Laboratuvar 6
TABLOLAR VE KISITLAR

Tablolar veritabanında yer alan en basit veri saklama yapılarıdır. Bir tablo temelde kendisini oluşturan sütun ve bu sütunların özelliklerinden oluşmuş iki boyutlu bir yapıdır. İlişkisel veritabanlarında tablolar arasında da çeşitli ilişkiler oluşturulmuştur.
Kısıtlar ise veritabanındaki veri bütünlüğünü sağlamak amacıyla oluşturulmuş kurallar olarak tanımlanabilir. Kısıtlar sayesinde veri bütünlüğünün bozulması engellenmiş olur.

TABLOLAR

Tablo ilişkisel veritabanlarında verileri saklamak için kullanılan sütunlar ve satırlardan oluşan yapıdır. Her tablonun kendisine ait özel bir ismi vardır. Bir tablo için kullanılan bir isim ayni şema içerisinde başka bir tabloya daha verilemez. Tabloda yer alan sütunların veritipi, büyüklük vs. gibi özellikleri vardır. Yine bir tablo içerisinde ayni sütun ismi birden fazla sütuna verilemez.

TABLO OLUŞTURMAK

Tablo oluşturmak için gerekli SQL cümlesi sözdizimi aşağıda verilmiştir.
CREATE TABLE <tablo_adı> (
sütun_1 veritipi [DEFAULT ifade] [Kısıt],
…);

VERİTİPLERİ

CHAR(sayı): Sabit uzunluktaki alfasayısal verilerin tutulabildiği alanlar için kullanılır. Oracle 7 ve saha önceki sürümler için bu alanın uzunluğu en fazla 255 karakter olabilir. Oracle 8 ve sonrasında 2000 karakter uzunluğundadır. Eğer, sayı ile ifade numaradan daha kısa uzunlukta veriler girilirse Oracle kaydın soonuna boşluk ekleyerek sabit uzunluğa kadar getirir.

VARCHAR2(sayı): Değişken uzunluklu alfasayısal verilerin tutulduğu alanlar için kullanılır. Oracle 7 ve önceki sürümlerinde 2000 karakter, Oracle 8 ve sonraki sürümlerinde 4000 karakter uzunluğunda bilgi girilebilir.

NUMBER(n,p): Tamsayı ve Geçel sayılar için kullanılan sayısal veri tipidir. Tam kısım en fazla 38 basamak olabilir. Ondalık kısmın basamak sayısı da -84 ile 127 arasında değişmektedir. Number veri tipinden türetilmiş int[eger], dec[imal], smallint ve real veri tipleri de kullanılabilmektedir.

DATE: Tarih tutan alanlar için kullanılır. Bu tip alanlarda, tarih bilgileri ve saat bilgileri tutulabilir. Tarih formatları Oracle yüklenirken seçilen dile göre değişir. Amerikan standartı için ‘DD-MON-YY’ dir. Yani bir tarih ’03-MAY-01’ şeklide görünür. NLS_DATE_FORMAT parametresi ile tarih formatı değiştirilebilir. Tarihsel alanlar üzerinde aritmetiksel işlemler yapılabilir. Sistem tarihi SYSDATE fonksiyonu kullanılarak öğrenilebilir. Sayısal veya karakter olarak tanımlı bir alandaki veriler TO_DATE fonksiyonu ile tarih tipine çevrilebilir.

LONG: 2 GB’a kadar bilgi tutabilen karakter alanlar için kullanılır. Bir tabloda bu tipten ancak bir adet alan tanımlanabilir. Long veri tipine sahip alanlar için index oluşturulamaz.

NOT1: Oracle'da boolean veri tipi yoktur. Bunun için char(1) ya da number(1) şeklinde tanımlama yapılıp kullanılabilir.

NOT2: Bir tablonun alanları kendi veri tipine uygun değerler alabildiği gibi bir de NULL değer alabilirler. NULL değeri sayısal oalrak 0'dan ve karakter olarak da boşluk karakterden (' ') farklıdır.

