CPSC 304 Project Cover Page

Milestone #: 2

Date: Oct. 25, 2021

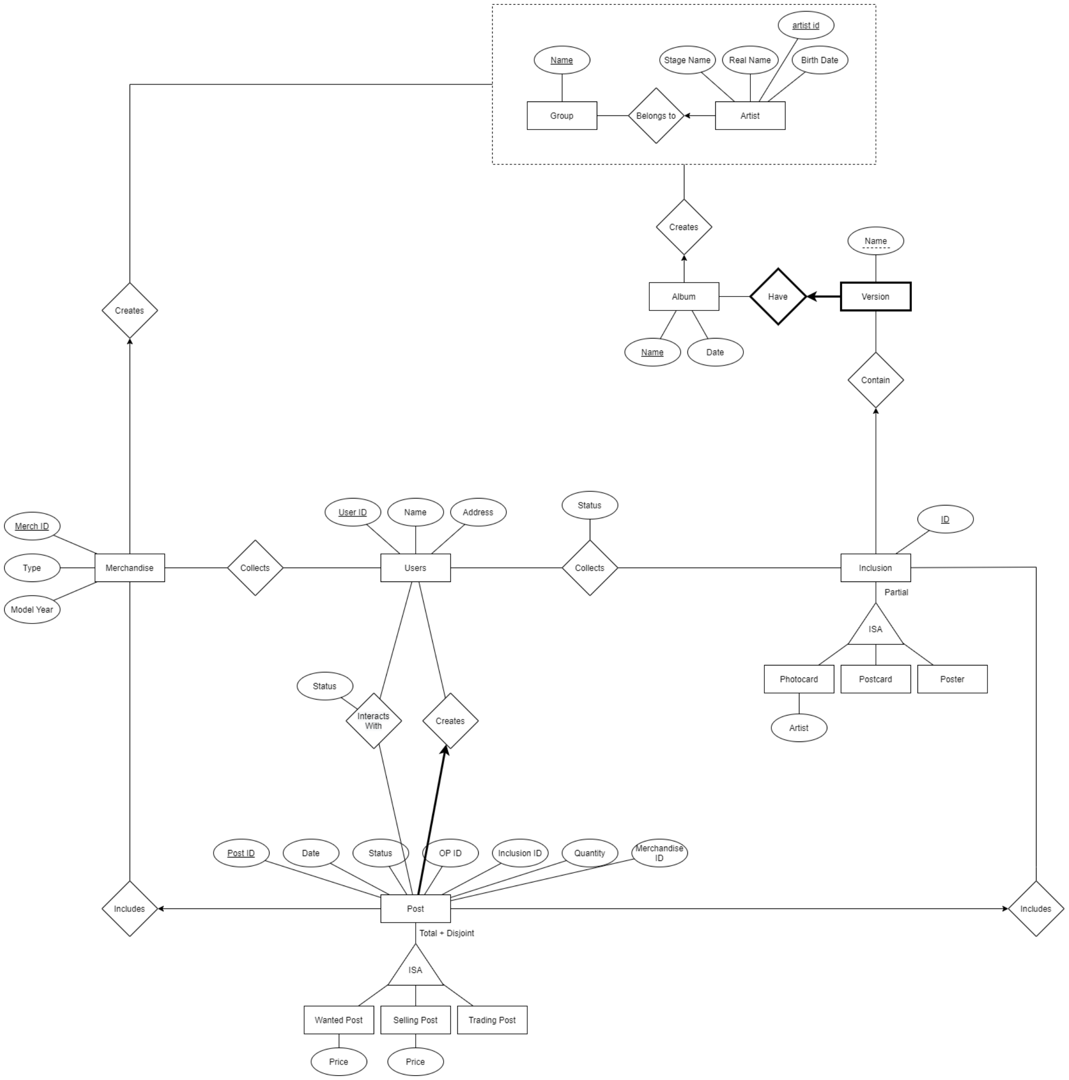
Group Number: 98

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Naomi Chen	52224094	o6v1b	nathanielle.chen@gmail.com
David (Taekyu) Kang	36304153	c2v0b	tk.kang@outlook.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

2. ER Diagram



Changes made:

1. Group Creates Merchandise updated to be one-to-many relationship
    a. Reason: If a user knows the merchandise, they would know the group
2. Group Creates Album updated to be one-to-many relationship
    a. Reason: If a user knew the album, the user would also know which group made it
3. Added Merchandise ID Attribute to Post Entity
    a. Reason: A necessary attribute added
4. Post Includes Merchandise updated to be many-to-one relationship
    a. Reason: Each post will only concern a single merchandise
5. Post Includes Inclusion updated to be many-to-one relationship
    a. Reason: Each post will only concern a single Inclusion
6. Users Create Post updated to be one-to-many relationship with total participation
    a. Reason: Every post must have an OP and if I know the post, I know the OP

