

**STORYTEL APPLICATION**

**INDIVIDUAL PROJECT**

# **ITC-6000 Database Management Systems**

**PRAPARNA MOHARANA**

**12/20/2021**

# **Introduction**

The individual project idea that I have chosen is the ‘Storytel' application which is an audio and e- book application associated with books. In the application, end-users can listen to audiobooks for the books, they are interested to know about and read e-books if they want.

The significance of this project is to help people know the content of the book without reading it. People can listen to audiobooks while doing multitasking and while travelling reading books are difficult as it strains the eye with a constant moment of vehicle, people can listen to audiobooks and even read the e-book along with it if they wish to.

The reason, I chose the ‘Storytel’ application, I listen to audiobooks more than reading a book.

# **Database System Analysis and Design**

## **User Profiles**

Storytel contains a database of audiobooks along with e-books. The database of audiobooks can be accessed by users in multiple ways for the books they are interested in.

The user needs to sign up if they have not created an account and provide details of name, country, email id, mobile number, subscription plan the end-user wants to choose. After signing up, process the user needs to enter the username and password for the login process and for further authentication need to enter a passcode.

In the ‘Storytel’ application, the user can access a wide variety of audiobooks and e-books too. The audiobook can be filtered according to preferred languages. Kid modes help filter content that is to be not read by kids which are controlled by entering the passcode. Recommendation or Book tips gives an overview of bestsellers, new books and popular books. Series helps to discover unique stories that the ‘Storytel app’ has created which are divided into one-hour episodes each. Categories contain books of various genres such as adventure, action, fantasy & sci-fi. A bookshelf is like a personal library to store books, the user can store books or if the user is listening for a longer time, it automatically gets added to the book self. The user can share the book title in their Instagram account as well as gift an audiobook to a friend. The friend can listen to only one book from a wide variety of audiobooks.

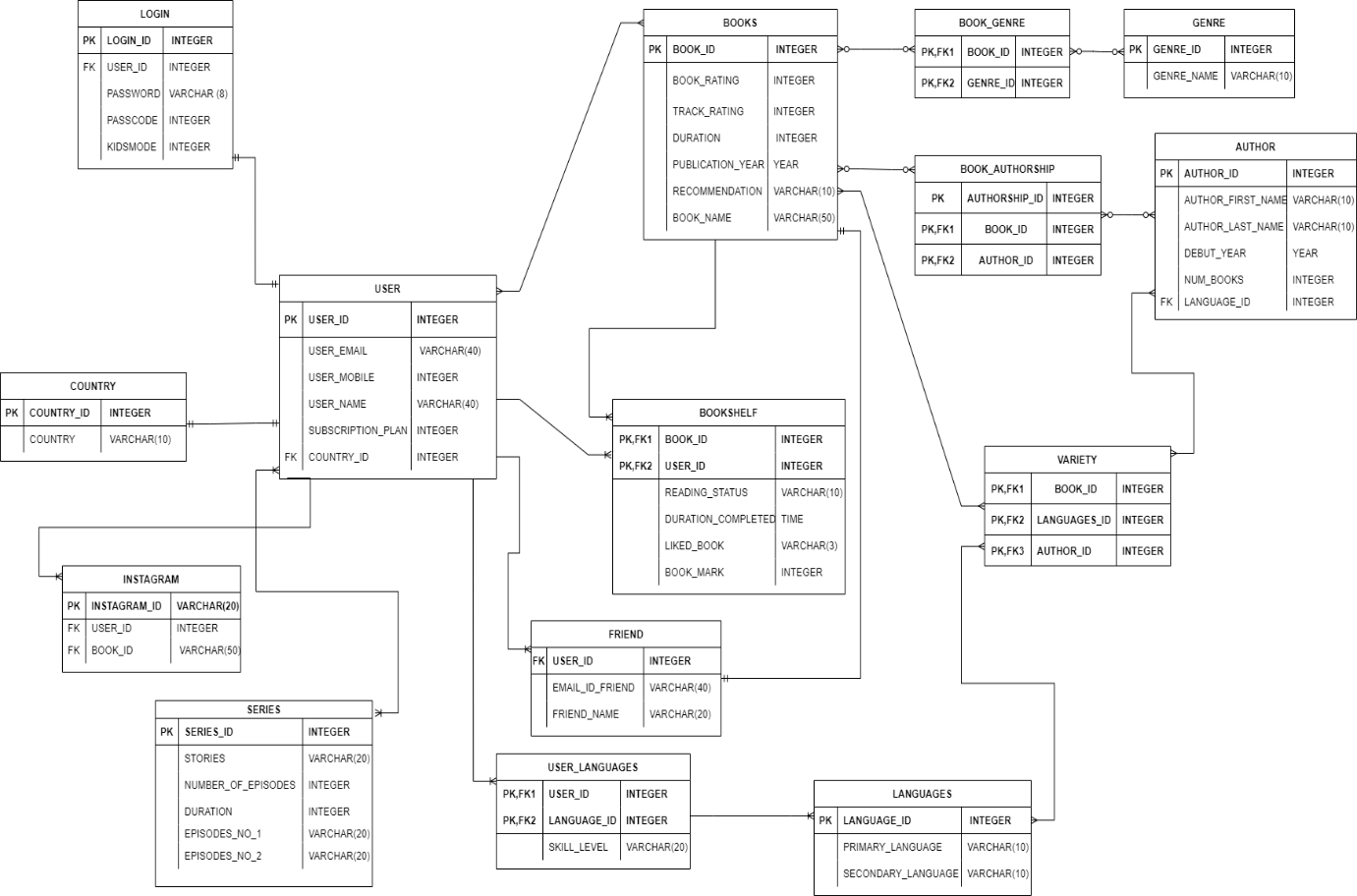
## **Business Rules**

1. One user enters login details once (one to one)
2. The one user only once signs up for a Storytel account. (One to one)
3. Many books can be filtered into a many language (many to many)
4. One book listened for a longer time or liked by the end-user is stored in the personal bookshelf in such a way many books can be stored (one to many)
5. One book listened for a longer time or liked by the end-user is stored in the personal bookshelf in such a way many books can be stored (one to many)
6. Once apply Kid modes is filtered by entering a passcode to view many kids’ related book database. (One to many)
7. One User can gift to many friends. (One to many)
8. One friend can listen to only one book of choice from a variety of books. (One to one)
9. Many Users can access many series to listen to a variety of stories. (One to many)
10. One user shares the book title to one Instagram. (One to one)
11. One user belongs from one country (One to one)
12. Many Books can have multiple authors (Many to many)
13. Many users can read and listen in multiple languages. (Many to many)
14. Many books can have multiple genres. (Many to Many)
15. Many users can access many books. (Many to Many)

**Database Design-Storytel Application**

User entity is the foundation entity which contains information related to users such as user id, user name, user email, user number, subscription plan taken by them. Country entity give us information which country the user belongs to. Login entity contains login details of user such password, passcode, kids’ mode they selected for while accessing the application. Instagram entity helps user to share book, they are correctly interested in. Series entity contains inbuilt stories of Storytel application along with their details. Friend entity helps user to gift their friend where friend can just access a book from book entity. The Book entity holds information related to books. User and Books entity are connected by a joining entity that is the Bookshelf which contains details about reading status is the user currently active or not, duration completed denotes till which duration user has listen to audio book and replay from that point Liked book attribute stores the book in shelf if the users read for long time or liked by user and book mark stores page number for user in e-books. A book entity can have more than one author for a book. The Book entity and authors entity are connected through book authorship joining entity. The book can have multiple genres, the genre and book entity are connected by book genre joining entity. Variety is a joining entity which is connecting to Language entity which contains list of languages. For helping user to get access to multiple languages, user and languages entity is connected through user languages joining entity which contains attribute as skill level.

