University of British Columbia

Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: October 20th 2023 Group Number: 23

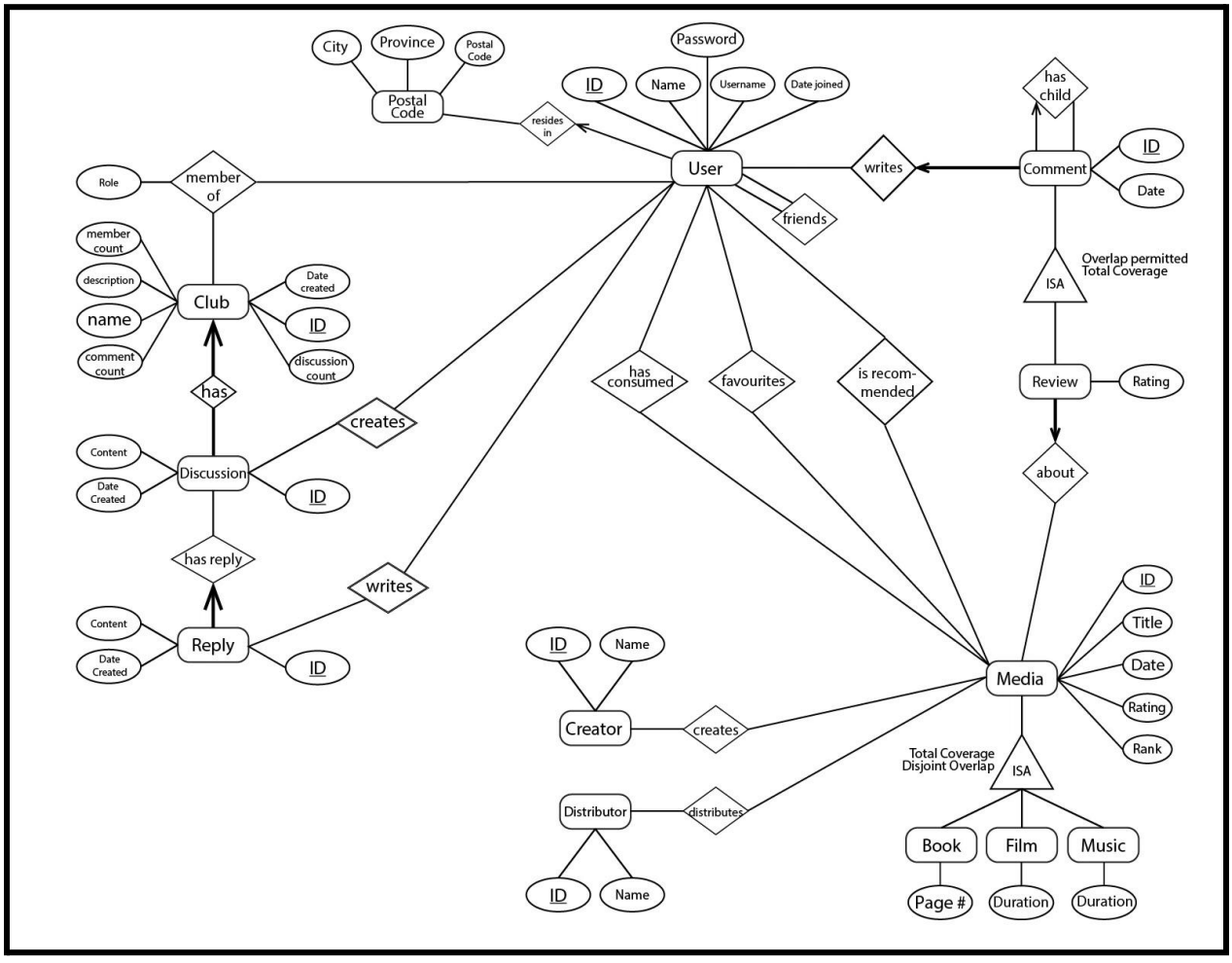
Name	Student #	CS Alias	E-mail Address
Samuel Meester	71841639	j9d4c	sam@meester.xyz
Sean Lin	79391983	g5y8v	seannlinn@gmail.com
Bryan Seo	13749536	w0i3b	bbokyu23@gmail.com

Project Description

The goal of our application is to create a social media platform where users can track various types of media such as books, movies, or music that they have consumed. It also allows users to view other users' libraries, friend them, and receive recommendations for new media across categories (e.g., users who listened to this music also read this book). It will be similar to platforms like GoodReads or MyAnimeList, but different in that it allows more than one type of media. The application will have basic features like rating, reviewing, and posting discussions on any topic of the user's choice.

