İSTANBUL ÜNİVERSİTESİ
CERRAHPAŞA

BIMU3064-Veri Tabanı Yönetim Sistemleri

İBRAHİM BAŞAR YARGICI
1306191467

Ödev3
Veritabanı = PostgreSQL

(12.12.2021)

İÜC. Bilgisayar Müh, Veritabanı Programlama Ödevi (Ödev 3),

- AKSIS'ten 10.12.2021 Cuma gecesi saat 24:00'ye kadar teslim edilmelidir.
- Ödev 10 puandır. Geç ödev kabul edilmez.

Student (sid, name, birthPlace, did, gpa) // ogrenci(ogrenci-no, adi, dogum-yeri, bolum-no)
Take (sid, cid, grade) // ders-al(ogrenci-no, ders-kodu, notu)
Course (cid, title, credits, did, studentCount) // ders(ders-kodu, adi, kredisi, bolum-no, ogrSayisi)
Department (did, name) // bolum(bolum-no, adi)
Teacher (tid, name, birthPlace, did) // hoca(hoca-no, adi, dogum-yeri, bolum-no)
Teach (tid, cid) // ders-ver(hoca-no, ders-kodu)

Aşağıdaki soruları cevaplayarak kaynak kodlarını .java/.php/.sql dosyaları olarak ve test sonuçlarını bir DOCX dosyasına copy-paste yaparak AKSIS'ten paylaşınız.

1. (3 puan) artist(id, name, gender, status, field) tablosundaki sanatçıları listeleyen, yeni sanatçi ekleyen, var olan bir sanatçıyı silen, ve var olan bir sanatçının bilgilerini güncelleyen PHP uygulamasını aksis.php örneğini değiştirerek yazınız ve test ediniz. Artist tablosundaki alanların türleri, varsa aldıkları farklı değerler ve ekleme/güncelleme HTML formu içinde INPUT element olarak TYPE türü aşağıdaki listede verilmiştir:
    a. id [int, input type=text],
    b. Name [varchar(20), input type=text],
    c. Gender [char(1): 'F','M', input type=radio],
    d. Status [smallint: 0: False, 1:True, input type=checkbox],
    e. Field [enum: 1: 'Music', 2:'Painting', 3:'Calligraphy', input type=select]

Bu soruyu cevaplamak için xampp'ı kurup, xampp/xampp-control-panel'den apache ve mysql sunucularını çalıştırmak, sonrasında xampp/htdocs/odev klasorunu oluşturup burada index.php diye programı yazmak ve chrome içinden http://localhost/odev2/index.php adresini yazarak test etmeniz gerekmektedir.

2. (3 puan) 1. Soruyu bir Java console programıyla yeniden yazınız. Kayıt ekleme ve güncellemede HTML formu kullanamayacağınızdan değerleri console'dan teker teker elle girilecek şekilde programlayınız. Bu programı compile etmek için PostgreSQL için JDBC driver'i indirip programla aynı klasöre yerleştirmeniz gerekmektedir.
3. (2 puan) take tablosuna kayıt eklenince, silinince veya bir kayıttaki cid yada sid değiştirildiğinde course tablosundaki studentCount alanını gerektiğinde güncelleyen trigger yada triggerları yazınız. Bu triggerları 4 adet komutu (INSERT, DELETE, UPDATE(sid), UPDATE(cid)) çalıştırarak studentCount alanının güncellenip güncellenmediği test ediniz.
4. (2 puan) Cid'leri verilen 2 dersi de alan öğrencilerin kayıtlarını tablo olarak döndüren ortakOgrenciSayisi(cid1, cid2) stored funtion'ı yazınız. Bu function'ı örnek bir SELECT komutunda kullanarak test ediniz.

Veri Tabanı ve Veri Tabanı Şeması Hazırlığı

```
drop table IF EXISTS Teach;
drop table IF EXISTS Take;
drop table IF EXISTS Teacher;
drop table IF EXISTS Course;
drop table IF EXISTS Student;
drop table IF EXISTS Department;



create table Department
        (did numeric(5) not null,
        name varchar(30) not null,
        primary key(did));

create table Student
        (sid numeric(5) not null,
        name varchar(30) not null,
        birthplace varchar(50),
        did numeric(5),
        gpa numeric(5),
        foreign key (did) references Department(did),
        primary key(sid));

create table Course
        (cid numeric(5) not null,
        title varchar(30) not null,
        credits numeric(2),
        did  numeric(5),
        studentCount numeric(5),
        foreign key (did) references Department(did),
        primary key(cid));
```

```
create table Teacher
        (tid numeric(5) not null,
        fname varchar(30) not null,
        birthplace varchar(50),
        did numeric(5),
        foreign key (did) references Department(did),
        primary key(tid));

create table Take
        (sid numeric(5) not null,
        cid numeric(5) not null,
        grade float,
        foreign key (sid) references Student(sid),
        foreign key (cid) references Course(cid),
        primary key (sid,cid));

create table Teach
 (tid numeric(5) not null,
 cid numeric(5) not null,
 foreign key (tid) references Teacher(tid),
 foreign key (cid) references Course(cid),
 primary key (tid,cid));

insert into Department values (1,  'Comp. Eng.');
insert into Department values (2,  'Elec. Eng.');
insert into Department values (3,  'Env. Eng.');
insert into Department values (4,  'Ind. Eng.');

insert into Student values (1,      'Ali',      'istanbul',     1, 1.2);
insert into Student values (2,      'Ahmet',    'ankara',       1, 1.2);
insert into Student values (3,      'Leyla',    'izmir',        1, 1.2);
```

```
insert into Student values (4,    'Can',     'manisa',    2, 2.2);
insert into Student values (5,    'Aziz',    'istanbul',  2, 2.2);
insert into Student values (6,    'Talat',   'izmir',     3, 3.2);
insert into Student values (7,    'Kamuran', 'adana',     3, 3.2);
insert into Student values (8,    'Turgut',  'bursa',     4, 4.2);
insert into Student values (9,    'Oznur',   'bolu',      2, 2.2);
insert into Student values (10,   'Pelin',   'izmir',     4, 4.2);
insert into Student values (11,   'Savas',   'izmir',     4, 4.2);

insert into Course values (1,    'database',                        3, 1, 12);
insert into Course values (2,    'operating system',                3, 1, 12);
insert into Course values (3,    'Introduction to Programming', 4,  1, 12);
insert into Course values (4,    'introduction to electronic',      2, 2, 22);
insert into Course values (5,    'statistic',                       4, 4, 42);
insert into Course values (6,    'circuit theory',                  3,2, 22);
insert into Course values (7,    'introduction to environment',   3,3, 32);
insert into Course values (8,    'operation research',              3,4, 42);
insert into Course values (9,    'summer practice',                 2,4, 42);
insert into Course values (10,   'summer practice',                 3,3, 32);
insert into Course values (11,   'summer practice',                 3,1, 12);
insert into Course values (12,   'summer practice',                 3,2, 22);

insert into Teacher values (1,    'Selami',    'amasya',1);
insert into Teacher values (2,    'Cengiz',    'istanbul',1);
insert into Teacher values (3,    'Derya',     'mersin',1);
insert into Teacher values (4,    'Dogan',     'istanbul',2);
insert into Teacher values (5,    'Ayten',     'istanbul',3);
insert into Teacher values (6,    'Tahsin',    'izmir', 4);
insert into Teacher values (7,    'Selcuk',    'amasya',4);
```

```
insert into Teach values (1,1);
insert into Teach values (3,	2);
insert into Teach values (2,	3);
insert into Teach values (4,	4);
insert into Teach values (7,	5);
insert into Teach values (4,	6);
insert into Teach values (5,	7);
insert into Teach values (6,	8);
insert into Teach values (7,	9);
insert into Teach values (5,	10);
insert into Teach values (1,	11);
insert into Teach values (4,	12);

insert into Take values (1,  1,	3);
insert into Take values (1,  3,	2.5);
insert into Take values (1,  4,	3.5);
insert into Take values (1,  6,	3);
insert into Take values (1,  9,	4);
insert into Take values (1,  10,	3);
insert into Take values (2,  1,	4);
insert into Take values (2,  2,	4);
insert into Take values (2,  3,	4);
insert into Take values (2,  4,	4);
insert into Take values (2,  5,	4);
insert into Take values (2,  6,	4);
insert into Take values (2,  7,	4);
insert into Take values (2,  8,	4);
insert into Take values (2,  9,	4);
insert into Take values (2,  10,	3);
insert into Take values (2,  11,	4);
insert into Take values (3,  1,	4);
```

```
insert into Take values (3,  2,      4);
insert into Take values (3,  3,      4);
insert into Take values (3,  4,      4);
insert into Take values (3,  5,      4);
insert into Take values (3,  6,      4);
insert into Take values (3,  7,      4);
insert into Take values (3,  8,      4);
insert into Take values (3,  9,      4);
insert into Take values (3,  10,     3);
insert into Take values (3,  11,     3.5);
insert into Take values (4,  1,      2.5);
insert into Take values (4,  5,      1.5);
insert into Take values (5,  11,     3.5);
insert into Take values (5,  1,      3);
insert into Take values (5,  5,      1.5);
insert into Take values (6,  2,      4);
insert into Take values (7,  5,      1.5);
insert into Take values (7,  1,      2.5);
insert into Take values (7,  8,      1.5);
insert into Take values (7,  2,      3);
insert into Take values (8,  2,      3.5);
insert into Take values (8,  7,      1.5);
insert into Take values (10,2,       4);
insert into Take values (10,8,       3);
insert into Take values (11,8,       1);
```