3. Relational Schema

KEY: underline = primary key, **bold = foreign key**

1. Artist(artistID:char, **groupName:char**, stageName:char, realName:char, birthDate:date)
2. Group (groupName:char)
3. Album(albumName:char**, groupName: char**, **versionName: char,** dateCreated:date)
4. AlbumVersion(**albumName:char**, versionName:char)
5. Inclusion(inclusionID:int, **albumName:char, versionName:char**)
6. Photocard(**inclusionID:int, artistID:char**)
7. Postcard(**inclusionID:int**)
8. Poster(**inclusionID:int**)
9. Users(userID:int, userName:char, userAddress:char)
10. CollectInclusion(**userID:int, inclusionID**, status)
11. CollectMerchandise(**userID:int, merchID: int**)
12. Merchandise(merchID:int, **groupName: char**, merchType:char, modelYear:date)
13. WantedPost(postID:int, price:int)
14. SellingPost(postID:int, price:int)
15. TradingPost(postID:int)
16. Post(postID:int, **opId:int, inclusionID:int, merchID:int,** date:date, status:char, inclusionCategory:char, quantity:int)

4. Functional Dependencies

1. Group (trivial)
    a. groupName -> groupName
2. Artist
    a. artistId -> groupName, stageName, realName, birthDate
3. Album
    a. albumName -> groupName, dateCreated
4. AlbumVersion (trivial)
    a. albumName, versionName -> albumName, versionName
5. Inclusion
    a. inclusionId -> albumName, versionName
6. Photocard
    a. inclusionId -> artistId
7. Postcard (trivial)
8. Poster(trivial)
9. Users
    a. userId -> userName, userAddress
10. CollectInclusion
    a. userId, inclusionId -> status
11. CollectMerchandise (trivial)
12. Merchandise
    a. merchId -> groupName, merchType, modelYear
13. WantedPost

    a. postId -> price

14. SellingPost
    a. postId -> price
15. TradingPost (trivial)
16. Post
    a. postId -> opId, inclusionID, merchID, date, status, quantity
    b. inclusionId -> inclusionCategory

5. Normalization

Before normalization:

Post(postId:int, **opId:int, inclusionId:int, merchID:int,** date:date, status:char, inclusionCategory:char, quantity:int)

FDs
postId -> opId, inclusionID, merchID, date, status, quantity
inclusionId -> inclusionCategory

This table violates 3NF and BCNF because of inclusionId -> inclusionCategory FD. inclusionID is not a super key and inclusionCategory is not part of a minimal key. Here are the steps to normalize it into BCNF.

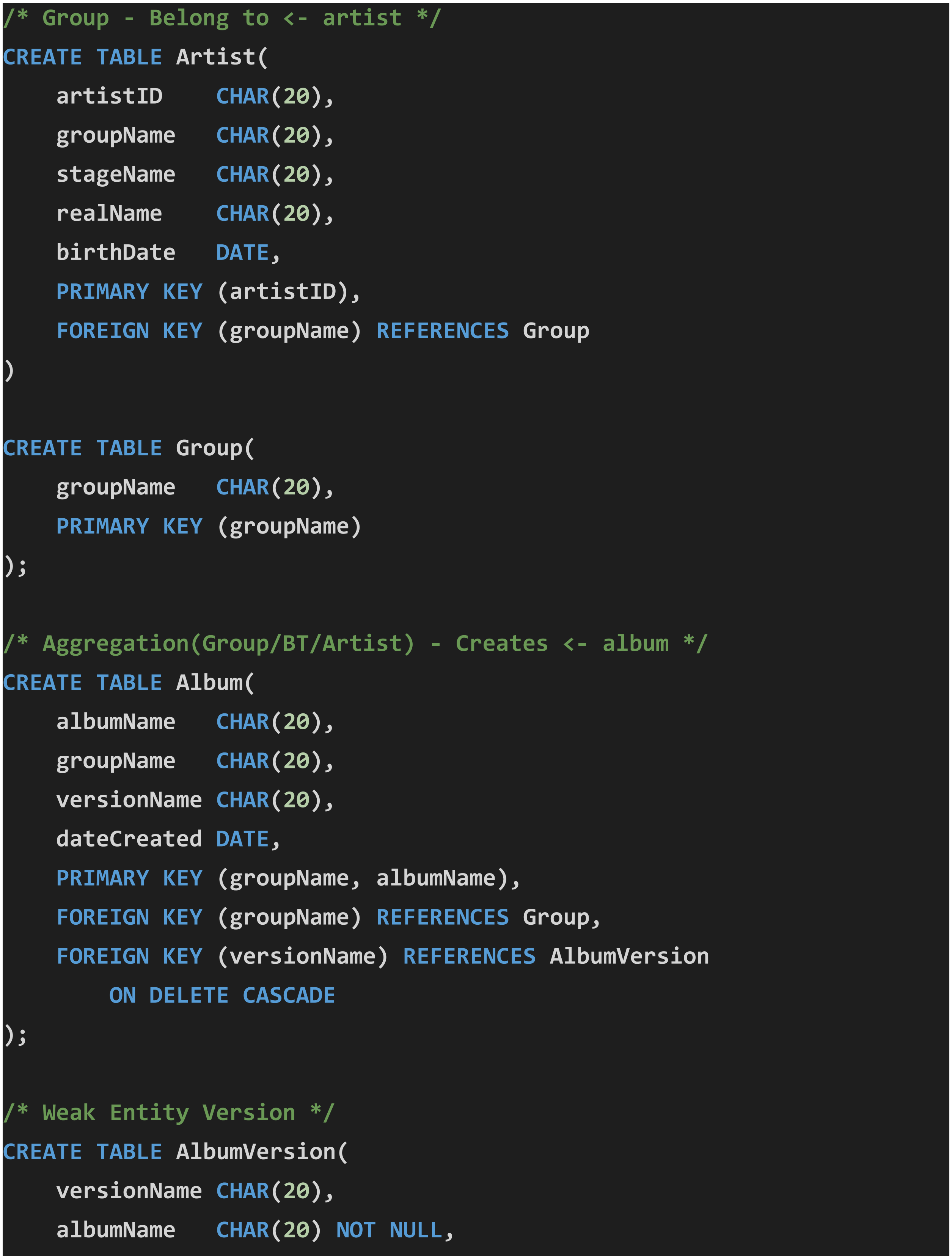
[postId, opId, merchID, date, status, quantity [**[inclusionId]**] inclusionCategory]