Updated ER Diagram

Note: This ER Diagram has been updated to reflect the entities and functional dependencies added for normalization below.



Relational Schema + FDs

Underline: Primary Key
Bold: Foreign Key
Highlight: Candidate Key

1. User(ID: RAW(16), Name: VARCHAR2, Username: VARCHAR2, Date: DATETIME, Birthday:DATETIME, Postal_Code:VARCHAR, City: VARCHAR, Province: VARCHAR)
    - ID→Name, Username, Date, Birthday, Postal_Code, City, Province
    - Username→ID, Name, Date, Birthday, Postal_Code, City, Province
    - Postal_Code → City, Province (non-key FD)

2. Friends(**user_one_ID**: RAW(16), **user_two_ID**: RAW(16), StartDate: DATETIME)
    - user_one_ID, user_two_ID→StartDate

3. has_consumed(**user_ID**: RAW(16), **media_ID**: Integer)

4. Favourites(**user_ID**: RAW(16), **media_ID**: Integer)

5. is_recommended(**user_ID**: RAW(16), **media_ID**: Integer)

6. Comment(ID: Integer, *Date*: DATETIME, ***author_ID***: RAW(16), *Content*: VARCHAR2, Rating: Float, **media_ID**: Integer):
    - ID→Date, author_ID, Content, Rating, media_ID

7. has_child(**parent_ID:** Integer, **child_ID**: Integer)

8. Media(ID: Integer, Book_ID: Integer, Film_ID: Integer, Music_ID: Integer)
    - ID → Book_ID, Film_ID, Music_ID

9. Book(ID: Integer, Title: VARCHAR2, Date:DATETIME, Rating: Float, Ranking: Integer, Page_Count: Integer, ISBN: Integer)
    - ID→Title, Date, Rating, ISBN, Ranking, Page_Count
    - ISBN→ID, Title, Date, Rating, Ranking, Page_Count

- Rating→Ranking (non-key FD)

10. Film(ID: Integer, Title: VARCHAR2, Date: DATETIME, Rating: Float, Duration: DATETIME, Ranking: Integer)
    - ID→Title, Date, Rating, Ranking
    - Rating→Ranking (non-key FD)

11. Music(ID: Integer, Title: VARCHAR2, Date: DATETIME, Rating: Float, Album: VARCHAR2, Duration: DATETIME, Ranking: Integer)
    - ID→Title, Date, Album, Rating, Ranking
    - Rating→Ranking (non-key FD)

12. Creator(ID: Integer, Name: VARCHAR2)
    - ID→Name

13. creates(**Media_ID**: Integer, **Creator_ID**: Integer)

14. Distributor(ID: Integer, Name: VARCHAR)
    - ID→Name

15. distributes(**Media_ID**: Integer, **Distributor_ID**: Integer)

16. Club(ID: Integer, Name: VARCHAR2, Date_Created: DATETIME, Description: VARCHAR2, Club_Popularity: Integer, Club_Activity: Integer, Member_Count: Integer, Comment_Count: Integer, Discussion_Count: Integer)
    - ID→Name, Date_Created, Description, Club_Popularity, Member_Count, Comment_Count, Discussion_Count
    - Name→ID, Date_Created, Description, Club_Popularity, Member_Count, Comment_Count, Discussion_Count
    - Comment_Count, Discussion_Count → Club_Activity (non-key FD)
    - Member_Count → Club_Popularity (non-key FD)

17. member_of(**Club_ID:** Integer, **User_ID:** RAW(16), Role: Integer)

18. Discussions(ID: Integer, Title: VARCHAR2, Date: DATETIME, Content: VARCHAR2, **Club_ID:** Integer, **Author_ID**: RAW(16))
    - ID→Date, Content, Club_ID, Author_ID

