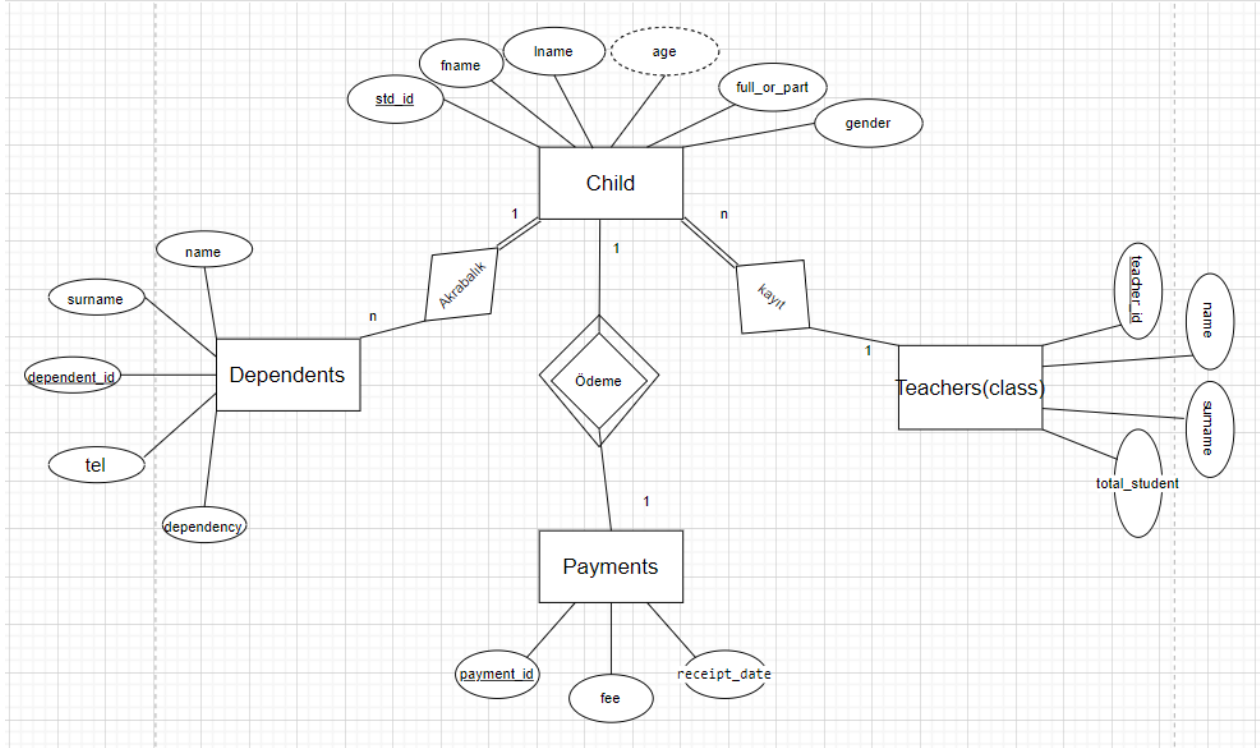


```
self.c = self.conn.cursor()
self.c.execute("CREATE TABLE IF NOT EXISTS teachers(teacher_id INTEGER CHECK (teacher_id >= 2 AND teacher_id <=5) PRIMARY KEY, name
self.c.execute(
    """CREATE TABLE IF NOT EXISTS child (
std_id integer PRIMARY KEY,
fname char(20),
lname char(20),
age INTEGER CHECK (age >=2 AND age <= 5) REFERENCES teachers(teacher_id),
full_or_part char(20),
gender char(1)
)""")

self.c.execute("CREATE TABLE IF NOT EXISTS payments(payment_id INTEGER PRIMARY KEY REFERENCES child(std_id) ON DELETE CASCADE, fee
self.c.execute("CREATE TABLE IF NOT EXISTS dependent(dependent_id INTEGER REFERENCES child(std_id) ON DELETE CASCADE, name char(20)
```

**Er diagram:



**Maddeler:

Madde 1) Oluşturacağınız veritabanı en az 4 tablo içermelidir. Her tabloda en az 10 kayıt bulunmalıdır.