**Entity Relationship Diagram**



# **Database Implementation**

# **Database Schema**

# CREATE TABLE IF NOT EXISTS "USER" (

# "USER\_ID" INTEGER,

# "USER\_NAME" TEXT,

# "USER\_EMAIL" TEXT,

# "USER\_MOBILE" INTEGER,

# "SUBSCRIPTION\_PLAN" INTEGER,

# "COUNTRY\_ID" INTEGER,

# FOREIGN KEY("COUNTRY\_ID")

# REFERENCES "COUNTRY"("COUNTRY\_ID"),

# PRIMARY KEY("USER\_ID")

# );

# CREATE TABLE IF NOT EXISTS "COUNTRY" (

# "COUNTRY\_ID" INTEGER,

# "COUNTRY" TEXT,

# PRIMARY KEY("COUNTRY\_ID")

# );

# CREATE TABLE IF NOT EXISTS "LOGIN" (

# "LOGIN\_ID" INTEGER,

# "USER\_ID" INTEGER,

# "PASSWORD" TEXT,

# "PASSCODE" INTEGER,

# "KIDSMODE" TEXT,

# FOREIGN KEY("USER\_ID") REFERENCES "USER"("USER\_ID"),

# PRIMARY KEY("LOGIN\_ID")

# );

# CREATE TABLE IF NOT EXISTS "BOOKS" (

# "BOOK\_ID" INTEGER,

# "BOOK\_RATING" REAL,

# "TRACK\_RATING" INTEGER,

# "DURATION(INHOURS)" TEXT,

# "PUBLICATION\_YEAR" INTEGER,

# "RECOMMENDATION" TEXT,

# "BOOK\_NAME" TEXT,

# PRIMARY KEY("BOOK\_ID")

# );

# CREATE TABLE IF NOT EXISTS "BOOK\_GENRE" (

# "BOOK\_ID" INTEGER,

# "GENRE\_ID" INTEGER,

# FOREIGN KEY("GENRE\_ID") REFERENCES "GENRE"("GENRE\_ID"),

# FOREIGN KEY("BOOK\_ID") REFERENCES "BOOKS"("BOOK\_ID"),

# PRIMARY KEY("BOOK\_ID","GENRE\_ID")

# );

# CREATE TABLE IF NOT EXISTS "GENRE" (

# "GENRE\_ID" INTEGER,

# "GENRE" TEXT,

# PRIMARY KEY("GENRE\_ID")

# );

# CREATE TABLE IF NOT EXISTS "BOOK\_AUTHORSHIP" (

# "AUTHORSHIP\_ID" INTEGER,

# "BOOK\_ID" INTEGER,

# "AUTHOR\_ID" INTEGER,

# FOREIGN KEY("BOOK\_ID") REFERENCES "BOOKS"("BOOK\_ID"),

# FOREIGN KEY("AUTHOR\_ID") REFERENCES "AUTHOR"("AUTHOR\_ID"),

# PRIMARY KEY("AUTHORSHIP\_ID","BOOK\_ID","AUTHOR\_ID")

# );

# CREATE TABLE IF NOT EXISTS "AUTHOR" (

# "AUTHOR\_ID" INTEGER,

# "AUTHOR\_FIRST\_NAME" TEXT,

# "AUTHOR\_LAST\_NAME" TEXT,

# "DEBUT\_YEAR" INTEGER,

# "NUM\_BOOKS" INTEGER,

# "LANGUAGE\_ID" INTEGER,

# FOREIGN KEY("LANGUAGE\_ID") REFERENCES "VARIETY"("LANGUAGE\_ID"),

# PRIMARY KEY("AUTHOR\_ID")

# );

# CREATE TABLE IF NOT EXISTS "VARIETY" (

# "BOOK\_ID" INTEGER,

# "LANGUAGE\_ID" INTEGER,

# "AUTHOR\_ID" INTEGER,

# FOREIGN KEY("BOOK\_ID") REFERENCES "BOOKS"("BOOK\_ID"),

# FOREIGN KEY("LANGUAGE\_ID") REFERENCES "LANGUAGE"("LANGUAGE\_ID"),

# FOREIGN KEY("AUTHOR\_ID") REFERENCES "AUTHOR"("AUTHOR\_ID"),

# PRIMARY KEY("BOOK\_ID","LANGUAGE\_ID","AUTHOR\_ID")

# );

# CREATE TABLE IF NOT EXISTS "LANGUAGE" (

# "LANGUAGE\_ID" INTEGER,

# "PRIMARY\_LANGUAGE" TEXT,

# "SECONDARY\_LANGUAGE" TEXT,

# PRIMARY KEY("LANGUAGE\_ID")

# );

# CREATE TABLE IF NOT EXISTS "BOOKSHELF" (

# "BOOK\_ID" INTEGER,

# "USER\_ID" INTEGER,

# "READING\_STATUS" TEXT,

# "DURATION\_COMPLETED" TEXT,

# "LIKED\_BOOK" TEXT,

# "BOOK\_MARK" INTEGER,

# FOREIGN KEY("BOOK\_ID") REFERENCES "BOOKS"("BOOK\_ID"),

# FOREIGN KEY("USER\_ID") REFERENCES "INSTAGRAM"("INSTAGRAM\_ID"),

# PRIMARY KEY("BOOK\_ID","USER\_ID")

# );

# CREATE TABLE IF NOT EXISTS "INSTAGRAM" (

# "INSTAGRAM\_ID" TEXT,

# "USER\_ID" INTEGER,

# "BOOKID" INTEGER,

# FOREIGN KEY("USER\_ID") REFERENCES "USER"("USER\_ID"),

# PRIMARY KEY("INSTAGRAM\_ID")

# );

# CREATE TABLE IF NOT EXISTS "FRIEND" (

# "USER\_ID" TEXT,

# "EMAIL\_ID\_FRIEND" TEXT,

# "FRIEND\_NAME" TEXT,

# FOREIGN KEY("USER\_ID") REFERENCES "USER"("USER\_ID")

# );

# CREATE TABLE IF NOT EXISTS "SERIES" (

# "SERIES\_ID" INTEGER,

# "STORIES" TEXT,

# "NUMBER\_OF\_EPISODES" INTEGER,

# "DURATION" TEXT,

# "EPISODES\_NO\_1" TEXT,

# "EPISODES\_NO\_2" TEXT,

# "EPISODES\_NO\_3" TEXT,

# "EPISODES\_NO\_4" TEXT,

# "EPISODES\_NO\_5" TEXT,

# PRIMARY KEY("SERIES\_ID")

# );

# CREATE TABLE IF NOT EXISTS "USER\_LANGUAGES" (

# "USER\_ID" INTEGER,

# "LANGUAGE\_ID" INTEGER,

# "SKILL\_LEVEL" TEXT,

# FOREIGN KEY("USER\_ID") REFERENCES "USER"("USER\_ID"),

# FOREIGN KEY("LANGUAGE\_ID") REFERENCES "LANGUAGE"("LANGUAGE\_ID"),

# PRIMARY KEY("USER\_ID","LANGUAGE\_ID")

# );

# INSERT INTO "USER" VALUES (101,'JAMES','aa@gmail.com',7037890523,99,510);

# INSERT INTO "USER" VALUES (102,'JOHN','bb@gmail.com',3056774211,200,502);

# INSERT INTO "USER" VALUES (103,'KELLY','cc@gmail.com',3434562143,99,506);

# INSERT INTO "USER" VALUES (104,'JAMES','dd@gmail.com',9056540976,45,502);

# INSERT INTO "USER" VALUES (105,'HELEN','ee@gmail.com',8964321134,99,501);

# INSERT INTO "USER" VALUES (106,'DAISY','ff@gmail.com',9673322005,200,506);

# INSERT INTO "USER" VALUES (107,'HELLY','gg@gmail.com',9778945311,300,509);

# INSERT INTO "USER" VALUES (108,'DAVID','hh@gmail.com',9654675301,99,503);

# INSERT INTO "USER" VALUES (109,'SALLY','ii@gmail.com',4998908900,45,507);

# INSERT INTO "USER" VALUES (110,'RICKY','jj@gmail.com',6879099833,99,508);

# INSERT INTO "USER" VALUES (111,NULL,NULL,NULL,NULL,NULL);

# INSERT INTO "USER" VALUES (112,NULL,NULL,NULL,NULL,NULL);

# INSERT INTO "USER" VALUES (113,NULL,NULL,NULL,NULL,NULL);

# INSERT INTO "USER" VALUES (114,NULL,NULL,NULL,NULL,NULL);

# INSERT INTO "USER" VALUES (115,NULL,NULL,NULL,NULL,NULL);

# INSERT INTO "USER" VALUES (116,NULL,NULL,NULL,NULL,NULL);

# INSERT INTO "USER" VALUES (117,NULL,NULL,NULL,NULL,NULL);

# INSERT INTO "USER" VALUES (118,NULL,NULL,NULL,NULL,NULL);

# INSERT INTO "USER" VALUES (119,NULL,NULL,NULL,NULL,NULL);

# INSERT INTO "USER" VALUES (120,NULL,NULL,NULL,NULL,NULL);

# INSERT INTO "USER" VALUES (121,NULL,NULL,NULL,NULL,NULL);

# INSERT INTO "USER" VALUES (122,NULL,NULL,NULL,NULL,NULL);

# INSERT INTO "USER" VALUES (123,NULL,NULL,NULL,NULL,NULL);

# INSERT INTO "USER" VALUES (124,NULL,NULL,NULL,NULL,NULL);

# INSERT INTO "USER" VALUES (125,NULL,NULL,NULL,NULL,NULL);

# INSERT INTO "USER" VALUES (126,NULL,NULL,NULL,NULL,NULL);

# INSERT INTO "USER" VALUES (127,NULL,NULL,NULL,NULL,NULL);

# INSERT INTO "COUNTRY" VALUES (501,'BELGIUM');

# INSERT INTO "COUNTRY" VALUES (502,'BRAZIL');

# INSERT INTO "COUNTRY" VALUES (503,'BULGARIA');

# INSERT INTO "COUNTRY" VALUES (504,'COLOMBIA');

# INSERT INTO "COUNTRY" VALUES (505,'DENMARK');

# INSERT INTO "COUNTRY" VALUES (506,'EQYPT');

# INSERT INTO "COUNTRY" VALUES (507,'FINLAND');

# INSERT INTO "COUNTRY" VALUES (508,'GERMANY');

# INSERT INTO "COUNTRY" VALUES (509,'ITALY');

# INSERT INTO "COUNTRY" VALUES (510,'INDIA');

# INSERT INTO "LOGIN" VALUES (301,101,'33##11',1236,'NO');

# INSERT INTO "LOGIN" VALUES (302,102,'$21%%',9823,'YES');

# INSERT INTO "LOGIN" VALUES (303,103,'Akn(I',984,'YES');

# INSERT INTO "LOGIN" VALUES (304,104,'BF924',1237,'NO');

# INSERT INTO "LOGIN" VALUES (305,105,'Ajyli',3456,'YES');

# INSERT INTO "LOGIN" VALUES (306,106,'Vyu\_s',2143,'NO');

# INSERT INTO "LOGIN" VALUES (307,107,'Qtyss12',5064,'YES');

# INSERT INTO "LOGIN" VALUES (308,108,'JJB893',2175,'YES');

# INSERT INTO "LOGIN" VALUES (309,109,'$23175',3537,'NO');

# INSERT INTO "LOGIN" VALUES (310,110,'&34RF',9842,'YES');

# INSERT INTO "BOOKS" VALUES (201,4.0,4,'09:33',2010,'BEST\_SELLERS','2\_STATES:THE\_STORY\_OF\_MY\_MARRIAGE');

# INSERT INTO "BOOKS" VALUES (202,4.1,3.9,'10:30',2011,'SIMILAR\_TITLES','The\_One\_We\_Fell\_in\_Love\_With');

# INSERT INTO "BOOKS" VALUES (203,0.0,0,'06:22',2009,'LATEST\_RELEASES','Rise\_Up,Sista');

# INSERT INTO "BOOKS" VALUES (204,3.8,3.9,'07:34',2007,'COMICS\_FOR\_GROWNUPS','Penguins\_Hate\_Stuff');

# INSERT INTO "BOOKS" VALUES (205,4.3,4,'04:00',2002,'ENGLISH\_TOP\_LIST','The\_Alchemist');

# INSERT INTO "BOOKS" VALUES (206,4.1,3.5,'19:00',2001,'THRILLERS','Come\_My\_Comapanion');

# INSERT INTO "BOOKS" VALUES (207,0.0,0,'08:07',2008,'LATEST\_RELEASES','The Girls` Book of Priesthood');

# INSERT INTO "BOOKS" VALUES (208,4.0,4.5,'01:00',2011,'BEST\_SELLERS','The\_Power\_of\_Your\_Subconsious\_Mind');

# INSERT INTO "BOOKS" VALUES (209,4.7,4.9,'03:00',2008,'BEST\_SELLERS','How\_to\_talk\_to\_Anybody,Anytime,Anywhere');

# INSERT INTO "BOOKS" VALUES (210,4.2,4.5,'04:00',2001,'BEST\_SELLERS','Ikigai:The\_Japanese\_Secret\_Philosphy\_for\_a\_Happy\_Healthy\_Life');

