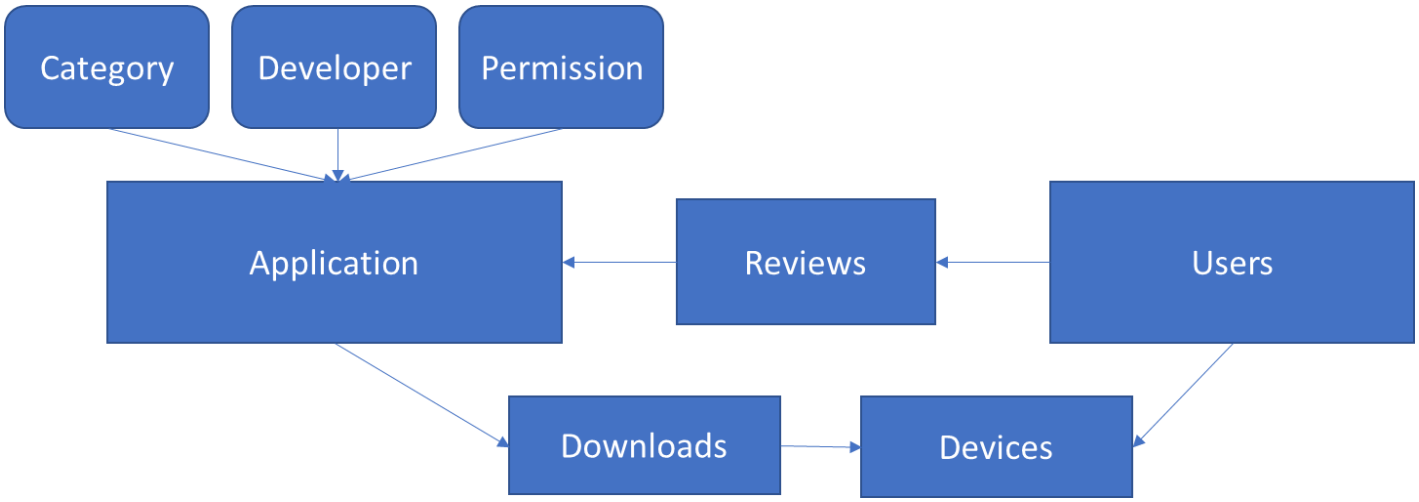
Apps Store Database Design Project

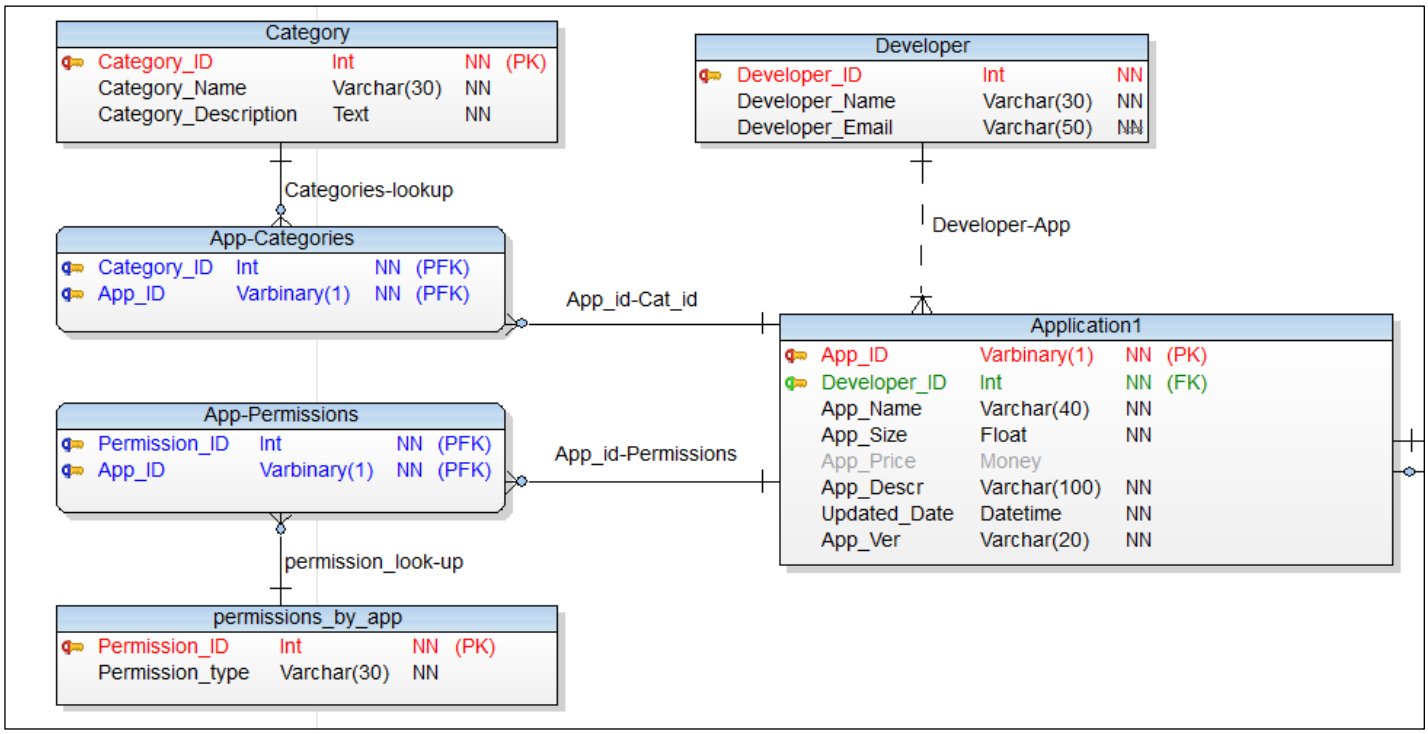
By
Charchil Kasliwal
Spring - 2018

Table of Content

1. High-Level Diagram



2. Cluster Application



Must have Business Rule – from Professor

In App Advertisements – Not an entity but an indicator stating application shows adds is appropriate	In-app ads as an attribute in Application
Language support is optional scope (showing screens and descriptions in another language)	
Applications will be listed in multiple categories	Already exist
Applications should be able to show similar or companion applications	Already exist
Geography and language will be captured for applications	Already exist
An application should list the devices support	Already exist
An application should list any phone operating system requirements	Added an attribute in Application entity
Applications should be flagged if they are in violation of our guidelines	Added an attribute in Application entity
Users can have multiple devices	Already exist
Applications that are free or paid will be tracked	Already exist
Users can download applications to multiple devices	Already exist
Versions of applications must be maintained	Already exist
Subscription information is associated one of two ways in the Appstore or in the application	Already exist- choosen in-app
Suggestions from friends and peers is optional scope	Can be done, if user allows contacts access
Application is a cluster of entities	Already exist
System is a cluster of entities	Already exist - Inscope
Download or Purchase is a cluster of entities	Already exist - Inscope
Items such as books and movies are out of scope	Already exist - out of scope
Attributes will have ratings and reviews (by version)	Already exist - app date, download date & review date recorded
User is a cluster that will also have an account associated with it	Already exist
All applications will be purchased even if there is no charge associated with them	Already exist
Application statistic will be generated from querying the tables in the system	Already exist
System will be a cluster containing information to support the running and maintenance of the app store	
Attributes of the application that need to be captured include Phone RAM (minimum) and OS (minimum)	Added attribute in Application entity
need to be store and will likely need to be archived	Already exist - Inscope
Some applications will have in app purchases that are charged and paid for by the store	
Network of friends is maintained in the app store	Can be done, if user allows contacts access
User passwords should be encrypted in some manner	Done
Support information should be captured for the application this could include contact info and websites	Already exist - app attribute
Discount information is not needed	
A history entity to track user's tendencies for search is optional scope	
All accounts must have a payment method on file	Already exist - Can be null
Credit cards are the only acceptable form of payment	Already exist

- **2.1 Application:**

This table would be used to store properties of each app. Following are the attributes and their purpose respectively.

Entity: Application1

Attributes	Data type	NULL/NOT NULL	Description/Purpose	Sample Data
App_ID	INT	NOT NULL	Primary key, unique	1199124545
Developer_id	INT	NOT NULL	Foreign key	2310
App_name	Varchar(40)	NOT NULL	Name of the application	Offroad Drive Dessert
App_description	Varchar(max)	NOT NULL	Description of various features of the app and new added features in an update.	Offroad is a game that test your ability to drive an All-terrain-Vehicle…
App_size	INT	NOT NULL	Float (MBs)	5.67 MB
Last_update_date	Date	NOT NULL	This would be used to send notification to the users according to the date of purchase/download.	04-22-18 09:00:00
App_price	Money	NOT NULL	Price of an app	$ 45
Supported_device	Varchar	NOT NULL	Each app have different applications depends on the device it's been downloaded to.	1

Sample Rows:

App_ID	Developer_id	App_name	App_description	App_size	Last_update_date	Ver	App_price	Device_supported
10103345454	23232	Snapchat	New way of texting..	69.40	04-22-17 09:00:00	1.08	0.00	1
83730383539	34343	CamScanner (license)	The worlds number 1 mobile	34.33	03-22-14 09:04:00	1.01	1.99	1
84938303030	84303	Coinbase	Buy crypto currency	15.64	03-22-14 22:04:00	1.09	0.00	3

Business Rules:

	In Scope	Out of scope
App_Id	-9 digit, unique integer app_id -Every update in an app gets a new app-Id. -Previous version gets archived in portioned table for the application -App_Id would be different for different device type	Choosing an App_Id would be out of scope for developer. App_Id would be allocated automatically.
App_size	Would have to be in MBs	
App_price	-If free then please enter 0.00 -Default would be in United States dollar (will be converted according to country chosen)	App store won't allow monthly subscription. Can be done in app purchase.
Update_date	mm-dd-yy hh:mm:ss	No other formats accepted
Version	Incrementing values starts from 1.00 (Ex-between - (1.01 -9.99)	No other format accepted
Device_supported	Mobile, watch, tablet, laptop, web Each application will have to specify device_types it supports.	

SQLQuery4.sql - DE...R86RFT\charc (53))* | SQLQuery5.sql - DE...R86RFT\charc (60))*

```
SELECT * FROM [dbo].[payment_details]
SELECT * FROM [dbo].[Application-devices_type]
SELECT * FROM [dbo].[Application1]
SELECT * FROM [dbo].[Device]
SELECT * FROM [dbo].[Downloads]
SELECT * FROM [dbo].[App-Categories]
SELECT * FROM [dbo].[Reviews]
```

110 %

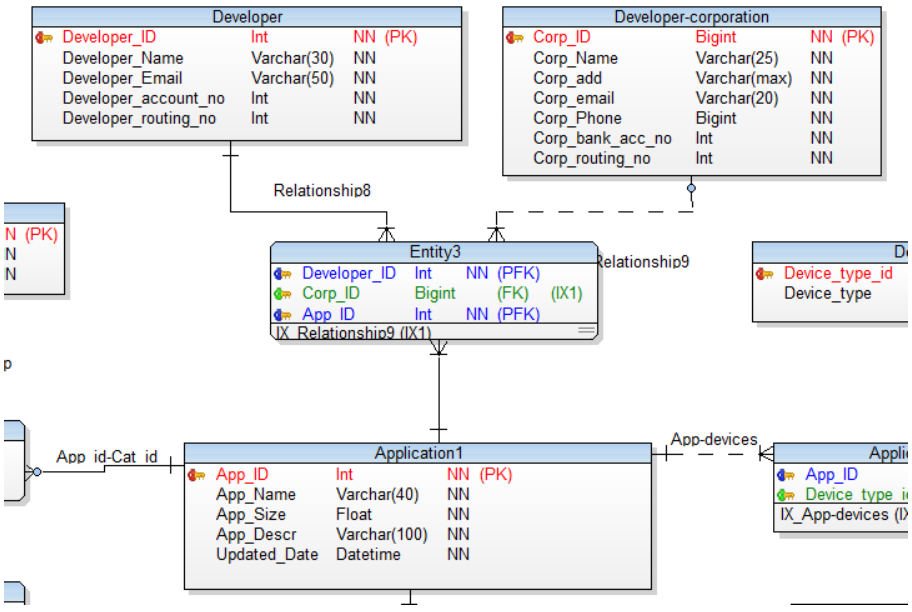
Results | Messages

	App_ID	Developer_ID	App_Name	App_Size	App_Price	App_Descr	Updated_Date	App_Ver
1	10101010	64849	Snapchat	68.3	0.00	New way of sexting, texting and social media	2018-02-03 09:00:03.000	1.01
2	20202020	64849	Gmail	29.3	0.00	Share your pictures	2016-04-18 08:00:00.000	1.13
3	30303030	65832	Facebook	76.2	3.50	Give charity to mark zukerburg	2017-02-17 09:09:00.000	1.01
4	40404040	65832	Facebook	73.2	2.50	share pictures with your loved once	2018-02-23 09:00:00.000	1.04
5	50505050	64849	All Terrian Game	64.73	4.09	play a nice game	2018-02-02 00:00:00.000	1.03
6	60606060	64849	All Terrian Game	62.02	0.00	Free trial version, grab it!!	2017-02-02 00:00:00.000	1.01
7	70707070	65832	Instagram	63.21	0.00	Share-edit high quality photos	2017-03-01 00:00:00.000	1.01

Query executed successfully. | DESKTOP-OR86RFT\SQLDEV16 (1... | DESKTOP-OR86RFT\charc ... | AppFinal | 00:00:00 | 7 rows

- **2.2 Developer**

  It contains all the information of developer. Normalized Application Entity to form a Developer table, to reduce data repetition in application entity.