	std_id [PK] integer	fname character (20)	lname character (20)	age integer	full_or_part character (20)	gender character (1)
/	/	ali /	bugday8	4	sabahcı	M
8	8	ali9	bugday9	4	oglenci	M
9	9	ali10	bugday10	4	tumGun	M
10	11	ali12	bugday12	5	oglenci	M

Total rows: 10 of 10 Query complete 00:00:00.045

Madde2) Tablolarınızda primary key ve foreign key kısıtlarını kullanmalısınız.

```
self.c.execute("CREATE TABLE IF NOT EXISTS teachers(teacher_id INTEGER CHECK (teacher_id >= 2 AND teacher_id <=5) PRIMARY KEY, name char(20), salary INTEGER CHECK (salary >= 2000 AND salary <= 5000))")
self.c.execute(
    """CREATE TABLE IF NOT EXISTS child (
        std_id integer PRIMARY KEY,
        fname char(20),
        lname char(20),
        age INTEGER CHECK (age >=2 AND age <= 5) REFERENCES teachers(teacher_id),
        full_or_part char(20),
        gender char(1)
    )""")

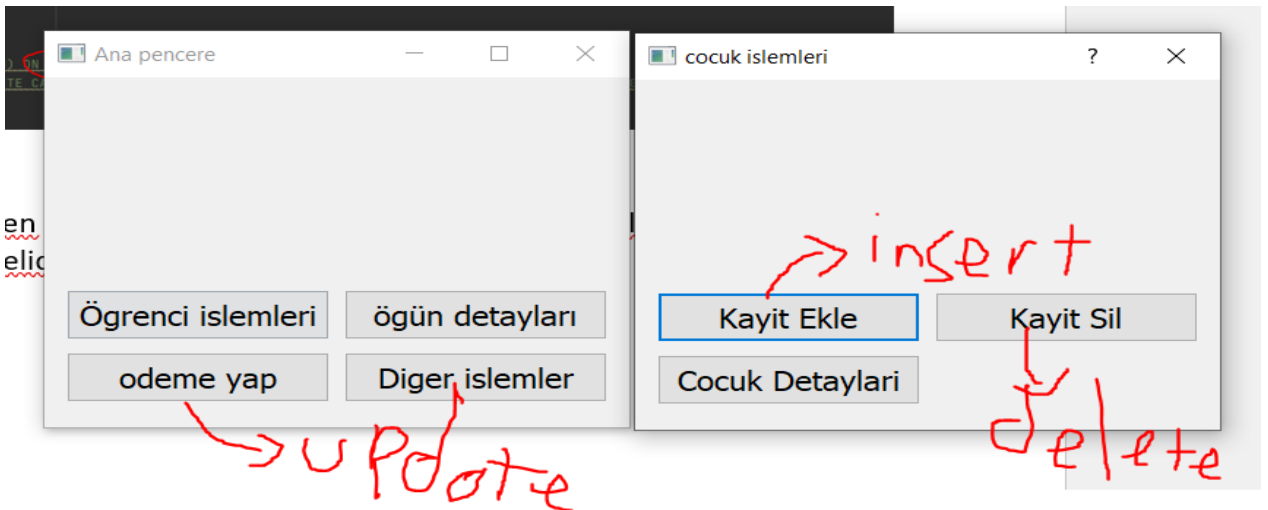
self.c.execute("CREATE TABLE IF NOT EXISTS payments(payment_id INTEGER PRIMARY KEY REFERENCES child(std_id) ON DELETE CASCADE, fee INTEGER, receipt_date DATE)")
self.c.execute("CREATE TABLE IF NOT EXISTS dependent(dependent_id INTEGER REFERENCES child(std_id) ON DELETE CASCADE, name char(20), surname char(20), tel char(12), dependency char(20))")
```

Madde 3) En az bir tabloda silme kısıtı ve sayı kısıtı olmalıdır.