# INSERT INTO "BOOKS" VALUES (211,3.4,4.4,'08:00',2004,'BEST\_SELLERS','Five\_Point\_Someone');

# INSERT INTO "BOOK\_GENRE" VALUES (201,401);

# INSERT INTO "BOOK\_GENRE" VALUES (202,402);

# INSERT INTO "BOOK\_GENRE" VALUES (203,402);

# INSERT INTO "BOOK\_GENRE" VALUES (204,403);

# INSERT INTO "BOOK\_GENRE" VALUES (205,402);

# INSERT INTO "BOOK\_GENRE" VALUES (206,404);

# INSERT INTO "BOOK\_GENRE" VALUES (207,401);

# INSERT INTO "BOOK\_GENRE" VALUES (208,402);

# INSERT INTO "BOOK\_GENRE" VALUES (209,405);

# INSERT INTO "BOOK\_GENRE" VALUES (210,406);

# INSERT INTO "GENRE" VALUES (401,'ROMANCE');

# INSERT INTO "GENRE" VALUES (402,'FICTION');

# INSERT INTO "GENRE" VALUES (403,'NON-FICTION');

# INSERT INTO "GENRE" VALUES (404,'THRILLER');

# INSERT INTO "GENRE" VALUES (405,'ECONOMY');

# INSERT INTO "GENRE" VALUES (406,'RELIGION');

# INSERT INTO "BOOK\_AUTHORSHIP" VALUES (801,201,601);

# INSERT INTO "BOOK\_AUTHORSHIP" VALUES (802,202,602);

# INSERT INTO "BOOK\_AUTHORSHIP" VALUES (803,203,603);

# INSERT INTO "BOOK\_AUTHORSHIP" VALUES (804,204,604);

# INSERT INTO "BOOK\_AUTHORSHIP" VALUES (805,205,605);

# INSERT INTO "BOOK\_AUTHORSHIP" VALUES (806,206,606);

# INSERT INTO "BOOK\_AUTHORSHIP" VALUES (807,207,607);

# INSERT INTO "BOOK\_AUTHORSHIP" VALUES (808,208,608);

# INSERT INTO "BOOK\_AUTHORSHIP" VALUES (809,209,609);

# INSERT INTO "BOOK\_AUTHORSHIP" VALUES (810,210,610);

# INSERT INTO "AUTHOR" VALUES (601,'Chetan','Bhagat',2001,6,701);

# INSERT INTO "AUTHOR" VALUES (602,'Paige','Toon',2002,8,704);

# INSERT INTO "AUTHOR" VALUES (603,'Kristine','Simelda',2003,4,702);

# INSERT INTO "AUTHOR" VALUES (604,'Greg','Stones',2005,1,706);

# INSERT INTO "AUTHOR" VALUES (605,'Paulo','Coelho',2004,5,703);

# INSERT INTO "AUTHOR" VALUES (606,'Himardri','Gupta',2005,3,701);

# INSERT INTO "AUTHOR" VALUES (607,'Louise','Rowland',2003,1,710);

# INSERT INTO "AUTHOR" VALUES (608,'Joseph','Murphy',2001,5,706);

# INSERT INTO "AUTHOR" VALUES (609,'Chris','Widener',2001,6,707);

# INSERT INTO "AUTHOR" VALUES (610,'Marie','Xue',2002,1,706);

# INSERT INTO "VARIETY" VALUES (201,701,601);

# INSERT INTO "VARIETY" VALUES (202,704,602);

# INSERT INTO "VARIETY" VALUES (203,702,603);

# INSERT INTO "VARIETY" VALUES (204,706,604);

# INSERT INTO "VARIETY" VALUES (205,703,605);

# INSERT INTO "VARIETY" VALUES (206,701,606);

# INSERT INTO "VARIETY" VALUES (207,710,607);

# INSERT INTO "VARIETY" VALUES (208,706,608);

# INSERT INTO "VARIETY" VALUES (209,707,609);

# INSERT INTO "VARIETY" VALUES (210,706,610);

# INSERT INTO "LANGUAGE" VALUES (701,'ENGLISH','DUTCH');

# INSERT INTO "LANGUAGE" VALUES (702,'ENGLISH','PORTUGESE');

# INSERT INTO "LANGUAGE" VALUES (703,'ENGLISH','BULGARIAN');

# INSERT INTO "LANGUAGE" VALUES (704,'ENGLISH','SPANISH');

# INSERT INTO "LANGUAGE" VALUES (705,'ENGLISH','DANISH');

# INSERT INTO "LANGUAGE" VALUES (706,'ENGLISH','ARABIC');

# INSERT INTO "LANGUAGE" VALUES (707,'ENGLISH','SWEDISH');

# INSERT INTO "LANGUAGE" VALUES (708,'ENGLISH','GERMAN');

# INSERT INTO "LANGUAGE" VALUES (709,'ENGLISH','ITALIAN');

# INSERT INTO "LANGUAGE" VALUES (710,'ENGLISH','HINDI');

# INSERT INTO "BOOKSHELF" VALUES (201,101,'ACTIVE','02:00','YES',20);

# INSERT INTO "BOOKSHELF" VALUES (202,101,'NOT-ACTIVE','03:12','YES',56);

# INSERT INTO "BOOKSHELF" VALUES (203,104,'ACTIVE','00:30','YES',67);

# INSERT INTO "INSTAGRAM" VALUES ('James',101,207);

# INSERT INTO "INSTAGRAM" VALUES ('John90',102,202);

# INSERT INTO "INSTAGRAM" VALUES ('Kelly78',103,203);

# INSERT INTO "INSTAGRAM" VALUES ('james\_12',104,204);

# INSERT INTO "INSTAGRAM" VALUES ('HeLlY\*\*90',105,204);

# INSERT INTO "INSTAGRAM" VALUES ('Daisy$$',106,206);

# INSERT INTO "FRIEND" VALUES ('101','12w@gmail.com','ETHANS');

# INSERT INTO "FRIEND" VALUES ('106','23s@gmail.com','GABRILA');

# INSERT INTO "FRIEND" VALUES ('110','34t@gmail.com','JABRILA');

# INSERT INTO "FRIEND" VALUES (NULL,NULL,NULL);

# INSERT INTO "FRIEND" VALUES (NULL,NULL,NULL);

# INSERT INTO "FRIEND" VALUES (NULL,NULL,NULL);

# INSERT INTO "SERIES" VALUES (901,'SATYR\_OF\_THE\_SUBWAY',5,'10:00','A\_Prayer\_For\_Sax','Behe-Moth','Come\_To\_My\_Bed,My\_Pretty','Consider\_the\_Tree','Mercury\_Woman');