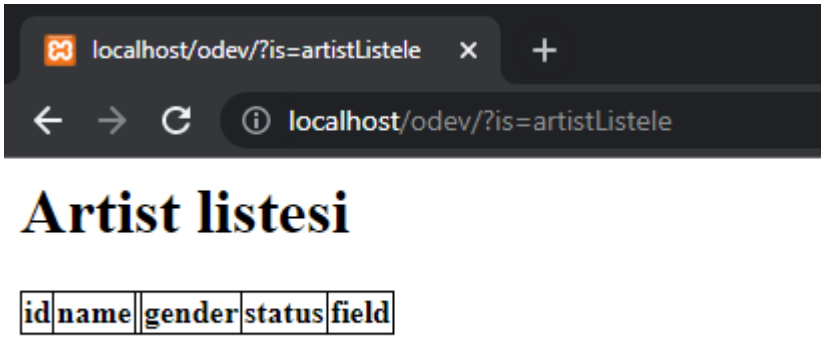
Cevaplar

Soru 1:

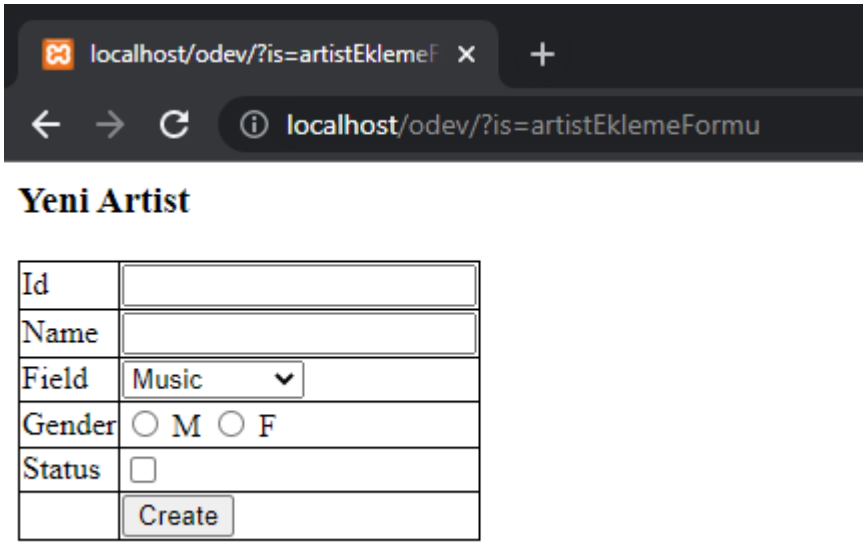
PHP kodunu index.php içerisinde görebilirsiniz ya da Github üzerinden bakmak isterseniz:

https://github.com/basarYargici/DatabaseManagementSystems/blob/main/HW3/index.php

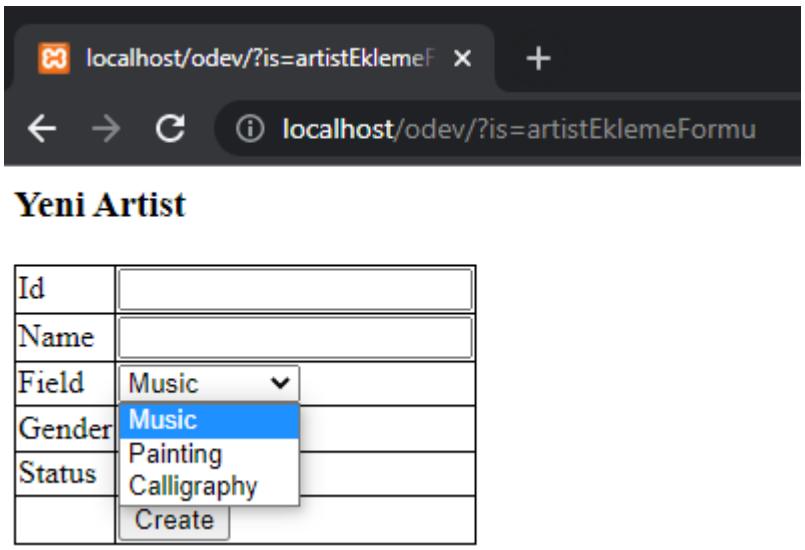
Artist Listele:



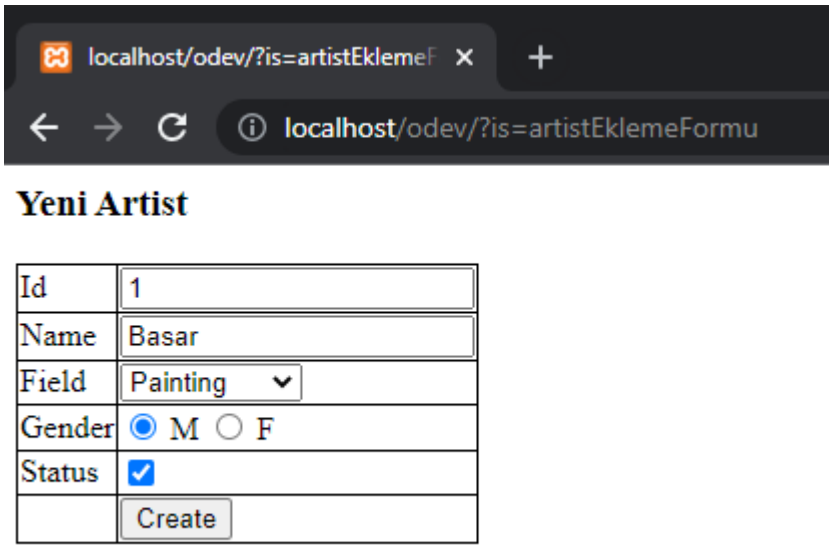
Artist Ekleme Formu:



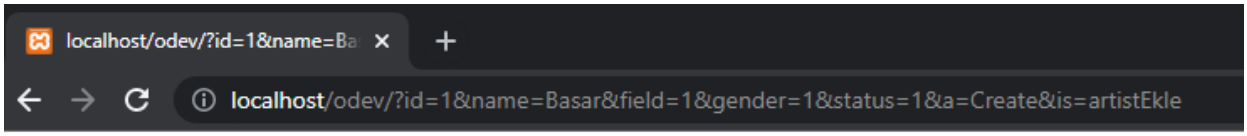
Fieldlar:



Form Doldurma:



Create butonuna tıklayınca:

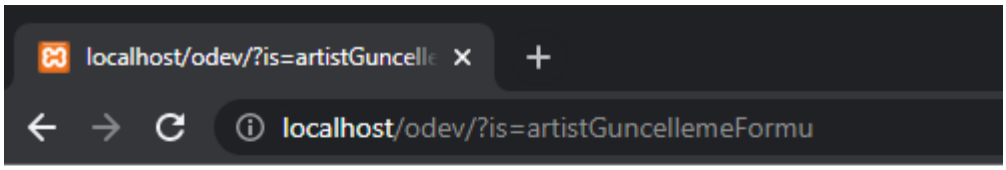


SQL: INSERT INTO artist2 VALUES(1, 'Basar', '1', '1', '1');

Artist listesi

id	name	gender	status	field
1	Basar	1	1	Music

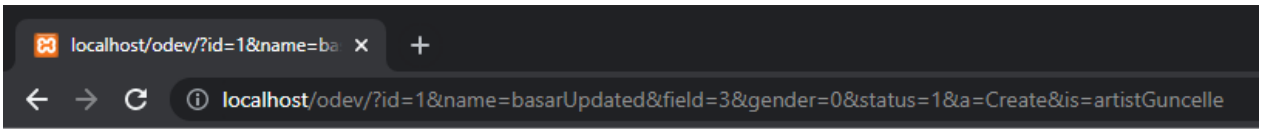
Artist Güncelleme Formu:



Yeni Artist

Id	
Name	
Field	Music
Gender	M F
Status	
	Create

Artist Güncelleme:



SQL: UPDATE artist2 SET name='basarUpdated', gender='0', status='1', field=3 WHERE id=1;

Artist listesi

id	name	gender	status	field
1	basarUpdated	0	1	Calligraphy

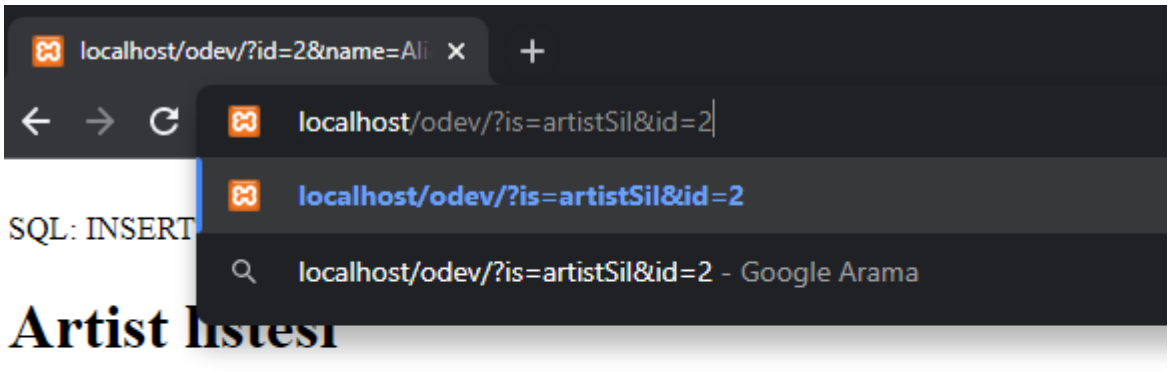
Artist Ekleme:

SQL: INSERT INTO artist2 VALUES(2, 'Ali', '1', '1', '3');

Artist listesi

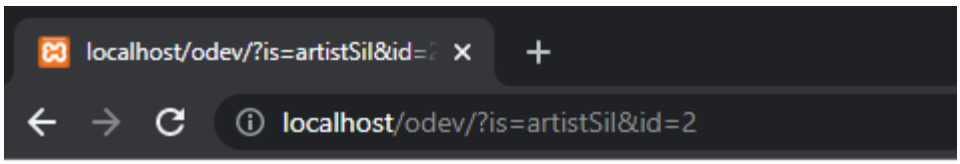
id	name	gender	status	field
1	basarUpdated	0	1	Calligraphy
2	Ali	1	1	Calligraphy