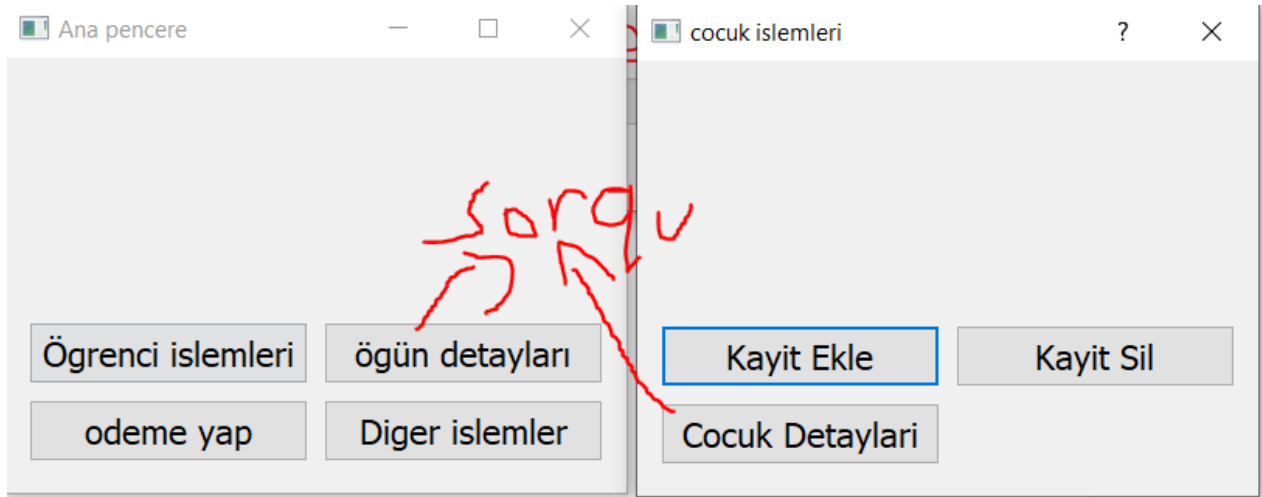
```
teacher_id >= 2 AND teacher_id <=5) PRIMARY KEY, name char(20), surname char(20), total_student INTEGER CHECK (total_student <= 30))")

KEY REFERENCES child(std_id) ON DELETE CASCADE, fee INTEGER, receipt_date DATE)")
REFERENCES child(std_id) ON DELETE CASCADE, name char(20), surname char(20), tel char(12), dependency char(20))")
```

Madde 4) Arayüzden en az birer tane insert, update ve delete işlemi gerçekleştirilebilmelidir.



Madde 5) Arayüzden girilecek bir değere göre ekrana sonuçların listelendiği bir sorgu yazmalısınız.



Madde 6) Arayüzden çağrılan sorgulardan en az biri “view” olarak tanımlanmış olmalıdır.

```
1 - 1,  
create or replace view ogrenciler_view as  
select fname, lname, fee, reciept_date  
from child, payments  
where std_id = payment_id;  
  
create or replace view yarim_gun as  
select fname, lname, fee, reciept_date  
from child, payments  
where full_or_part <> 'tumGun' and std_id = payment_id;  
  
create or replace view tam_gun as  
select fname, lname, fee, reciept_date  
from child, payments  
where full_or_part = 'tumGun' and std_id = payment_id;
```

Madde 7) En az bir adet “sequence” oluşturmali ve arayüzden yapılacak insert sırasında ilgili sütundaki değerlerin otomatik olarak atanmasını sağlamalısınız.

```
self.c.execute("""CREATE SEQUENCE IF NOT EXISTS mysequence
                INCREMENT 1
                START 1;""")
```

Madde 8) Arayüzden çağrılan sorgulardan en az birinde union veya intersect veya except kullanmış olmalısınız.

```
DECLARE
    ogrs ogrenciler[];
    i INTEGER;
    kayit_gecersiz CURSOR FOR
    select y.fname, y.lname, y.fee, y.receipt_date
    from yarim_gun y, tam_gun tam
    where y.fee = 250 and tam.fee = 500
    union
    select fname, lname, fee, receipt_date
    from ogrenciler_view
    where (EXTRACT (YEAR FROM receipt_date) + 1) < EXTRACT (YEAR FROM CURRENT_DATE);
BEGIN
    i = 1;
```

