Öğrenci ders tablosunun gereksinimleri:

bir öğrenci bir dersi aldığında bu tabloya verisi eklenecek. geçemez ve kalırsa tekrar aldığında dönemkodu değiştirilerek tabloya pk dışındaki satırları boş olan fakat devam zorunluluğu bilgisi önceki kaydın devamsizlik oranına göre belirlenerek otomatik atanacak şekilde bir yeni kayıt daha eklenecek. (eğer öğrenci o derse zaten kayıtlıysa dönem kodu 1den 2ye çıkacak 3. kez alıyosa 3e çıkacak otomatik tetiklenerek.)

öğrenci dersi ilk aldığında devam zorunluluğu olacak.

eğer öğrenci aynı dersi önceki alışında devam zorunluluğunu tamamladıysa sonraki alışlarında zorunlu olmayacak ama tamamlamadıysa (yani devam oranı düşükse) sonraki alışında da devam zorunlu olacak.

Devam oranı devamsızlığın toplam dönemlik ders saatine oranlanacak ve devam zorunluluğu yüzdesi hesaplanarak gereken yüzdenin altında kalan öğrenci devamsızlıktan dolayı doğrudan dersten kalacak. (FF)

Devam zorunluluğunu tamamlayan öğrencinin final notu girildiğinde hem finalinin hem de vize%40+final%60 ortalamasının 50 puanın üstünde olması durumunda öğrencinin harf notu AA BA BB CB CC harf notlarından birine ait olacak ve dersi geçti sayılacak. Bunlardan biri 50 puanın altında kaldığında öğrencinin harf notu FF düşecek ve kaldı sayılacak.

Eğer öğrencinin final notu veya ortalaması 50 nin altında kaldıysa büt notu girildiğinde yeni ortalaması vize%40 + büt%60 olarak hesaplanacak, final devreden çıkacak.

Derslerin varsayılan devam zorunluluğu oranı %30 olacak ve bir dersten o dönem içinde varsayılan olarak 42 saat olacak. (yani devamsızlık 13’ten fazla olursa öğrenci devam zorunluluğunu geçemiyor ve dersten kalıyor.)

Ortalama, harf notu ve geçti geçmedi durumunu hesaplayarak tabloya verilerini yazan bir trigger planlanmalı.

Bu trigger ilk olarak öğrencinin devamsizlik sayısını oranlayarak 13 saati geçmişse otomatik olarak FF yazmalı. Geçmemişse diğer ortalama hesaplarına geçebilir.

“Aynı öğrenci aynı derse bir daha kayıt yapmak istediğinde”, bir önceki dönemkodu ile aynı dersi almış olduğu eski kayıtta devam zorunluluğunu geçip geçmediğine bakan ve eğer devam zorunluluğu sürüyorsa yeni kaydın devam zorunlu kısmındaki zorunluluğu ZORUNLU (TRUE) yapan bir trigger yazılmalı.

-- 1. Tablo yapısı varsayımı:

-- ogrenci\_ders(

-- ogr\_no INT,

-- ders\_id INT,

-- donemkodu VARCHAR,

-- vizenotu DECIMAL,

-- finalnotu DECIMAL,

-- butunlemenotu DECIMAL,

-- devamsizlik INT,

-- devamzorunlulugu BOOLEAN,

-- ortalama DECIMAL,

-- harfnotu VARCHAR(2),

-- gectimi BOOLEAN

-- )

-- 2. Fonksiyon: Ortalama hesaplama, harf notu belirleme, devam kontrolü

CREATE OR REPLACE FUNCTION hesapla\_ogrenci\_durum()

RETURNS TRIGGER AS $$

DECLARE

toplam\_saat CONSTANT INT := 42;

devamsizlik\_limiti CONSTANT INT := 13;

devamsizlik\_orani NUMERIC;

BEGIN

-- Devamsızlık oranı hesapla

devamsizlik\_orani := NEW.devamsizlik::NUMERIC / toplam\_saat;

-- Devamsızlık limiti aşılmışsa FF

IF NEW.devamsizlik > devamsizlik\_limiti THEN

NEW.harfnotu := 'FF';

NEW.gectimi := FALSE;

RETURN NEW;

END IF;

-- Ortalama hesapla

IF NEW.butunlemenotu IS NOT NULL THEN

NEW.ortalama := NEW.vizenotu \* 0.4 + NEW.butunlemenotu \* 0.6;

ELSE

NEW.ortalama := NEW.vizenotu \* 0.4 + NEW.finalnotu \* 0.6;

END IF;

-- Geçme kontrolü

IF NEW.finalnotu >= 50 AND NEW.ortalama >= 50 THEN

NEW.gectimi := TRUE;

ELSIF NEW.butunlemenotu IS NOT NULL AND NEW.ortalama >= 50 THEN

NEW.gectimi := TRUE;

ELSE

NEW.gectimi := FALSE;

END IF;

-- Harf notu

IF NEW.ortalama >= 88 THEN

NEW.harfnotu := 'AA';

ELSIF NEW.ortalama >= 77 THEN

NEW.harfnotu := 'BA';

ELSIF NEW.ortalama >= 66 THEN

NEW.harfnotu := 'BB';