R1(postId, opId, merchID, date, status, quantity, inclusionID)
R2(inclusionId, inclusionCategory)

This is now in BCNF. Tables as follows:

Post(postId:int, **opId:int, inclusionId:int, merchID:int,** date:date, status:char, quantity:int)
InclusionCategory(inclusionId:int, inclusionCategory:char)

6. SQL DDL

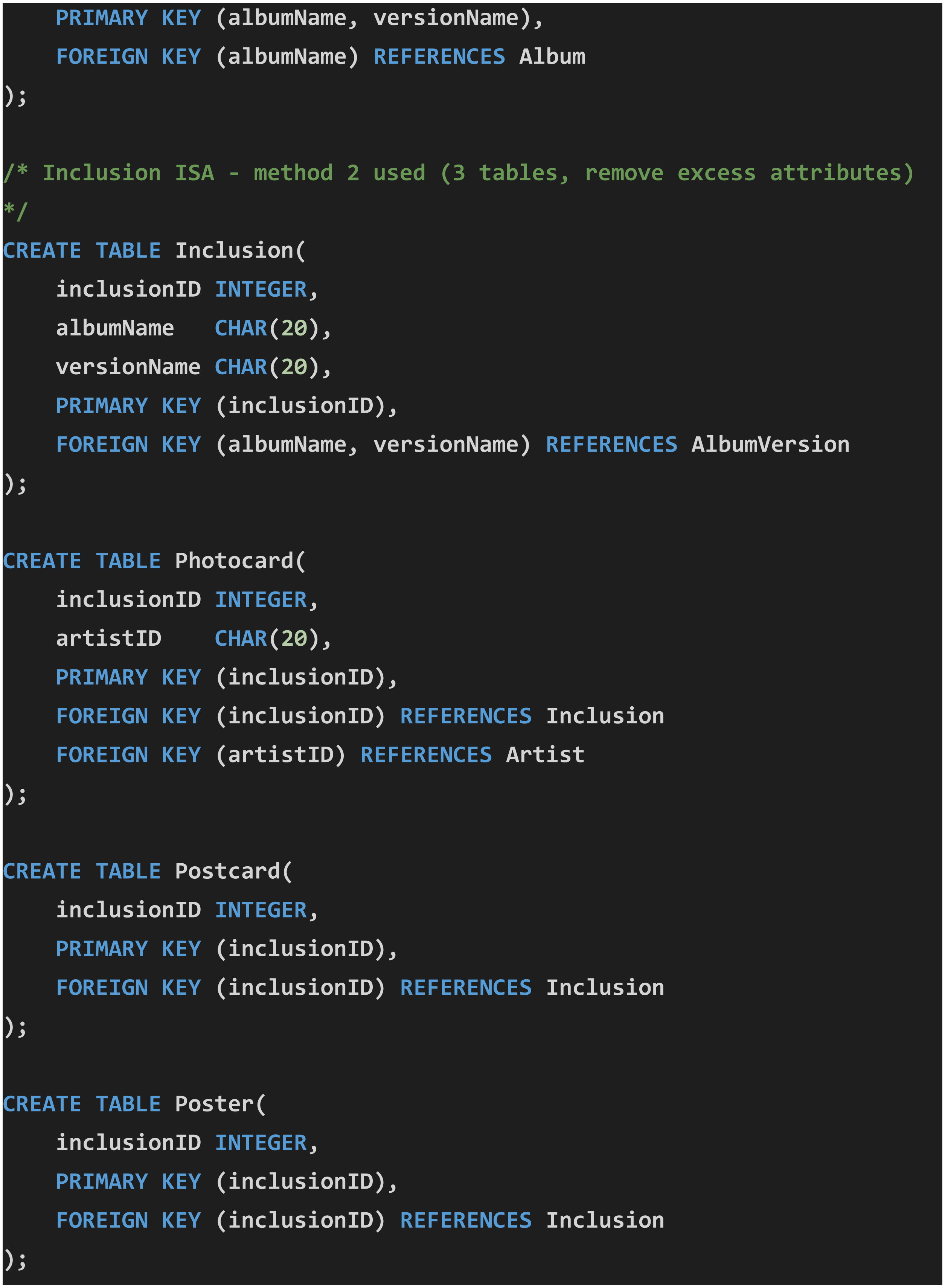


```sql
/* Group - Belong to <- artist */
CREATE TABLE Artist(
    artistID    CHAR(20),
    groupName   CHAR(20),
    stageName   CHAR(20),
    realName    CHAR(20),
    birthDate   DATE,
    PRIMARY KEY (artistID),
    FOREIGN KEY (groupName) REFERENCES Group
)

CREATE TABLE Group(
    groupName   CHAR(20),
    PRIMARY KEY (groupName)
);

/* Aggregation(Group/BT/Artist) - Creates <- album */
CREATE TABLE Album(
    albumName   CHAR(20),
    groupName   CHAR(20),
    versionName CHAR(20),
    dateCreated DATE,
    PRIMARY KEY (groupName, albumName),
    FOREIGN KEY (groupName) REFERENCES Group,
    FOREIGN KEY (versionName) REFERENCES AlbumVersion
        ON DELETE CASCADE
);

/* Weak Entity Version */
CREATE TABLE AlbumVersion(
    versionName CHAR(20),
    albumName   CHAR(20) NOT NULL,
```