19. Reply(ID: Integer, Date: DATETIME, Content: VARCHAR2, **Discussion_ID:** Integer, **Author_ID**: RAW(16))
    - ID→Date, Content, Discussion_ID

Normalization to BCNF

User Decomposition to User_1, User_2

User(ID: RAW(16), Name: VARCHAR2, Username: VARCHAR2, Date: DATETIME, Birthday:DATETIME, Postal_Code:VARCHAR, City: VARCHAR, Province: VARCHAR)

Decompose on: Postal_Code →City, Province

1. User_1(ID: RAW(16), Name: VARCHAR2, Username: VARCHAR2, Date: DATETIME, Birthday:DATETIME, **Postal_Code**:VARCHAR)
2. User_2(Postal_Code: VARCHAR, City: VARCHAR, Province: VARCHAR)

   Note: In DDL statements, these tables are referred to as "User" and "Postal_Code" respectively.

Club Decomposition to Club_2, Club_3, Club_4

Club(ID: Integer, Name: VARCHAR2, Date_Created: DATETIME, Description: VARCHAR2, Club_Popularity: Integer, Club_Activity: Integer, Member_Count: Integer, Comment_Count: Integer, Discussion_Count: Integer)

Decompose on: Comment_Count, Discussion_Count → Club_Activity

1. Club_1(ID: Integer, Name: VARCHAR2, Date_Created: DATETIME, Description: VARCHAR2, Club_Popularity: Integer, Member_Count: Integer, **Comment_Count**: Integer, **Discussion_Count**: Integer)

   Decompose on: Member_Count → Club_Popularity

   a. Club_3(ID: Integer, Name: VARCHAR2, Date_Created: DATETIME, Description: VARCHAR2, **Comment_Count**: Integer, **Discussion_Count**: Integer, **Member_Count**: Integer)
   b. Club_4(Member_Count: Integer, Club_Popularity: Integer)

   c. Club_2(Comment_Count: Integer, Discussion_Count: Integer, Club_Activity: Integer)

Book Decomposition

Book(ID: Integer, Title: VARCHAR2, Date:DATETIME, ISBN: Integer, Rating: Float, Ranking: Integer, Page_Count: Integer)

Decompose on Rating → Ranking

1. Book(ID: Integer, Title: VARCHAR2, Date:DATETIME, ISBN: Integer, **Rating**: Float, Page_Count: Integer)
2. Book(Rating: Float, Ranking: Integer)

Film Decomposition

Film(ID: Integer, Title: VARCHAR2, Date: DATETIME, Rating: Float, Duration: DATETIME, Ranking: Integer)

Decompose on Rating → Ranking

1. Film(ID: Integer, Title: VARCHAR2, Date: DATETIME, **Rating**: Float, Duration: DATETIME)
2. Film(Rating: Float, Ranking: Integer)

Music Decomposition

Music(ID: Integer, Title: VARCHAR2, Date: DATETIME, Rating: Float, Album: VARCHAR2, Duration: DATETIME, Ranking: Integer)

Decompose on Rating → Ranking

1. Music(ID: Integer, Title: VARCHAR2, Date: DATETIME, **Rating**: Float, Album: VARCHAR2, Duration: DATETIME)
2. Music(Rating: Float, Ranking: Integer)

Note: Original relation of Book, Music, and Film violates BCNF due to FD: Rating → Ranking. However, since Rating is floating-point value, the FD will result in minimal duplications. In the implementation, we will explore options of decomposing it, or leaving as is. We have also broken up the media ISA relationship into three tables, Book, Film, and Music, to reflect concrete table inheritance. The Media table allows

comments to reference a “media” id which is then translated to a Book, Film, or Music ID without having to specify which sub-table the lookup should occur in.