KISITLAR (CONSTRAINTS)

Kısıtlar veritabanı üzerinde iş kurallarını zorunlu kılmak ve tablolar arasında ilişki kurarak veri bütünlüğünü sağlamak amacıyla oluşturulur. İş kurallarını veritabanı tetikçileri ve uygulama yazılımı ile de sağlamak mümkündür. Bütünlük kısıtları veritabanına hatalı veri girişini engeller.
Oracle beş çeşit bütünlük kısıtını desteklemektedir:

- **NOT NULL:** NULL değerlerin girilmesini engeller. Tek bir sütun için çalışır, yani her bir sütun için ayrı ayrı belirtilmesi gerekir. Orcle varsayılan değer olarak NULL değerlerin girişine izin verir.
- **CHECK:** Kısıtta belirtilen koşulun sağlanıp sağlanmadığını kontrol eder. Eğer koşul sağlanmıyorsa veri girişine izin vermez.
- **UNIQUE:** İlgili sütun ya da sütunlara aynı kayıttan sadece bir tane girilmesine izin verir. Birden fazla aynı kayıta izin vermez.
- **PRIMARY KEY:** Bir tablodaki her bir kaydı benzersiz olacak şekilde belirtir. Yani birincil anahtar olarak belirtilen sütunlara aynı kaydın tekrar girilmesine ve NULL değer girişine izin vermez. Bir tabloda sadece bir tane PRIMARY KEY kısıtı olabilir.
- **FOREIGN KEY:** Tablolar arasında ortak sütunları kullanarak ana-çocuk ilişkisi kurar. Tanımlanan bir FOREIGN KEY diğer tablodaki PRIMARY KEY ya da UNIQUE sütuna referans verir.

NOT: UNIQUE kısıtı ile PRIMARY KEY kısıtı birçok yönden birbirine benzer ancak UNIQUE kısıtı NULL değer girişine izin verirken PRIMARY KEY kısıtı NULL değer girişine izin vermez.

KISITLARIN OLUŞTURULMASI

Kısıtlar CREATE TABLE veya daha sonra bahsedeceğimiz ALTER TABLE cümleleri ile oluşturulurlar. Kısıtımız sadece bir sütun ile ilgili ise sütun seviyesinde bu kısıt oluşturulabilir. Eğer kısıtımız birden fazla sütun içeriyor ise tablo seviyesinde bu kısıt oluşturulabilir; sütunlar parantez içerisinde ve aralarında virgül olacak şekilde verilmelidir.
Kısıt oluştururken kısıta bir isim verilmesi gerekir. Eğer bu isim verilmezse Oracle otomatik olarak SYS_Cn formatında bir isim verecektir. Burada n Integer veritipinde bir sayıdır. Örneğin SYS_C003290, SYS_C003199 sistem tarafında oluşturulmuş birer kısıttır.

NOT: Kısıtları oluştururken bu kısıtlara anlamlı birer isim vermek güzel bir alışkanlıktır. Daha sonra geliştirme veritabanı ile ürün veritabanı tablo karakteristikleri vs. karşılaştırılmak istendiği zaman tutarsız sistem tarafından oluşturulmuş isimler bu karşılaştırılmayı oldukça zor bir hale getirecektir.
Sütun seviyesinde ve tablo saviyesinde kısıt oluşturma ile ilgili genel sözdizimi aşağıda verilmiştir.

Sütun Seviyesi:
<sütun_adı> [CONSTRAINT <kısıt adı>] kısıt tipi,

Tablo Seviyesi:
... <sütunN>,
[CONSTRAINT <kısıt adı>] kısıt tipi (<sütun1>,<sütun2>,...)

Bu bölümün devamında her bir kısıt tipinin nasıl oluşturulduğu örneklerle açıklanacaktır.

NOT NULL Kısıtı

NOT NULL Kısıtı sütun seviyesinde tanımlanan bir kısıttır. Tablo seviyesinde tanımlanamaz. NOT NULL kısıtı ilgili sütuna NULL değer girişine izin vermez, tabloya bir kayıt eklenirken mutlaka ilgili sütun verisi belirtilmelidir. NOT NULL kıstının sözdizimi aşağıda verilmiştir:

[CONSTRAINT <kısıt adı>] [NOT] NULL

Aşağıdaki örnekte NOT NULL kısıtı olan iki sütunlu bir tablo oluşturulmaktadır.

```
SQL> CREATE TABLE test_kisit (
2       sirano NUMBER(4) CONSTRAINT test_kisit_sirano_nn NOT NULL,
3       tarih date NOT NULL);

Table created.

SQL> DESC test_kisit
 Name        Null?      Type
 SIRANO      NOT NULL   NUMBER(4)
 TARIH       NOT NULL   DATE
```

Bu örnekte SIRANO sütunu için oluşturulan NOT NULL kısıtı için *test_kisit_sirano_nn* ismi verilmiştir. TARIH sütunu için oluşturulan NOT NULL kısıtı için herhangi bir isim verilmemiştir. Bu nedenle bu kısıt için ORACLE otomatik olarak SYS_C ile başlayan bir isim verecektir.