# INSERT INTO "SERIES" VALUES (902,'RADHA-THE\_SUPERHERO\_VAMPIRE',5,'10:00','PART1:S01E01','PART2:S01E02','PART3:S01E03','PART4:S01E04','PART5:S01E05');

# INSERT INTO "SERIES" VALUES (903,'BAABLI\_GHOSH\_IS\_ALIVE',5,'10:00','PART1:S01E01','PART2:S01E02','PART3:S01E03','PART4:S01E04','PART5:S01E05');

# INSERT INTO "SERIES" VALUES (904,'LITTLE\_DURGA',3,'12:00','PART1:S01E01','PART2:S01E02','PART3:S01E03',NULL,NULL);

# INSERT INTO "SERIES" VALUES (905,'INDIA\_UNCHARTED',2,'10:00','JALLIANWALA\_BAGH','INDIAN\_REVOLUTION',NULL,NULL,NULL);

# INSERT INTO "USER\_LANGUAGES" VALUES (101,710,'Profession\_proficient');

# INSERT INTO "USER\_LANGUAGES" VALUES (102,702,'Beginner');

# INSERT INTO "USER\_LANGUAGES" VALUES (103,706,'Limited\_proficient');

# INSERT INTO "USER\_LANGUAGES" VALUES (104,702,'Native\_Language');

# INSERT INTO "USER\_LANGUAGES" VALUES (105,701,'Profession\_proficient');

# INSERT INTO "USER\_LANGUAGES" VALUES (106,706,'Beginner');

# INSERT INTO "USER\_LANGUAGES" VALUES (107,709,'Limited\_proficient');

# INSERT INTO "USER\_LANGUAGES" VALUES (108,703,'Native\_Language');

# INSERT INTO "USER\_LANGUAGES" VALUES (109,707,'Profession\_proficient');

# INSERT INTO "USER\_LANGUAGES" VALUES (110,708,'Limited\_proficient');

# CREATE VIEW [KIDSMODE] AS

# SELECT USER\_NAME,USER\_EMAIL,USER\_MOBILE,KIDSMODE

# FROM LOGIN

# JOIN USER ON LOGIN.USER\_ID=USER.USER\_ID;

## **Data Query Implementation**

***Business Statement 1:* Users can check for specific Genre of book and view details of the book.**

**SELECT** BOOK\_NAME, GENRE

**FROM** BOOKS

**JOIN** BOOK\_GENRE **ON** BOOKS.BOOK\_ID=BOOK\_GENRE.BOOK\_ID

**JOIN** GENRE **ON** GENRE.GENRE\_ID=BOOK\_GENRE.GENRE\_ID

**WHERE** GENRE = 'FICTION'

***Business Statement 2:*** **The user can search for specific book name and view details about it.**

**SELECT** BOOK\_NAME, (AUTHOR\_FIRST\_NAME||""||AUTHOR\_LAST\_NAME) **AS** AUTHOR\_FULLNAME, BOOK\_RATING, TRACK\_RATING

**FROM** BOOKS

**JOIN** BOOK\_AUTHORSHIP **ON** BOOKS.BOOK\_ID=BOOK\_AUTHORSHIP.BOOK\_ID

**JOIN** AUTHOR **ON** BOOK\_AUTHORSHIP.AUTHOR\_ID=AUTHOR.AUTHOR\_ID

**WHERE** BOOK\_NAME **LIKE** '%ALCHEMIST%';

***Business Statement 3:*** **The user can view recommendations such Best Sellers, Latest releases and view list of books related to Recommendation by genre and publication year.**

**SELECT** BOOK\_NAME, RECOMMENDATION, GENRE, PUBLICATION\_YEAR

**FROM** BOOKS

**JOIN** BOOK\_GENRE **ON** BOOKS.BOOK\_ID=BOOK\_GENRE.BOOK\_ID

**JOIN** GENRE **ON** GENRE.GENRE\_ID=BOOK\_GENRE.GENRE\_ID

**WHERE** RECOMMENDATION = 'BEST\_SELLERS'

**ORDER** **BY** PUBLICATION\_YEAR **AND** GENRE

***Business Statement 4:*** **User can update Instagram id as user might change Instagram id in future.**

**UPDATE** INSTAGRAM

**SET** INSTAGRAM\_ID ='HeLlY\*\*90'

**WHERE** USER\_ID = 105;

***Business Statement 5:*** **User can see for top 3 highest rated books by genre.**

**SELECT** BOOK\_NAME, BOOK\_RATING, GENRE

**FROM** BOOKS

**JOIN** BOOK\_GENRE **ON** BOOKS.BOOK\_ID=BOOK\_GENRE.BOOK\_ID

**JOIN** GENRE **ON** GENRE.GENRE\_ID=BOOK\_GENRE.GENRE\_ID

**ORDER** **BY** BOOK\_RATING **DESC**

**LIMIT** 3

***Business Statement 6:*** **Users can view the List of Stories and information related to it.**

**SELECT** STORIES, NUMBER\_OF\_EPISODES, **DURATION**, EPISODES\_NO\_1, EPISODES\_NO\_2, EPISODES\_NO\_3, EPISODES\_NO\_5

**FROM** SERIES;

***Business Statement 7:* Admin can view which User who have taken subscription plans 99.**

**SELECT** USER\_NAME, USER\_EMAIL, USER\_MOBILE, SUBSCRIPTION\_PLAN

**FROM** **USER**

**WHERE** SUBSCRIPTION\_PLAN == 99

***Business Statement 8:*** **Displaying the books which have less rating than three and not rated books as newly released (EXCEPT clause is used to display the result which could not be displayed in second SELECT statement).**

**SELECT** BOOK\_NAME, BOOK\_RATING, GENRE

**FROM** BOOKS

**JOIN** BOOK\_GENRE **ON** BOOKS.BOOK\_ID=BOOK\_GENRE.BOOK\_ID

**JOIN** GENRE **ON** GENRE.GENRE\_ID=BOOK\_GENRE.GENRE\_ID

**EXCEPT**

**SELECT** BOOK\_NAME, BOOK\_RATING, GENRE

**FROM** BOOKS

**JOIN** BOOK\_GENRE **ON** BOOKS.BOOK\_ID=BOOK\_GENRE.BOOK\_ID

**JOIN** GENRE **ON** GENRE.GENRE\_ID=BOOK\_GENRE.GENRE\_ID

**GROUP** **BY** BOOK\_NAME **HAVING** BOOK\_RATING > 3.0

***Business Statement 9:*** **The Admin can view user who have select kids’ mode filter without invading the privacy of users.**