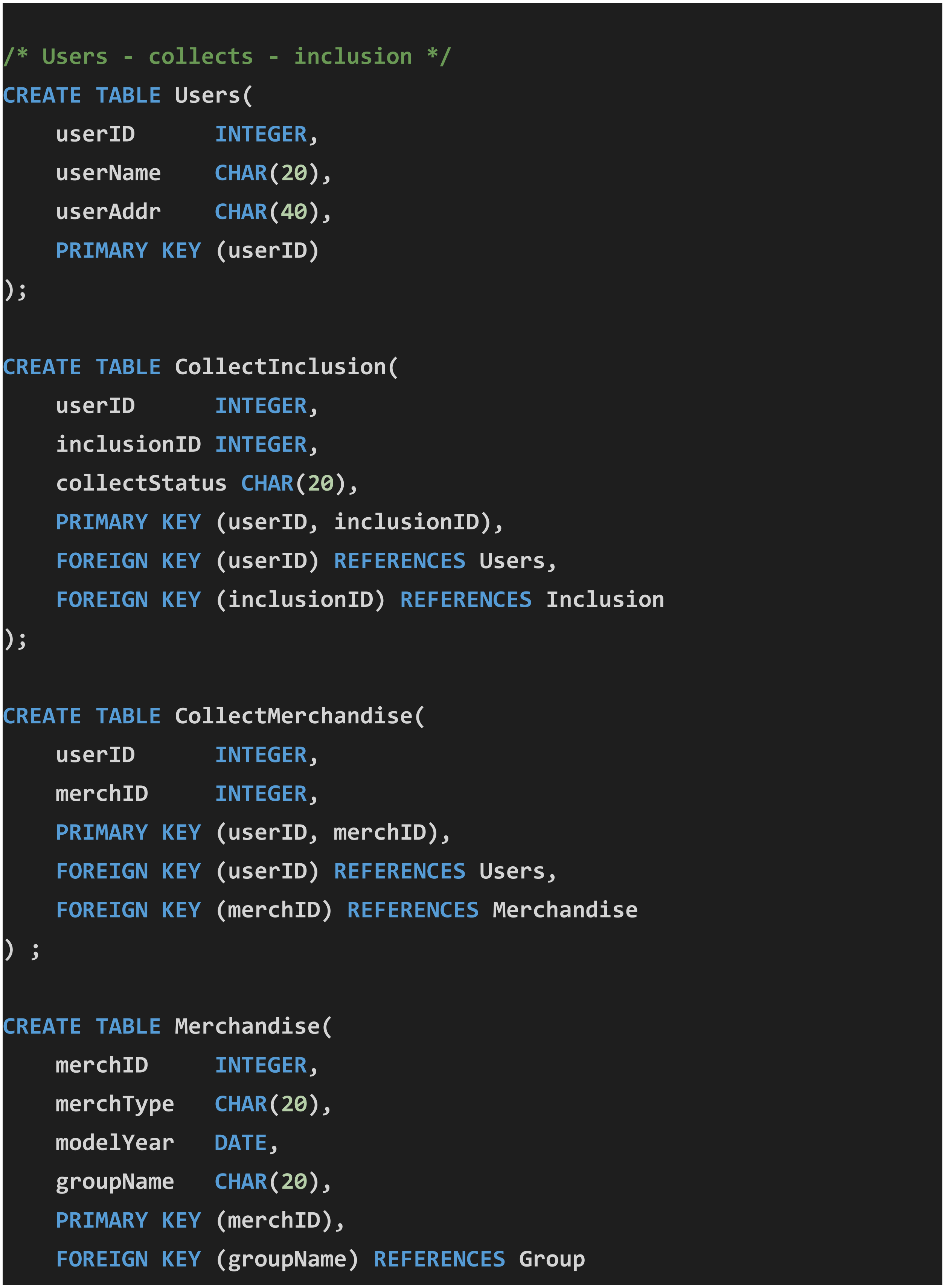
```sql
    PRIMARY KEY (albumName, versionName),
    FOREIGN KEY (albumName) REFERENCES Album
);

/* Inclusion ISA - method 2 used (3 tables, remove excess attributes)
*/
CREATE TABLE Inclusion(
    inclusionID INTEGER,
    albumName   CHAR(20),
    versionName CHAR(20),
    PRIMARY KEY (inclusionID),
    FOREIGN KEY (albumName, versionName) REFERENCES AlbumVersion
);

CREATE TABLE Photocard(
    inclusionID INTEGER,
    artistID    CHAR(20),
    PRIMARY KEY (inclusionID),
    FOREIGN KEY (inclusionID) REFERENCES Inclusion
    FOREIGN KEY (artistID) REFERENCES Artist
);

CREATE TABLE Postcard(
    inclusionID INTEGER,
    PRIMARY KEY (inclusionID),
    FOREIGN KEY (inclusionID) REFERENCES Inclusion
);

CREATE TABLE Poster(
    inclusionID INTEGER,
    PRIMARY KEY (inclusionID),
    FOREIGN KEY (inclusionID) REFERENCES Inclusion
);
```