CHECK Kısıtı

Check kısıtını sütun ya da tablo seviyesinde tanımlamak mümkündür. Check kısıtı bir koşul cümlesi içerir. İlgili sütuna ya da tabloya veri girişi yapılırken sadece bu koşula uyan kayıtların süruna ya da tabloya eklenmesine izin verir. CHECK kısıtının sözdizimi aşağıda verilmiştir:

[CONSTRAINT <kısıt adı>] CHECK (<koşul>)

CHECK kısıtı oluştururken aşağıdaki maddelerin dikkate alınması gerekir:

- CHECK cümlesi sonucu mutlaka doğru-yanlış şekilde bool bir değer olmalıdır.
- CHECK cümlesi aynı satırda yer alan diğer sütunlara referans verebilir.
- CHECK koşulu oluşturulurken ortam fonksiyonları (SYSDATE, USER, USERENV VE UID) ve sözde sütunlar (ROWNUM, CURRVAL, NEXTVAL VE LEVEL) kullanılamaz.
- Bir sütuna ait birden fazla CHECK kısıtı olabilir.

UNIQUE KEY (TEKİL ANAHTAR) Kısıtı

UNIQUE kısıtı bir tablodaki bir veya birden fazla sütun ile oluşturulur ve kısıtı oluşturan sütun ya da sütun grubuna aynı verilerin girilmesini engeller. UNIQUE kısıtı NULL değer girişine izin vermez.

UNIQUE kısıtı tek bir sütun ile oluşturulduğu zaman ilgili sütuna aynı veri sadece bir kez girilebilir. Tek bir sütun içeren UNIQUE kısıtı oluşturmak için kullanılan sözdizimi aşağıda verilmiştir:

[CONSTRAINT <kısıt adı>] UNIQUE

UNIQUE kısıtı tek bir sütun ile oluşturulduğu gibi birden fazla sütun içerecek şekilde komposit bir yapıda da oluşturulabilir. Komposit yapıda oluşturulan UNIQUE kısıtı için en fazla 32 sütun kullanılabilir. Bu durumda ilgili sütun grubu için aynı veri grubu sadece bir kez girilebilir. Bu şekildeki bir kısıt tablo seviyesinde tanımlanmalıdır. Bununla ilgili sözdizimi aşağıda verilmiştir:

[CONSTRAINT <kısıt adı>] UNIQUE (<sütun1>,<sütun2>,...)

NOT: Oracle UNIQUE ksıtını oluşturan sütunlar için unique indeks oluşturur. Eğer unique ya da unique olmayan aynı sütunlardan ve aynı sütun sırasında bir indeks önceden oluşturulmuş ise bu var olan indeks kullanılır.

PRIMARY KEY (BİRİNCİL ANAHTAR) Kısıtı

PRIMARY KEY kısıtı UNIQUE kısıtına oldukça benzemektedir. UNIQUE kısıtında bahsedilen özelliklerden NULL değer girişine izin vermesi haricinde diğer tüm özellikler PRIMARY KEY kısıtında da vardır. UNIQUE kısıtı NULL değer girişine izin verirken PRIMARY KEY kısıtı NULL değer girişine izin vermez.

NOT: Bir tabloda sadece bir adet PRIMARY KEY kısıtı olabilir.

PRIMARY KEY kısıtının sütun seviyesi sözdizimi aşağıda verilmiştir:

[CONSTRAINT <kısıt adı>] PRIMARY KEY

PRIMARY KEY kısıtının tablo seviyesi sözdizimi de aşağıda yer almaktadır:

[CONSTRAINT <kısıt adı>] PRIMARY KEY (<sütun1>,<sütun2>,...)

PRIMARY KEY kısıtı oluşturulduğu zaman Oracle kısıtta yer alan her bir sütun için bir UNIQUE indeks ve NOT NULL kısıtı oluşturur.

FOREIGN KEY (YABANCI ANAHTAR) Kısıtı

FOREIGN KEY kısıtı ilişkisel bütünlük kısıtı olarak bilinir. Bu kısıtta bir tablodaki PRIMARY KEY ya da UNIQUE KEY ile diğer bir tablodaki ya da aynı tablodaki diğer sütun ya da sütun kombinezonu ile bir ilişki oluşturulur. Bu ilişki ana-çocuk ilişkisidir. Bu sayede eğer FOREIGN KEY kısıtı olan tabloya (çocuk) NULL harici bir kayıt eklenirken ilgili kaydın ana tobloda yer alması koşulu aranır. Ana tablo ile çocuk tablo arasında FOREIGN KEY kısıtı oluşturulacak sütunların veritipleri aynı olmalıdır.