Artist Sil:



id	name	gender	status	field
1	basarUpdated	0	1	Calligraphy
2	Ali	1	1	Calligraphy

2 Numaralı Artist Silindi:



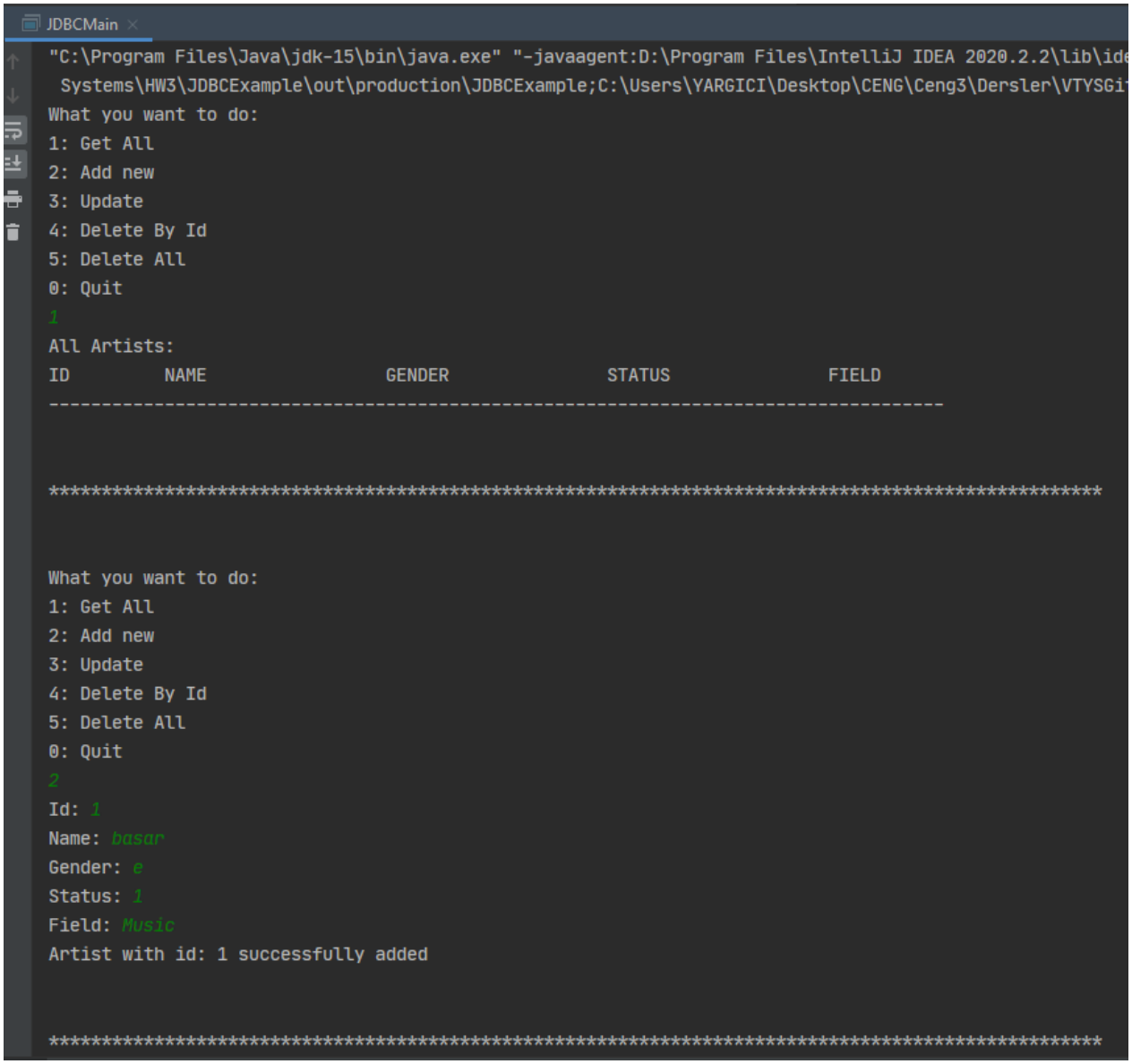
Artist listesi

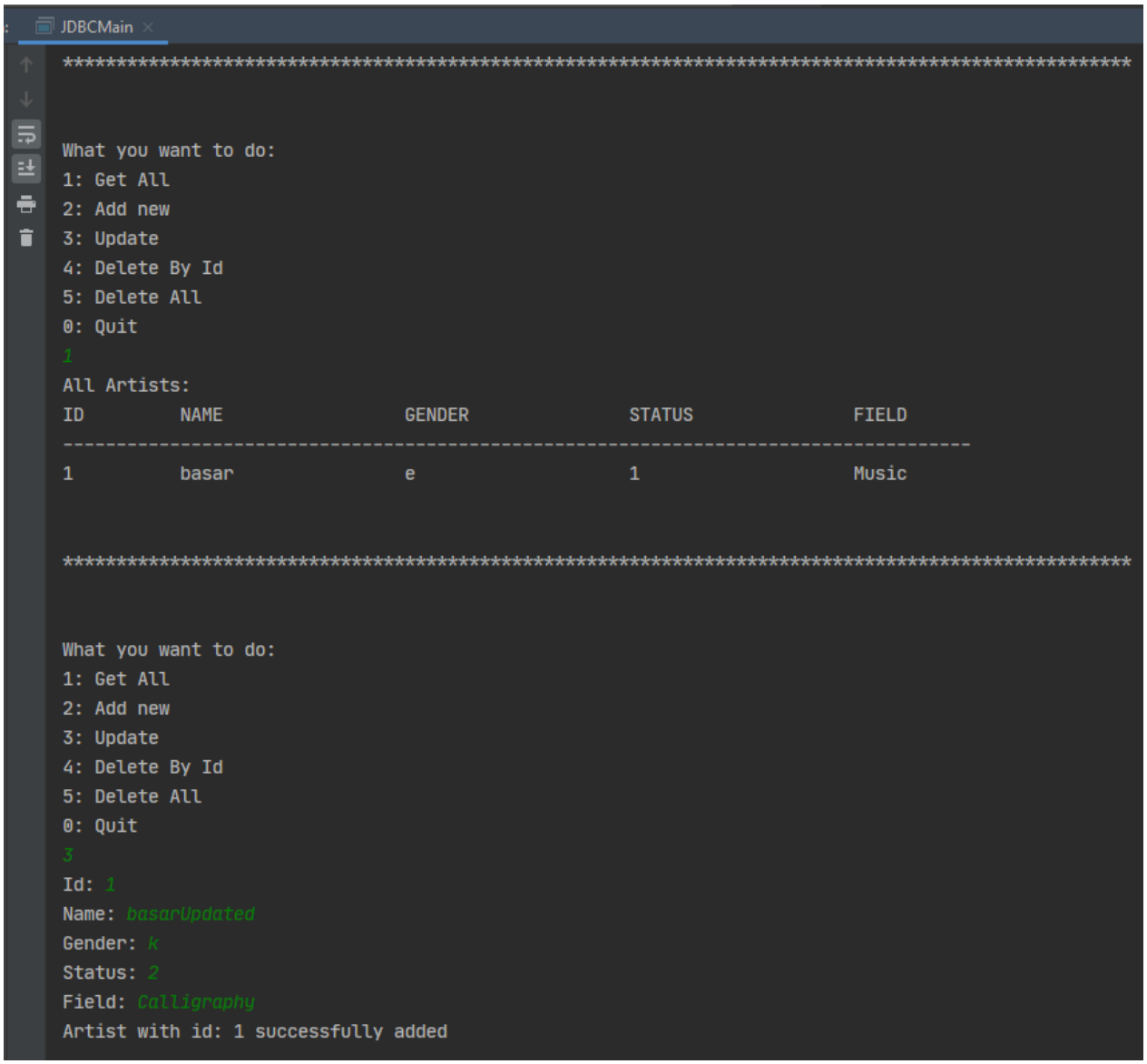
id	name	gender	status	field
1	basarUpdated	0	1	Calligraphy