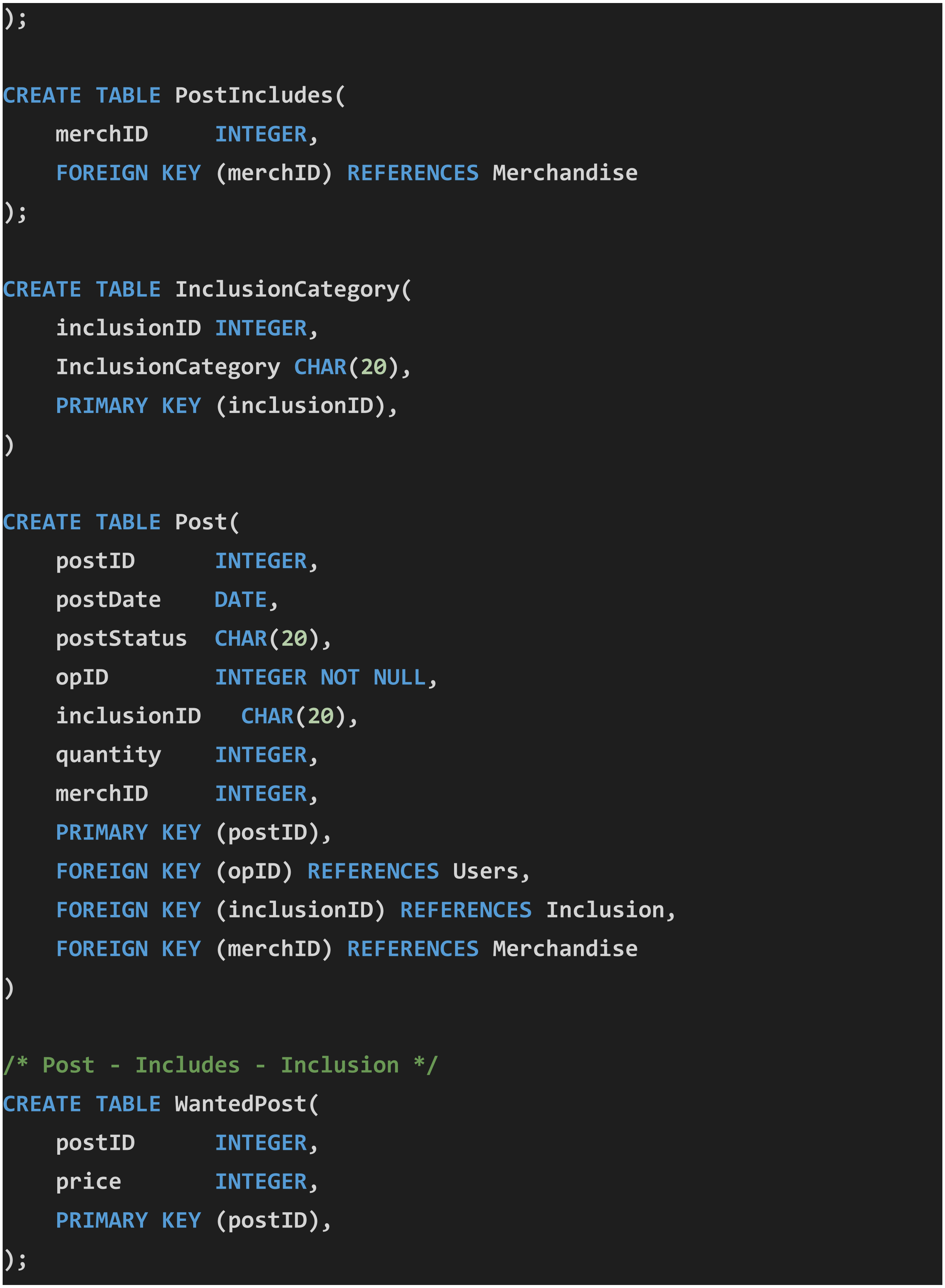


```sql
/* Users - collects - inclusion */
CREATE TABLE Users(
    userID      INTEGER,
    userName    CHAR(20),
    userAddr    CHAR(40),
    PRIMARY KEY (userID)
);

CREATE TABLE CollectInclusion(
    userID      INTEGER,
    inclusionID INTEGER,
    collectStatus CHAR(20),
    PRIMARY KEY (userID, inclusionID),
    FOREIGN KEY (userID) REFERENCES Users,
    FOREIGN KEY (inclusionID) REFERENCES Inclusion
);

CREATE TABLE CollectMerchandise(
    userID      INTEGER,
    merchID     INTEGER,
    PRIMARY KEY (userID, merchID),
    FOREIGN KEY (userID) REFERENCES Users,
    FOREIGN KEY (merchID) REFERENCES Merchandise
) ;

CREATE TABLE Merchandise(
    merchID     INTEGER,
    merchType   CHAR(20),
    modelYear   DATE,
    groupName   CHAR(20),
    PRIMARY KEY (merchID),
    FOREIGN KEY (groupName) REFERENCES Group
```



