

SQL DİLİ

VERİ İŞLEME DİLİ (DML)

SQL KATEGORİLERİ

İŞLEV	AÇIKLAMA
DDL (Data Definition Language – Veri Tanımlama Dili)	Veritabanı şemasını tanımlamak ve veritabanındaki veritabanı nesnelerinin yapısını oluşturmak ve değiştirmek için kullanılır.
DQL (Data Query Language - Veri Sorgu Dili)	Veritabanındaki verileri sorgulamak ve sonuçları listelemek için kullanılır.
DML (Data Manipulation Language - Veri İşleme Dili)	Veritabanında bulunan verilerin değiştirilmesi (güncellenmesi, eklenmesi, silinmesi) için kullanılır.
DCL (Data Control Language – Veri Kontrol Dili)	Grup ve kullanıcılara veritabanı sisteminde verilen izinleri ve konan yasakları düzenlemek amacıyla kullanılır.
TCL (Transaction Control Language – İşlem Kontrol Dili)	Veritabanındaki işlemlerle ilgili bütünlük vb. için kullanılır.

Kaynak: "İlişkisel Veritabanı Tasarlama ve SQL ile Geliştirme", Dr. Selçuk Kıran, Nobel Akademik Yayıncılık

VERİ İŞLEME DİLİ (DML)

Select, Insert, Update, Delete

Select

- Veritabanındaki verileri seçmeyi sağlar.

Insert

- Veritabanına yeni kayıt ekler.

Update

- Veritabanında bulunan kayıtlar üzerinde değişiklik (güncelleme) yapmak için kullanılır.

Delete

- Veritabanından veri silmek için kullanılır.

Kaynak: <http://www.cansuayvazguven.com>

SELECT INTO

Belirli bir tablodan yeni oluşturulacak bir tabloya veri kopyalamak için kullanılır. Tüm alanlar eklenmek istendiğinde * kullanılırken, spesifik alanlar için alan isimleri belirtilir.

Eğer yeni oluşturulacak tablonun farklı bir database'de oluşturulması isteniyorsa IN 'databaseAdı' eklenir.

Kaynak: <http://https://bukettbodur.wordpress.com/2020/08/14/sql-egitim-serisi-5-select-into-insert-into-insert-into-select/>

SELECT INTO

```
SELECT *  
INTO yenitablo IN 'databaseAdı'  
FROM eskitablo  
WHERE koşul
```

```
SELECT alan1, alan2, alan3, ...  
INTO yenitablo  
FROM eskitablo  
WHERE koşul
```

Kaynak: <http://https://bukettbodur.wordpress.com/2020/08/14/sql-egitim-serisi-5-select-into-insert-into-insert-into-select/>

SELECT INTO

```
select*into Categories_copy  
from Categories where  
CategoryID>5
```

	CategoryID	CategoryName	Summary	Description	Picture
1	6	Meat/Poultry	NULL	Prepared meats	0x151C2F00020000000D000E0014002100FFFFFFFF4269746...
2	7	Produce	NULL	Dried fruit and bean curd	0x151C2F00020000000D000E0014002100FFFFFFFF4269746...
3	8	Seafood	NULL	Seaweed and fish	0x151C2F00020000000D000E0014002100FFFFFFFF4269746...
4	9	FastFood	NULL	hamburger, potatoes	NULL

Kaynak: <http://https://bukettbodur.wordpress.com/2020/08/14/sql-egitim-serisi-5-select-into-insert-into-insert-into-select/>

INSERT INTO

- Insert Into komutu ilgili tabloya kayıt atabilmemize yarayan bir komuttur.

```
INSERT INTO tabloAdı (alan1, alan2,alan3,..)  
VALUES (değer1, değer2,değer3,..)
```

- Tablo adından sonra alanları hangi sıra ile eklendiye değerler de aynı sıra ile eklenmelidir

Kaynak: <http://https://bukettbodur.wordpress.com/2020/08/14/sql-egitim-serisi-5-select-into-insert-into-insert-into-select/>

INSERT INTO

- Eğer tablodaki tüm alanlar için değer girilecekse tablo adından sonra parantez içinde alanların belirtilmesine gerek yoktur.