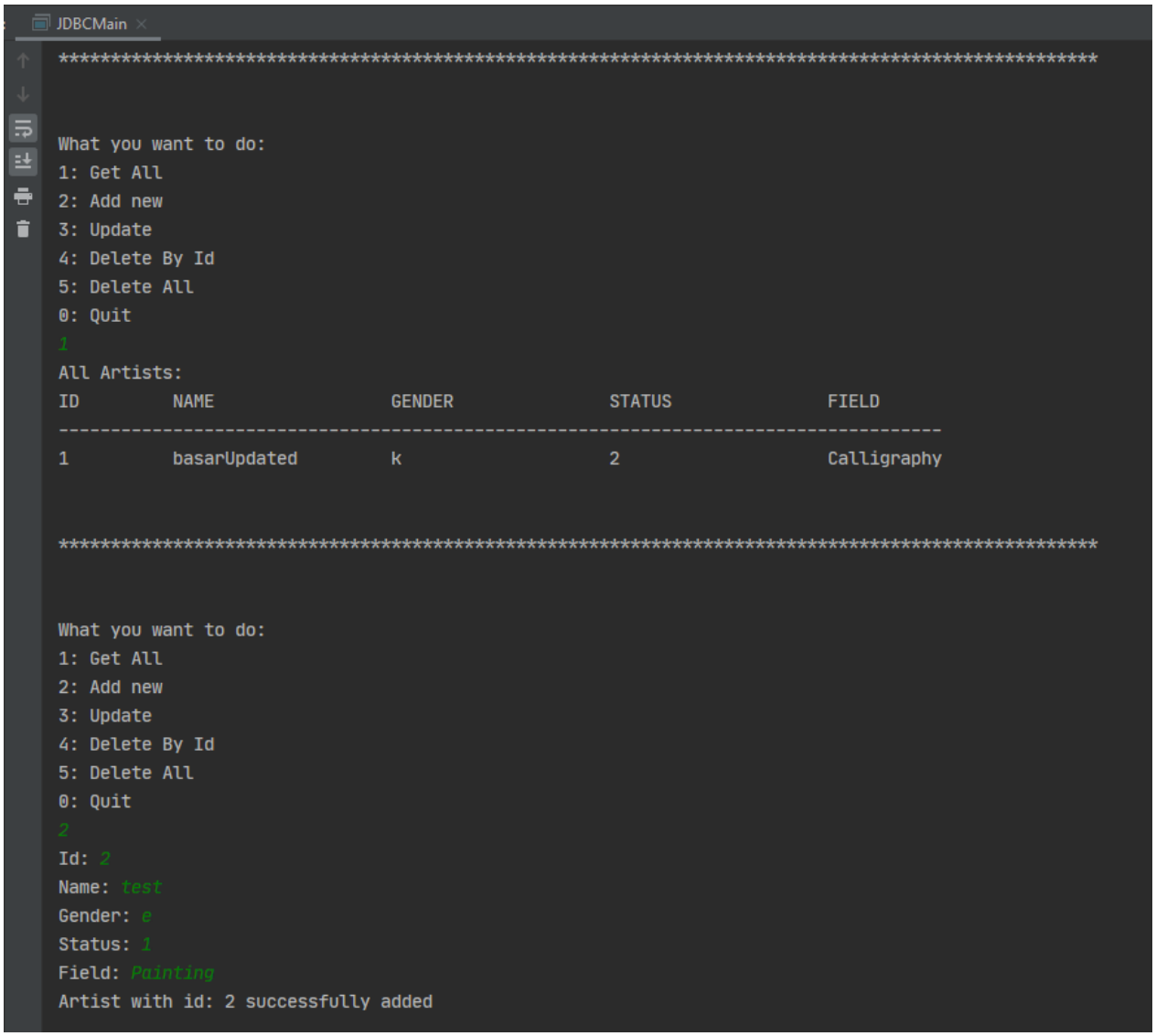
Sorgu 2:

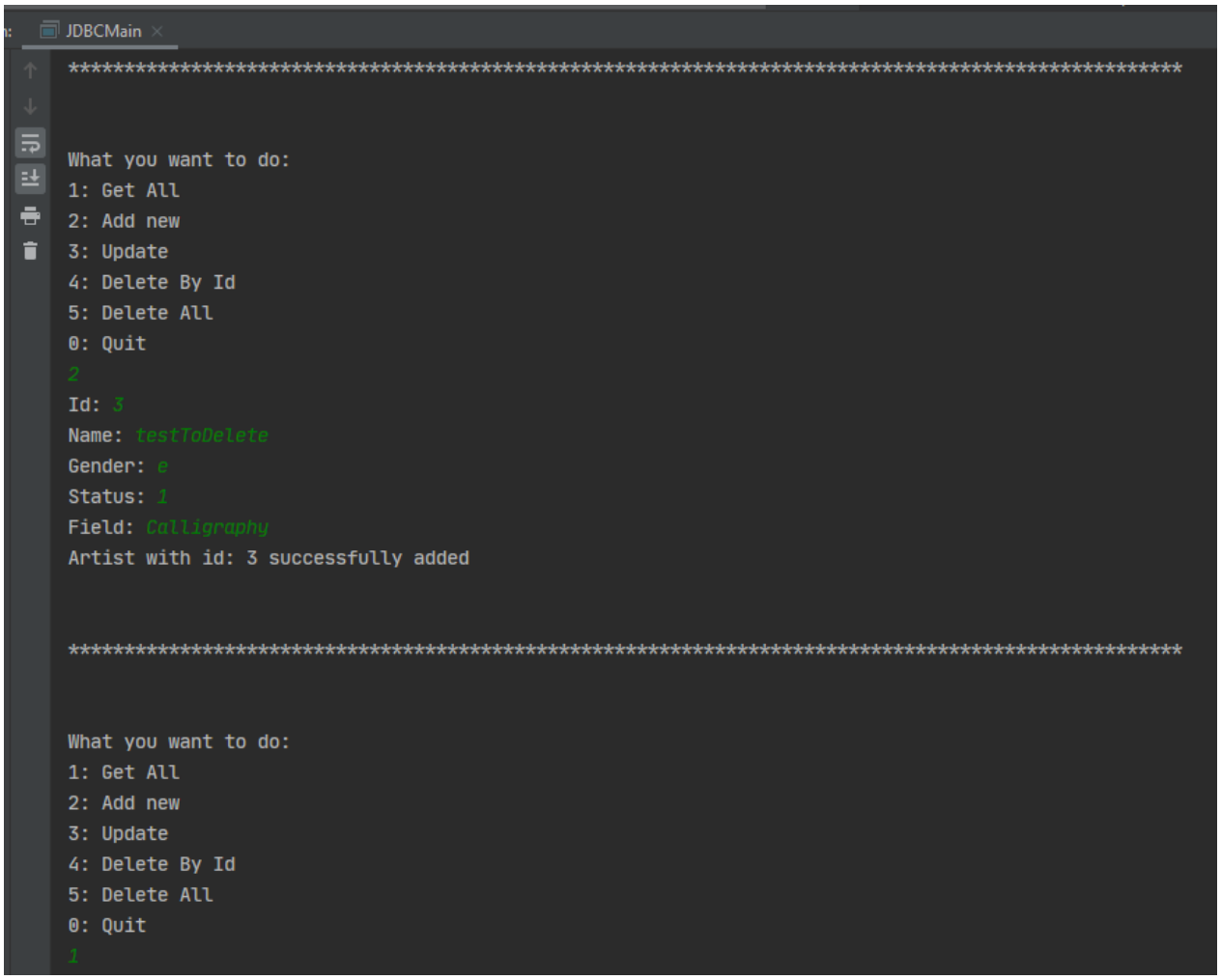
Console çıktıları aşağıdadır. Java kodunu JDBCExample projesi-dosyası içerisinde görebilirsiniz ya da Github üzerinden bakmak isterseniz:

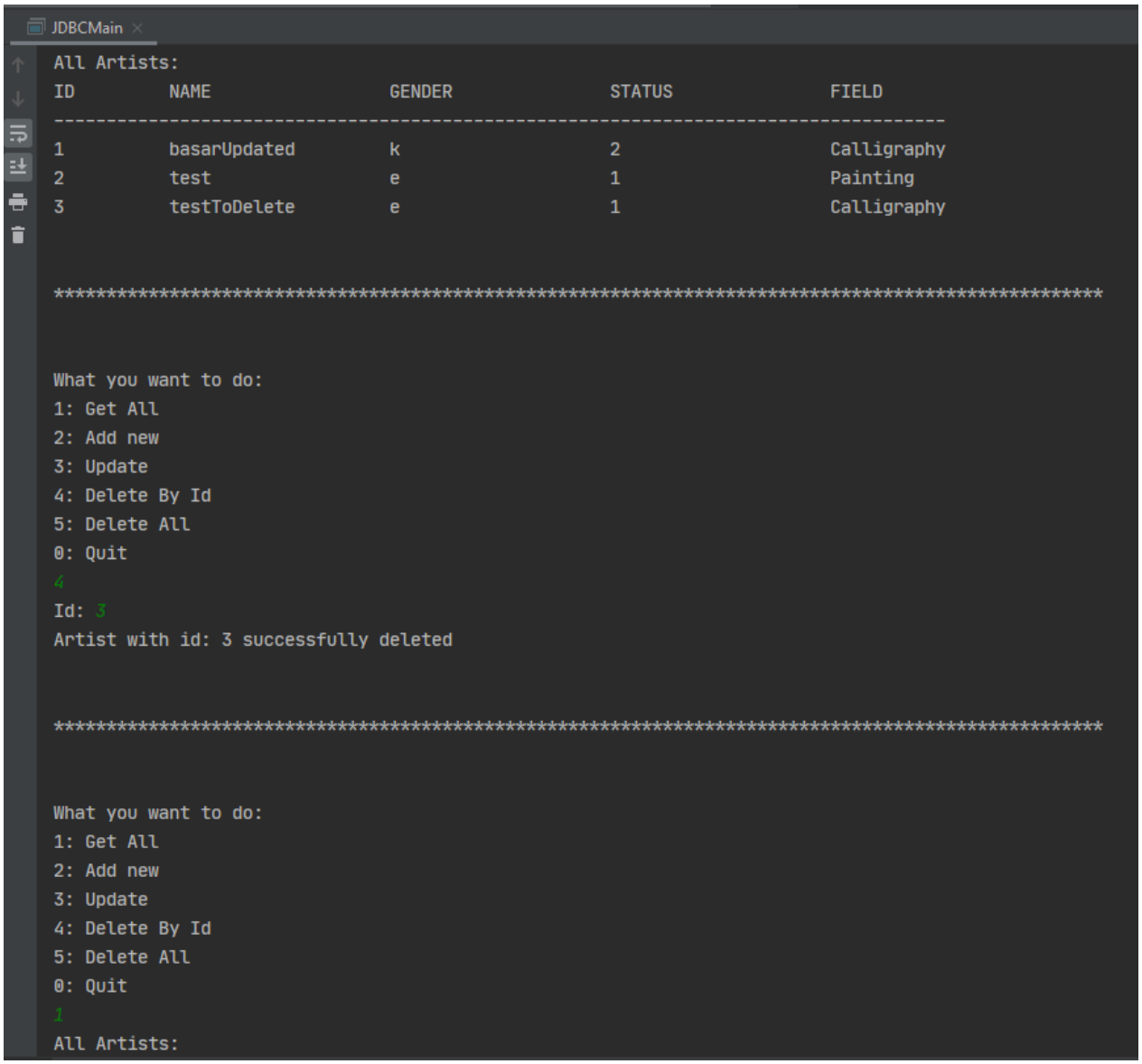
https://github.com/basarYargici/DatabaseManagementSystems/tree/main/HW3/JDBCExample/src