SQL DDL

```
CREATE TABLE "Postal_Code" (
      "postal_code" varchar(6) PRIMARY KEY,
      "city" varchar(255) NOT NULL,
      "province" varchar(255) NOT NULL
);

CREATE TABLE "User" (
      "id" RAW(16) DEFAULT SYS_GUID() PRIMARY KEY,
      "username" varchar(255) UNIQUE NOT NULL,
      "name" varchar(255) NOT NULL,
      "date" TIMESTAMP,
      "password" varchar(255) NOT NULL,
      "postal_code" varchar(6),
      CONSTRAINT "postal_code_fk" FOREIGN KEY ("postal_code")
REFERENCES "Postal_Code"("postal_code") ON DELETE SET NULL
);


CREATE TABLE "Book_Ranking" (
      "rating" FLOAT PRIMARY KEY,
      "ranking" INTEGER
);

CREATE TABLE "Film_Ranking" (
      "rating" FLOAT PRIMARY KEY,
      "ranking" INTEGER
);

CREATE TABLE "Music_Ranking" (
      "rating" FLOAT PRIMARY KEY,
      "ranking" INTEGER
);

CREATE TABLE "Book" (
      "id" NUMBER GENERATED by default on null as IDENTITY PRIMARY
KEY,
      "title" varchar(255) NOT NULL,
      "date" date,
      "rating" FLOAT,
      "isbn" integer,
      "page_length" integer,
      CONSTRAINT "book_ranking_fk0" FOREIGN KEY ("rating") REFERENCES
"Book_Ranking"("rating") ON DELETE CASCADE
```

```
);

CREATE TABLE "Film" (
     "id" NUMBER GENERATED by default on null as IDENTITY PRIMARY
KEY,
     "title" varchar(255) NOT NULL,
     "date" date,
     "rating" FLOAT,
     "duration" integer,
     CONSTRAINT "film_ranking_fk0" FOREIGN KEY ("rating") REFERENCES
"Film_Ranking"("rating") ON DELETE CASCADE
);

CREATE TABLE "Music" (
     "id" NUMBER GENERATED by default on null as IDENTITY PRIMARY
KEY,
     "title" varchar(255) NOT NULL,
     "date" date,
     "rating" FLOAT,
     "album" varchar(255),
     "duration" integer,
     CONSTRAINT "music_ranking_fk0" FOREIGN KEY ("rating")
REFERENCES "Music_Ranking"("rating") ON DELETE CASCADE
);

CREATE TABLE "Media" (
     "id" NUMBER GENERATED by default on null as IDENTITY PRIMARY
KEY,
     "book_id" integer,
     "film_id" integer,
     "music_id" integer,
     CONSTRAINT "Media_fk_book" FOREIGN KEY ("book_id") REFERENCES
"Book"("id") ON DELETE CASCADE,
     CONSTRAINT "Media_fk_film" FOREIGN KEY ("film_id") REFERENCES
"Film"("id") ON DELETE CASCADE,
     CONSTRAINT "Media_fk_music" FOREIGN KEY ("music_id") REFERENCES
"Music"("id") ON DELETE CASCADE,
     CONSTRAINT "Media_key_unique" CHECK (     ("book_id" is NOT
NULL and "film_id" is NULL and "music_id" is NULL)
                                          or  ("book_id" is NULL and
"film_id" is NOT NULL and "music_id" is NULL)
                                          or  ("book_id" is NULL and
"film_id" is NULL and "music_id" is NOT NULL))
);
```

```
/*Comments*/

CREATE TABLE "Comment" (
     "id" NUMBER GENERATED by default on null as IDENTITY PRIMARY
KEY,
     "date" TIMESTAMP,
     "rating" integer,
     "author_id" RAW(16),
     "media_id" integer,
     CONSTRAINT "Comment_author_fk" FOREIGN KEY ("author_id")
REFERENCES "User"("id") ON DELETE CASCADE,
     CONSTRAINT "Comment_media_fk" FOREIGN KEY ("media_id")
REFERENCES "Media"("id") ON DELETE CASCADE
);

CREATE TABLE "has_child" (
     "parent_id" integer NOT NULL,
     "child_id" integer NOT NULL,
     CONSTRAINT "has_child_parent_fk" FOREIGN KEY ("parent_id")
REFERENCES "Comment"("id") ON DELETE CASCADE,
     CONSTRAINT "has_child_child_fk" FOREIGN KEY ("child_id")
REFERENCES "Comment"("id") ON DELETE CASCADE,
     CONSTRAINT "recursive_child" CHECK ("parent_id" != "child_id")
);

/*User*/

CREATE TABLE "favourites" (
     "user_id" RAW(16),
     "media_id" integer,
     CONSTRAINT "favourites_user_fk" FOREIGN KEY ("user_id")
REFERENCES "User"("id") ON DELETE CASCADE,