**DROP** **VIEW** **IF** **EXISTS** KIDSMODE

**CREATE** **VIEW** [KIDSMODE] **AS**

**SELECT** USER\_NAME, USER\_EMAIL, USER\_MOBILE, KIDSMODE

**FROM** LOGIN

**JOIN** **USER** **ON** LOGIN.USER\_ID=USER.USER\_ID

**SELECT** USER\_NAME, USER\_EMAIL, USER\_MOBILE, KIDSMODE **FROM** [KIDSMODE];

***Business Statement 10:*** **Ranking of Books according to rating and genre.**

**SELECT** BOOK\_NAME, BOOK\_RATING, GENRE,

**RANK** () **OVER** (

**PARTITION** **BY** GENRE

**ORDER** **BY** BOOK\_RATING) Ranking

**FROM** BOOKS

**JOIN** BOOK\_GENRE **ON** BOOKS.BOOK\_ID=BOOK\_GENRE.BOOK\_ID

**JOIN** GENRE **ON** GENRE.GENRE\_ID=BOOK\_GENRE.GENRE\_ID

***Business Statement 11:*** **Admin can view user country and their proficiency level of language as well as language used by them to access the storytel application.**

**SELECT** USER\_NAME, COUNTRY, PRIMARY\_LANGUAGE, SECONDARY\_LANGUAGE, SKILL\_LEVEL

**FROM** **USER**

**JOIN** USER\_LANGUAGES **ON** USER.USER\_ID=USER\_LANGUAGES.USER\_ID

**JOIN** **LANGUAGE** **ON** USER\_LANGUAGES.LANGUAGE\_ID=LANGUAGE.LANGUAGE\_ID

**JOIN** COUNTRY **ON** COUNTRY.COUNTRY\_ID=USER.COUNTRY\_ID

***Business Statement 12:* User can view ranking of books according to specific genre.**

WITH Genre\_Rank AS (**SELECT** BOOK\_NAME, BOOK\_RATING, GENRE

**FROM** BOOKS

**JOIN** BOOK\_GENRE **ON** BOOKS.BOOK\_ID=BOOK\_GENRE.BOOK\_ID

**JOIN** GENRE **ON** GENRE.GENRE\_ID=BOOK\_GENRE.GENRE\_ID

**WHERE** GENRE = 'FICTION')

**SELECT** BOOK\_NAME, BOOK\_RATING, GENRE,

**RANK** () **OVER** (

**PARTITION** **BY** GENRE

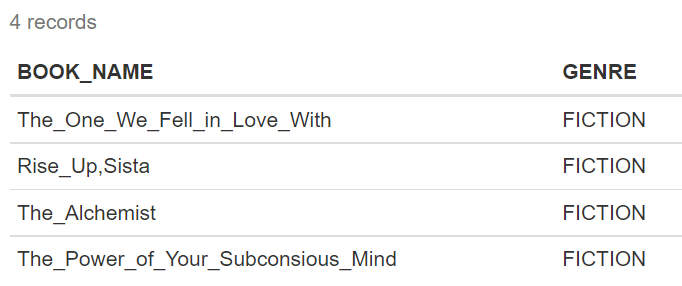
**ORDER** **BY** BOOK\_RATING **DESC**) Ranking

**FROM** Genre\_Rank

# **Analytics and Reports**

**Report 1:**

Users can view the results of books based on different genre. In instance given below, ‘Fiction’ genre is specifically searched and results are produced.



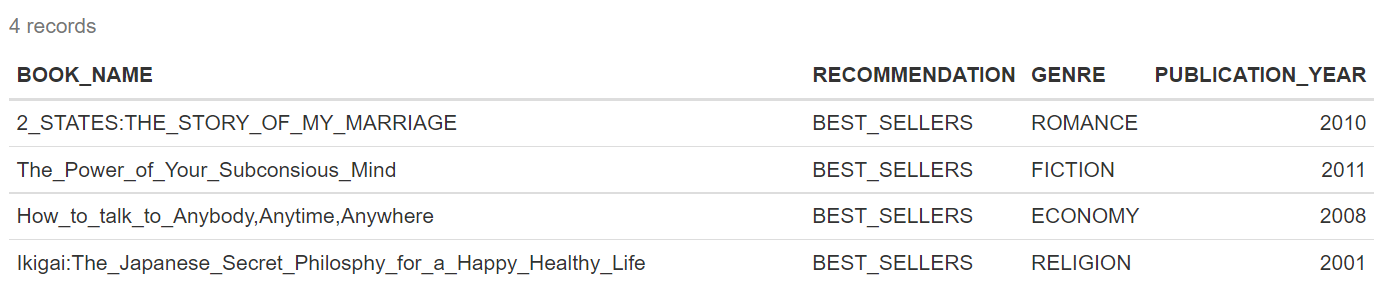
**Report 2:**

The user can search for a particular book name, they are interested in and view the details of books. In instance given below, “The Alchemist” is search and results is provided with details about the book.



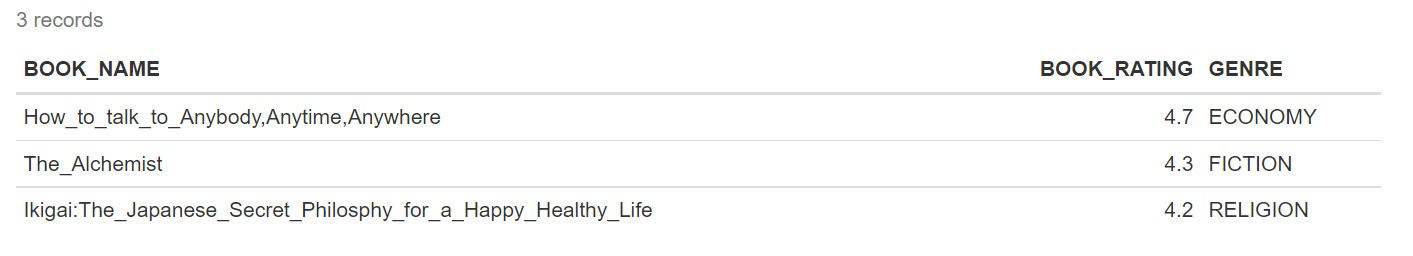
**Report 3:**

The user can view recommendations related to books such as latest releases, popular among kids. In below instance, recommendation for best sellers is provided along with the book name and their publication year as well as genre.



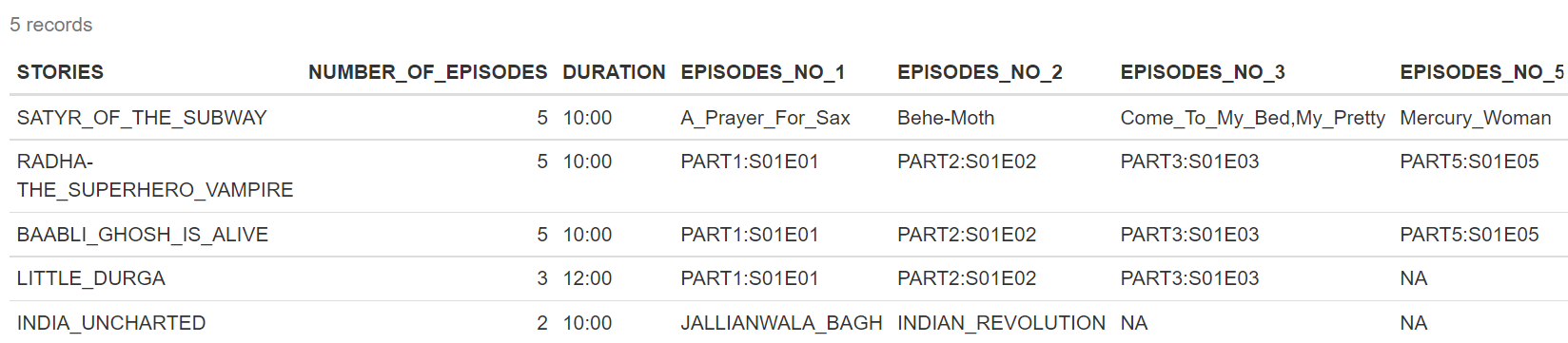
**Report 4:**

User can look for top 3 rated books along with their genre type. It`s important to mention genre of book along with book name and rating because it will give an idea to user which books are top 3 rated depending on interest of their genre.