Entity: Developer

Attributes	Data type	NULL/NOT NULL	Description/Purpose	Sample Data
Developer_id	INT	NOT NULL	Foreign key	23182
Developername	Varchar	NOT NULL	Name of the developer/ company (attribute)	Logic Miracle
Developer_email	varchar	NOT NULL	Users and app store can use to contact the developers.	logicmiracle@gmail.com

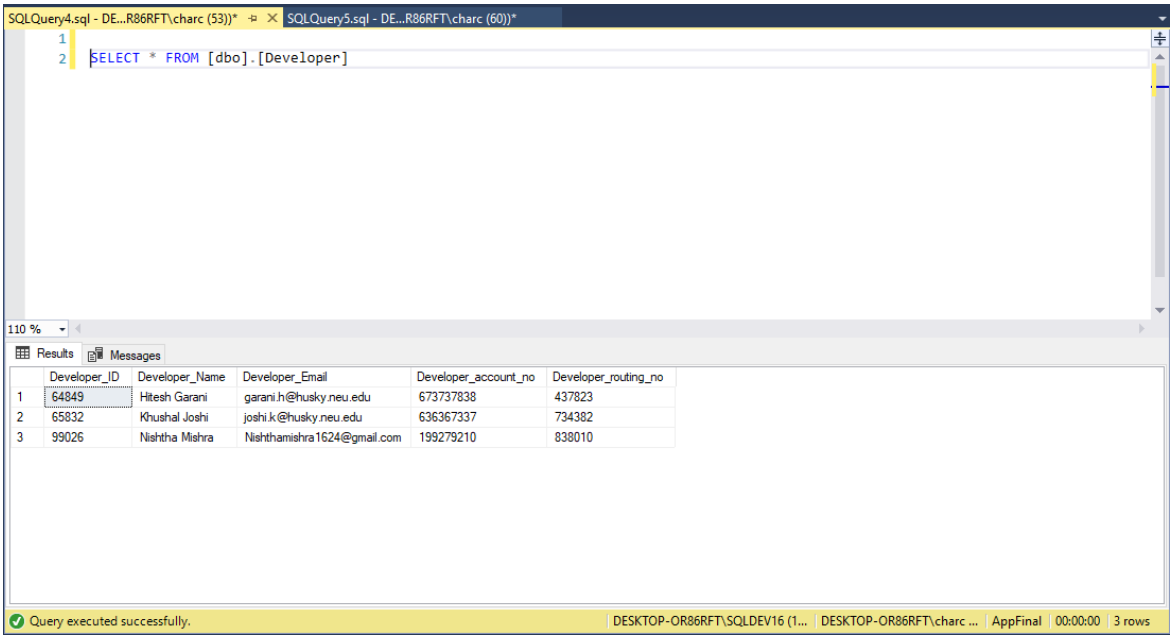
Developer_account_no+Routing_no	INT	NOT NULL	Would be used to pay the developer	01138900+839020939
Attributes	**Data type**	**NULL/NOT NULL**	**Description/Purpose**	**Sample Data**
Corp name	Varchar	NOT NULL	Name of the developer/ company (attribute)	Logic INC
Corp_email	varchar	NOT NULL	Users and app store can use to contact the developers.	logic@gmail.com
Corp_account_no+Routing_no	INT	NOT NULL	Would be used to pay the developer	01138900+839020939
Corp_address	Varcharc	NOT NULL	Address of Corporation	56 Charles gate E, Boston

Example Data:

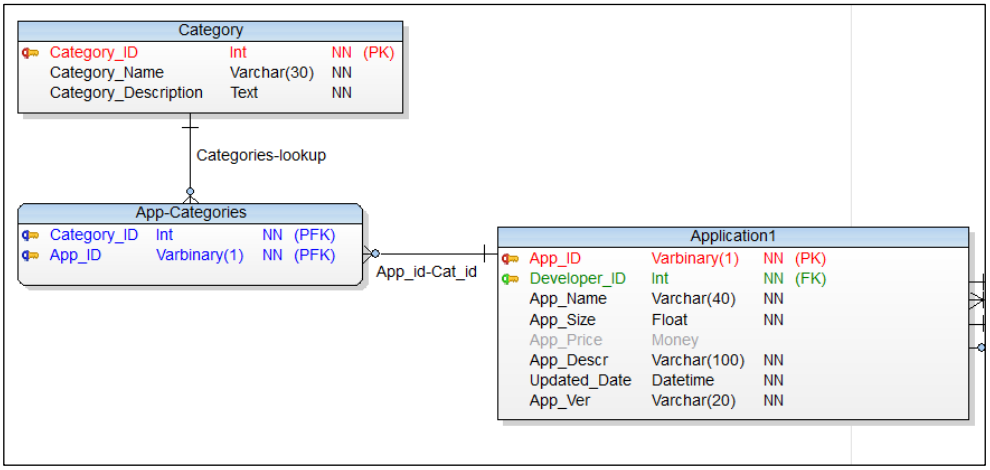
Developer_id	Developername	Developer_email	Developer_account_no+Routing_no
99026	Nishtha Mishra	Nishthamishra1624@gmail.com	199279210+838010
65832	Khushal Joshi	Joshi.k@husky.neu.edu	637302338+739200
64849	Hitesh Garani	Garani.h@husky.neu.edu	728202820+830302

Business Rules:

	In Scope	Out of scope
Developer_ID	Unique 5 digit incrementing integers, allocation by app store.	No scope of developer choosing their own developer_Id
Developer_account details	Mandatory, for paying the developers once number of downloads reaches to a certain level	
Developer_Email	A public email to report error report and bugs	



• 2.3 App- Category cluster



• 2.3-1 App-Categories

Tables would use to describe each application under what category. Followings can be the attribute.

Entity: App

Attributes	Data type	NULL/NOT NULL	Purpose	Sample Data
App_ID	INT	NOT NULL	Primary Foreign Key from Application table	10101993
Category_ID	INT	NOT NULL	Primary Foreign Key from Categories	003

Example data:

App_id	Category_ID
10103345454	004
10103345454	001
10103345454	001
83730383539	003
83730383539	002
84938303030	005

Business Rule: In-scope:

- One app can fall in many category, therefore used as a Foreign key.

• 2.3-2 Categories_type

Table would be used to describe each Category_ID with category name so that we can use category ID in the App_in_categories table as Primary foreign key.

Attributes	Data type	NULL/NOT NULL	Purpose	Sample data
Category_ID	Varchar	NOT NULL	Primary Foreign Key	003
Category_Name	Varchar	NOT NULL	Saves name of each category	Gaming

Example data:

Category_ID	Category_Name
001	Adult
002	Student
003	Office
004	Finance
005	Business
006	Travel

Business Rule: In-scope:

- Category_Id: incrementing primary key, Developer needs to choose all the category in which the apps fall.

SQLQuery4.sql - DE...R86RFT\charc (53))* | SQLQuery5.sql - DE...R86RFT\charc (60))*

```
SELECT * FROM [dbo].[Category]
SELECT * FROM [dbo].[App-Categories]
```

110 %

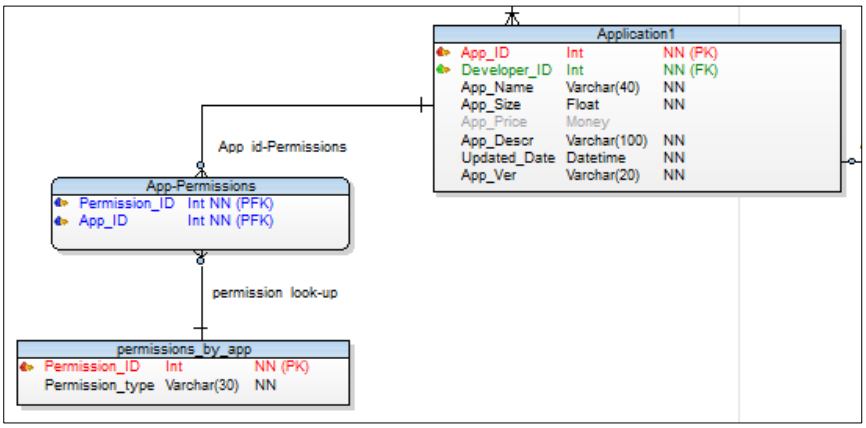
Results | Messages

	Category_ID	Category_Name
1	101	Games
2	102	Social Media
3	103	Utilities
4	104	photography
5	105	Education
6	106	Cooking
7	107	Adult
8	108	Productivity

	Category_ID	App_ID
1	103	10101010
2	104	10101010
3	103	20202020
4	105	20202020
5	107	20202020
6	104	30303030
7	106	30303030
8	107	30303030
9	101	50505050
10	102	50505050
11	101	60606060

Query executed successfully. | DESKTOP-OR86RFT\SQLDEV16 (1... | DESKTOP-OR86RFT\charc ... | AppFinal | 00:00:00 | 21 rows

- ## 2.4 App Permission



- ## 2.4-1 Permissions_by_apps

  Entity contains information of each app requires what all kinds of permission from user. Like location, access to gallery, access to inbox etc

Entity: App-permissions

Attributes	Data type	NULL/NOT NULL	Purpose	Sample Data
permission_ID	INT	NOT NULL	Primary Foreign Key from permissions	98
App_ID	INT	NOT NULL	Primary Foreign Key from Application	10101993

Example Data:

App_id	Permission_ID
10103345454	99
10103345454	98
10103345454	97
83730383539	95
83730383539	99
84938303030	97

Business Rules: In-scope

- Every app can have zero, one or many access permissions from the user device.

- **2.4-2 Permission_types**

  Reference Table for Permission table

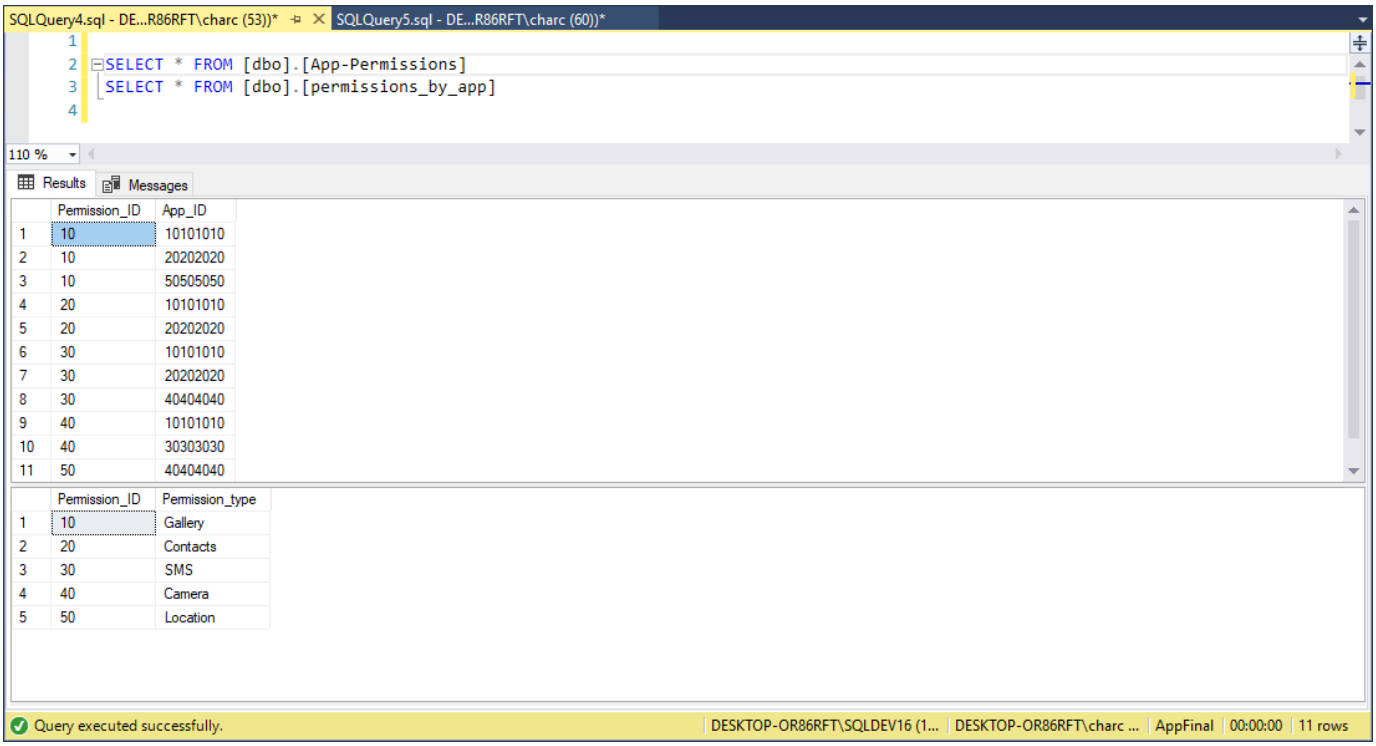
Attributes	Data type	NULL/NOT NULL	Purpose	Sample Data
permission_ID	INT	NOT NULL	Primary Foreign Key	98
Permission_type	Varchar	NOT NULL	Saves the kind of permission detail	Contacts