     CONSTRAINT "favourites_media_fk" FOREIGN KEY ("media_id")
REFERENCES "Media"("id") ON DELETE CASCADE
);

CREATE TABLE "is_recommended" (
     "user_id" RAW(16),
     "media_id" integer,
     CONSTRAINT "is_recommended_user_fk" FOREIGN KEY ("user_id")
REFERENCES "User"("id") ON DELETE CASCADE,
     CONSTRAINT "is_recommended_media_fk" FOREIGN KEY ("media_id")
REFERENCES "Media"("id") ON DELETE CASCADE
);
```

```
CREATE TABLE "has_consumed" (
     "user_id" RAW(16),
     "media_id" integer,
     CONSTRAINT "has_consumed_user_fk" FOREIGN KEY ("user_id")
REFERENCES "User"("id") ON DELETE CASCADE,
     CONSTRAINT "has_consumed_media_fk" FOREIGN KEY ("media_id")
REFERENCES "Media"("id") ON DELETE CASCADE
);

/*Media*/

CREATE TABLE "Creator" (
     "id" NUMBER GENERATED by default on null as IDENTITY PRIMARY
KEY,
     "name" varchar(255) NOT NULL,
     "birthdate" date
);

CREATE TABLE "creates" (
     "media_id" integer,
     "creator_id" integer,
     CONSTRAINT "Creates_fk0" FOREIGN KEY ("media_id") REFERENCES
"Media"("id") ON DELETE CASCADE,
     CONSTRAINT "Creates_fk1" FOREIGN KEY ("creator_id") REFERENCES
"Creator"("id") ON DELETE CASCADE
);

CREATE TABLE "Distributor" (
     "id" NUMBER GENERATED by default on null as IDENTITY PRIMARY
KEY,
     "name" varchar(255)

);

CREATE TABLE "distributes" (
     "media_id" integer,
     "distributor_id" integer,
     CONSTRAINT "Distributes_fk0" FOREIGN KEY ("media_id")
REFERENCES "Media"("id") ON DELETE CASCADE,
     CONSTRAINT "Distributes_fk1" FOREIGN KEY ("distributor_id")
REFERENCES "Distributor"("id") ON DELETE CASCADE
);

/*Clubs*/
```

```
CREATE TABLE "Activity" (
      "comment_count" integer,
      "discussion_count" integer,
      "activity" integer,
    CONSTRAINT "Activity_pk" PRIMARY KEY ("comment_count",
"discussion_count")
);

CREATE TABLE "Popularity" (
      "member_count" integer PRIMARY KEY,
      "popularity" integer
);

CREATE TABLE "Club" (
      "id" NUMBER GENERATED by default on null as IDENTITY PRIMARY
KEY,
      "date" TIMESTAMP,
      "name" varchar(255) UNIQUE NOT NULL,
      "description" varchar(255),
      "comment_count" integer,
      "discussion_count" integer,
      "member_count" integer,
      CONSTRAINT "club_fk0" FOREIGN KEY ("comment_count",
"discussion_count") REFERENCES "Activity"("comment_count",
"discussion_count") ON DELETE CASCADE,
      CONSTRAINT "club_fk1" FOREIGN KEY ("member_count") REFERENCES
"Popularity"("member_count") ON DELETE CASCADE
);

CREATE TABLE "member_of" (
      "user_id" RAW(16),
      "club_id" integer,
      "role" integer,
      CONSTRAINT "member_of_user_fk" FOREIGN KEY ("user_id")
REFERENCES "User"("id") ON DELETE CASCADE,
      CONSTRAINT "member_of_club_fk" FOREIGN KEY ("club_id")
REFERENCES "Club"("id") ON DELETE CASCADE
);

CREATE TABLE "Discussion" (
      "id" NUMBER GENERATED by default on null as IDENTITY PRIMARY
KEY,
      "date" TIMESTAMP,
      "club_id" integer,
      "author_id" RAW(16),
```

```
     "content" varchar(255) NOT NULL,
     CONSTRAINT "Discussion_club_fk" FOREIGN KEY ("club_id")
REFERENCES "Club"("id") ON DELETE CASCADE,
     CONSTRAINT "Discussion_author_fk" FOREIGN KEY ("author_id")
REFERENCES "User"("id") ON DELETE CASCADE
);

CREATE TABLE "Reply" (
     "id" NUMBER GENERATED by default on null as IDENTITY PRIMARY
KEY,
     "date" TIMESTAMP,
     "discussion_id" integer,
     "author_id" RAW(16),
     "content" varchar(255) NOT NULL,
     CONSTRAINT "Reply_discusison_fk" FOREIGN KEY ("discussion_id")
REFERENCES "Discussion"("id") ON DELETE CASCADE,
     CONSTRAINT "Reply_author_fk" FOREIGN KEY ("author_id")
REFERENCES "User"("id") ON DELETE CASCADE
);
```