****

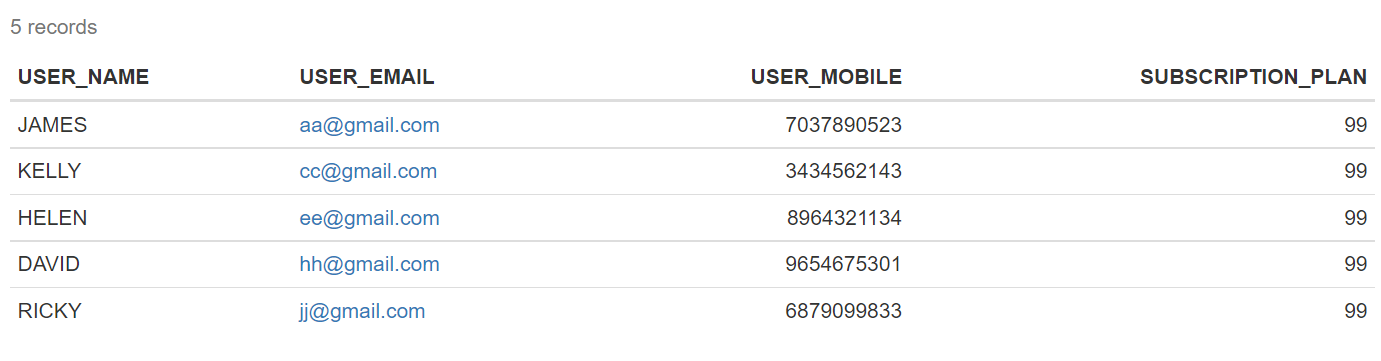
**Report 5:**

User can access inbuilt stories of storytel application and see the list of stories in the application along with details such as Stories name, Number of Episodes present in the stories, duration of stories and titles of each episode.

****

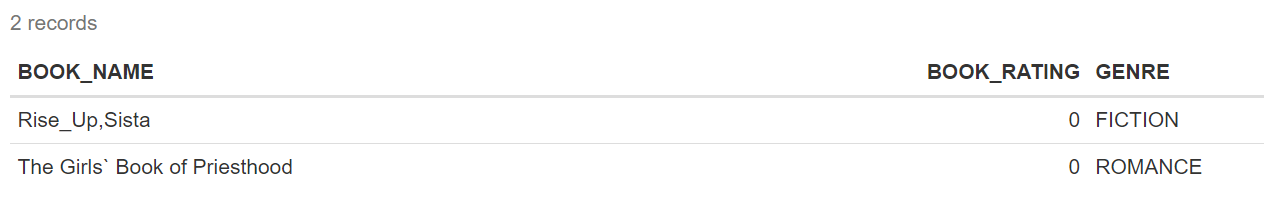
**Report 6:**

Admin can view details related to subscription plans taken by each user and filter according to different plan provided by Storytel application for marketing analysis. In the below instance, admin can view subscription plans of users with 99.



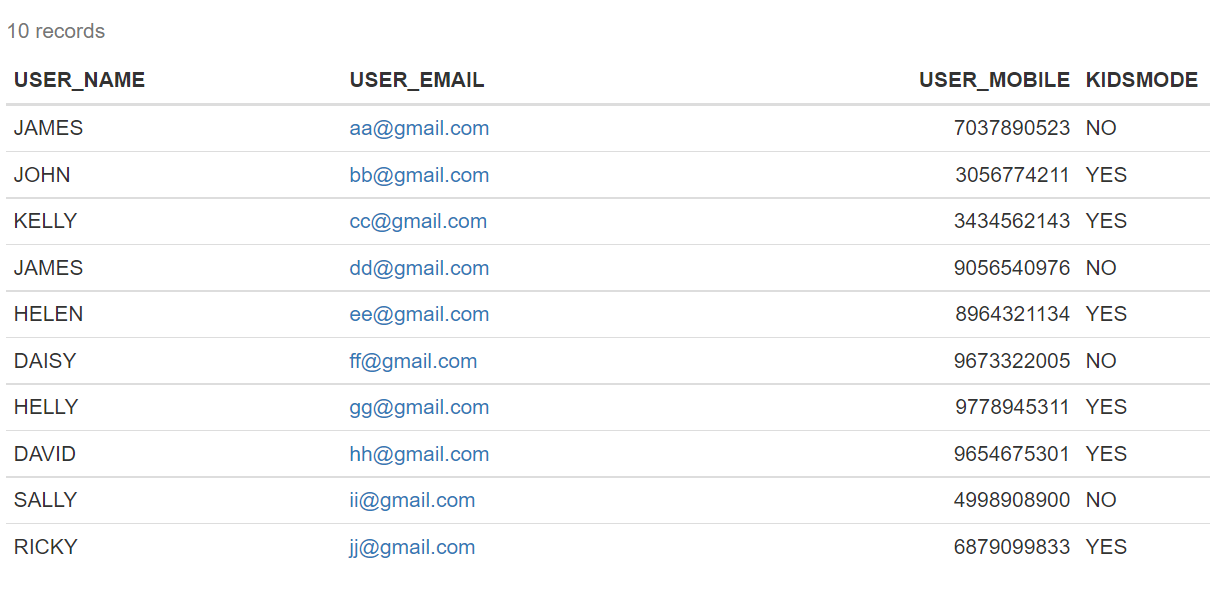
**Report 7:**

For marketing purposes, it important to have an idea about the books with rating less than three and not rated books. So that these books can be focused upon and find ways to improve their rating by ratings received as feedback and for newly released books marketing team can promote them.