ELSIF NEW.ortalama >= 56 THEN

NEW.harfnotu := 'CB';

ELSIF NEW.ortalama >= 50 THEN

NEW.harfnotu := 'CC';

ELSE

NEW.harfnotu := 'FF';

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

-- 3. Trigger: güncellemede çalışacak

DROP TRIGGER IF EXISTS trg\_hesapla\_ogrenci\_durum ON ogrenci\_ders;

CREATE TRIGGER trg\_hesapla\_ogrenci\_durum

BEFORE INSERT OR UPDATE ON ogrenci\_ders

FOR EACH ROW

EXECUTE FUNCTION hesapla\_ogrenci\_durum();

-- 4. Devam zorunluluğu aktarımı ve donemkodu artırımı

CREATE OR REPLACE FUNCTION devam\_zorunlulugu\_aktar()

RETURNS TRIGGER AS $$

DECLARE

onceki\_kayit RECORD;

BEGIN

SELECT \* INTO onceki\_kayit FROM ogrenci\_ders

WHERE ogr\_no = NEW.ogr\_no AND ders\_id = NEW.ders\_id

ORDER BY donemkodu::INT DESC

LIMIT 1;

IF FOUND THEN

NEW.donemkodu := (onceki\_kayit.donemkodu::INT + 1)::TEXT;

IF onceki\_kayit.devamzorunlulugu = TRUE AND onceki\_kayit.devamsizlik > 13 THEN

NEW.devamzorunlulugu := TRUE;

ELSE

NEW.devamzorunlulugu := FALSE;

END IF;

ELSE

NEW.donemkodu := '1';

NEW.devamzorunlulugu := TRUE;

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

-- 5. Trigger: öğrenci aynı dersi tekrar aldığında çalışacak

DROP TRIGGER IF EXISTS trg\_devam\_zorunlulugu\_aktar ON ogrenci\_ders;

CREATE TRIGGER trg\_devam\_zorunlulugu\_aktar

BEFORE INSERT ON ogrenci\_ders

FOR EACH ROW

EXECUTE FUNCTION devam\_zorunlulugu\_aktar();

Bu triggerları test etmek için veritabanına veri girişi yaptık:

-- Öğrenci 1: Devamsızlığı 5 saat (devam zorunluluğu tamamlandı)

INSERT INTO ogrenci\_ders (ogr\_no, ders\_id, donemkodu)

VALUES (1001, 1, '1');

-- Öğrenci 2: Devamsızlığı 15 saat (devam zorunluluğunu geçemedi)

INSERT INTO ogrenci\_ders (ogr\_no, ders\_id, donemkodu)

VALUES (1002, 1, '1');

----------------------------------------------------

UPDATE ogrenci\_ders

SET vizenotu = 50

WHERE ogr\_no = 1001 AND ders\_id = 1 AND donemkodu = '1';

UPDATE ogrenci\_ders

SET vizenotu = 50

WHERE ogr\_no = 1002 AND ders\_id = 1 AND donemkodu = '1';

-----------------------------------------------------

UPDATE ogrenci\_ders

SET finalnotu = 40

WHERE ogr\_no = 1001 AND ders\_id = 1 AND donemkodu = '1';

UPDATE ogrenci\_ders

SET finalnotu = 40

WHERE ogr\_no = 1002 AND ders\_id = 1 AND donemkodu = '1';

------------------------------------------------------

-- Öğrenci 1001'nin 1 numaralı dersi için devamsızlık 13 saat oldu

UPDATE ogrenci\_ders

SET devamsizlik = 13

WHERE ogr\_no = 1001 AND ders\_id = 1 AND donemkodu = '1';

-- Öğrenci 1002'nin 1 numaralı dersi için devamsızlık 14 saat oldu

UPDATE ogrenci\_ders

SET devamsizlik = 15

WHERE ogr\_no = 1002 AND ders\_id = 1 AND donemkodu = '1';

-------------------------------------------------------

UPDATE ogrenci\_ders

SET butunlemenotu = 60

WHERE ogr\_no = 1001 AND ders\_id = 1 AND donemkodu = '1';

UPDATE ogrenci\_ders

SET butunlemenotu = 60

WHERE ogr\_no = 1002 AND ders\_id = 1 AND donemkodu = '1';

--------------------------------------------------------

UPDATE ogrenci\_ders

SET devamsizlik = 15

WHERE ogr\_no = 1001 AND ders\_id = 1 AND donemkodu = '1';

--------------------------------------------------------

UPDATE ogrenci\_ders

SET devamsizlik = 12

WHERE ogr\_no = 1001 AND ders\_id = 1 AND donemkodu = '1';

UPDATE ogrenci\_ders

SET devamsizlik = 12

WHERE ogr\_no = 1002 AND ders\_id = 1 AND donemkodu = '1';

--------------------------------------------------------

INSERT INTO ogrenci\_ders (ogr\_no, ders\_id)

VALUES (1002, 1);

---------------------------------------------------------

INSERT INTO ogrenci\_ders (ogr\_no, ders\_id)

VALUES (1001, 1);

---------------------------------------------------------

TRUNCATE TABLE ogrenci\_ders RESTART IDENTITY CASCADE;