Insert Statements

```
INSERT INTO "User" VALUES(NULL, 'abc', 'John', '2023-10-20 00:00:01',
'cba', 'A1A1A1');
INSERT INTO "User" VALUES(NULL, 'bcd', 'Ron', '2023-10-20 00:00:02',
'dcb', 'A1A1A1');
INSERT INTO "User" VALUES(NULL, 'cde', 'Bon', '2023-10-20 00:00:03',
'edc', 'A1A1A1');
INSERT INTO "User" VALUES(NULL, 'def', 'Jeff', '2023-10-20 00:00:04',
'fed', 'A1A1A1');
INSERT INTO "User" VALUES(NULL, 'efg', 'Jerry', '2023-10-20
00:00:05', 'gfe', 'A1A1A1');
```

Note: User ID is null, generated on insert.

```
INSERT INTO "favourites" VALUES(4256, 809);
INSERT INTO "favourites" VALUES(2345, 345);
INSERT INTO "favourites" VALUES(3465, 475);
INSERT INTO "favourites" VALUES(1754, 264);
INSERT INTO "favourites" VALUES(2955, 163);

INSERT INTO "is_recommended" VALUES(4256, 809);
INSERT INTO "is_recommended" VALUES(4256, 456);
INSERT INTO "is_recommended" VALUES(4256, 264);
INSERT INTO "is_recommended" VALUES(4256, 724);
INSERT INTO "is_recommended" VALUES(4256, 385);


INSERT INTO "has_consumed" VALUES(4256, 957);
INSERT INTO "has_consumed" VALUES(4256, 475);
INSERT INTO "has_consumed" VALUES(4256, 368);
INSERT INTO "has_consumed" VALUES(4256, 308);
INSERT INTO "has_consumed" VALUES(4256, 275);


INSERT INTO "Postal_Code" VALUES('A1A1A1', 'Vancouver', 'British
Columbia');
INSERT INTO "Postal_Code" VALUES('A2A2A2', 'Richmond', 'British
Columbia');
INSERT INTO "Postal_Code" VALUES('A3A3A3', 'Burnaby', 'British
Columbia');
INSERT INTO "Postal_Code" VALUES('A4A4A4', 'Coquitlam', 'British
Columbia');
INSERT INTO "Postal_Code" VALUES('A5A5A5', 'Calgary', 'Alberta');
```

```
INSERT INTO “Book_Ranking” VALUES(5.0, 1);
INSERT INTO “Book_Ranking” VALUES(4.0, 2);
INSERT INTO “Book_Ranking” VALUES(3.0, 3);
INSERT INTO “Book_Ranking” VALUES(2.0, 4);
INSERT INTO “Book_Ranking” VALUES(1.0, 5);

INSERT INTO “Film_Ranking” VALUES(5.0, 1);
INSERT INTO “Film_Ranking” VALUES(4.0, 2);
INSERT INTO “Film_Ranking” VALUES(3.0, 3);
INSERT INTO “Film_Ranking” VALUES(2.0, 4);
INSERT INTO “Film_Ranking” VALUES(1.0, 5);

INSERT INTO “Music_Ranking” VALUES(5.0, 1);
INSERT INTO “Music_Ranking” VALUES(4.0, 2);
INSERT INTO “Music_Ranking” VALUES(3.0, 3);
INSERT INTO “Music_Ranking” VALUES(2.0, 4);
INSERT INTO “Music_Ranking” VALUES(1.0, 5);

INSERT INTO “Book” VALUES(NULL, ‘Defend the Past’, ‘2023-10-10’, 5.0,
1234567890123, 333);
INSERT INTO “Book” VALUES(NULL, ‘The Horizon of Avalon’,
‘2023-09-10’, 4.0, 1234567890124, 345);