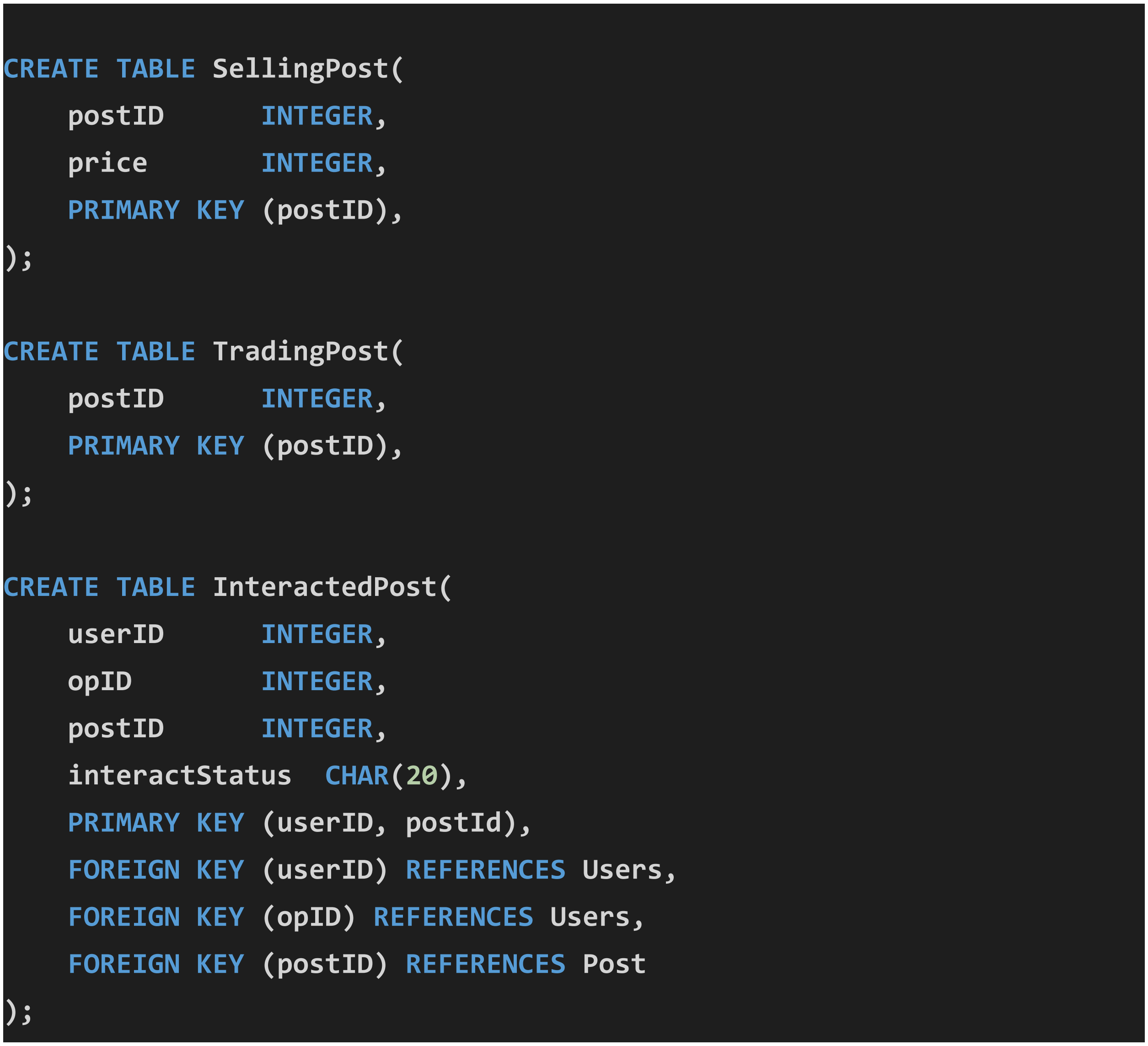
```
);

CREATE TABLE PostIncludes(
    merchID     INTEGER,
    FOREIGN KEY (merchID) REFERENCES Merchandise
);

CREATE TABLE InclusionCategory(
    inclusionID INTEGER,
    InclusionCategory CHAR(20),
    PRIMARY KEY (inclusionID),
)

CREATE TABLE Post(
    postID      INTEGER,
    postDate    DATE,
    postStatus  CHAR(20),
    opID        INTEGER NOT NULL,
    inclusionID   CHAR(20),
    quantity    INTEGER,
    merchID     INTEGER,
    PRIMARY KEY (postID),
    FOREIGN KEY (opID) REFERENCES Users,
    FOREIGN KEY (inclusionID) REFERENCES Inclusion,
    FOREIGN KEY (merchID) REFERENCES Merchandise
)

/* Post - Includes - Inclusion */
CREATE TABLE WantedPost(
    postID      INTEGER,
    price       INTEGER,
    PRIMARY KEY (postID),
);
```