Example Data:

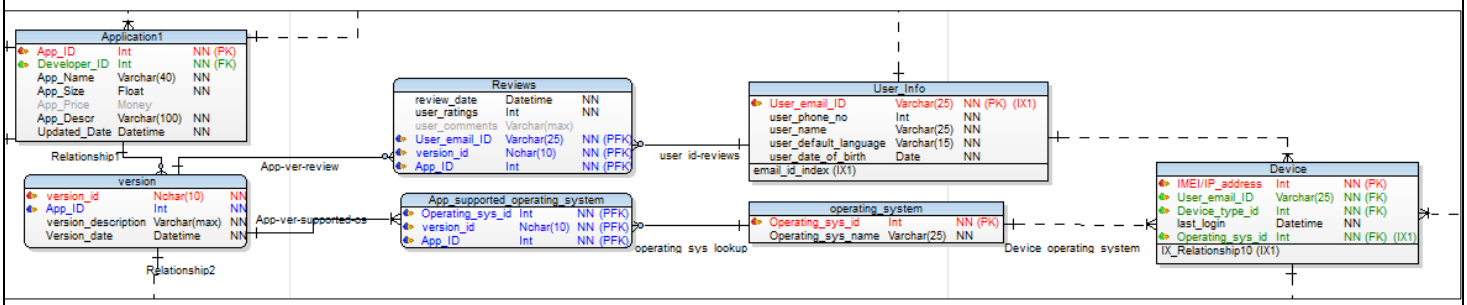
Permission_ID	Category_Name
99	Gallery
98	Contacts
97	Location
96	Camera
95	SMS

Business Rules: In-scope

- Permission_ID: 2-digit unique Integer key of every type of permission. Its look up table for permissions.



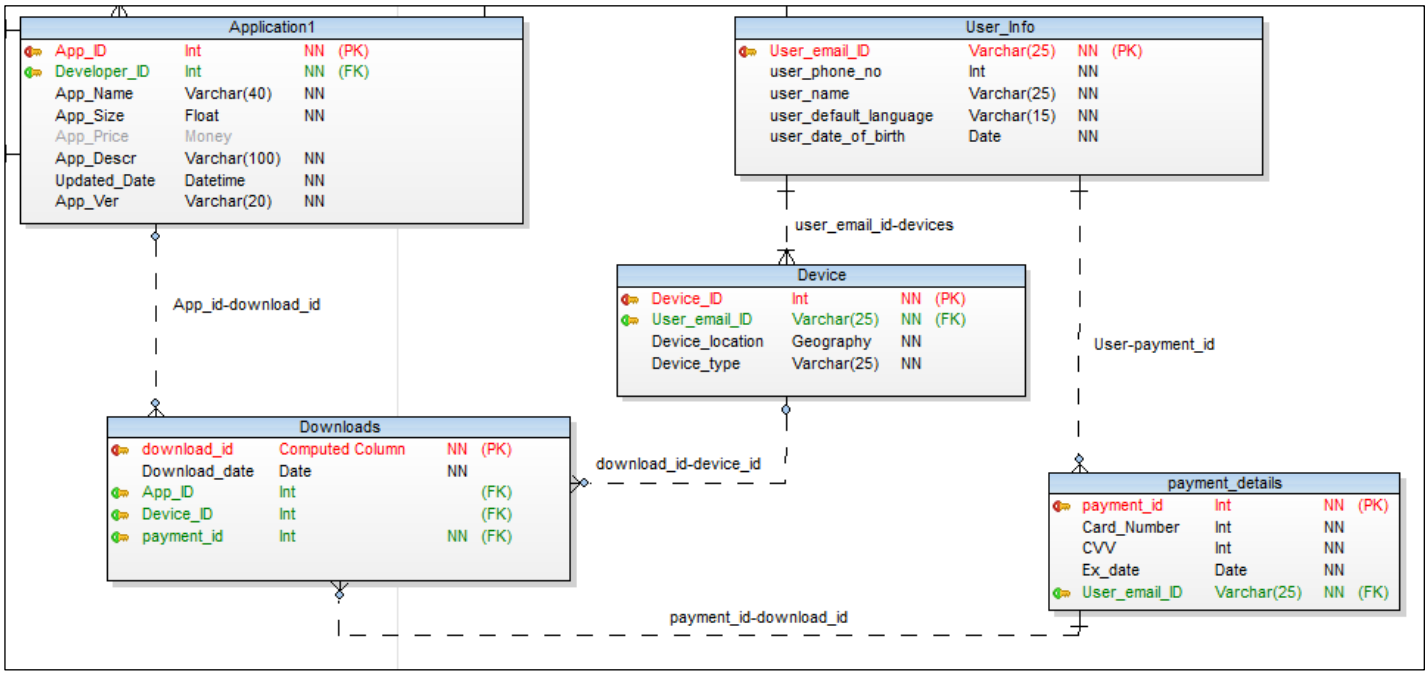
• 2.5 App Supported operating system



Updated Features:

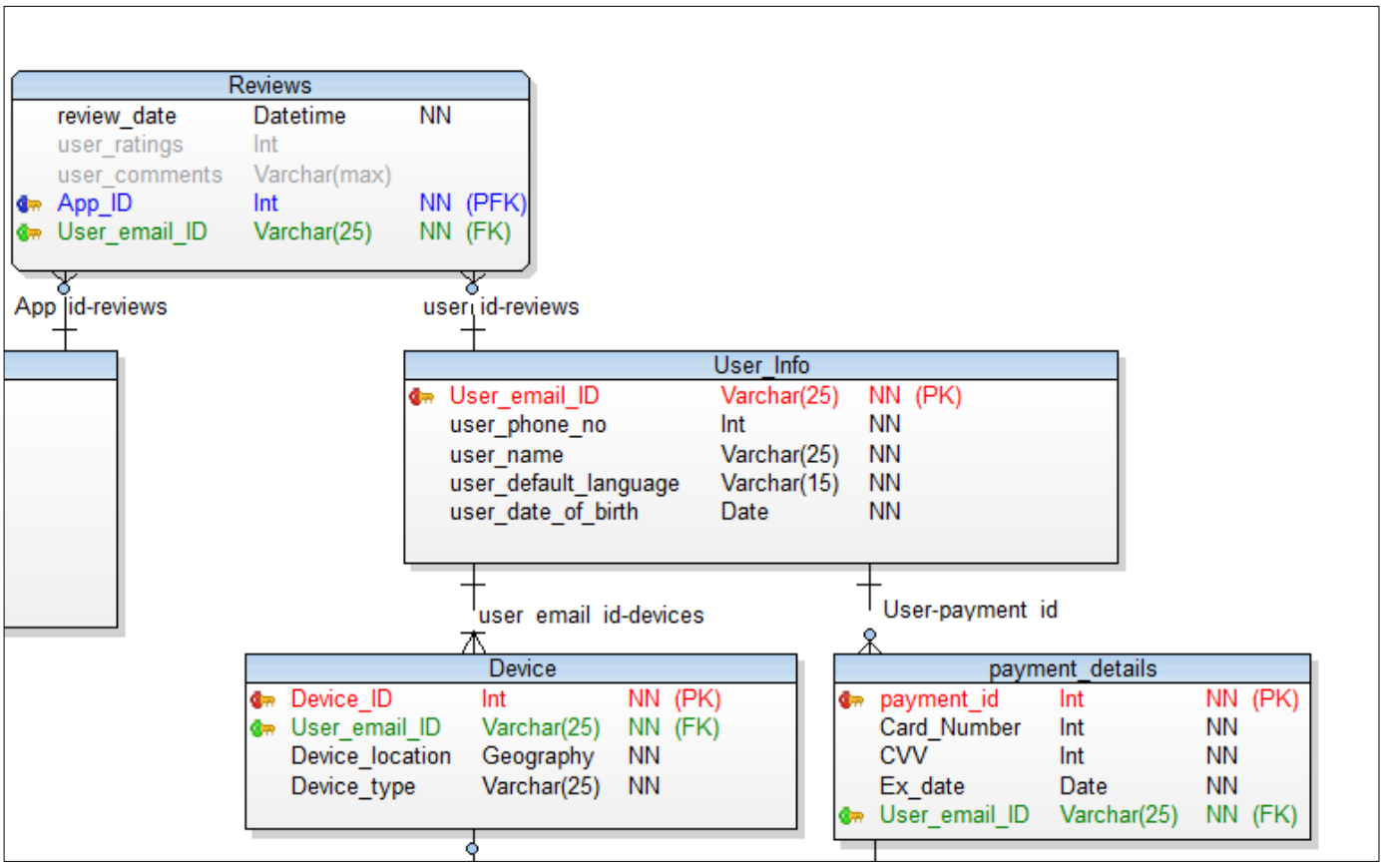
- App can only be downloaded if device OS matches the application-version supported OS.
- New model enables us to track reviews based on Application versions.

- **3. User Cluster**



- **3.1 User Entity**

This table would be used to store the users' information. Following are the attributes for this table and the purpose is stated respectively.



Entity: User_Info

Attributes	Data type	NULL/NOT NULL	Purpose	Sample data
User_email_ID	Varchar(25)	NOT NULL	Primary key, unique	Kasliwal.c@husky.neu.edu
User_phone_no	INT	NOT NULL	To use to send verification codes and OTPs	8572309984
User_name	Varchar	NOT NULL	Used to show names on reviews and ratings	Charchil Kasliwal
User_language	varchar	NOT NULL	Use app version according to this attribute	English
User_date_of_birth	varchar	NOT NULL	To use appropriate age group for the application	10-10-1993
User_Country	varchar	NOT NULL	To put filters on services available as per the applications	China

Example Data:

User_email_Id	User_phone_no	User_name	User_language	date_of_birth	country
kasliwal.c@husky.neu.edu	8572309984	Charchil	English	10-10-1993	US
Joshi.k@husky.neu.edu	8573490408	Khushal	Hindi	23-02-2001	India
info@jashmetrology.com	7739383937	Jash	French	12-12-2007	France

Business Rules:

	In-scope	Out of scope
User_email	Email id is used as primary key, since the email-id is itself an unique attribute. -User cannot change the email id. User must create a new account if they need to change their email_id	No increment integer value, as thousands of billion users out there, computing a unique incrementing key would be slow.
User_phone	User is only allowed to enter 10 digits, country code can be referred by using reference table for the below mentioned country	No more than 10 integers would be allowed
Date_of_birth	Restricts users' access to certain apps in appropriate for certain age	
User_language	Used by application to deliver an appropriate version of their app.	Default language would be English
User_country	For user authorization, database is going to take country code for mobile verification	
Added a non-clustered index on Email_id which would be numeric		

SQLQuery4.sql - DE...R86RFT\charc (53))* | SQLQuery5.sql - DE...R86RFT\charc (60))*

```
SELECT * FROM [dbo].[User_Info]
SELECT * FROM [dbo].[payment_details]
SELECT * FROM [dbo].[Reviews]
```

110 %

Results | Messages

	User_email_ID	user_phone_no	user_name	user_default_language	user_date_of_birth
1	charchil.kasliwal@gmail	8572309984	charchil kasliwal	English	1993-10-10
2	kushal.j@husky	9200888832	Khushal	Hindi	2007-03-19
3	mishra.n@husky	7383838383	Mishra	English	1991-09-09
4	shah.ala@husky	9128921828	alay	English	1991-09-01