INSERT INTO “Book” VALUES(NULL, ‘The Shadows in the Sea’,
‘2023-08-10’, 3.0, 1234567890125, 256);
INSERT INTO “Book” VALUES(NULL, ‘Cage the Truth’, ‘2023-07-10’, 2.0,
1234567890126, 367);
INSERT INTO “Book” VALUES(NULL, ‘Winged Eden’, ‘2023-06-10’, 1.0,
1234567890127, 453);

INSERT INTO “Film” VALUES(NULL, ‘Clone of the Stars’, ‘2023-10-11’,
5.0, 75);
INSERT INTO “Film” VALUES(NULL, ‘Mercenaries from Outer Space’,
‘2023-09-11’, 4.0, 76);
INSERT INTO “Film” VALUES(NULL, ‘Armies and Agents’, ‘2023-08-11’,
3.0, 78);
INSERT INTO “Film” VALUES(NULL, ‘Statues of Alien Life’,
‘2023-07-11’, 2.0, 96);
INSERT INTO “Film” VALUES(NULL, ‘Intelligence in the End’,
‘2023-06-11’, 1.0, 56);

INSERT INTO “Music” VALUES(NULL, ‘Cocktails Lies’, ‘2023-10-12’, 5.0,
‘Home of Out Here’, 200);
INSERT INTO “Music” VALUES(NULL, ‘Hurt by Teenage’, ‘2023-09-12’,
4.0, ‘Love For Power’, 202);
```

```
INSERT INTO “Music” VALUES(NULL, ‘10/10 Reflections’, ‘2023-08-12’,
3.0, ‘Pure Commission’, 205);
INSERT INTO “Music” VALUES(NULL, ‘Melancholy Crash’, ‘2023-07-12’,
2.0, ‘A Time of Birds’, 310);
INSERT INTO “Music” VALUES(NULL, ‘Golden White Noise’, ‘2023-06-12’,
1.0, ‘Mozart Destiny’, 30);

INSERT INTO “Media” VALUES(NULL, 12, NULL, NULL);
INSERT INTO “Media” VALUES(NULL, NULL, 34, NULL);
INSERT INTO “Media” VALUES(NULL, NULL, NULL, 2453);
INSERT INTO “Media” VALUES(NULL, NULL, 364, NULL);
INSERT INTO “Media” VALUES(NULL, NULL, NULL, 456);

INSERT INTO “Creator” VALUES(NULL, ‘Jeffery’, 1999-4-12);
INSERT INTO “Creator” VALUES(NULL, ‘Amanda’, 1999-4-30);
INSERT INTO “Creator” VALUES(NULL, ‘Phil’, 1999-4-21);
INSERT INTO “Creator” VALUES(NULL, ‘Dan’, 1999-4-22);
INSERT INTO “Creator” VALUES(NULL, ‘Matt’, 1999-4-15);

INSERT INTO “creates” VALUES(1423, 5623);
INSERT INTO “creates” VALUES(2345, 7465);
INSERT INTO “creates” VALUES(3465, 2475);
INSERT INTO “creates” VALUES(4765, 3856);
INSERT INTO “creates” VALUES(5867, 1543);

INSERT INTO “Distributor” VALUES(NULL, ‘Johnathan’);
INSERT INTO “Distributor” VALUES(NULL, ‘Perry’);
INSERT INTO “Distributor” VALUES(NULL, ‘Chris’);
INSERT INTO “Distributor” VALUES(NULL, ‘Avery’);
INSERT INTO “Distributor” VALUES(NULL, ‘Matthew’);

INSERT INTO “distributes” VALUES(6785, 4675);
INSERT INTO “distributes” VALUES(6354, 5678);
INSERT INTO “distributes” VALUES(3564, 2345);
INSERT INTO “distributes” VALUES(3857, 3568);
INSERT INTO “distributes” VALUES(1234, 9585);


INSERT INTO “Comment” VALUES(NULL, ‘2023-10-19 00:00:01’, 5.0, 2344,