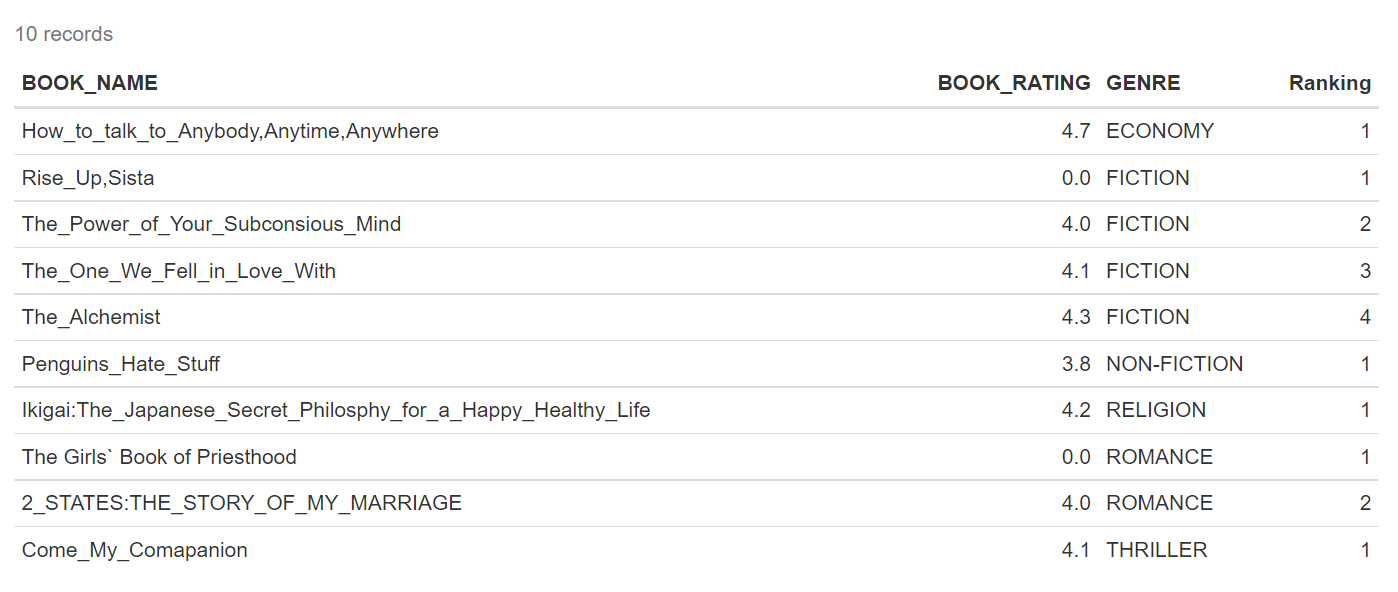
**Report 8:**

The admin can view user who have opted for kids’ mode option without invading the privacy of users. As kids’ mode attribute is present in Login entity which may hinder the privacy of user if access directly as it contains password and passcode of users. Therefore, Kids mode is accessed by view as one table.



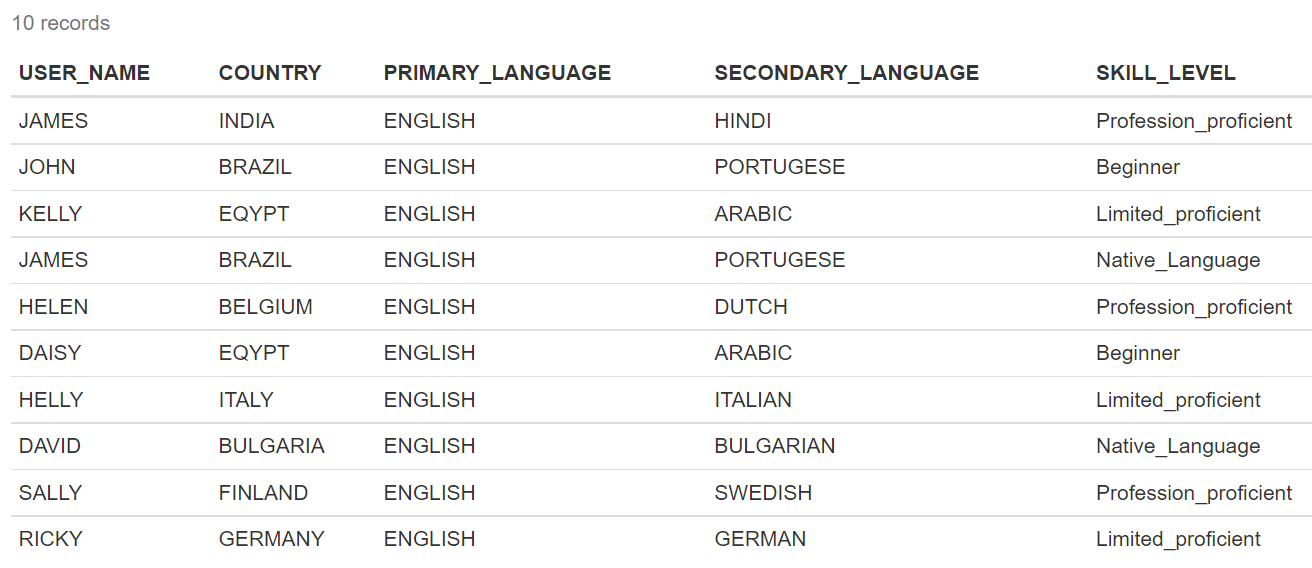
**Report 9:**

Book is ranked according to rating along with type of genre. In below instance rank clause is used to see the ranking of books which is partition by genre for each and every book.



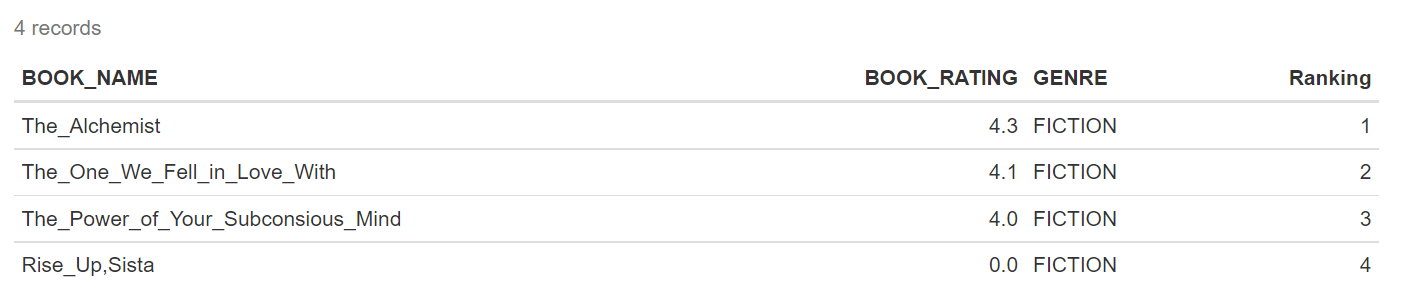
**Report 10:**

Admin can view user information based on country, proficiency level in language along with language used by them to access the application. This will give analytical insights about the targeted audience of the application.



**Report 11:**

**User can view ranking of books according to specific genre. In below instance, ranking is performed just on “Fiction” genre.**

****

**Project Wrap-up and Future Considerations**

Storytel application is a good idea for spreading knowledge and making it entertaining for end users. The individual project helps me to understand the challenge to design entities relationship design which could be user friendly and domain expertise was important aspect to design the Storytel application. It was a challenge to build up queries which would even work for future data which was the major challenge.