INSERT INTO tabloAdı

VALUES (değer1,değer2,değer3,..)

- Insert ederken dikkat edilecek en önemli şey Identity olan sütunlara veri girilmemesi gerektiğidir. Çünkü bu sütunlar her kayıta sırayla yeni bir sayı atmaktadır.

Kaynak: <http://https://bukettbodur.wordpress.com/2020/08/14/sql-egitim-serisi-5-select-into-insert-into-insert-into-select/>

INSERT INTO

- INSERT INTO** Categories (CategoryName, Description) **VALUES** ('FastFood','hamburger, potatoes')

	CategoryID	CategoryName	Summary	Description	Picture
1	1	Beverages	NULL	Soft drinks, coffees, teas, beers, and ales	0x151C2F000200000000D000E0014002100FFFFFFF4269746...
2	2	Condiments	NULL	Sweet and savory sauces, relishes, spreads, and ...	0x151C2F000200000000D000E0014002100FFFFFFF4269746...
3	3	Confections	NULL	Desserts, candies, and sweet breads	0x151C2F000200000000D000E0014002100FFFFFFF4269746...
4	4	Dairy Products	NULL	Cheeses	0x151C2F000200000000D000E0014002100FFFFFFF4269746...
5	5	Grains/Cereals	NULL	Breads, crackers, pasta, and cereal	0x151C2F000200000000D000E0014002100FFFFFFF4269746...
6	6	Meat/Poultry	NULL	Prepared meats	0x151C2F000200000000D000E0014002100FFFFFFF4269746...
7	7	Produce	NULL	Dried fruit and bean curd	0x151C2F000200000000D000E0014002100FFFFFFF4269746...
8	8	Seafood	NULL	Seaweed and fish	0x151C2F000200000000D000E0014002100FFFFFFF4269746...
9	9	FastFood	NULL	hamburger, potatoes	NULL

Kaynak: <http://https://bukettbodur.wordpress.com/2020/08/14/sql-egitim-serisi-5-select-into-insert-into-insert-into-select/>

INSERT INTO SELECT

Her defasında tek tek kayıt eklemek uzun zaman isteyen bir işlemdir. Eğer eklenecek veriler başka bir tabloda bulunuyorsa direkt tablolar arası kopyalama yapılabilir. Bunun için INSERT INTO SELECT komutu kullanılmaktadır.

Kaynak: <http://https://bukettbodur.wordpress.com/2020/08/14/sql-egitim-serisi-5-select-into-insert-into-insert-into-select/>

INSERT INTO SELECT

INSERT INTO tablo2 (alan1, alan2, alan3, ...)

SELECT alan1, alan2, alan3, ... **FROM** tablo1

WHERE koşul

Tablodaki tüm alanlar kopyalanacaksa alan isimlerinin belirtilmesine gerek yoktur.

INSERT INTO tablo2

SELECT * FROM tablo1

WHERE koşul;

Kaynak: <http://https://bukettbodur.wordpress.com/2020/08/14/sql-egitim-serisi-5-select-into-insert-into-insert-into-select/>

INSERT INTO SELECT

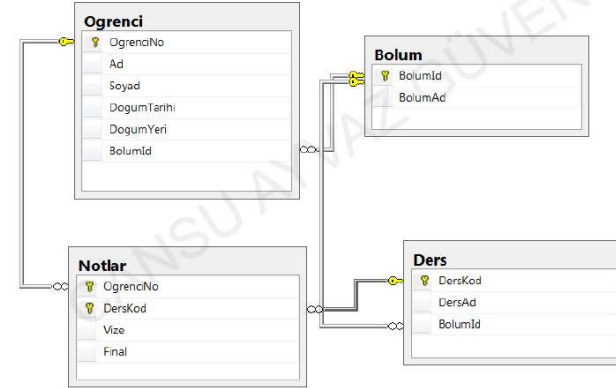
Farklı ya da benzer tablolar arasında kopyalama işlemi gerçekleştirilebilir ama belirtilen alanlar mutlaka aynı data tipinde ve aynı sırada ve sayıda olmalıdır.