12);
INSERT INTO “Comment” VALUES(NULL, ‘2023-10-19 00:00:02’, 4.0, 3456,
12);
INSERT INTO “Comment” VALUES(NULL, ‘2023-10-19 00:00:03’, 3.0, 6758,
12);
```

```
INSERT INTO “Comment” VALUES(NULL, ‘2023-10-19 00:00:04’, 2.0, 3758,
12);
INSERT INTO “Comment” VALUES(NULL, ‘2023-10-19 00:00:05’, 1.0, 5987,
12);

INSERT INTO “has_child” VALUES(1423, 3645);
INSERT INTO “has_child” VALUES(7456, 2745);
INSERT INTO “has_child” VALUES(1863, 2865);
INSERT INTO “has_child” VALUES(1364, 3769);
INSERT INTO “has_child” VALUES(4573, 4679);

INSERT INTO "Activity" VALUES (217, 13, 3);
INSERT INTO "Activity" VALUES (432, 12, 7);
INSERT INTO "Activity" VALUES (465, 4, 5);
INSERT INTO "Activity" VALUES (3645, 345, 55);
INSERT INTO "Activity" VALUES (25, 6, 4);

INSERT INTO "Club" VALUES(NULL, '2023-10-20 12:00:00', 'Fantasy Novel
Club', 'A club for escapists of reality', 3, 32, 10);
INSERT INTO "Club" VALUES(NULL, '2023-10-21 14:30:00', 'Otaku Club',
'Watch 25 hours of anime every single day', 0, 0, 15);
INSERT INTO "Club" VALUES(NULL, '2023-10-22 18:45:00', 'Anime
Enthusiasts Club', 'Another anime club', 0, 0, 5);
INSERT INTO "Club" VALUES(NULL, '2023-10-23 10:15:00', 'Manga Heart
Heart Club', 'Exploring the world of manga', 5, 3, 20);
INSERT INTO "Club" VALUES(NULL, '2023-10-24 15:20:00', 'Rock Music
Enthusiasts', 'Staying updated on rock music in the 1610s', 2, 4,
30);

INSERT INTO “member_of” VALUES(283476, 8490, 2);
INSERT INTO “member_of” VALUES(283476, 2543, 7);
INSERT INTO “member_of” VALUES(283476, 2345, 5);
INSERT INTO “member_of” VALUES(283476, 7654, 8);
INSERT INTO “member_of” VALUES(283476, 6875, 4);

INSERT INTO “Discussion” VALUES(NULL, ‘2023-10-18 00:00:01’, 123,
3452, ‘hi’)
INSERT INTO “Discussion” VALUES(NULL, ‘2023-10-18 00:00:02’, 234,
3452, ‘bye’)
INSERT INTO “Discussion” VALUES(NULL, ‘2023-10-18 00:00:03’, 345,
3452, ‘yes’)
INSERT INTO “Discussion” VALUES(NULL, ‘2023-10-18 00:00:04’, 456,
3452, ‘no’)
INSERT INTO “Discussion” VALUES(NULL, ‘2023-10-18 00:00:05’, 156,
3452, ‘this’)
```

```
INSERT INTO “Reply” VALUES(NULL, ‘2023-10-17 00:00:01’, 0249387, 589,
‘hello’);
INSERT INTO “Reply” VALUES(NULL, ‘2023-10-17 00:00:02’, 0249389, 589,
‘goodbye’);
INSERT INTO “Reply” VALUES(NULL, ‘2023-10-17 00:00:03’, 0249388, 589,
‘yes yes’);
INSERT INTO “Reply” VALUES(NULL, ‘2023-10-17 00:00:04’, 0249386, 589,
‘no no’);
INSERT INTO “Reply” VALUES(NULL, ‘2023-10-17 00:00:05’, 0249385, 589,
‘that’);
```