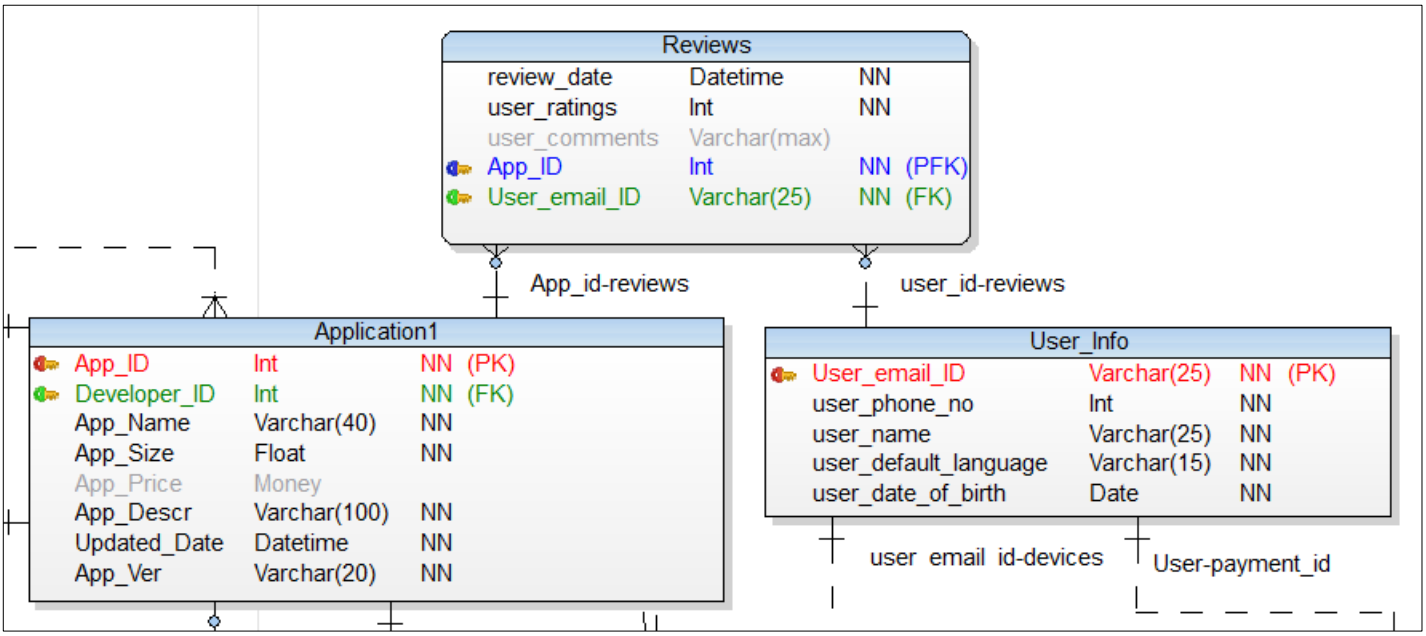
	payment_id	Card_Number	CVV	Ex_date	User_email_ID
1	1234	1212232334344545	12	2021-09-01	charchil.kasliwal@gmail
2	2345	1234234534564567	134	2022-08-01	kushal.j@husky
3	2436	1234567812345678	123	2030-02-01	charchil.kasliwal@gmail
4	9876	3333444455556666	424	2019-09-01	mishra.n@husky
5	9999	8888999900001111	232	2019-09-09	shah.ala@husky

	review_date	user_ratings	user_comments	App_ID	User_email_ID
1	2012-10-10 00:00:00.000	4	This is a nice app. Must download	10101010	charchil.kasliwal@gmail
2	2013-04-13 00:00:00.000	1	Crap!	20202020	kushal.j@husky
3	2018-01-12 00:00:00.000	2	NULL	30303030	charchil.kasliwal@gmail
4	2018-02-01 00:00:00.000	5	I use this app daily.. bla bla	10101010	kushal.j@husky
5	2018-01-11 00:00:00.000	4	My phone hangs	30303030	mishra.n@husky

Query executed successfully. | DESKTOP-OR86RFT\SQLDEV16 (1... | DESKTOP-OR86RFT\charc ... | AppFinal | 00:00:00 | 14 rows

- **3.2 Reviews:**

This table would be used to store reviews from user and the ratings



Entity: Reviews

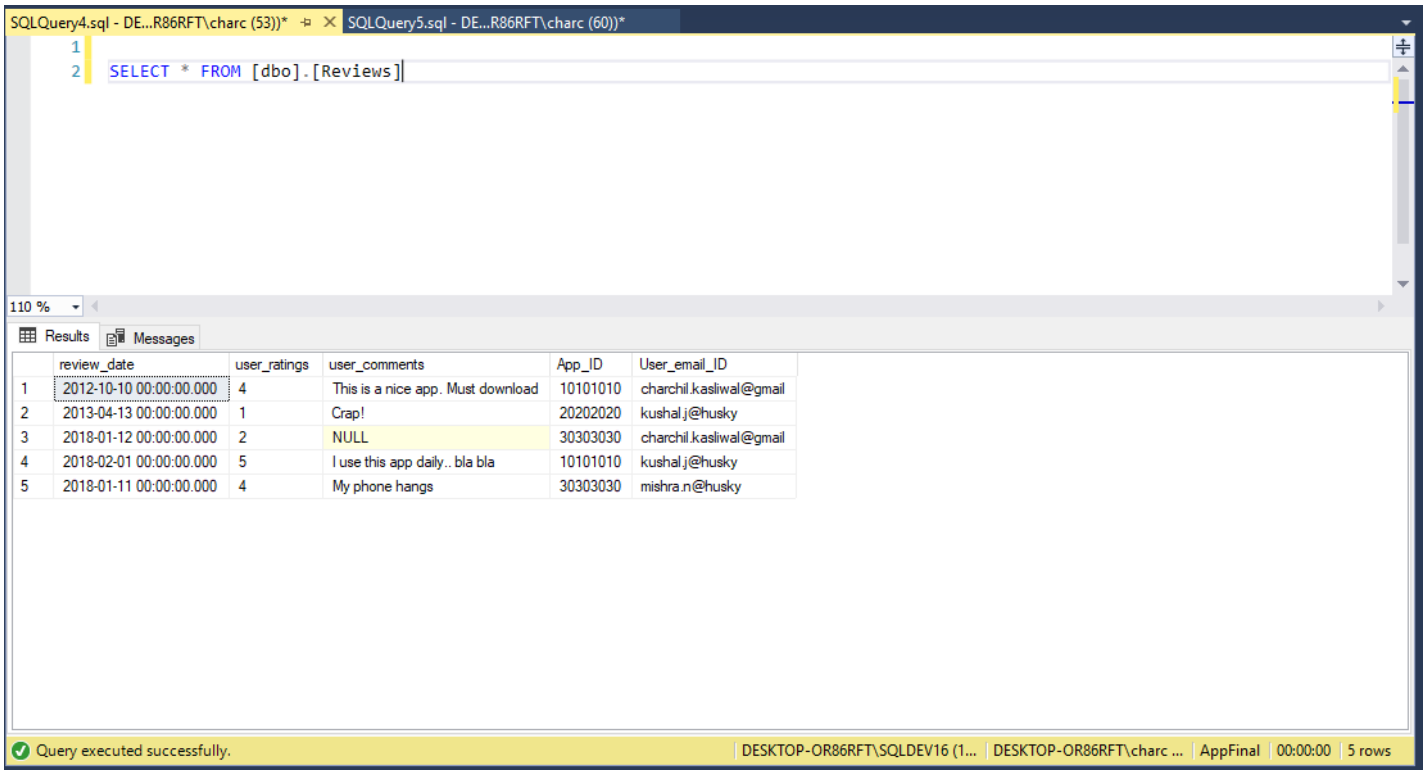
Attributes	Data type	NULL/NOT NULL	Purpose	Sample Data
App_ID	INT	NOT NULL	Primary Foreign Key	10101993
User_email_ID	Varchar	NOT Null	Foreign key with user email Id	Kasliwal.c@husky.neu.edu
User_ratings	INT	NOT NULL	Ratings pertaining to each app by each user	4
User_comments	INT	NULL	Saves comments on each review, can be null	This app is amazing, must download

Example Data:

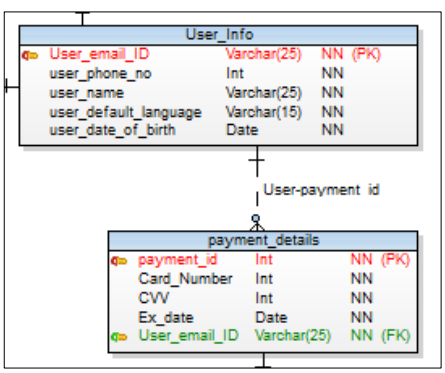
App_ID	User_email_Id	User_ratings	User_comments
738393837290	Kasliwal.c@husky.neu	4	This app is freaking awesome
513036561548	Joshi.k	2	Fuck this app.
412154651252	Joey_triibiani	5	NULL

Business rules:

	In Scope	Out of scope
User_ratings	Ratings Should be out of 5	Cannot accept values in float
User_comment	• Each rating have a comment associated to describe more about the app and more suggestions for downloaders and developers. • Comments can be Null	• Cannot exceed more than 640 character. • Images and videos in comments cannot be saved in the databse.
App_Id	-Used as primary foreign key, also linked as identifying relation with application. If application leaves the database, all reviews associated would be removed as well. - If a user leaves the database, their comments and review will still be present.	



- **3.3 User-Payment details**



Entity: payment_info

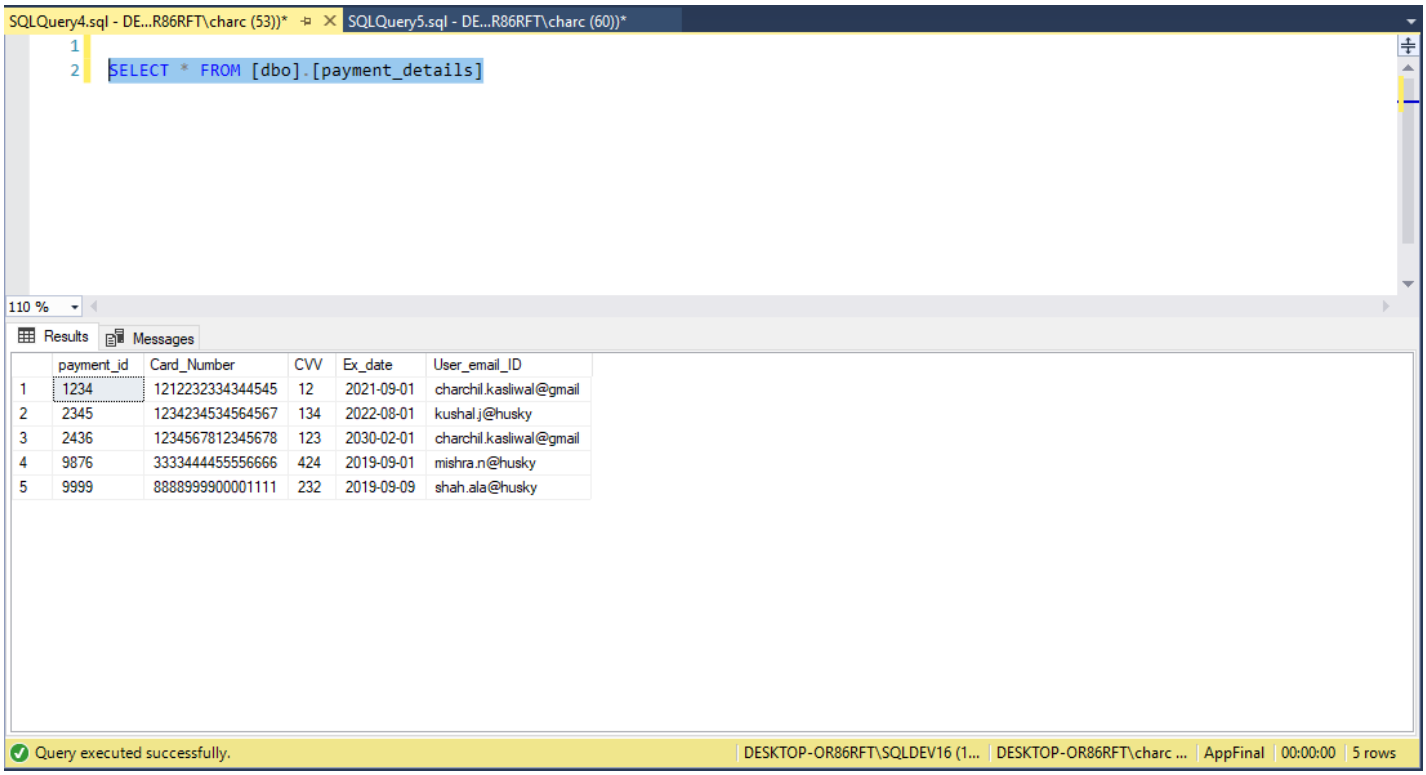
Attributes	Data type	NULL/NOT NULL	Purpose	Sample data
Payment_id	INT	NOT NULL	Primary key, unique	CV8849
Card_number	INT	NOT NULL	Used for Payments, and refund.	1234 8484 3453 4555
CVV	INT	NOT NULL	Security code! For bank	223
Expiry Date	Date	NOT NULL	For bank (mm/yyyy)	08/2022
User_Email_id	varchar	NOT NULL	Foreign key	Kasliwal.c@husky.neu.edu