Örneğin; CINSIYETLER tablosunda **E** ve **K** şeklinde iki kayıt olsun ve PERSONEL_NUFUS tablomuzda CINSIYET tablosunda referans veren bir FOREIGN KEY olsun. Bu durumda PERSONEL_NUFUS tablosunun CINSIYET sütununa **E** ve **K** harici bir değer girişine izin verilmeyecektir.

FOREIGN KEY kısıtı sütun ve tablo seviyesinde tanımlanabilir. Aşağıda sütun ve tablo seviyesine FOREIGN KEY kısıtı oluşturma ile ilgili sözdizimi görülmektedir:

Sütun seviyesinde;

[CONSTRAINT <kısıt adı>] REFERENCES [<şema>.]<tablo> [(<sütun1>,<sütun2>,...)]
[ON DELETE {CASCADE | SET NULL}]

Tablo seviyesinde;

[CONSTRAINT <kısıt adı>] FOREIGN KEY (<sütun1>,<sütun2>,...) REFERENCES [<şema>.]<tablo> [(<sütun1>,<sütun2>,...)] [ON DELETE {CASCADE | SET NULL}]

Aşağıdaki örnekte CINSIYETLER ve PERSONEL_NUFUS tablosu oluşturulmakta ve PERSONEL_NUFUS tablosundan (çocuk) CINSIYETLER tablosuna (ana) FK_PERSONEL_NUFUS FOREIGN KEY kısıtı oluşturulmaktadır. Daha sonra CINSIYETLER tablosuna **E** ve **K** iki değer giriliyor. Son INSERT cümlesinde ise PERSONEL_NUFUS tablosunun CINSIYET alanına **D** girişine izin verilmediği görülmektedir.

```
SQL> CREATE TABLE cinsiyetler (
2       cinsiyet CHAR(1) CONSTRAINT cinsiyetler_cinsiyet_pk PRIMARY KEY);
Table created.

SQL> CREATE TABLE personel_nufus(
2       personel_id NUMBER(10) CONSTRAINT personel_nufus_personelid_pk PRIMARY KEY,
3       dogum_yeri VARCHAR2(15),
4       dogum_tarihi date,
5       cinsiyet  CHAR(1) NOT NULL);
Table created.



SQL> ALTER TABLE personel_nufus
2 ADD CONSTRAINT personel_nufus_FK FOREIGN KEY(cinsiyet) REFERENCES cinsiyetler(cinsiyet);
Table altered.

SQL> INSERT INTO  cinsiyetler VALUES('E');
1 row created.

SQL> INSERT INTO  cinsiyetler VALUES('K');
1 row created.

SQL>COMMIT;
Commit complete.

SQL> INSERT INTO  personel_nufus
      VALUES(1, 'ANKARA', '10-MAY-1970','E');
1 row created.
```

```
SQL> INSERT INTO  personel_nufus
      VALUES(1, 'KONYA', '22-MAY-1980','D');
INSERT INTO  personel_nufus
*
ERROR at line 1:
ORA-02291:integrity constraint (IK. personel_nufus_FK) violated – parent key not found
```

Eğer CINSIYET alanına NOT NULL kısıtı konmuş olsaydı CINSIYET alanına NULL değer girişi yapabilirdik. FOREIGN KEY kısıtının NULL girişini engellemediğine dikkat ediniz.

Aşağıdaki örnekte PERSONEL_NUFUS tablosunun CINSIYET alanındaki NOT NULL kısıtı kaldırılmakta ve CINSIYET alanını NULL değer içeren bir kayıt eklemektedir.

```
SQL> ALTER TABLE personel_nufus
2       MODIFY (cinsiyet NULL)
Table altered.

SQL> INSERT INTO  personel_nufus
      VALUES(3, 'ADANA', '12-MAY-1972',NULL);
1 row created.
```

FOREIGN KEY kısıtı oluşturulurken kullanılan ON DELETE cümlesi ana tabloda bir kayıt silindiği zaman yapılacak işlemi belirtmektedir. ON DELETE CASCADE kullanılır ise ana tabloda bir kayıt silinirse çocuk tablodaki bununla ilgili kayıtlar otomatik olarak silinir. ON DELETE SET NULL kullanılır ise ana tabloda bir kayıt silinirse çocuk tablodaki bununla ilgili kayıtlara otomatik olarak NULL değer atanır. Eğer ON DELETE cümlesi kısıt oluşturulurken kullanılmaz ise Oracle çocuk tabloda kayıt varken ana tablodan kayıt silinmesine izin vermez.

