Sabanci University 2023-2024 Spring CS306 Group Project Phase2

TeamUp Music Streaming Platform

Osman Şah Yılmaz, 31316

Description

In Phase 2, we were told that we should choose 2 tables per person, but we also have tables with 3 or 4 relations. In other words, we cannot create 2 tables and insert them because it references the 3rd or 4th table with the id. That's why we can't do it without using common tables. That is, one of us chose user-subscription, the other chose artist-song, and then for the rest, we also need the song table to select (review-user)-song duo, and the song table also needs artist and genre. Likewise for selecting the song-album duo. That's why we created the other tables required for the two different tables we chose, but everyone executed the steps from the two main tables they chose. Yücel Saygın hoca knows the situation and confirmed it.

Step 1:

I choose Artists and Songs_Belong_to table to work with. (Genres and albums table also included as I explained in description)

Creating Tables:

```
CREATE TABLE Artists (
   art_id CHAR(5) NOT NULL,
   name CHAR(50),
   biography TEXT,
   monthly_listening INTEGER,
   PRIMARY KEY (art_id)
);

CREATE TABLE Genres (
 gid CHAR(5) NOT NULL,
 gname CHAR(50),
 PRIMARY KEY (gid)
);

CREATE TABLE Songs_Belong_to (
   sid CHAR(5) NOT NULL,
   length INTEGER,
   sname CHAR(50),
   art_id CHAR(5) NOT NULL,
   gid CHAR(5) NOT NULL,
   PRIMARY KEY (sid),
   FOREIGN KEY (art_id) REFERENCES Artists(art_id) ON DELETE CASCADE,
   FOREIGN KEY (gid) REFERENCES Genres(gid) ON DELETE CASCADE
);
```

CREATE TABLE Albums_consisted_of (

alb_id CHAR(5) NOT NULL,

aname CHAR(50),

sid CHAR(5) NOT NULL,

PRIMARY KEY (alb_id)

);

Step 2:

Here are the functional dependencies for each table:

Artists Table:

art_id → name, biography, monthly_listening

Songs_Belong_to Table:

sid

sid → \text{length, sname, art_id, gid}

Both tables are in BCNF because:

- In the **Artists** table, **art_id** is the primary key, and all attributes are functionally dependent on it.
- In the **Songs_Belong_to** table, **sid** is the primary key, and all attributes are functionally dependent on it. Moreover, **art_id** and **gid** are foreign keys, but there are no partial dependencies or transitive dependencies violating BCNF.