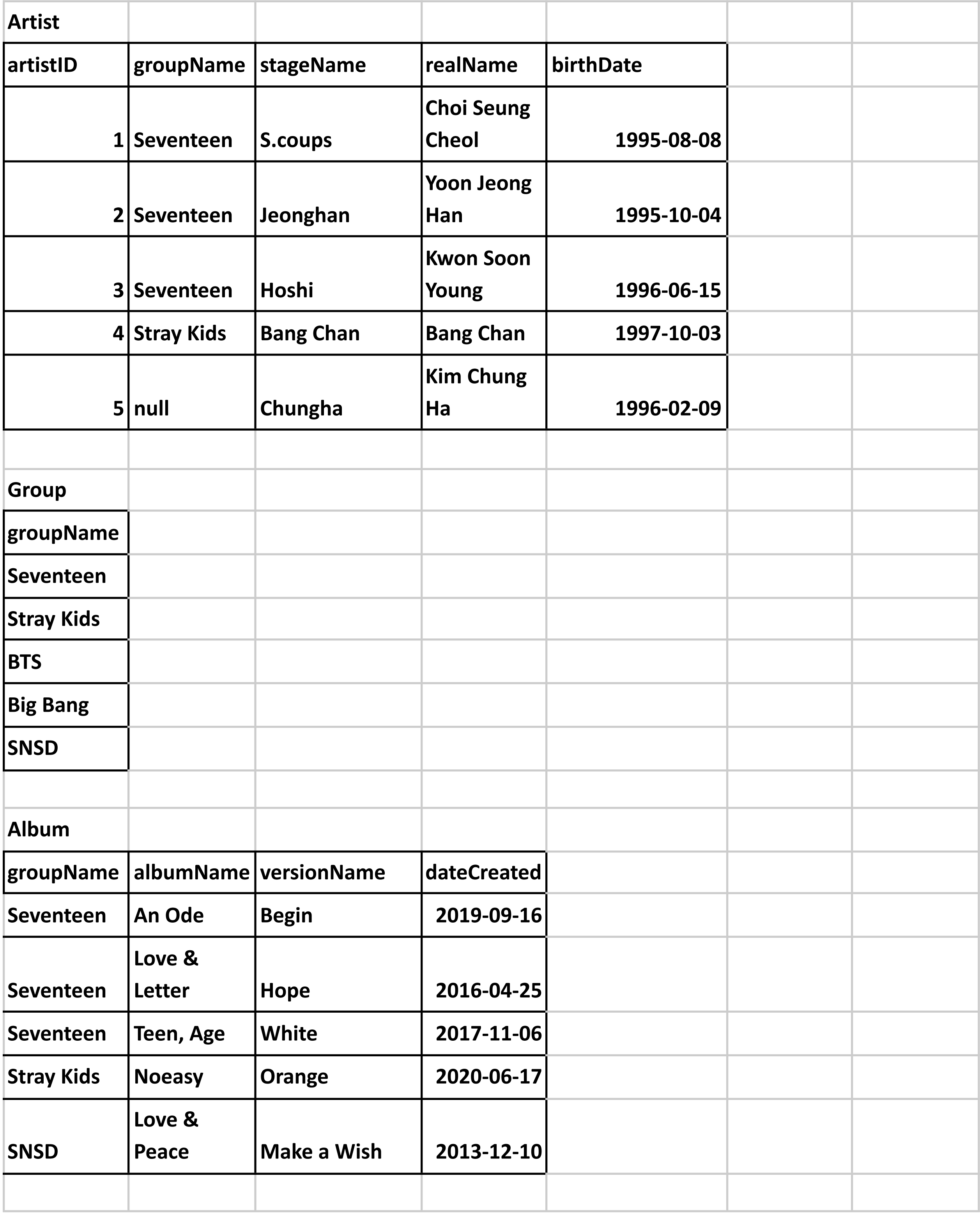


```
CREATE TABLE SellingPost(
    postID      INTEGER,
    price       INTEGER,
    PRIMARY KEY (postID),
);

CREATE TABLE TradingPost(
    postID      INTEGER,
    PRIMARY KEY (postID),
);

CREATE TABLE InteractedPost(
    userID      INTEGER,
    opID        INTEGER,
    postID      INTEGER,
    interactStatus  CHAR(20),
    PRIMARY KEY (userID, postId),
    FOREIGN KEY (userID) REFERENCES Users,
    FOREIGN KEY (opID) REFERENCES Users,
    FOREIGN KEY (postID) REFERENCES Post
);
```