NOT: USER_CONSTRAINTS, ALL_CONSTRAINTS, USER_CONS_COLUMNS ve ALL_CONS_COLUMNS görüntülerini kullanarak kısıtlar hakkında bilgi alınabilir.

Aşağıda basit bir tablo oluşturma cümlesi görülmektedir.

```
SQL> CREATE TABLE kitaplar(
  2  kitap_no NUMBER(4) CONSTRAINT kitaplar_kitap_no_pk PRIMARY KEY,
  3  kitap_adi VARCHAR2(20),
  4  yazar_adi VARCHAR2(30),
  5  fiyati NUMBER(10,2));

Table created.
```

Yukarıdaki örnekte KITAPLAR tablosu oluşturulmaktadır. CREATE TABLE kelimelerinden hemen sonra gelen isim tablo adıdır. Tabloya ait sütunlar ve bu sütunların özellikleri parantezler içerisinde verilir. Bu örnekte KITAPLAR tablosunun dört adet sütunu olduğu görülmektedir. Bu sütunlar sırasıyla KITAP_NO, KITAP_ADI, YAZAR_ADI VE FIYATI'dır. Buu sütunlardan ikisi VARCHAR2 veritipinde ve ikisi de NUMBER veritipindedir. Her bir sütun için bir veritipi belirtmek zorunludur.

Bir tablo oluştururken tablo ismi ve sütunların isimlerinden başka aşağıdakileri de belirtmek mümkündür.

- Sütunlar için varsayılan değerler belirtilebilir.
- Tablo ve/veya sütunlar için kısıtlar belirtilebilir.
- Tablonun tipi belirtilebilir: İlişkisel, geçici, indeks organizeli, dışsal veya nesnel.
- Tablonun depolama özelliği belirtilebilir.
- Tablonun yar alacağı tablo boşluğu belirtilebilir.
- Herhangi bir bölümleme veya lat bölümleme belirtilebilir.

NOT: Her tabloda en fazla 1000 sütun olabilir ve bunlar nesne-isimlendirme kurallarına uygun olmalıdır. AS altsorgu cümleleri kullanılarak sütun isimlendirmesi yapılmadan tablo oluşturulabilir. Tablolar genelde verisiz bir şekilde oluşturulur ve kayıtlar INSERT cümleleri ile eklenir.

TABLO VE SÜTUNLARIN İSİMLENDİRİLMESİ

Tablo ve sütun isimleri her bir tabloyu veya her bir tablo içerisindeki sütunları belirlemek için kullanılır. Tablo ve sütun isimlerinin olabildiğince ne ile ilgili olduğunu tanımlayıcı olması kullanım açısından önemlidir. Tablo ve sütun isimlendirme kuralları aşağıda verilmiştir:

- Tablo ve sütun isimleri en az 1 en fazla 30 karakter olabilir.
- Her bir isim bir harf ile başlamalıdır ve isim içerisinde sayı kullanılabilir.
- İsimlerde özel işaretlerden sadece dolar işaretine ($), diyez işaretine (#) ve alt çizgi işaretine (_) izin verilir.
- İsimlendirmede büyük küçük harf fark etmez. Oracle bütün karakterleri büyük harfe çevirerek saklar. Eğer küçük büyük harf duyarlı olması istenirse isim tırnak işareti (“) içerisinde verilmelidir.
- Veritabanında bir nesne tarafından kullanılan bir isim bir başka nesneye verilemezç (Farklı şemalarda aynı isim kullanılabilir.)