- **INSERT INTO** Suppliers(CompanyName, ContactName,Address,city,Country)
SELECT CompanyName, ContactName,Address,city,Country **FROM**
Customers **WHERE** CustomerID='ANTON'

Results	Messages	SupplierID	CompanyName	ContactName	ContactTitle	Address	City	Region	PostalCode	Country	Phone	Fax	HomePage
1		31	Antonio Moreno Taqueria	Antonio Moreno	NULL	Mataderos 2312	México D.F.	NULL	NULL	Mexico	NULL	NULL	NULL

Kaynak: <http://https://bukettbodur.wordpress.com/2020/08/14/sql-egitim-serisi-5-select-into-insert-into-insert-into-select/>

Örnek Veritabanı



Kaynak: <http://www.cansuayvazguven.com>

UPDATE

Bir tablonun içerdiği değerler değiştirilmek istendiğinde UPDATE deyimi kullanılır.

UPDATE tablo

SET sütun1 =değer1, sütun2=değer2,...

WHERE şart

Kaynak: <http://www.cansuayvazguven.com>

UPDATE

Örnek: Ogrenci tablosunda Bolum Numarası 3 olan öğrencinin doğum yerini Sivas olarak değiştirmek için,

```
UPDATE Ogrenci
SET DogumYeri = 'Sivas'
WHERE BolumId = 3
```

Kaynak: <http://www.cansuayvazguven.com>

DELETE

Tabloda yer alan bir ya da daha fazla satırı silmek için DELETE deyimi kullanılır.

*DELETE [FROM] tablo
[WHERE şart]*

Kaynak: <http://www.cansuayvazguven.com>

DELETE

Örnek: OGRENCI tablosunun tüm kayıtlarını silmek için,

Delete From Ogrenci

Örnek: OGRENCI tablosunda öğrenci numarası 12 olan öğrenci kaydını silmek için,

Delete From Ogrenci
Where OgrenciNo = 12

Kaynak: <http://www.cansuayvazguven.com>

VERİ KONTROL DİLİ (DCL) Grant, Revoke, Deny

Veri Kontrol Dili (DCL), ilişkisel veritabanlarında depolanan verilere erişimi ve izin kontrolünü yönetmek için kullanılan Yapılandırılmış Sorgu Dilinin (SQL) bir alt kümesidir. SQL'in önemli bir bileşeni olan DCL, bir veritabanı sistemindeki kullanıcıların güvenlik ayarlarını, ayrıcalıklarını ve erişim düzeylerini tanımlamaya ve düzenlemeye odaklanır. DCL komutları, veritabanı yöneticilerinin hassas verileri koruyabilmesini, veri gizliliğini koruyabilmesini ve veritabanı ortamında yetkilendirme kurallarını uygulayabilmesini sağlar.

Kaynak: <https://appmaster.io/tr/glossary/veri-kontrol-dili-dcl>

DCL En İyi Uygulamalar

1. En Az Ayrıcalık İlkesi: Bu ilke, kullanıcılara yalnızca görevlerini yerine getirmek için gereken minimum ayrıcalıkların verilmesi gerektiğini öne sürer. Yöneticiler, ayrıcalıkları yalnızca görevlerini yerine getirmek için gereken kişilerle sınırlandırarak, yetkisiz erişim veya kazara veri kaybı riskini azaltabilir.
2. Rol Tabanlı Erişim Kontrolü: Veritabanı yöneticileri, bireysel kullanıcılar yerine rollere dayalı ayrıcalıklar atamalıdır. Bu yaklaşım, kullanıcı erişiminin yönetimini basitleştirir ve erişim kontrol politikalarının kuruluş genelinde tutarlı bir şekilde uygulanmasını sağlar.