Madde 9) Sorgularınızın en az biri aggregate fonksiyonlar içermeli, having ifadesi kullanılmalıdır.

```
CREATE OR REPLACE FUNCTION calculate_me

DECLARE
    sabahci INTEGER;
    oglenci INTEGER;
    tumGun INTEGER;
BEGIN
    select count(*) into sabahci
    from child
    group by full_or_part
    having full_or_part = 'sabahci';
```

Madde 10) Arayüzden girilen değerleri parametre olarak alıp ekrana sonuç döndüren 3 farklı SQL fonksiyonu tanımlamış olmalısınız. Bu fonksiyonların en az birinde “record” ve “cursor” tanımı-kullanımı olmalıdır.

1.

```
CREATE OR REPLACE FUNCTION calculate_meal(out kahvalti INTEGER, out oglenYemegi INTEGER) AS $$

DECLARE
    sabahci INTEGER;
    oglenci INTEGER;
    tumGun INTEGER;
BEGIN
    select count(*) into sabahci
    from child
    group by full_or_part
    having full_or_part = 'sabahci';

    select count(*) INTO oglenci
    from child
    where full_or_part='oglenci';

    select count(*) INTO tumGun
    from child
    where full_or_part='tumGun';

    kahvalti := sabahci + tumGun;
    oglenYemegi := oglenci + tumGun;
END;
```

2.

```
self.c.execute("""
    CREATE OR REPLACE FUNCTION odeme_yap(id_num INTEGER,miktar INTEGER) RETURNS VOID AS $$

    DECLARE
        borc INTEGER;
    BEGIN
        select fee INTO borc
        from payments
        where payment_id = id_num;

        borc = borc - miktar;

        update payments set fee = borc where payment_id = id_num;
    END;

    $$LANGUAGE plpgsql;
""")
```

3.

```
self.c.execute("""
    CREATE OR REPLACE FUNCTION calculate_record() RETURNS ogrenciler[] AS $$

    DECLARE
        ogrs ogrenciler[];
        i INTEGER;
        kayit_gecersiz CURSOR FOR
        select y.fname, y.lname, y.fee, y.reciept_date
        from yarim_gun y, tam_gun tam
        where y.fee = 250 and tam.fee = 500
        union
        select fname, lname, fee, reciept_date
        from ogrenciler_view
        where (EXTRACT (YEAR FROM reciept_date) + 1) < EXTRACT (YEAR FROM CURRENT_DATE);
    BEGIN
        i = 1;
        create or replace view ogrenciler_view as
        select fname, lname, fee, reciept_date
        from child, payments
        where std_id = payment_id;

        create or replace view yarim_gun as
        select fname, lname, fee, reciept_date
        from child, payments
        where std_id = payment_id and fee = 250;
    END;
    $$

```

Madde 11) 2 adet trigger tanımlamalı ve arayüzden girilecek değerlerle tetiklemelisiniz. Trigger'ın çalıştığına dair arayüze bilgilendirme mesajı döndürülmelidir.

1. Trigger her bir öğrenci eklendiğinde öğretmenin öğrenci sayısını bir artırıyor.

```
self.c.execute("""
    CREATE OR REPLACE FUNCTION trig_inc() RETURNS TRIGGER AS $$
    BEGIN
        update teachers set total_student = total_student + 1 where new.age = teacher_id;
        return new;
    END;

    $$LANGUAGE plpgsql;
""")

self.c.execute("""
    CREATE TRIGGER increment_student
    AFTER INSERT ON child
    FOR EACH ROW execute procedure trig_inc()
""")

```

2.Trigger ogrencinin akraba numarasının numeric degerler dışında deger içerip icermediğini control ediyor.

```
self.c.execute("""
    CREATE OR REPLACE FUNCTION trig_numeric() RETURNS TRIGGER AS $$
    BEGIN
        IF new.tel not like '%[^0-9]%' THEN
            return new;
        END IF;
        return null;
    END;
    $$LANGUAGE plpgsql;
""")
self.c.execute("""
    CREATE TRIGGER check_numeric
    AFTER INSERT ON dependent
    FOR EACH ROW execute procedure trig_numeric()
""")
```

SON