Bir tablodaki sütunları ve bunların veri tiplerini vs. görmek için SQL*Plus komutlarından **DESCRIBE** (DESC) kullanılabilir. Sözdizimi DESCRIBE <tablo_adı> şeklindedir. Tablo isimlendirmesi ve DESC koumutu için aşağıdaki örnekleriinceleyiniz:

1. İki sütundan oluşan TestTablosu oluşturuluyor.

```
SQL> CREATE TABLE testtablosu(
  2  kolon_1 NUMBER,
  3  kolon_2 varchar2(5));

Table created.
```

2. DESCRIBE komutu il tablo sütunları ve veri tipleri görüntüleniyor.

```
SQL> DESC testtablosu
 Name                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 KOLON_1                            NUMBER
 KOLON_2                            VARCHAR2(5)
```

3. USER_TABLES görüntüsü kullanılarak oluşturduğumuz tablo sorgulanıyor. Sorguda tablo isminde küçük harfler kullanıldığı için isim doğru olsa bile küçük-büyük harf farklılığından dolayı sorgu sonnucunda birşey dönmüyor.

```
SQL> select table_name
  2  from user_tables
  3  where table_name='TestTablosu';

no rows selected
```

4. Yine USER_TABLES görüntüsü kullanılarak oluşturduğumuz tablo sorgulanıyor ancak bu sefer tablo ismi tamamen büyük harflerden oluşuyor.

```
SQL> select table_name
  2  from user_tables
  3  where table_name='TESTTABLOSU';
TABLE_NAME
------------------------------
TESTTABLOSU
```

TABLOLARIN DEĞİŞTİRİLMESİ

Bir tablo oluşturulduktan sonra çeşitli nedenlerle tablo yapısında değişiklik yapmak gerekir. Tablolar üzerinde yapılabilecek değişiklikler genel olarak şunlardır:

- Sütun tanımın ve sütunun varsayılan değerini değiştirilmesi
- Yeni bir sütun eklenmesi
- Var olan bir sütunun düşürülmesi (silinmesi)
- Tablo tanımının değiştirilmesi
- Tablo kısıtlarının değiştirilmesi, yeni kıstıların eklenmesi ya da var olan kısıtların düşürülmesi
- Tablonun isminin değiştirilmesi ya da tablonun düşürülmesi

Tabloların değiştirilmesi için ALTER TABLE cümleleri kullanılır.

TABLOYA SÜTUN EKLEME

Var olan bir tabloya sütun eklemek için kullanılan sözdizimiaşağıda görülmektedir:

ALTER TABLO[<şema>.]<tablo adı>
ADD <sütun_tanımı>;

Tabloya yeni bir sütun eklendiği zaman daima tablonun en sonuna eklenir. Yeni sütun eklendiği zaman tabloda var olan önceki kayıtların bu sütununa ait değerleri NULL olur.

Eğer tabloya birden fazla sütun eklenecek ise sütun tanımları parantez içerisinde verilmeli ve virgülle birbirinden ayrılmalıdır. Eğer yeni eklenen sütun için varsayılan değer belirtilmiş ise (DEFAULT değer varsa) tabloda yer aln tüm kayıtlara otomatik olarak varsayıln değer atanır.

Tabloya yeni bir sütun eklerken eğer tabloda daha önceden kayıt var ise NOT NULL kısıtı belirtilemez. NOT NULL kısıtı eklemek için aşağıdaki aşamalar takip edilmelidir:

- **A.** Tabloya yeni sütunu NOT NULL kısıtı olmadan ekleyiniz.
- **B.** Yeni eklenen sütunu ilgili veriler ile güncelleyiniz.
- **C.** NOT NULL kısmını ekleyiniz.

Eğer NOT NULL kısıtı varsayılan değer ile birlikte belirtilirse tabloda kayıt olsa bile NOT NULL kısıtını eklemek mümkün olur.

SÜTUNLARIN DEĞİŞTİRİLMESİ

Tabloda var olan bir sütunu değiştirmek için kullanılan sözdizimi aşağıda verilmiştir:

ALTER TABLO[<şema>.]<tablo adı>
MODIFY <sütun_adı><sütunun yeni özellikleri>;

Yukarıdaki sözdiziminde < sütunun yeni özellikleri> kısmında, veritipi, varsayılan değer veya sütun kısıtları yer alabilir. Değiştirilmek istenen özellikler burada belirtilir. Belirtilmeyen özellikler değişmeden kalır. Eğer birden fazla sütunun özellikleri değiştirilecek ise bunları parantez içinde virgülle ayırarak yazmak gerekir.

NOT: MODIFY cümlesinde yer alan DEFAULT değer sadece tabloya yeni eklenen kayıtları etkiler. Tabloda önceden yer alan NULL değerleri etkilemez. Bir sütuna ait varsayılan değerleri, yani DEFAULT değeri kaldırmak istenirse sütun DEFAULT NULL ile yeniden tamamlanmalıdır.