Sample Data:

Payment_id	User_email_Id	Card Number	CVV	Ex_date
CV1229	kasliwal.c@husky.neu.edu	1516 5156 6718 6789	678	04/2033
DB6338	Joshi.k@husky.neu.edu	1516 5156 6718 6679	748	08/2019
VA7388	info@jashmetrology.com	1516 5156 6718 6347	566	09/2018

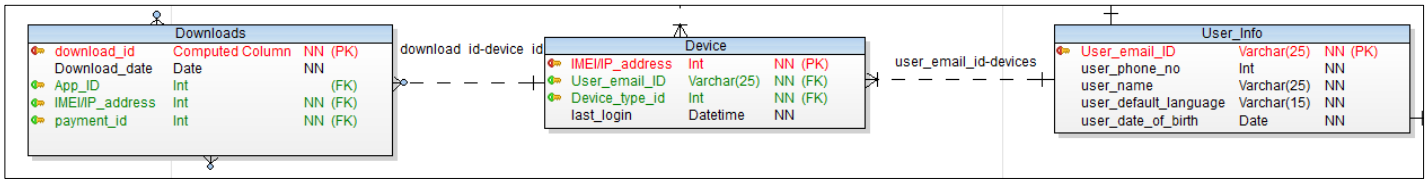
Business Rules:

	In Scope	Out of scope
Card_Number	-Must be 16 digits. -Can only use one card for one payment,	-Internet banking transactions -cannot use two or more cards to pay for an app.
User_email_id	Can have many card associated with one user_id.	
Expiry date	-It must be a date in Future and have format of mm/yyyy	No other date formats would be accepted



- **3.4 Users_devices**

  This table will have information pertaining to the user and applications. Device location and device geography will be used by applications to provide services accordingly.



Entity: Devices

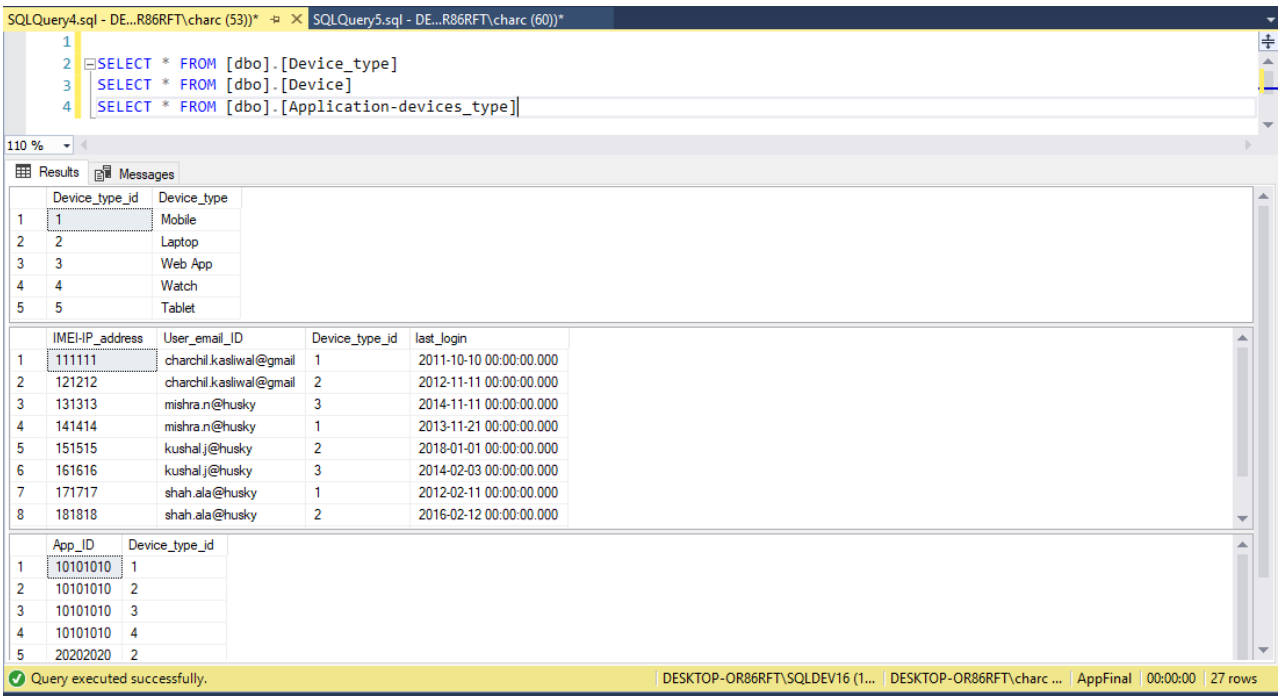
Attributes	Data type	NULL/NOT NULL	Purpose	Sample Data
IEMI/IP_Address	Varchar	NOT NULL	Primary key, IMEI Unique itself	891921621100
User_email_ID	Varchar	NOT NULL	Primary Foreign key with users	Kasliwal.c@husky.neu.edu
Last_Login	Datetime	NOT NULL	Would be used to track if there is any change in the users on the same device.	04-22-17 09:00:00
Device_type	varchar	NOT NULL	Device Id would be used to give device the app version accordingly	1

Example Data:

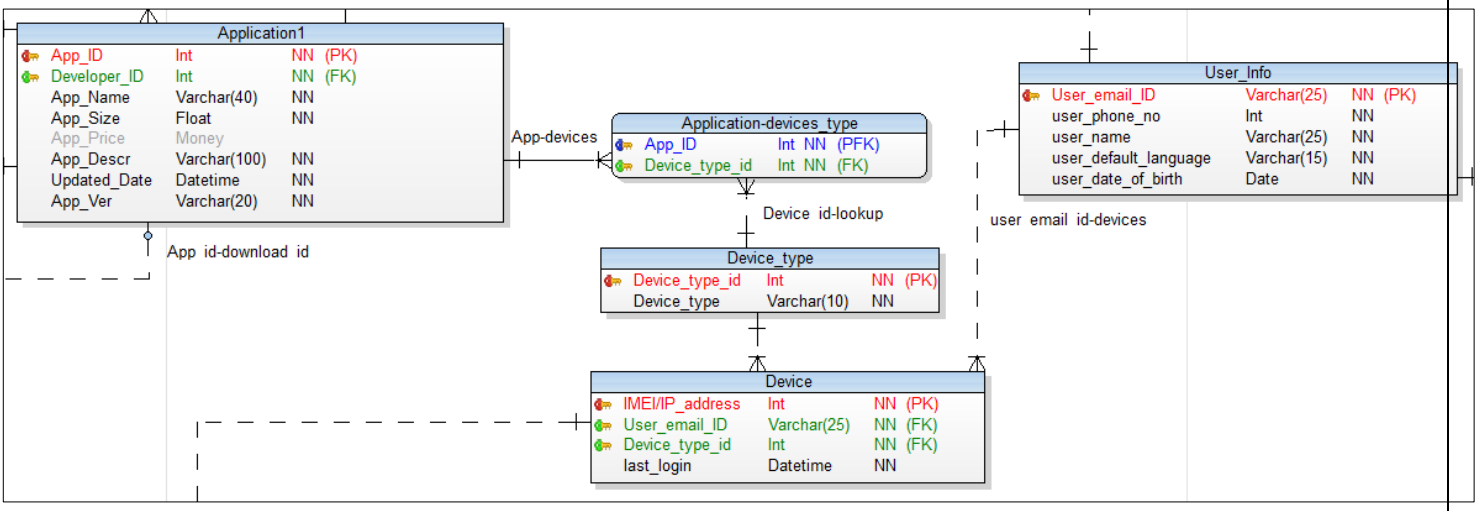
IMEI/IP_Address	User_email_id	Device_type
748493037392	Kasliwal.c@husky.neu.edu	1
739930238390	Kasliwal.c@husky.neu.edu	3
738302829272	Joshi.k@husky.neu.edu	1
546378932932	nishthamishra1624@gmail.com	2
648303739373	nishthamishra1624@gmail.com	1

Business Rules:

- If application is same for all devices, then it is allowed on one or more types of devices with the same Application_id.
- If application is different in the size and different in the features, it will have to have a new app_id
- Last_login date must be a time stamp when user log in their account on the app store.



- **4 Device Cluster:**



- **4.1- Device_types**

  Stores the types of devices for the reference to other tables

  Example data:

Device_type_id	Device_type
1	Mobile
2	Tablet
3	Watch
4	Ipod
5	Laptop
6	Web-application

- **4.2 App_Device_supported**

  Stores information for each device that an app can support.

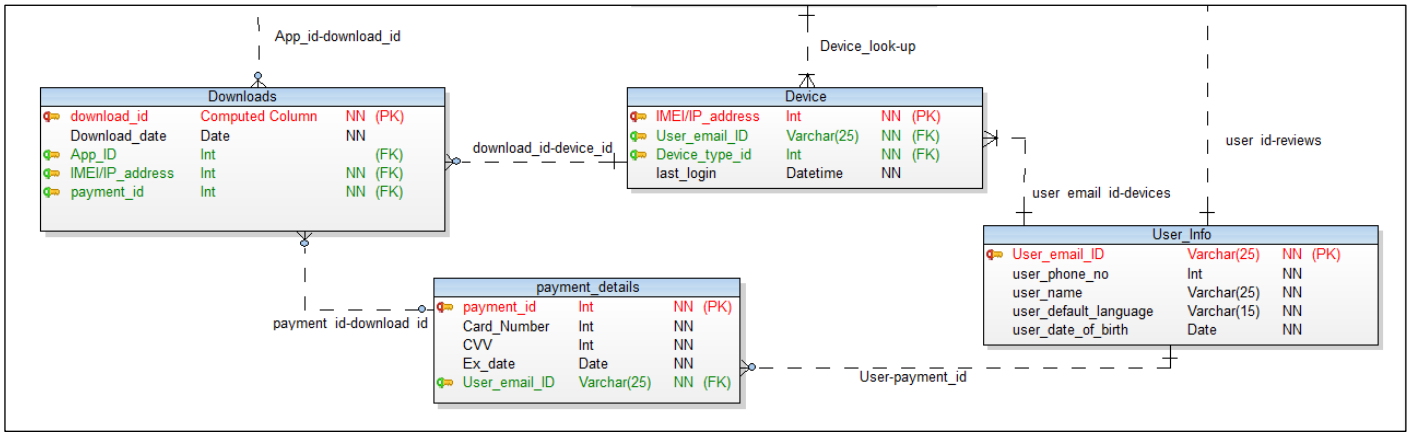
  Example data:

App_id	Device_id
10103345454	1
10103345454	2
10103345454	3
83730383539	2
83730383539	4
84938303030	2

Please Refer above image for actual data entered in SQL

- **5 Downloads**

  It contains details of every device downloading the app.



Entity: Download:

Attributes	Data type	NULL/NOT NULL	Purpose	Sample Data
Download_id	Computed Column	NOT NULL	Primary Key	623829262392@891921621100
App_id	Int	NOT NULL	Foreign Key	623829262392
IMEI/IP Address	Int	NOT NULL	Foreign Key	891921621100
Payment_id	Int	NOT NULL	Describes the terms of payement. Example: $0.00	CC090909
Download_date	Datetime	NOT NULL	Timestamp, UTC	02-02-2018 04:02:00

Example data:

Download_id	App_id	IMEI/IP Address	Payment_id	Download_date
111111111@748493037392	111111111	748493037392	CC0000	02-17-2018 04:02:00
222222222@739930238390	222222222	739930238390	DD9329	02-10-2018 04:02:00
234234234@738302829272	234234234	738302829272	MD6747	02-09-2018 04:02:00
111111111@546378932932	111111111	546378932932	CC0000	02-03-2018 04:02:00
236575093@648303739373	236575093	648303739373	YU7893	02-04-2018 04:02:00

Business Rules:

	In Scope	Out of scope
Download_id	Computed coloumn (App_id + '@' + IMEI/IP_Address	We won't be able to put a cluster index.
Payment_id	Every app purchased, would have to be paid separately.	No payment other than card payment, Could be debit card or credit. No payment through internet banking
Download_date	• After a user uninstall an app for which he already has paid would be linked through user_Id via IMEI & User_id in another table. So, whenever user returns and download with the same user-Id will not have to pay again. • If a user-id changes on a device, it will be updated in device-user table, hence will have to pay again for the app. • If a user downloads an app again, records would be updated with latest download date by replacing previous record.	