7. Tuples



Artist

artistID	groupName	stageName	realName	birthDate
1	Seventeen	S.coups	Choi Seung Cheol	1995-08-08
2	Seventeen	Jeonghan	Yoon Jeong Han	1995-10-04
3	Seventeen	Hoshi	Kwon Soon Young	1996-06-15
4	Stray Kids	Bang Chan	Bang Chan	1997-10-03
5	null	Chungha	Kim Chung Ha	1996-02-09

Group

groupName
Seventeen
Stray Kids
BTS
Big Bang
SNSD

Album

groupName	albumName	versionName	dateCreated
Seventeen	An Ode	Begin	2019-09-16
Seventeen	Love & Letter	Hope	2016-04-25
Seventeen	Teen, Age	White	2017-11-06
Stray Kids	Noeasy	Orange	2020-06-17
SNSD	Love & Peace	Make a Wish	2013-12-10

AlbumVersion

versionName	albumName
Begin	An Ode
Hope	An Ode
White	Teen, Age
Orange	Teen, Age
Make a Wish	Going Seventeen

Inclusion

inclusionID	albumName	versionName
1	An Ode	Begin
2	An Ode	Hope
3	Teen, Age	White
4	Teen, Age	Orange
5	Going Seventeen	Make a Wish

Photocard

inclusionID	artistID
1	1
2	2
3	3
4	4
5	5

Postcard

inclusionID
10
11

12
13
14

Poster

inclusionID
21
22
23
24
25

Users

userID	userName	userAddr
1	labufanda	772 Bellvue Drive
2	h1histone	123 Handworth Str
3	langdon	23 Hackim Road
4	nomnomnom	132-2 Longdrive Str
5	pandapandapanda	12 Slament Drive

CollectInclusion

userID	inclusionID	collectStatus
1	1	Collected
2	1	Collected
3	1	Wanted
4	2	Wanted
4	5	Wanted

CollectMerchandise

userId	merchID
1	1
2	2
3	3
4	3
5	3

Merchandise

merchID	merchType	groupName	modelYear
1	hoodie	Seventeen	2013
2	bag	Stray Kids	2016
3	tote	BTS	2013
4	bottle	Big Bang	2020
5	bottle	SNSD	2008

PostIncludes

postID	merchID
1	1
2	2
3	2
4	2
5	3

Post

postID	postDate	postStatus	opID	inclusionID	quantity	merchID
1	2021-10-10	Active	1	1	1	-
2	2021-10-17	Active	1	-	1	1

3	2021-10-18	Active	1	-	2	3
4	2021-10-20	Closed	3	2	1	-
5	2021-10-22	Closed	4	-	2	3

InclusionCategory

InclusionID	InclusionCategory
1	Hoodies
2	Hoodies
3	Cups
4	Cups
5	Shirt

WantedPost

postID	price
1	20
2	14
3	23.5
4	12
5	5.5

SellingPost

postID	price
1	20
2	14
3	23.5
4	12
5	5.5

TradingPost

postID

1
2
3
4
5

InteractedPost

userID	opID	postID	interactStatus
1	1	1	Active
2	4	5	Non-Active
3	1	3	Active
4	3	22	Active
5	2	21	Active

8. Queries

We have written insert, delete, update queries only because we have not learned the other query types in class yet.

1. Insertion: Add a user to the system.
2. Insertion: Add a new album to the system
3. Insertion: Add a new group to the system
4. Insertion: Add new merchandise to the system
5. Insertion: Add new inclusions to the system
6. Insertion: Add a new inclusion to a user’s collection
7. Insertion: Create a wanted/selling/trading post
8. Deletion: Delete a wanted/selling/trading post
9. Deletion: Delete something from a user collection
10. Update: Update the status of a wanted/selling/trading post
11. Update: Update the status of a user’s collections
12. Update a user’s information (name or address)