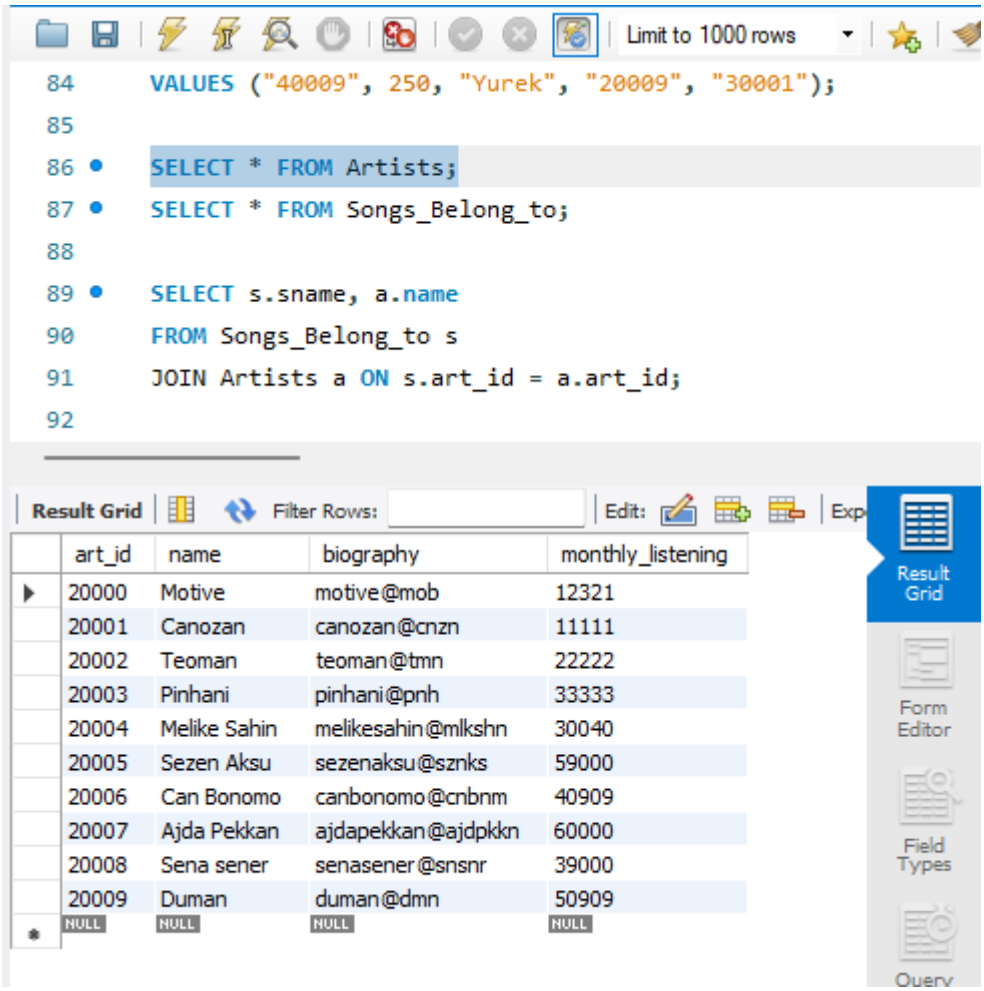
Step 3:

```
INSERT INTO Artists (art_id, name, biography, monthly_listening)
VALUES ("20000", "Motive", "motive@mob", 12321);
INSERT INTO Artists (art_id, name, biography, monthly_listening)
VALUES ("20001", "Canozan", "canozan@cnzn", 11111);
INSERT INTO Artists (art_id, name, biography, monthly_listening)
VALUES ("20002", "Teoman", "teoman@tmn", 22222);
INSERT INTO Artists (art_id, name, biography, monthly_listening)
VALUES ("20003", "Pinhani", "pinhani@pnh", 33333);
INSERT INTO Artists (art_id, name, biography, monthly_listening)
VALUES ("20004", "Melike Sahin", "melikesahin@mlkshn", 30040);
INSERT INTO Artists (art_id, name, biography, monthly_listening)
VALUES ("20005", "Sezen Aksu", "sezenaksu@sznks", 59000);
INSERT INTO Artists (art_id, name, biography, monthly_listening)
VALUES ("20006", "Can Bonomo", "canbonomo@cnbnm", 40909);
INSERT INTO Artists (art_id, name, biography, monthly_listening)
VALUES ("20007", "Ajda Pekkan", "ajdapekkan@ajdpkkn", 60000);
INSERT INTO Artists (art_id, name, biography, monthly_listening)
VALUES ("20008", "Sena sener", "senasener@snsnr", 39000);
INSERT INTO Artists (art_id, name, biography, monthly_listening)
VALUES ("20009", "Duman", "duman@dmn", 50909);

INSERT INTO Genres (gid, gname) VALUES ("30000", "Turkish Rap");
INSERT INTO Genres (gid, gname) VALUES ("30001", "Alternative");
INSERT INTO Genres (gid, gname) VALUES ("30002", "Acoustic Rock");
INSERT INTO Genres (gid, gname) VALUES ("30003", "Turkish Rock");