SQLQuery4.sql - DE...R86RFT\charc (53))* | SQLQuery5.sql - DE...R86RFT\charc (60))*

```
1
2  SELECT * FROM [dbo].[Downloads]
```

110 %

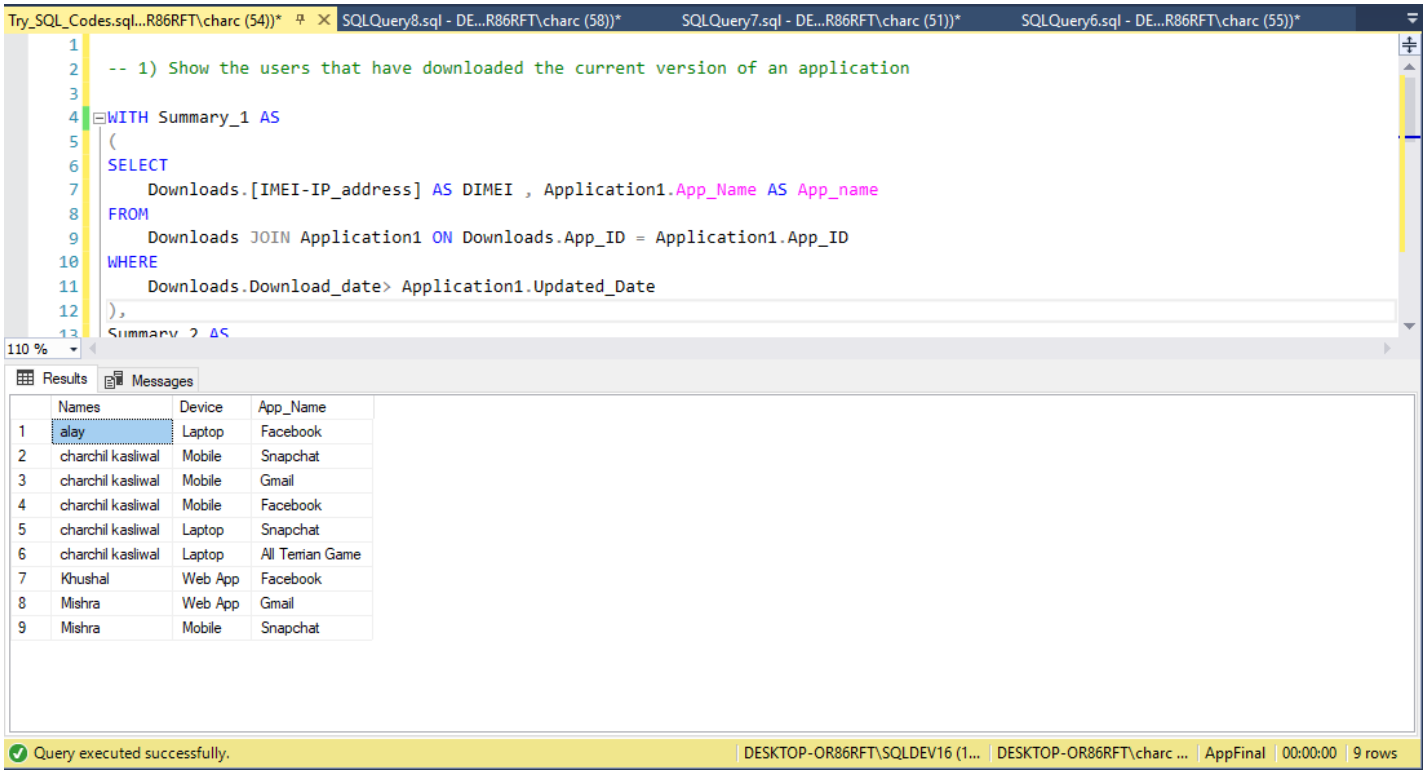
Results | Messages

	download_id	Download_date	App_ID	IMEI-IP_address	payment_id
1	10101010@111111	2018-03-01 00:00:00.000	10101010	111111	NULL
2	10101010@121212	2018-03-01 00:00:00.000	10101010	121212	NULL
3	10101010@141414	2018-02-20 00:00:00.000	10101010	141414	NULL
4	20202020@111111	2018-03-01 00:00:00.000	20202020	111111	NULL
5	20202020@131313	2018-03-28 00:00:00.000	20202020	131313	NULL
6	40404040@111111	2018-03-01 00:00:00.000	40404040	111111	1234
7	40404040@161616	2018-03-03 00:00:00.000	40404040	161616	2345
8	40404040@181818	2018-03-03 00:00:00.000	40404040	181818	9999
9	50505050@111111	2017-03-03 00:00:00.000	50505050	111111	1234
10	50505050@171717	2017-03-04 00:00:00.000	50505050	171717	9999
11	60606060@121212	2017-02-27 00:00:00.000	60606060	121212	1234

Query executed successfully. | DESKTOP-OR86RFT\SQLDEV16 (1... | DESKTOP-OR86RFT\charc ... | AppFinal | 00:00:00 | 11 rows

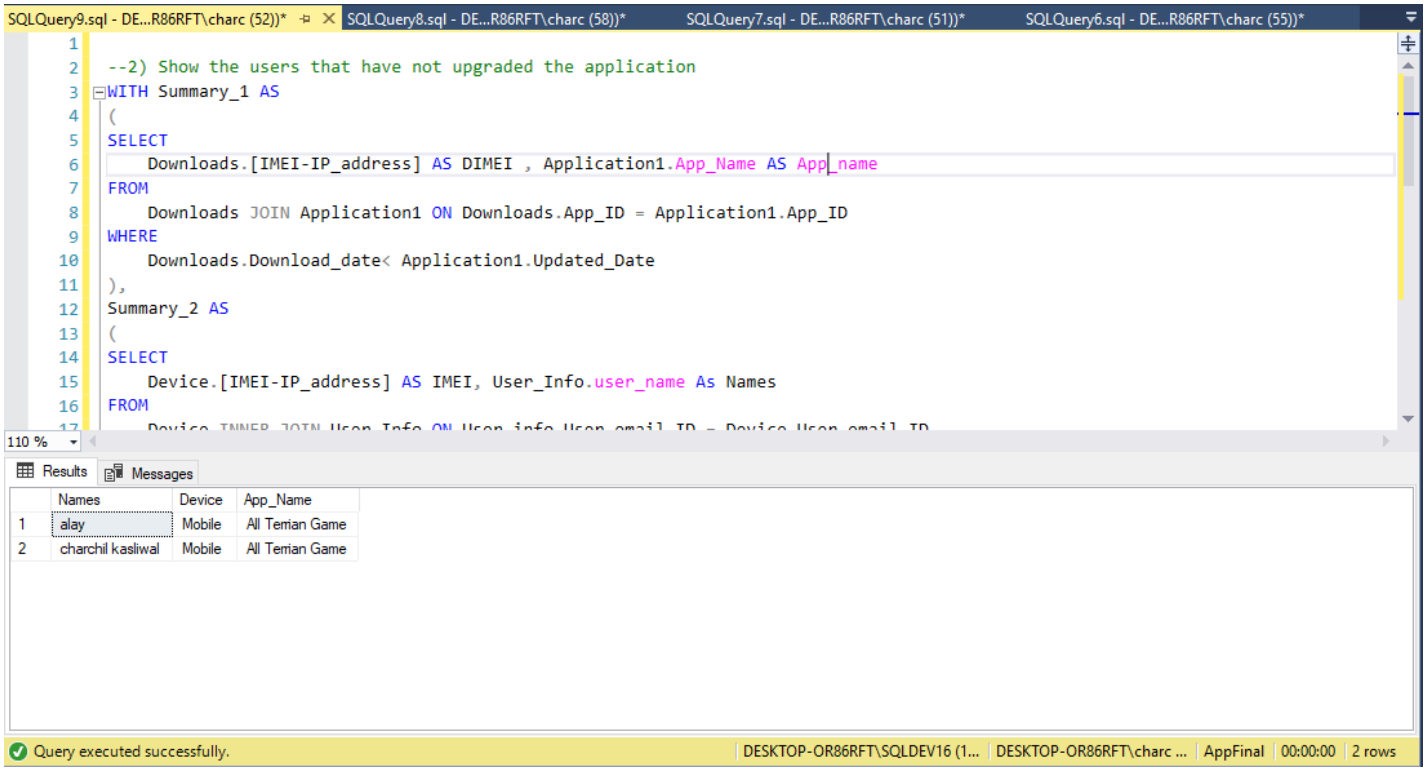
6. Data Model Answering All the Questions

```sql
-- 1) Show the users that have downloaded the current version of an application

WITH Summary_1 AS
(
SELECT
	Downloads.[IMEI-IP_address] AS DIMEI , Application1.App_Name AS App_name
FROM
	Downloads JOIN Application1 ON Downloads.App_ID = Application1.App_ID
WHERE
	Downloads.Download_date> Application1.Updated_Date
),
Summary_2 AS
(
SELECT
	Device.[IMEI-IP_address] AS IMEI, User_Info.user_name As Names
FROM
	Device INNER JOIN User_Info ON User_info.User_email_ID = Device.User_email_ID
),
Summary_3 AS
(
SELECT
	User_info.user_name AS Names2, Device_type.Device_type AS Device, Device.[IMEI-IP_address]
FROM
	[Device]  JOIN User_Info ON User_info.User_email_ID = Device.User_email_ID
			JOIN Device_type ON Device.Device_type_id = Device_type.Device_type_id
)
SELECT
	Summary_2.Names, Summary_3.Device, Summary_1.App_Name
FROM
	Summary_2 JOIN Summary_1 ON Summary_1.DIMEI = Summary_2.IMEI
			JOIN Summary_3 ON Summary_3.[IMEI-IP_address]= Summary_2.IMEI
ORDER BY Names
```

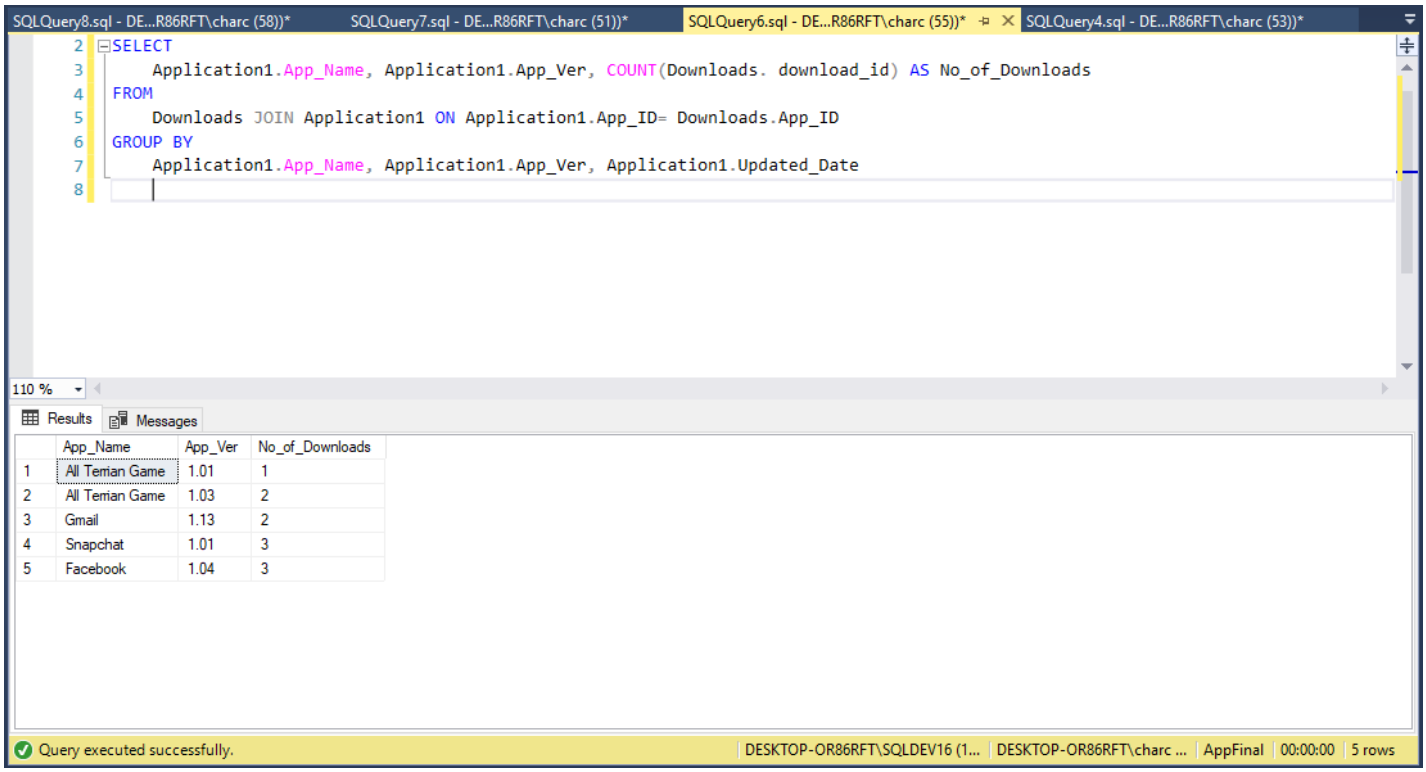


	Names	Device	App_Name
1	alay	Laptop	Facebook
2	charchil kasliwal	Mobile	Snapchat
3	charchil kasliwal	Mobile	Gmail
4	charchil kasliwal	Mobile	Facebook
5	charchil kasliwal	Laptop	Snapchat
6	charchil kasliwal	Laptop	All Terrian Game
7	Khushal	Web App	Facebook
8	Mishra	Web App	Gmail
9	Mishra	Mobile	Snapchat

