CSC675-02

3/14/22

Final Report

Members: Christopher Su, Konnor Nishimura, Wenye Guo, Jade Simien, and Xinrong Wen

We built our database in VisualStudioCode through SQLite Extension

DDL:

-Tables:

```
CREATE TABLE Users(
    uid INTEGER PRIMARY KEY,
    uname CHAR(20),
    udate DATE,
    uemail CHAR(30),
    uage INTEGER
);

CREATE TABLE Playlists(
    pid INTEGER PRIMARY KEY,
    pname CHAR(30),
    pfollowers INTEGER
);

CREATE TABLE Artists(
    aid INTEGER PRIMARY KEY,
    aname CHAR(20),
    adate DATE,
    afollowers INTEGER
);

CREATE TABLE Songs(
    sid INTEGER PRIMARY KEY,
    sname CHAR(30)
);
```

```sql
CREATE TABLE SongCreates(
    sid INTEGER,
    aid INTEGER,
    sday DATE,
    PRIMARY KEY (sid, aid),
    FOREIGN KEY (sid) REFERENCES Songs(sid),
    FOREIGN KEY (aid) REFERENCES Artists(aid)
);

CREATE TABLE PlayCreates(
    pid INTEGER,
    uid INTEGER,
    pday DATE,
    PRIMARY KEY (pid),
    FOREIGN KEY (pid) REFERENCES Playlists(pid),
    FOREIGN KEY (uid) REFERENCES Users(uid)
);

CREATE TABLE SongFav(
    sid INTEGER,
    uid INTEGER,
    PRIMARY KEY (sid, uid),
    FOREIGN KEY (sid) REFERENCES Songs(sid),
    FOREIGN KEY (uid) REFERENCES Users(uid)
);

CREATE TABLE ArtistFav(
    uid INTEGER,
    aid INTEGER,
    PRIMARY KEY (uid, aid),
    FOREIGN KEY (uid) REFERENCES Users(uid),
    FOREIGN KEY (aid) REFERENCES Artists(aid)
);

CREATE TABLE Contains(
    sid INTEGER,
    pid INTEGER,
    PRIMARY KEY (sid, pid),
    FOREIGN KEY (sid) REFERENCES Songs(sid),
```

```
    FOREIGN KEY (pid) REFERENCES Playlists(pid)
);
```

-Indexes:

```
CREATE INDEX idxage ON Users(uage);

--Check the index in Users Table:
.indices Users

-- Check if the statement is using index or not:
-- Get the name of the users who are older than 22:
EXPLAIN QUERY PLAN
SELECT u.uname, u.uage
FROM Users u
WHERE u.uage > 22;
```

-Views:

```
CREATE VIEW [User name and age] AS
SELECT uname, uage
FROM Users;

CREATE VIEW [Song and Artist] AS
SELECT sname, aname
FROM Songs s, Artists a, SongCreates sc
WHERE sc.sid = s.sid AND sc.aid = a.aid

CREATE VIEW playlist_song_names AS
SELECT pname, sname
FROM Playlists p, Songs s, Contains c
WHERE c.sid = s.sid AND c.pid = p.pid

CREATE VIEW snamerelease AS
SELECT sname, sday
FROM Songs s, SongCreates sc
```

```
WHERE s.sid = sc.sid

-- Show view:
SELECT * FROM [User name and age];
SELECT * FROM [Song and Artist];
SELECT * FROM playlist_song_names;
SELECT * FROM snamerelease;
```

Data to import:

Users

ui	uname	udate	uemail	uage
1000	chris	2000-10-03	chris@mail.com	22
1001	kenny	2010-03-03	kenny@mail.com	21
1002	jade	2014-02-10	jade@mail.com	35
1003	wenye	2019-09-04	wenye@mail.com	45
1004	konnor	2002-05-07	konnor@mail.com	19

```
INSERT INTO Users(uid, uname, udate, uemail, uage)
VALUES (1000, 'chris', '2000-10-03', 'chris@mail.com', 22),
       (1001, 'kenny', '2010-03-03', 'kenny@mail.com', 21),
       (1002, 'jade', '2014-02-10', 'jade@mail.com', 35),
       (1003, 'wenye', '2019-09-04', 'wenye@mail.com', 45),
       (1004, 'konnor', '2002-05-07', 'konnor@mail.com', 19);
```

Playlist

pid	pname	pfollowers
1000	pop	26

1002	edm	598
1003	anime	1234
1001	kpop	9876

```
INSERT INTO Playlists(pid, pname, pfollowers)
VALUES (1000, 'pop', 26),
       (1002, 'edm', 598),
       (1003, 'anime', 1234),
       (1001, 'kpop', 9876);
```

Artist

aid	aname	adate	afollowers
1100	keish	2020-03-10	10300
1200	niki	2021-03-15	30222
1300	bts	2014-10-30	300432
1400	maroon5	2004-09-01	1111113

```
INSERT INTO Artists(aid, aname, adate, afollowers)
VALUES (1100, 'keish', '2020-03-10', 10300),
       (1200, 'niki', '2021-03-15', 30222),
       (1300, 'bts', '2014-10-30', 300432),
       (1400, 'maroon5', '2004-09-01', 1111113);
```