SÜTUNLARIN DÜŞÜRÜLMESİ

Tabloda gereksiz yere oluşturulmuş ya da veritabanı tasarımındaki değişiklikler nedeniyle artık ihtiyaç kalmamış sütunlar tablodan düşürülür. Sütunun düşürülmesi ile tabloda yer alan sütunun tablodan çıkarılması ya da silinmesi ifade edilmektedir. Tabloda yer alan bir sütunu düşürmek ile ilgili sözdizimi aşağıda verilmiştir.

ALTER TABLO[<şema>.]<tablo adı>
DROP {COLUMN <sütun_adı>|<sütunların adı>}
[CASCADE CONSTRAINTS]

Sadece bir sütun düşürülecek ise DROP COLUMN kullanılır. Eğer birden fazla sütun düşürülecek ise DROP ifadesinden sonra parantez içerisinde virgülle ayrılmış biçimde sütun adları verilmelidir. Tablodan bir sütun düşürüldüğü zaman ilgili sütunun yer aldığı indeksler ve kısıtlarda düşürülür. Eğer düşürülen sütun çoklu-sütun kısıdında kullanılmış ise CASCADE CONSTRAINTS ifadesi de ilave edilir.

Zaman zaman sütunu tablodan hemen düşürmek yerine ilgili sütunu kullanım dışı bırakmak daha uygundur. Örneğin veritabanı üzerinde yoğun işlemlerin yapıldığı bir zamanda sütunu düşürmek yerine kullanım dışı bırakmak tercih edilebilir. Çünkü bir sütun düşürüldüğünde Oracle bu tabloyu yeniden oluşturur. Özellikle sütun düşürmek istediğimiz tabloda çok fazla veri varsa bu işlem uzun zaman alacaktır. Bu nedenle sütunu o an için kullanım dışı bırakıp daha sonra düşürmek uygun görülebilir. Kullanım dışı bırakılan bir sütun tablo tanımı içinde artık görülemez.

Aşagıda tablodaki bir sütunu kullanım dışı bırakmak için kullanılan sözdizimi görülmektedir:
ALTER TABLO[<şema>.]<tablo adı>
SET UNUSED {COLUMN <sütun_adı>|<sütunların adı>}

[CASCADE CONSTRAINTS]

Kullanım dışı bırakılan sütunu düşürmek için kullanılan sözdizimi aşağıda görülmektedir:

ALTER TABLO[<şema>.]<tablo adı>
DROP { UNUSED COLUMNS | COLUMNS CONTINUE}

COLUMNS CONTINUE	Daha önce yapılan bir sütun düşürme işlemi kesilmiş, yarıda kalmış ise düşürme işlemini tamamlamak için kullanılır.
UNUSED COLUMNS	Kullanım dışı bırakılan sütunları düşürmek için kullanılır. Düşürme işlemi yapılırken eğer birden fazla sütun kullanım dışı bırakılmış ise hepsi birden düşürülür. Sütunlar içerisinde seçim yapılamaz.

KISITLARIN DÜŞÜRÜLMESİ

Kısıtları düşürmek için ALTER TABLE cümlesi kullanılır. Herhangi bir kısıt, ismi belirtilmek suretiyle düşürülebilir.

ALTER TABLO[<şema>.]<tablo adı>
DROP CONSTRAINT <kısıt_adı>

NOT NULL kısıtını düşürmek için ALTER TABLE MODIFY cümlesi kullanılır.

ALTER TABLO[<şema>.]<tablo adı>
MODIFY <sütun_adı> NULL

UNIQUE kısıtını düşürmek için ALTER TABLE DROP UNIQUE cümlesi kullanılır. Eğer ilgili kısıta FOREIGN KEY referansı verilmiş ise ilgili FOREIGN KEY kısıtını da düşürmek için CASCADE kelimesi kullanılır. UNIQUE kısıtı düşürülürken UNIQUE kısıtını oluşturan sütunlar belirtilmelidir:

ALTER TABLO[<şema>.]<tablo adı>
DROP UNIQUE <sütun_adı1, sütun_adı2> [CASCADE]

PRIMARY KEY kısıtını düşürmek için ALTER TABLE DROP PRIMARY KEY cümlesi kullanılır. Eğer ilgili kısıta FOREIGN KEY referansı verilmiş ise FOREIGN KEY kısıtını da düşürmek için CASCADE kelimesi kullanılır.