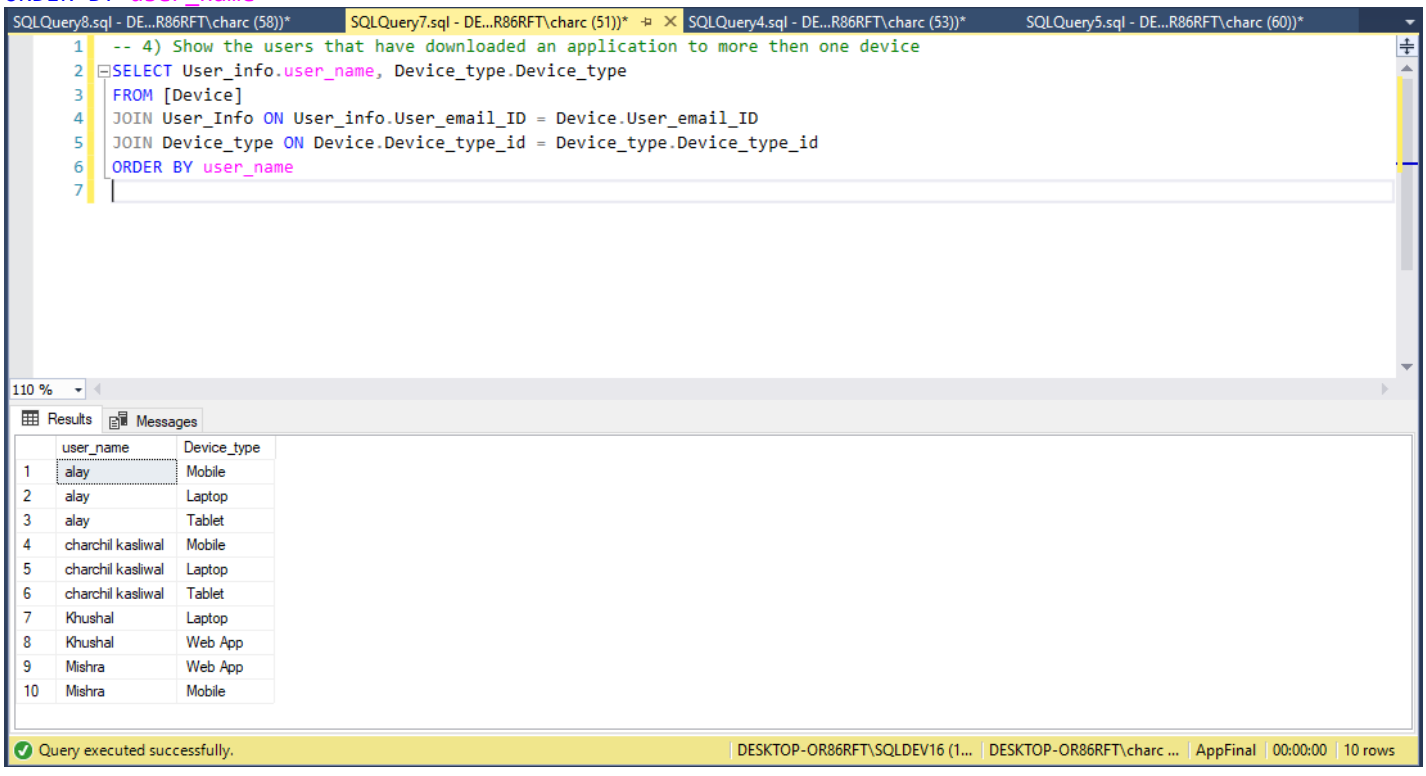
```sql
--2) Show the users that have not upgraded the application
WITH Summary_1 AS
(
SELECT
	Downloads.[IMEI-IP_address] AS DIMEI , Application1.App_Name AS App_name
FROM
	Downloads JOIN Application1 ON Downloads.App_ID = Application1.App_ID
WHERE
	Downloads.Download_date< Application1.Updated_Date
),
Summary_2 AS
(
SELECT
	Device.[IMEI-IP_address] AS IMEI, User_Info.user_name As Names
FROM
	Device INNER JOIN User_Info ON User_info.User_email_ID = Device.User_email_ID
),
Summary_3 AS
(
SELECT
	User_info.user_name AS Names2, Device_type.Device_type AS Device, Device.[IMEI-IP_address]
FROM
	[Device]  JOIN User_Info ON User_info.User_email_ID = Device.User_email_ID
		  JOIN Device_type ON Device.Device_type_id = Device_type.Device_type_id
)
SELECT
	Summary_2.Names, Summary_3.Device, Summary_1.App_Name
FROM
	Summary_2 JOIN Summary_1 ON Summary_1.DIMEI = Summary_2.IMEI
		  JOIN Summary_3 ON Summary_3.[IMEI-IP_address]= Summary_2.IMEI
ORDER BY Names
```



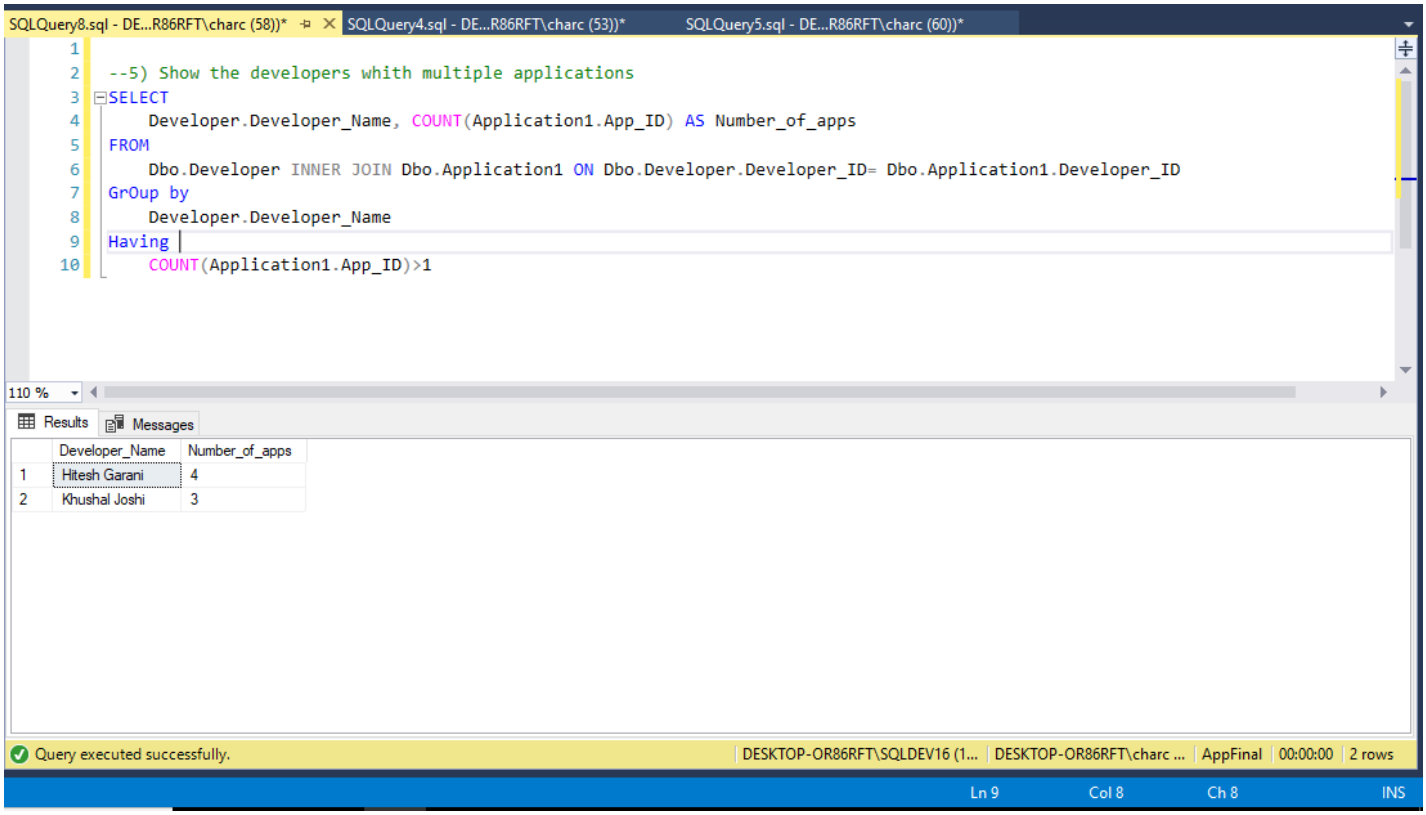
```sql
--3) Should the total count of all downloads for an application by version
SELECT
	Application1.App_Name, Application1.App_Ver, COUNT(Downloads. download_id) AS No_of_Downloads
FROM
	Downloads JOIN Application1 ON Application1.App_ID= Downloads.App_ID
GROUP BY
				Application1.App_Name, Application1.App_Ver, Application1.Updated_Date
```




```sql
-- 4) Show the users that have downloaded an application to more then one device
SELECT User_info.user_name, Device_type.Device_type
FROM [Device]
JOIN User_Info ON User_info.User_email_ID = Device.User_email_ID
JOIN Device_type ON Device.Device_type_id = Device_type.Device_type_id
ORDER BY user_name
```



```sql
--5) Show the developers whith multiple applications
SELECT
	Developer.Developer_Name, COUNT(Application1.App_ID) AS Number_of_apps
FROM
	Dbo.Developer INNER JOIN Dbo.Application1 ON Dbo.Developer.Developer_ID=
Dbo.Application1.Developer_ID
GrOup by
	Developer.Developer_Name
Having
	COUNT(Application1.App_ID)>1
```