INSERT INTO Genres (gid, gname) VALUES ("30004", "Blues");
INSERT INTO Genres (gid, gname) VALUES ("30005", "Classical");
INSERT INTO Genres (gid, gname) VALUES ("30006", "Pop");
INSERT INTO Genres (gid, gname) VALUES ("30007", "Country");
```

```
INSERT INTO Genres (gid, gname) VALUES ("30008", "Reggae");
INSERT INTO Genres (gid, gname) VALUES ("30009", "Indie");

INSERT INTO Songs_Belong_to (sid, length, sname, art_id, gid)
VALUES ("40000", 190, "LOADED", "20000", "30000");
INSERT INTO Songs_Belong_to (sid, length, sname, art_id, gid)
VALUES ("40001", 180, "Bazen", "20001", "30001");
INSERT INTO Songs_Belong_to (sid, length, sname, art_id, gid)
VALUES ("40002", 170, "Gonullcelen", "20002", "30002");
INSERT INTO Songs_Belong_to (sid, length, sname, art_id, gid)
VALUES ("40003", 160, "Nehirler Durmaz", "20003", "30003");
INSERT INTO Songs_Belong_to (sid, length, sname, art_id, gid)
VALUES ("40004", 240, "Pusulam Ruzgar", "20004", "30006");
INSERT INTO Songs_Belong_to (sid, length, sname, art_id, gid)
VALUES ("40005", 240, "Vay", "20005", "30006");
INSERT INTO Songs_Belong_to (sid, length, sname, art_id, gid)
VALUES ("40006", 235, "Tukeniyor Omrum", "20004", "30003");
INSERT INTO Songs_Belong_to (sid, length, sname, art_id, gid)
VALUES ("40007", 195, "Haykiracak Nefesim Kalmasa Bile", "20007", "30006");
INSERT INTO Songs_Belong_to (sid, length, sname, art_id, gid)
VALUES ("40008", 195, "Teni Tenime", "20008", "30001");
INSERT INTO Songs_Belong_to (sid, length, sname, art_id, gid)
VALUES ("40009", 250, "Yurek", "20009", "30001");
```