Kaynak: <https://appmaster.io/tr/glossary/veri-kontrol-dili-dcl>

DCL En İyi Uygulamalar

3. Düzenli Denetimler: Kullanıcı erişimini incelemek ve erişim kontrolündeki potansiyel riskleri veya tutarsızlıkları belirlemek için periyodik denetimler yapılmalıdır. Düzenli denetimler, kullanıcı ayrıcalıklarının güncel kalmasını ve rolleri ve sorumluluklarıyla tutarlı olmasını sağlamaya yardımcı olur.
4. Parçalı Erişim Kontrolü: Erişim kısıtlamalarının belirli veri nesnelerine ve kullanıcı gereksinimlerine göre uyarlanması sağlamak için DCL, veritabanı, şema, tablo ve sütun seviyeleri gibi birden fazla seviyede uygulanmalıdır. Parçalı erişim kontrolünün uygulanması, hassas veriler üzerinde daha hassas kontrole olanak tanır ve genel veri güvenliğini artırır.

Kaynak: <https://appmaster.io/tr/glossary/veri-kontrol-dili-dcl>

DCL En İyi Uygulamalar

5. Görünümlerin ve Saklı Prosedürlerin Kullanımı: Kullanıcının belirli veri işlemlerine erişimini kısıtlamak için belirli izinlerle görünümler ve saklı prosedürler oluşturulabilir. Yöneticiler, DCL ile birlikte görünümleri ve saklı prosedürleri kullanarak veri güvenliğini daha da artırabilir ve veri manipülasyonu üzerinde daha sıkı kontrol sağlayabilir.

Kaynak: <https://appmaster.io/tr/glossary/veri-kontrol-dili-dcl>

DCL Temel Komutları

- İki temel DCL komutu GRANT ve REVOKE'dur.
- GRANT kullanıcılara ve kullanıcı gruplarına belirli ayrıcalıklar vermek için kullanılır.
- REVOKE daha önce verilen ayrıcalıkları kaldırmak için kullanılır.
- Bu komutlar, veritabanı yöneticilerinin, okuma, yazma, güncelleme ve silme işlemleri de dahil olmak üzere kullanıcıların verilerle gerçekleştirebileceği işlemleri denetlemesine olanak tanır.

Kaynak: <https://appmaster.io/tr/glossary/veri-kontrol-dili-dcl>

Temel SQL Yetkileri

- **select**: tablodaki veya görünümdeki verileri okuma ve sorgulama yetkisi.
- **insert**: tabloya yeni veri girme yetkisi.
- **update**: SQL update komutu kullanarak verileri değiştirme yetkisi.
- **delete**: veri/satır silme yetkisi.
- **references**: tablo yaratırken yabancı anahtar oluşturma yetkisi.
- **all privileges**: tüm yetkilerin verilmesi için kullanılır.

Kaynak: Database System Concepts, Silberschatz, Korth and Sudarshan, 9. Baskı

Görünümler ve Yetkilendirme

- Kullanıcılar görünümleri oluşturan tablolar üzerinde yetkilendirilmeksizin sadece görünümler için yetkilendirilebilirler.
- Bunun yanı sıra, bir görünümün yaratıcısı , o görünümü oluşturan tablo veya tablolarda sahip olduğu yetkilerden daha fazlasına sahip olmaz.

Örneğin *musteri* ve *borc* tablolarını kullanarak *musteri-borc* görünümünü yaratan bir kullanıcı *musteri* ve *borc* tablolarında sadece **okuma (select)** yetkisine sahip ise oluşturduğu görünümde de sadece **okuma** yetkisine sahip olur.

Kaynak: Database System Concepts, Silberschatz, Korth and Sudarshan, 9. Baskı

GRANT

- **Grant ifadesi:**
Kullanıcının kayıtlar üzerinde işlem yapmasına ve SQL komutlarını çalıştırmasına izin verir.