Songs

sid	sname
1111	beside you
1112	spring day
1113	less of you
1114	daylight
1115	lowkey

1116	butter
1117	dynamite

```
INSERT INTO Songs(sid, sname)
VALUES (1111, 'beside you'),
       (1112, 'spring day'),
       (1113, 'less of you'),
       (1114, 'daylight'),
       (1115, 'lowkey'),
       (1116, 'butter'),
       (1117, 'dynamite');
```

SongCreates

sid	aid	sday
1111	1100	2020-04-02
1112	1300	2021-12-09
1113	1100	2021-06-03
1114	1400	2022-01-09
1115	1200	2021-04-03
1116	1300	2021-03-06
1117	1300	2021-05-02

```
INSERT INTO SongCreates(sid, aid, sday)
VALUES (1111, 1100, '2020-04-2'),
       (1112, 1300, '2021-12-09'),
       (1113, 1100, '2021-06-03'),
       (1114, 1400, '2022-01-09'),
       (1115, 1200, '2021-04-03'),
       (1116, 1300, '2021-03-06'),
       (1117, 1300, '2021-05-02');
```

PlayCreates

pid	uid	pday
1000	1004	2020-07-25
1002	1003	2021-01-03
1003	1002	2021-03-13

```
INSERT INTO PlayCreates(pid, uid, pday)
VALUES (1000, 1004, '2020-07-25'),
       (1002, 1003, '2021-01-03'),
       (1003, 1002, '2021-03-13');
```

SongFav

sid	uid
1111	1001
1111	1002
1113	1003
1115	1004
1117	1003
1117	1000

```
INSERT INTO SongFav(sid, uid)
VALUES (1111, 1001),
       (1111, 1002),
       (1113, 1003),
       (1115, 1004),
       (1117, 1003),
       (1117, 1000);
```

ArtistFav

uid	aid
1000	1100
1003	1200
1001	1300

1002	1400

```
INSERT INTO ArtistFav(uid, aid)
VALUES (1000, 1100),
       (1003, 1200),
       (1001, 1300),
       (1002, 1400);
```

Contains

sid	pid
1112	1000
1111	1003
1115	1001
1117	1002
1114	1000

```
INSERT INTO Contains(sid, pid)
VALUES (1112, 1000),
       (1111, 1003),
       (1115, 1001),
       (1117, 1002),
       (1114, 1000);
```

-SQL Queries:

```
-- Get each age and the number of people of each age who are 22 or older:
SELECT u.uage, count(*)
FROM Users u
WHERE u.uage >= 22 GROUP BY u.uage;
```

Result:

uage	count(*)
22	1
35	1
45	1

```
-- Get the name of playlists and its number of songs for playlists that have more than 1 songs
SELECT p.pname, count(s.sid)
FROM Playlists p, Contains c, Songs s
WHERE p.pid = c.pid AND c.sid = s.sid
GROUP BY p.pid, p.pname
HAVING count(s.sid) > 1;
```

Result:

pname	count(s.sid)
pop	2

```
-- Find the names of songs for which created by artist has more than 20000 followers
SELECT s.sname
FROM Songs s
WHERE s.sid IN (SELECT sc.sid
                FROM SongCreates sc, Artists a
                WHERE sc.aid = a.aid AND a.afollowers > 20000);
```

Result:

sname
spring day
daylight
lowkey
butter
dynamite

```sql
-- find the names of artists created the maximum number of songs
SELECT a.aname
FROM Artists a
WHERE a.aid IN (SELECT sc.aid
                FROM SongCreates sc
                GROUP BY sc.aid
                HAVING count(*) IN
                (SELECT MAX(mycount) FROM
                (SELECT count(*) AS mycount
                FROM SongCreates sc1
                GROUP BY sc1.aid)));
```

Result:

aname
bts

```sql
-- find names of artists who made their debut before 2021
SELECT a.aname
FROM Artists a
WHERE EXISTS (SELECT *
              FROM Artists a1
              WHERE a1.adate < '2021-01-01' AND a.aid = a1.aid);
```

Result:

aname
keish
bts
maroon5

```
-- find names of latest song
SELECT s.sname
FROM Songs s, SongCreates sc
WHERE s.sid = sc.sid
AND sc.sday IN (SELECT MAX(sday)
                FROM SongCreates);
```

Result:

sname
daylight

-Result for View's SELECT Queries:

[User name and age]:

uname	uage
chris	22
kenny	21
jade	35
wenye	45
konnor	19

[Song and Artist]:

sname	aname
beside you	keish
spring day	bts
less of you	keish
daylight	maroon5
lowkey	niki
butter	bts
dynamite	bts

Playlist_song_name:

pname	sname
pop	spring day
anime	beside you
kpop	lowkey
edm	dynamite
pop	daylight

Snamerelease:

sname	sday
beside you	2020-04-2
spring day	2021-12-09
less of you	2021-06-03
daylight	2022-01-09
lowkey	2021-04-03
butter	2021-03-06
dynamite	2021-05-02