Step 4:

SELECT * FROM Artists;



```
84      VALUES ("40009", 250, "Yurek", "20009", "30001");
85
86 •    SELECT * FROM Artists;
87 •    SELECT * FROM Songs_Belong_to;
88
89 •    SELECT s.sname, a.name
90      FROM Songs_Belong_to s
91      JOIN Artists a ON s.art_id = a.art_id;
92
```

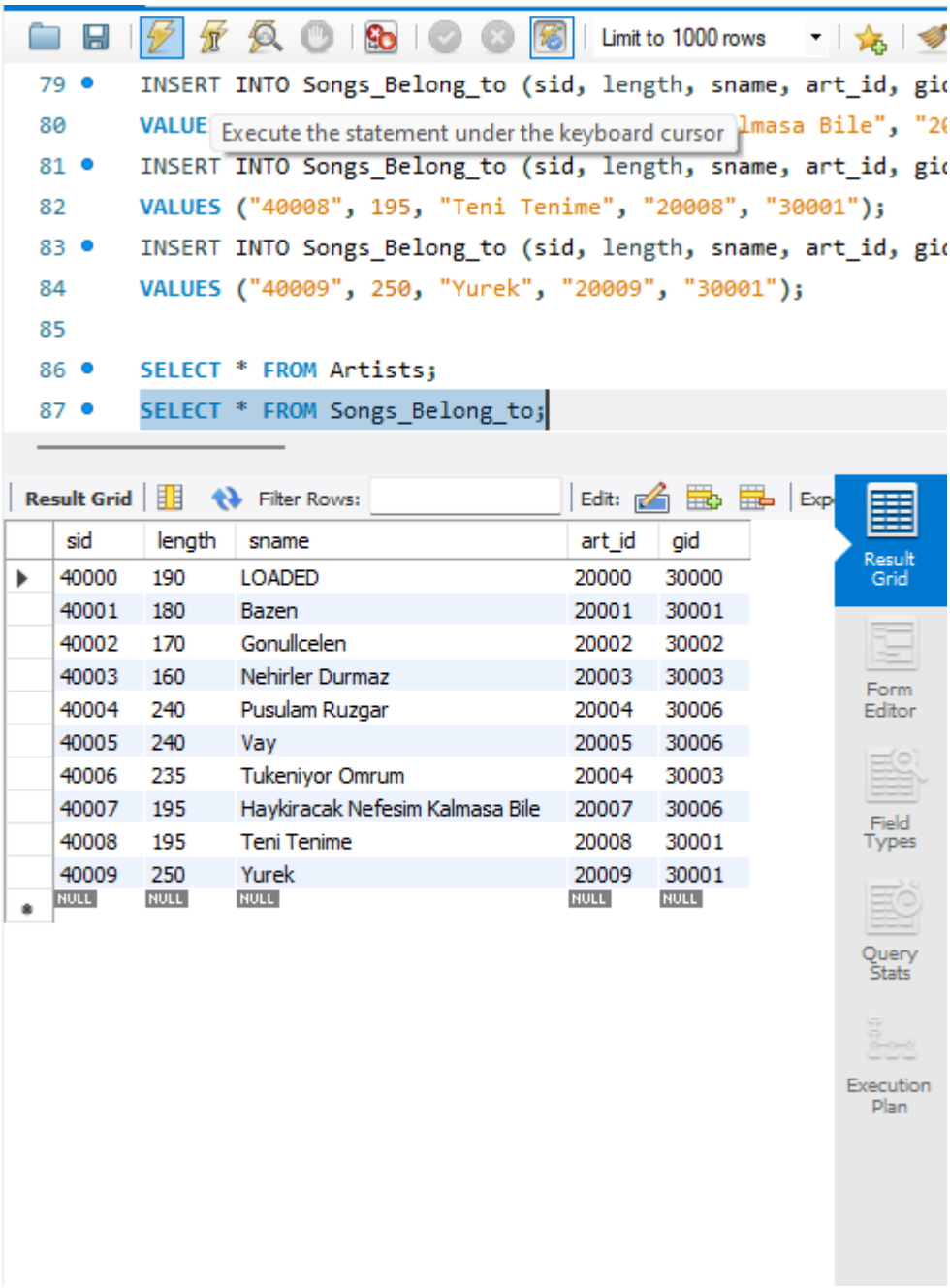
Limit to 1000 rows

Result Grid | Filter Rows: | Edit: | Exp

art_id	name	biography	monthly_listening
20000	Motive	motive@mob	12321
20001	Canozan	canozan@cnzn	11111
20002	Teoman	teoman@tmn	22222
20003	Pinhani	pinhani@pnh	33333
20004	Melike Sahin	melikesahin@mlkshn	30040
20005	Sezen Aksu	sezenaksu@sznks	59000
20006	Can Bonomo	canbonomo@cnbnm	40909
20007	Ajda Pekkan	ajdapekkan@ajdpkkn	60000
20008	Sena sener	senasener@snsnr	39000
20009	Duman	duman@dmn	50909
NULL	NULL	NULL	NULL

Result Grid
Form Editor
Field Types
Query

SELECT * FROM Songs_Belong_to;



sid	length	sname	art_id	gid
40000	190	LOADED	20000	30000
40001	180	Bazen	20001	30001
40002	170	Gonullcelen	20002	30002
40003	160	Nehirler Durmaz	20003	30003
40004	240	Pusulam Ruzgar	20004	30006
40005	240	Vay	20005	30006
40006	235	Tukeniyor Omrum	20004	30003
40007	195	Haykiracak Nefesim Kalmasa Bile	20007	30006
40008	195	Teni Tenime	20008	30001
40009	250	Yurek	20009	30001
NULL	NULL	NULL	NULL	NULL

Step 5:

Join Query in English: "Retrieve all songs along with their artist names."

Relational Algebra Equivalent :

π_{sname, name} (Songs_Belong_to ⋈_{Songs_Belong_to.art_id} = {Artists.art_id}{Artists})