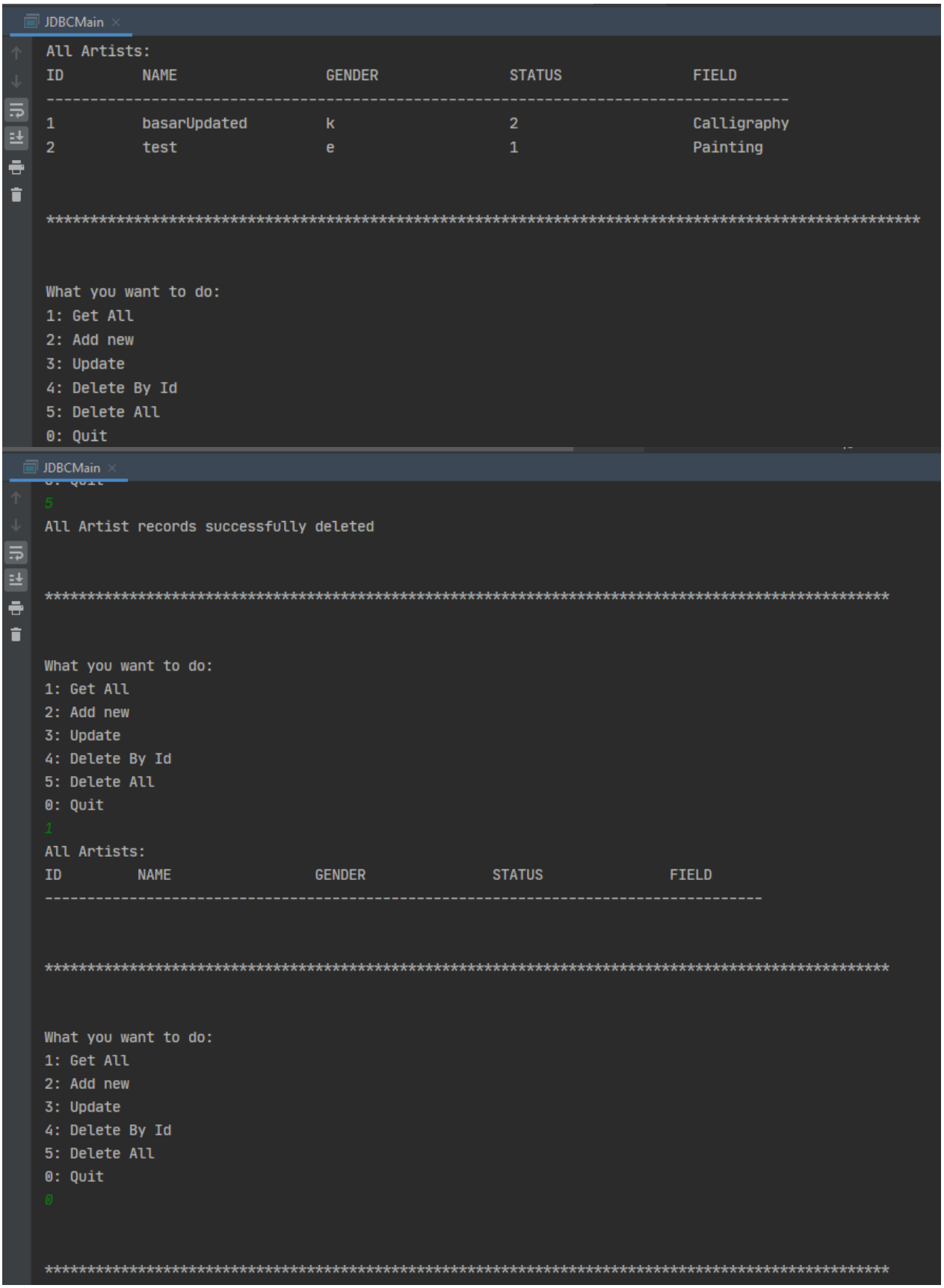



JDBCMain
****
What you want to do:
1: Get All
2: Add new
3: Update
4: Delete By Id
5: Delete All
0: Quit
1
All Artists:
ID NAME GENDER STATUS FIELD
1 basar e 1 Music
****
What you want to do:
1: Get All
2: Add new
3: Update
4: Delete By Id
5: Delete All
0: Quit
3
Id: 1
Name: basarUpdated
Gender: k
Status: 2
Field: Calligraphy
Artist with id: 1 successfully added


JDBCMain
****
What you want to do:
1: Get All
2: Add new
3: Update
4: Delete By Id
5: Delete All
0: Quit
1
All Artists:
ID NAME GENDER STATUS FIELD
1 basarUpdated k 2 Calligraphy
****
What you want to do:
1: Get All
2: Add new
3: Update
4: Delete By Id
5: Delete All
0: Quit
2
Id: 2
Name: test
Gender: e
Status: 1
Field: Painting
Artist with id: 2 successfully added


JDBCMain
****
What you want to do:
1: Get All
2: Add new
3: Update
4: Delete By Id
5: Delete All
0: Quit
2
Id: 3
Name: testToDelete
Gender: e
Status: 1
Field: Calligraphy
Artist with id: 3 successfully added
****
What you want to do:
1: Get All
2: Add new
3: Update
4: Delete By Id
5: Delete All
0: Quit
1


JDBCMain
All Artists:
ID NAME GENDER STATUS FIELD
1 basarUpdated k 2 Calligraphy
2 test e 1 Painting
3 testToDelete e 1 Calligraphy
What you want to do:
1: Get All
2: Add new
3: Update
4: Delete By Id
5: Delete All
0: Quit
4
Id: 3
Artist with id: 3 successfully deleted
What you want to do:
1: Get All
2: Add new
3: Update
4: Delete By Id
5: Delete All
0: Quit
1
All Artists:


JDBCMain
All Artists:
ID NAME GENDER STATUS FIELD
1 basarUpdated k 2 Calligraphy
2 test e 1 Painting
What you want to do:
1: Get All
2: Add new
3: Update
4: Delete By Id
5: Delete All
0: Quit
JDBCMain
5
All Artist records successfully deleted
What you want to do:
1: Get All
2: Add new
3: Update
4: Delete By Id
5: Delete All
0: Quit
1
All Artists:
ID NAME GENDER STATUS FIELD
What you want to do:
1: Get All
2: Add new
3: Update
4: Delete By Id
5: Delete All
0: Quit
0

```
Process finished with exit code 0
```

Soru 3:

```
-- Student ders alınca Course'deki studentCount'ı arttırmak için gereken trigger işlemleri
CREATE OR REPLACE Function add()
	RETURNS TRIGGER AS $$
			begin
				UPDATE course SET studentCount = studentCount+1 WHERE
cid = NEW.cid;
				return new;
			end; $$
	language plpgsql;

CREATE TRIGGER add_trigger
BEFORE INSERT
ON take
FOR EACH ROW
EXECUTE PROCEDURE add();



SELECT * FROM take WHERE cid =1;
SELECT * FROM course ORDER BY cid;

INSERT INTO take VALUES(10,1,2)
```