ALTER TABLO[<şema>.]<tablo adı>
DROP PRIMARY KEY [CASCADE]

TABLOLARIN DÜŞÜRÜLMESİ

Bazı zamanlarda değişik nedenlerle oluşturulan tabloların düşürülmesi gerekir. Tablo düşürme işlemi oldukça basittir. Aşağıda tablo düşürülmesi için gerekli sözdizimi görülmektedir:

DROP TABLE [<şema>.]<tablo adı> [CASCADE CONSTRAINTS]

Bir tablo düşürüldüğü zaman:

- Tablo tanımı ve tablo verileri silinir.
- Tablo ile ilişkili indeksler, kısıtlar, tetikçiler ve yetkilerde düşürülür.
- Tablo düşürüldükten sonra işlem geri alınamaz.
- Tabloya referans veren görüntüler, materyal görüntüler veya diğer prosedür veya fonksiyonlar düşürülemez. Sadece geçersiz olarak işaretlenir.
- Eğer tablodaki birincil anahtara ya da benzersizlik anahtarına referans veren bütünlük kısıtları varsa CASCADE CONSTRAINTS ifadesi ilave edilmelidir.

NOT: Eğer bir tabloyu düşürmeden sadece içeriğini silmek isteseniz tabloyu drop edip yeniden oluşturmak yerine TRUNCATE cümlesi kullanabilirsiniz.

TABLO ADININ DEĞİŞTİRİLMESİ

Tablo isimlerinin içerdiği veri ile ilişkili olması kullanım kolaylığı açısından önemlidir. Yoksa tablo sayısının oldukça fazlalaştığı bir sistemde tabloları ayırt etmek ve doğru sorguları yazmak oldukça zorlaşır. Tablo ismini değiştirmek için RENAME cümlesi kullanılır. Tablo adını değiştirmek için kullanılan RENAME cümleleri diğer veritabanı nesnelerinin adlarını değiştirmek için de kullanılır. Bununla ilgili sözdizimi aşağıda görülmektedir:

RENAME <eski_ad> TO <yeni_ad>

Tablo adı değiştirildiği zaman:

- Oracle otomatik olarak bütünlük kısıtlarını, indeksleri ve yetkileri yeni tabloya aktarır.
- Adı değiştirilen tabloyu kullanan tüm nesneler geçersiz konuma getirilir.

ORACLE'DA YER ALAN TABLOLAR

Oracle'da yer alan tabloları iki gruba ayırmak mümkündür.

- Kullanıcı tabloları ve görüntüleri
- Veri sözlüğü tabloları ve görüntüleri

Kullanıcı tabloları, veritabanı kullanıcıları tarafından oluşturulan tablolardır. Bu tablolar kullanıcılar tarafından yönetilir ve kullanıcılara ait verilerin saklanması için kullanılır.

Veri sözlüğü tabloları ve görüntüleri, Oracle veritabanı sunucusu tarafından oluşturulur ve yönetilir. Bu tablolarda veritabanı ile ilgili bilgiler yer alır. Veri sözlüğünde yer alan tabloların hepsine SYS (SYSTEM) kullanıcısı sahiptir. Temel tablolar genelde sorgular için pek kullanılmaz çünkü buradaki verileri anlamak zordur. Bu nedenle kullanıcılar genelde bu taboların içeriğinin daha formatlı bir şekilde sunulduğu görüntüleri kullanarak sorgulamaları yaparlar. Bu görüntülerde yer alan verileri anlamak daha kolaydır. Bu tablolarda yer alan verilere örnek olarak kullanıcı bilgileri, kullanıcı yetkileri, tablo bilgilerini verebiliriz.

Veri sözlüğü görüntüleri dört kategoride toplanmaktadır. Her kategori kullanım alanı ile ilgili olarak bir ön ek ile başlar. Bunlar aşağıdaki tabloda görülmektedir.

Veri Sözlüğü Görüntüleri Ön Ekleri ve Açıklamaları

Ön Ek	Açıklama
USER_	Bu görüntüler kullanıcı tarafından sahip olunan nesneler hakkında bilgi içerir.
ALL_	Bu görüntüler kullanıcı tarafından erişilebilen tüm tablolar (ilişkisel ve nesnel tablolar) hakkında bilgi içerir.
DBA_	Bu görüntülere erişim sınırlandırılmıştır. Sadece DBA rolüne sahip kullanıcılar tarafından erişilebilir.
V$	Bu görüntüler dinamik performans görüntüleri olarak adlandırılır. Veritabanı sunucusu, performans, bellek, kilitler vs. hakkında bilgiler içerir.

Kaynak: Murat Obalı – Oracle 10g – PUSULA

Derleyen: Şebnem Çoban