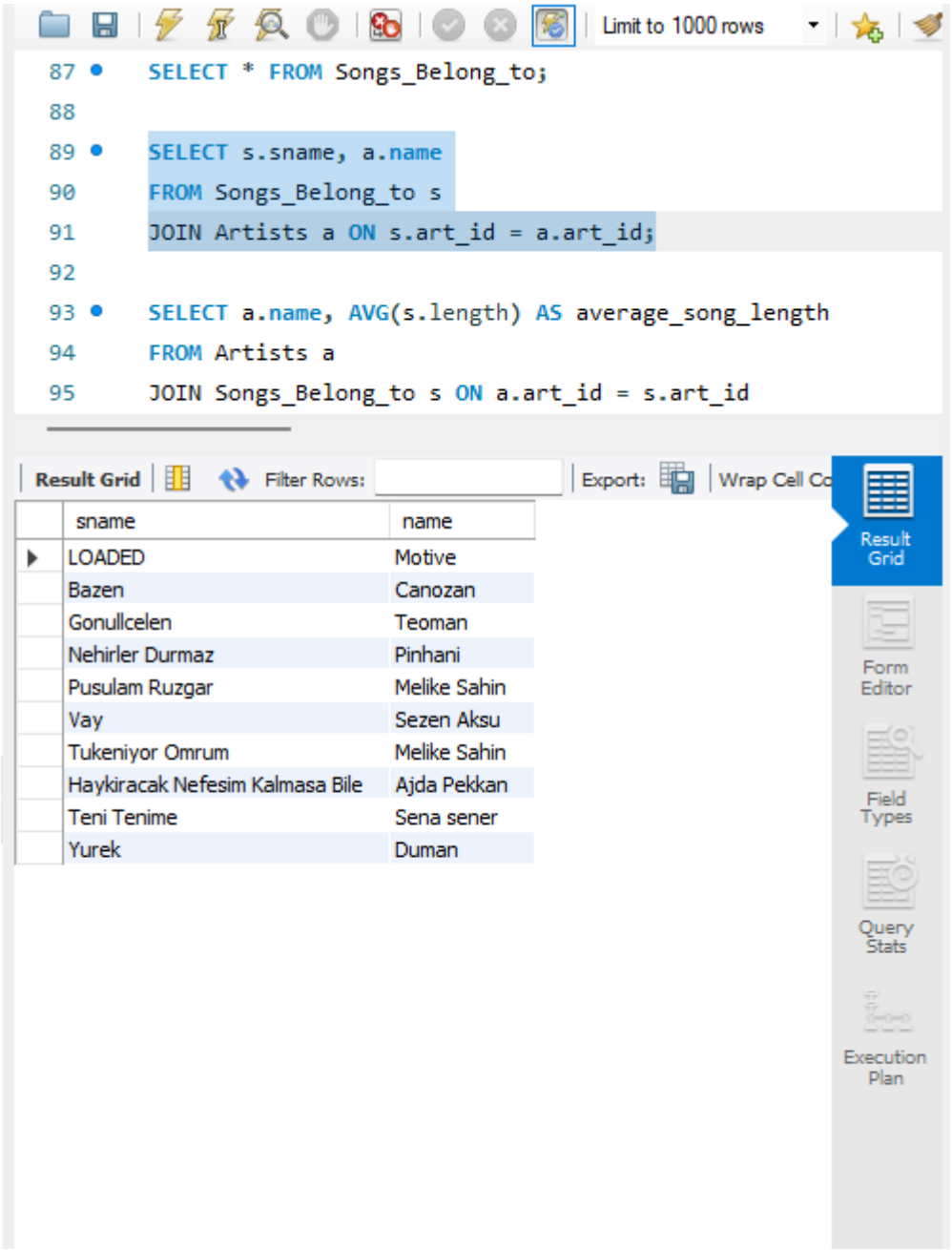
Step 6:

SQL Equivalent:

SELECT s.sname, a.name

FROM Songs_Belong_to s

JOIN Artists a ON s.art_id = a.art_id;



Step 7:

Group By Query in English: "Calculate the average song length for each artist, including the artist's name."