INSERT TEST

Cid = 1 olan kursu alanlar:

```
184  SELECT * FROM take WHERE cid =1;
185  SELECT * FROM course ORDER BY cid;
186
187  INSERT INTO take VALUES(10,1,2)
```

Data Output | Explain | Messages | Notifications

	sid [PK] numeric (5)	cid [PK] numeric (5)	grade double precision
1	1	1	3
2	2	1	4
3	3	1	4

Coursedeki 1 numaralı kurstaki öğrenci satısı:

```
184  SELECT * FROM take WHERE cid =1;
185  SELECT * FROM course ORDER BY cid;
186
187  INSERT INTO take VALUES(10,1,2)
```

Data Output | Explain | Messages | Notifications

	cid [PK] numeric (5)	title character varying (30)	credits numeric (2)	did numeric (5)	studentcount numeric (5)
1	1	database	3	1	13
2	2	operating system	3	1	12

Sid = 10 olan öğrencinin kursa kaydolması

```
184  SELECT * FROM take WHERE cid =1;
185  SELECT * FROM course ORDER BY cid;
186
187  INSERT INTO take VALUES(10,1,2)
```

Data Output | Explain | Messages | Notifications

```
INSERT 0 1

Query returned successfully in 36 msec.
```

Course id si 1 olan kursun “studentcount” field’ı 13 iken 14 oldu:

```
184  SELECT * FROM take WHERE cid =1;
185  SELECT * FROM course ORDER BY cid;
186
187  INSERT INTO take VALUES(10,1,2)
```

Data Output | Explain | Messages | Notifications

	cid [PK] numeric (5)	title character varying (30)	credits numeric (2)	did numeric (5)	studentcount numeric (5)
1	1	database	3	1	14
2	2	operating system	3	1	12
3	3	Introduction to Programming	4	1	12

DELETE TEST

```
-- Student ders bırakınca Course'deki studentCount'ı azaltmak  için gereken trigger
--  işlemleri
CREATE OR REPLACE Function delete()
        RETURNS TRIGGER AS $_$
                begin
                        UPDATE course SET studentCount = studentCount-1 WHERE
studentCount > 0 AND cid = OLD.cid;
                        RETURN OLD;
                end $_$
        language plpgsql;

CREATE TRIGGER delete_trigger
BEFORE DELETE
ON take
FOR EACH ROW
EXECUTE PROCEDURE delete();



SELECT * FROM take WHERE cid =1;
SELECT * FROM course ORDER BY cid;

DELETE FROM take WHERE (cid = 1 AND sid = 10)
```

Burada sid'si 10 numaralı olan Student'ın take kaydını silmeye çalışacağız:

```
218  SELECT * FROM take WHERE cid =1;
219  SELECT * FROM course ORDER BY cid;
220
221  DELETE FROM take WHERE (cid = 1 AND sid = 10)
```

Data Output | Explain | Messages | Notifications

	sid [PK] numeric (5)	cid [PK] numeric (5)	grade double precision
1	1	1	3
2	2	1	4
3	3	1	4
4	4	1	2.5
5	5	1	3
6	7	1	2.5
7	8	1	4
8	10	1	2

1 numaralı kursu alan sayısı = 18:

```
218  SELECT * FROM take WHERE cid =1;
219  SELECT * FROM course ORDER BY cid;
220
221  DELETE FROM take WHERE (cid = 1 AND sid = 10)
```

Data Output | Explain | Messages | Notifications

	cid [PK] numeric (5)	title character varying (30)	credits numeric (2)	did numeric (5)	studentcount numeric (5)
1	1	database	3	1	18
2	2	operating system	3	1	12
3	3	Introduction to Programming	4	1	12

Silme işlemi:

```
218  SELECT * FROM take WHERE cid =1;
219  SELECT * FROM course ORDER BY cid;
220
221  DELETE FROM take WHERE (cid = 1 AND sid = 10)
```

Data Output | Explain | Messages | Notifications

```
DELETE 1
```

Silme sonrası 1 numaralı kursu alan sayısı = 17:

```
218  SELECT * FROM take WHERE cid =1;
219  SELECT * FROM course ORDER BY cid;
220
221  DELETE FROM take WHERE (cid = 1 AND sid = 10)
```

Data Output | Explain | Messages | Notifications

	cid [PK] numeric (5)	title character varying (30)	credits numeric (2)	did numeric (5)	studentcount numeric (5)
1	1	database	3	1	17
2	2	operating system	3	1	12
3	3	Introduction to Programming	4	1	12

Soru 4:

```
CREATE FUNCTION ortakOgrenciSayisi(
    cid1 int,
    cid2 int
)
RETURNS int
language plpgsql
AS
$$
declare
   ogrenciSayisi integer;
BEGIN

        SELECT sid
        INTO ogrencisayisi
        FROM take
        WHERE cid IN (cid1,cid2)
        GROUP BY sid
        HAVING COUNT(distinct cid) = 2;

        RETURN ogrenciSayisi;
END;
$$;

SELECT * FROM take WHERE cid = 4 or cid = 5
SELECT ortakOgrenciSayisi(4,5);
```

Cid 4 ve 5 I alan öğrenci sayısı:

```
241  SELECT * FROM take WHERE cid = 4 or cid = 5
242  SELECT ortakOgrenciSayisi(4,5);
```

Data Output | Explain | Messages | Notifications

	sid [PK] numeric (5)	cid [PK] numeric (5)	grade double precision
1	1	4	3.5
2	2	4	4
3	2	5	4
4	3	4	4
5	3	5	4
6	4	5	1.5
7	5	5	1.5
8	7	5	1.5

```
241  SELECT * FROM take WHERE cid = 4 or cid = 5
242  SELECT ortakOgrenciSayisi(4,5);
```

Data Output | Explain | Messages | Notifications

	ortakogrencisayisi integer
1	2

Cid 3 ve 9 u alan öğrenci sayısı:

```
241  SELECT * FROM take WHERE cid = 3 or cid = 8
242  SELECT ortakOgrenciSayisi(3,8);
```

Data Output | Explain | Messages | Notifications

	sid [PK] numeric (5)	cid [PK] numeric (5)	grade double precision
1	1	3	2.5
2	2	3	4
3	2	8	4
4	3	3	4
5	3	8	4
6	7	8	1.5
7	10	8	3
8	11	8	1

```
242  SELECT ortakOgrenciSayisi(3,8);
```

Data Output | Explain | Messages | Notifications

	ortakogrencisayisi integer
1	2