7. Triggers, Transactions, Stored procedure

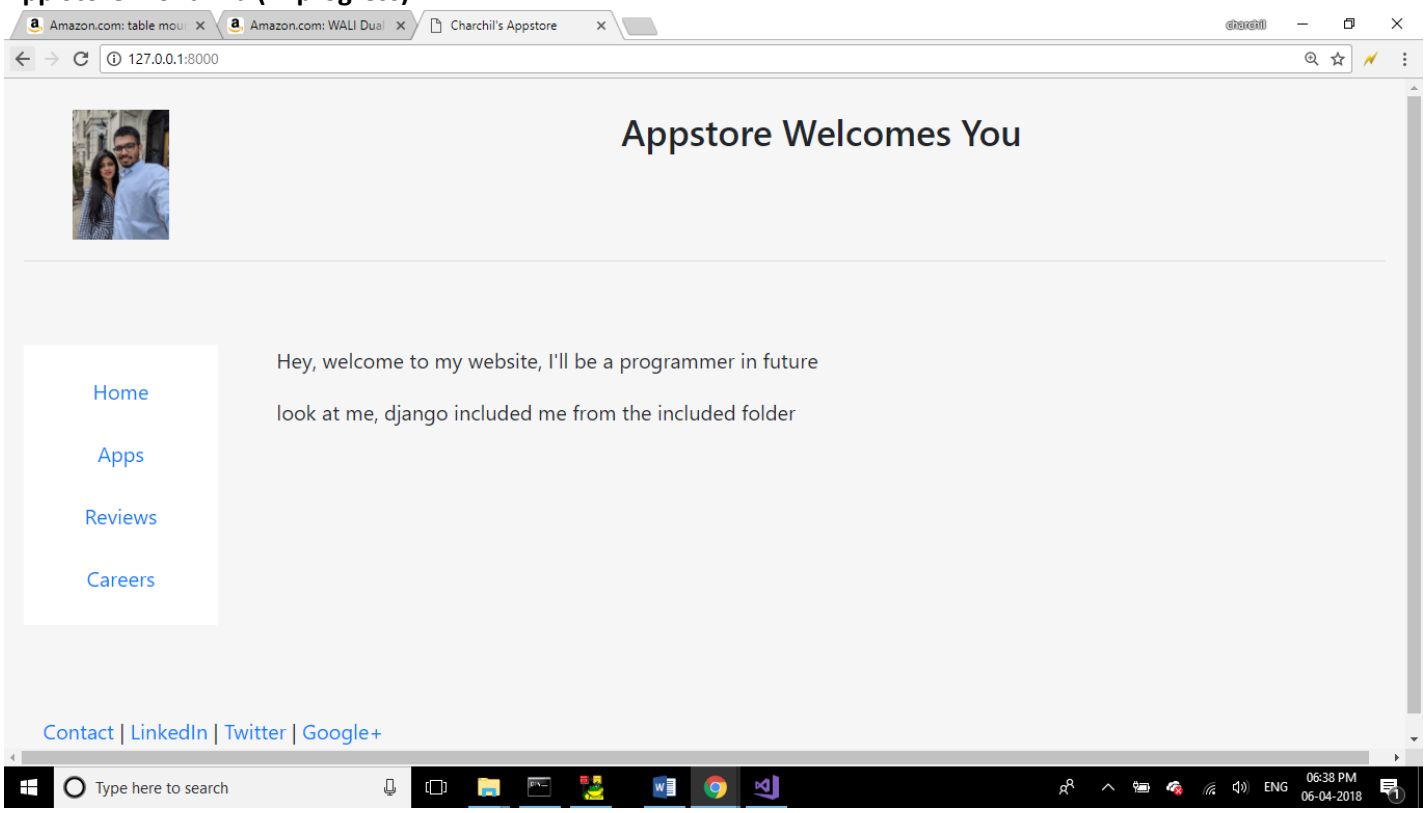
--Part1 – This Trigger would save logs for any app that got deleted or updated in the database.

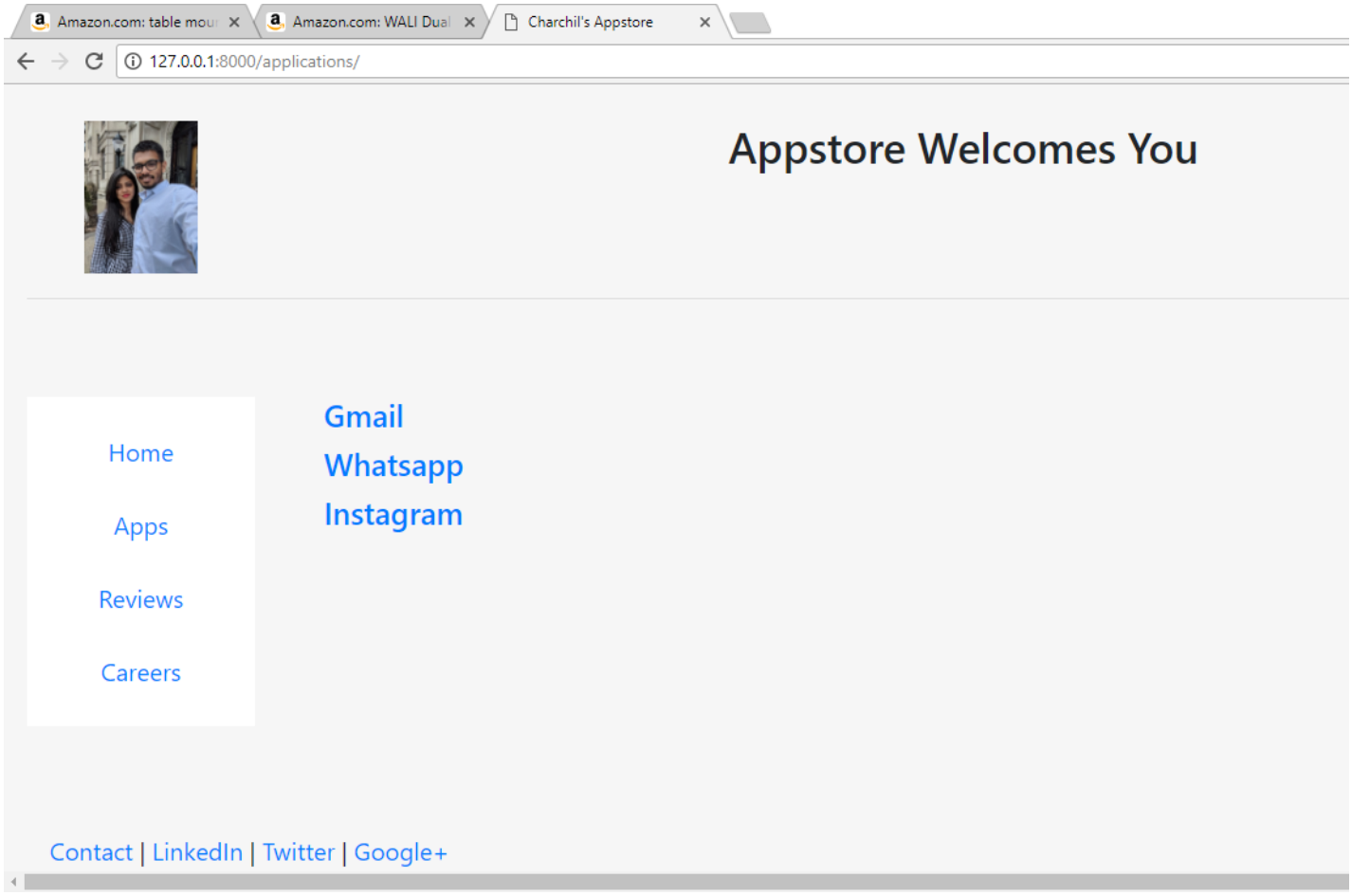
```
CREATE TABLE Logger_hw (new_msg varchar(220), time_stamp datetime, userid varchar(100), loginName varchar(100), ServerName varchar(50))


SELECT * FROM Logger_hw

SELECT * FROM authors

Drop trigger tr_updatedAuthor

create trigger tr_updatedAuthor

on authors

for update

as

DECLARE @newName1 VARCHAR(100)

DECLARE @newName2 Varchar (100)


select @newName1 = (select au_fname + ' ' + au_lname from INSERTED)

PRINT 'Updated Author To ' + @newName1

select @newName2 = (select au_fname + ' ' + au_lname from DELETED)

PRINT 'Updated Author From ' + @newName2

INSERT INTO Logger_hw values('updated', getdate(), @@SPID, SYSTEM_USER, @@SERVERNAME)


Update authors

SET au_fname = 'Charchil', au_lname = 'Kasliwal'

WHERE au_lname = 'White'


--Part 2


SELECT * FROM employee

DROP TABLE logger_emp
```

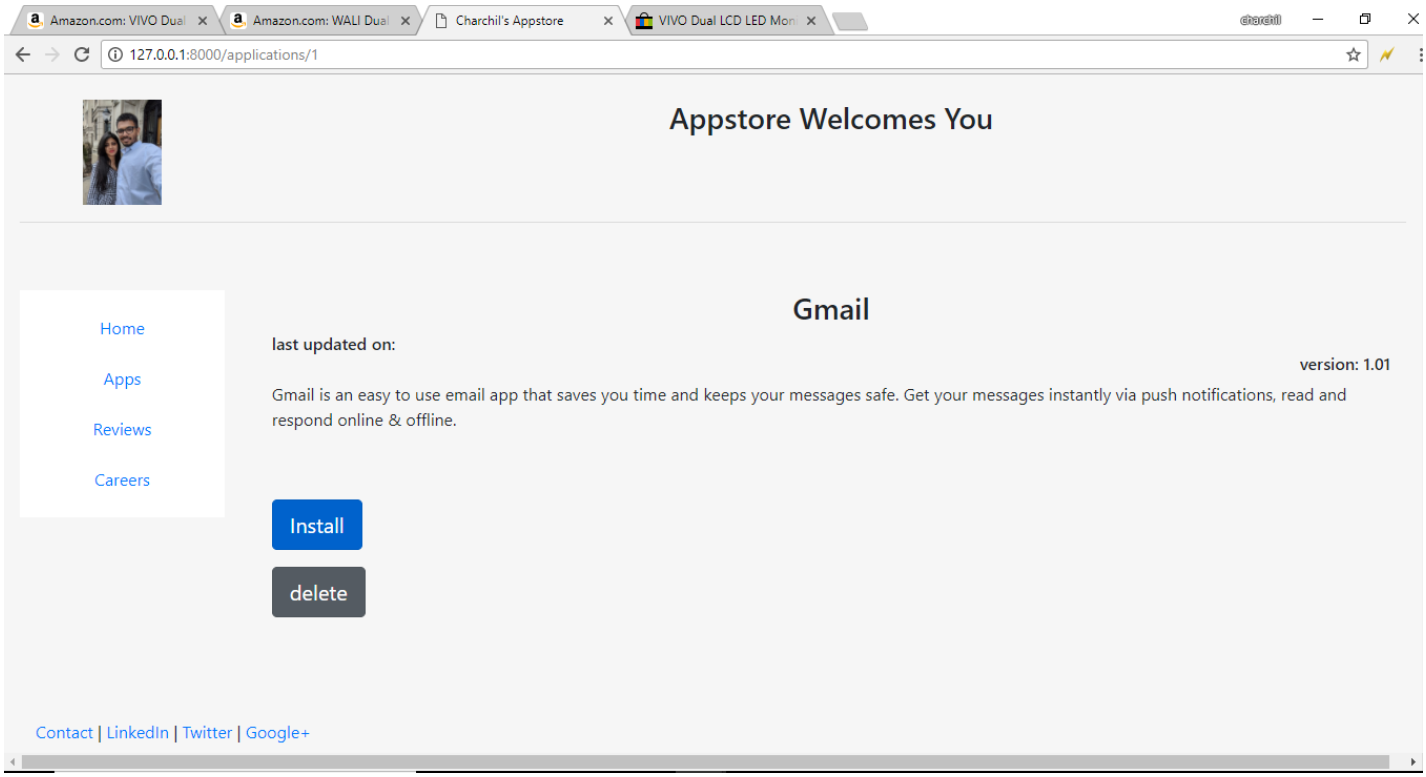
```
CREATE TABLE logger_emp (Activity varchar(200), emp_id varchar(100), old_id varchar(20), new_id varchar(20), old_lvl varchar(20), new_lvl varchar(20), time_stamp Datetime)

SELECT * FROM logger_emp


---- PART-2 --- Trigger to update a new table with Number of downloads for each app_id that got downloaded.

CREATE PARTITION SCHEME pf_appVersion

AS PARTITION pf_appVersion

ALL TO (App_Ver)



DROP TRIGGER tr_noofdownloads

CREATE TRIGGER tr_noofdownloads

ON dbo.Downloads

AFTER

INSERT AS


BEGIN


	Declare @count int = 0;

	SELECT @count = COUNT(download_id)

	FROM dbo.Downloads

	WHERE Downloads.[App_ID] =(SELECT App_ID FROM Inserted);


	BEGIN

	--SET @totalcount int = CAST(@count as int)


	Update [Downloads_manager]

	SET [No_of_downloads] = CAST (@count as int)

	--SET [App_ID] = CAST((SELECT App_ID FROM INSERTED) As Varchar(20))

	--WHERE [Downloads_manager].[App_ID] = (SELECT App_ID FROM INSERTED);

	END

	END;


	SELECT * FROM Downloads
```

```
        INSERT INTO Downloads Values ('10101010@161616','2018-04-10', '10101010','161616', NULL)

        SELECT * FROM Downloads_manager
PART 3 --- Create Backup table for New App version
USE AppFinal
GO

DECLARE @pf_appVersion varchar(20) =
        N' CREATE PARTITION FUNCTION pf_appVersion varchar(20)
        AS RANGE RIGHT FOR VALUES (';
DECLARE @i varchar(20) = 1.00;
WHILE @i<2.00
BEGIN
SET @pf_appVersion += CAST(@i as varchar(20)) + N', ';
SET @i +=.01;
END

SET @pf_appVersion += CAST (@i as varchar(20)) + N');';

EXEC sp_executesql @pf_appVersion;
GO
```

App Store Front End (In progress)







Revision History	Revision Date	Highlights
Revision 1	March 03, 2018	First Document on App Store
Revision 2	April 04, 2018	Model Finialized Entered sample data Query for Prof questions Screen Shots from SSMS Updated Business Rules
Revision	April 23, 2018	Updated Developer Cluster with Corporation Entity Updated Model Updated Business Rule Created Trigger, Backup table Front End (intro)