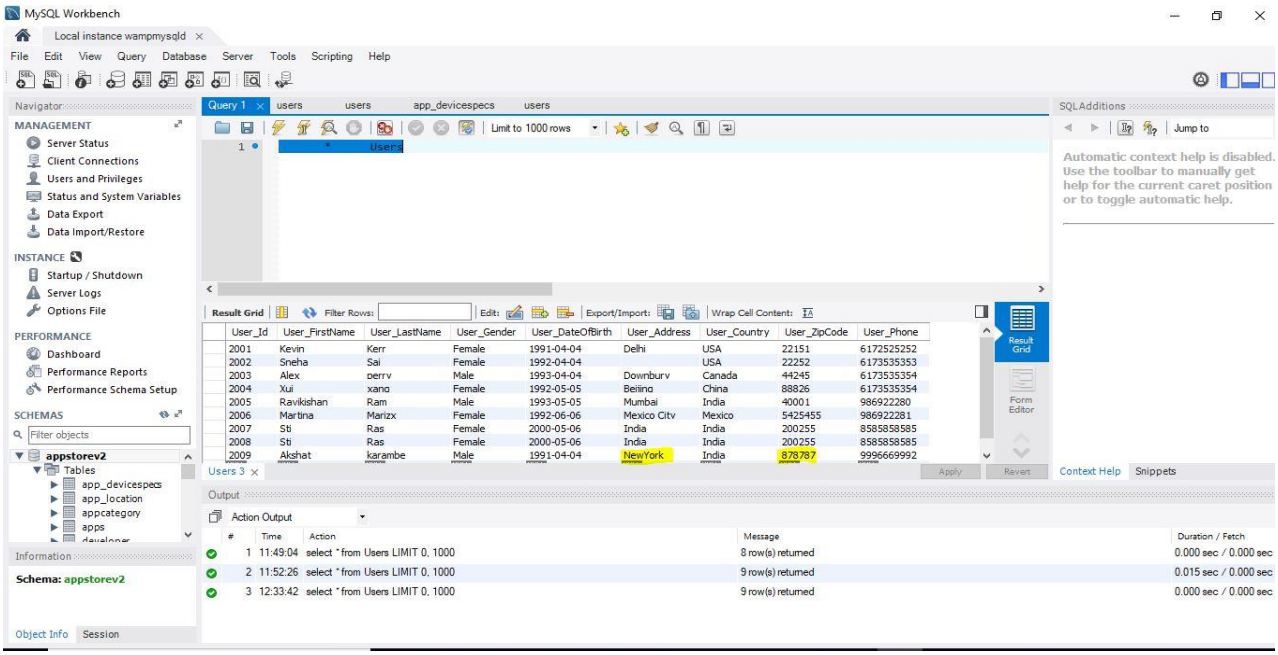
Country - India

Zipcode- 878787

Contact - 9996669992

Home

The Updated Address and Zipcode reflecting at the backend



After deleting a user from the front end.

Deleted!!!

Home

After deleting the user from the front end the user with the user id 2009 got deleted