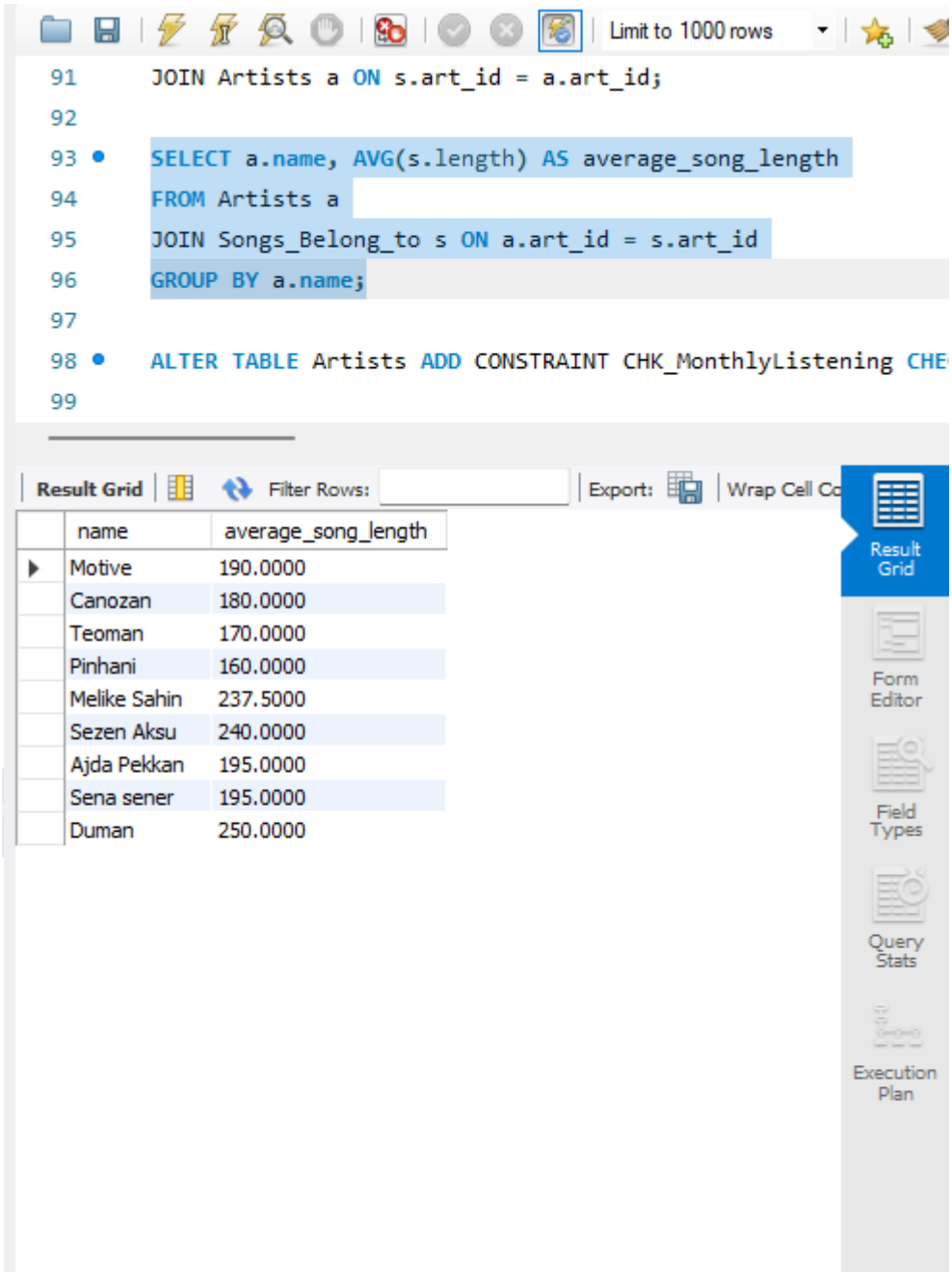
SQL Equivalent:

SELECT a.name, AVG(s.length) AS average_song_length

FROM Artists a

JOIN Songs_Belong_to s ON a.art_id = s.art_id

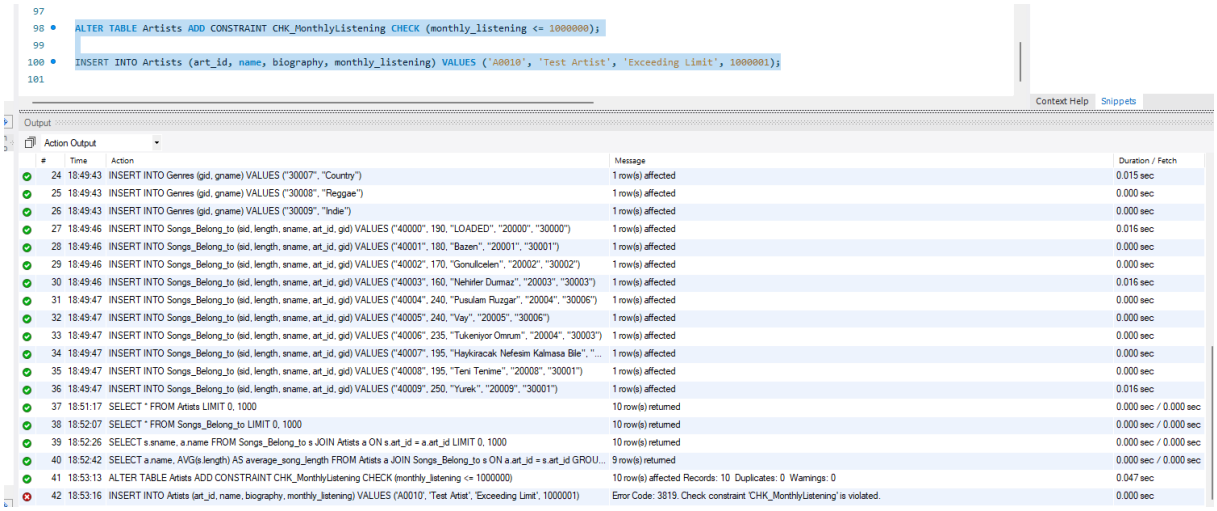
GROUP BY a.name;



Step 8:

ALTER TABLE Artists ADD CONSTRAINT CHK_MonthlyListening CHECK (monthly_listening <= 1000000);

INSERT INTO Artists (art_id, name, biography, monthly_listening) VALUES ('A0010', 'Test Artist', 'Exceeding Limit', 1000001);



```
97
98 •  ALTER TABLE Artists ADD CONSTRAINT CHK_MonthlyListening CHECK (monthly_listening <= 1000000);
99
100 •  INSERT INTO Artists (art_id, name, biography, monthly_listening) VALUES ('A0010', 'Test Artist', 'Exceeding Limit', 1000001);
101
```

Context Help | Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
24	18:49:43	INSERT INTO Genres (gid, gname) VALUES ("30007", "Country")	1 row(s) affected	0.015 sec
25	18:49:43	INSERT INTO Genres (gid, gname) VALUES ("30008", "Reggae")	1 row(s) affected	0.000 sec
26	18:49:43	INSERT INTO Genres (gid, gname) VALUES ("30009", "Indie")	1 row(s) affected	0.000 sec
27	18:49:46	INSERT INTO Songs_Belong_to (sid, length, sname, art_id, gid) VALUES ("40000", 190, "LOADED", "20000", "30000")	1 row(s) affected	0.016 sec
28	18:49:46	INSERT INTO Songs_Belong_to (sid, length, sname, art_id, gid) VALUES ("40001", 180, "Bazen", "20001", "30001")	1 row(s) affected	0.000 sec
29	18:49:46	INSERT INTO Songs_Belong_to (sid, length, sname, art_id, gid) VALUES ("40002", 170, "Gonullcelen", "20002", "30002")	1 row(s) affected	0.000 sec
30	18:49:46	INSERT INTO Songs_Belong_to (sid, length, sname, art_id, gid) VALUES ("40003", 160, "Nehirler Durmaz", "20003", "30003")	1 row(s) affected	0.016 sec
31	18:49:47	INSERT INTO Songs_Belong_to (sid, length, sname, art_id, gid) VALUES ("40004", 240, "Pusulam Ruzgar", "20004", "30006")	1 row(s) affected	0.000 sec
32	18:49:47	INSERT INTO Songs_Belong_to (sid, length, sname, art_id, gid) VALUES ("40005", 240, "Vay", "20005", "30006")	1 row(s) affected	0.000 sec
33	18:49:47	INSERT INTO Songs_Belong_to (sid, length, sname, art_id, gid) VALUES ("40006", 235, "Tukeniyor Omrum", "20004", "30003")	1 row(s) affected	0.000 sec
34	18:49:47	INSERT INTO Songs_Belong_to (sid, length, sname, art_id, gid) VALUES ("40007", 195, "Haykiracak Nefesim Kalmasa Bile", "...	1 row(s) affected	0.000 sec
35	18:49:47	INSERT INTO Songs_Belong_to (sid, length, sname, art_id, gid) VALUES ("40008", 195, "Teni Tenime", "20008", "30001")	1 row(s) affected	0.000 sec
36	18:49:47	INSERT INTO Songs_Belong_to (sid, length, sname, art_id, gid) VALUES ("40009", 250, "Yurek", "20009", "30001")	1 row(s) affected	0.016 sec
37	18:51:17	SELECT * FROM Artists LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
38	18:52:07	SELECT * FROM Songs_Belong_to LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
39	18:52:26	SELECT s.sname, a.name FROM Songs_Belong_to s JOIN Artists a ON s.art_id = a.art_id LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
40	18:52:42	SELECT a.name, AVG(s.length) AS average_song_length FROM Artists a JOIN Songs_Belong_to s ON a.art_id = s.art_id GROU...	9 row(s) returned	0.000 sec / 0.000 sec
41	18:53:13	ALTER TABLE Artists ADD CONSTRAINT CHK_MonthlyListening CHECK (monthly_listening <= 1000000)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.047 sec
42	18:53:16	INSERT INTO Artists (art_id, name, biography, monthly_listening) VALUES ('A0010', 'Test Artist', 'Exceeding Limit', 1000001)	Error Code: 3819. Check constraint 'CHK_MonthlyListening' is violated.	0.000 sec