- **Kullanım Şekli**

grant <izin listesi>

on <izin alanı>

to <kullanıcı listesi>

<izin alanı> : tablo veya görünüm ismi

<kullanıcı listesi> : yetkilendirilecek kullanıcılar

Kullanıcı ismi

public, tüm kullanıcıları yetkilendirmek için kullanılır

rol (daha sonraki kısımda anlatılacaktır)

Kaynak: Database System Concepts, Silberschatz, Korth and Sudarshan, 9. Baskı

GRANT

Örnek:

grant select on *sube* to U1

- Bir kullanıcının herhangi bir yetkilendirme yapabilmesi için, veri tabanı yöneticisi ya da o yetkiye sahip bir kullanıcı olması gerekir.
- Kullanıcı kendinde olmayan bir yetkiyi başka kullanıcılara veremez.

Kaynak: Database System Concepts, Silberschatz, Korth and Sudarshan, 9. Baskı

Yetkilendirme Yetkisi

with grant option: ifadesi bir kullanıcının kendine verilen bir yetkiyi başka kullanıcılara verebilmesi yetkisini (hakkını) verir..

Örnek:

grant select on sube to U_1 with grant option

U_1 isimli kullanıcıya *sube* tablosu için **select** yetkisini ve bu yetkiyi diğer kullanıcılara aktarma yetkisini verir.

Kaynak: Database System Concepts, Silberschatz, Korth and Sudarshan, 9. Baskı

Roller

- **Roller**, ortak yetkilere sahip olacak olan bir grup kullanıcının tek bir komutla yetkilendirilmesi için kullanılır.
- Yetkilendirme (ve yetki iptal etme) işlemi kullanıcılara yapıldığı şekilde yapılır.
- Roller kullanıcılara ve hatta başka rollere verilebilir.

Kaynak: Database System Concepts, Silberschatz, Korth and Sudarshan, 9. Baskı

Roller

Örnekler:

create role memur
create role mudur

grant select on sube to memur
grant update (mevduat_miktari) on hesap
to memur

grant all privileges on hesap to mudur

grant memur to mudur

grant memur to ayse, mehmet
grant mudur to hasan

Kaynak: Database System Concepts, Silberschatz, Korth and Sudarshan, 9. Baskı

REVOKE

Daha önce verilen tüm izinleri iptal eder.

- **Kullanım Şekli**

revoke < izin listesi >

on < izin alanı > **from** < kullanıcı listesi > [**restrict** | **cascade**]

Örnek: **revoke select on sube from U_1, U_2, U_3 cascade**

- Bir kullanıcıya ait bir iznin iptal edilmesi, o kullanıcı tarafından yetkilendirilen tüm kullanıcıların izinlerinin iptaline neden olur
 - Böyle bir durumda ardışık iptallerin engellenmesi için **restrict** kullanılır:
Örnek: **revoke select on sube from U_1, U_2, U_3 restrict**
- Bu durumda U_1, U_2, U_3 kullanıcıları tarafından yetkilendirilen kullanıcıların izinleri iptal olmaz.

Kaynak: Database System Concepts, Silberschatz, Korth and Sudarshan, 9. Baskı

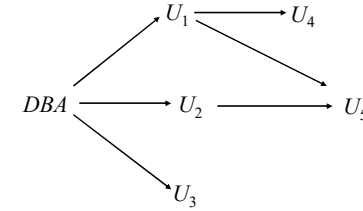
REVOKE

- <izin listesi > **all to** olarak yazıldığında kullanıcıya ait tüm izinler iptal edilir.
- If <kullanıcı listesi> **public** ise tüm kullanıcıların izinleri iptal olur.
- Eğer bir kullanıcı aynı yetkiyi birden fazla kullanıcı tarafından almış ise, kullanıcılardan birinin iznini iptal etmesi (veya ardışık izin iptalinden dolayı (cascade) iptal olması durumunda dahi, diğer kullanıcı tarafından verilen yetkiyi kullanmaya devam eder!
 - Veri tabanı Yöneticisi bu tip **çevrim** oluşturabilecek yetkilendirmeleri takip etmelidir ve oluşumlarını engellemelidir!

Kaynak: Database System Concepts, Silberschatz, Korth and Sudarshan, 9. Baskı

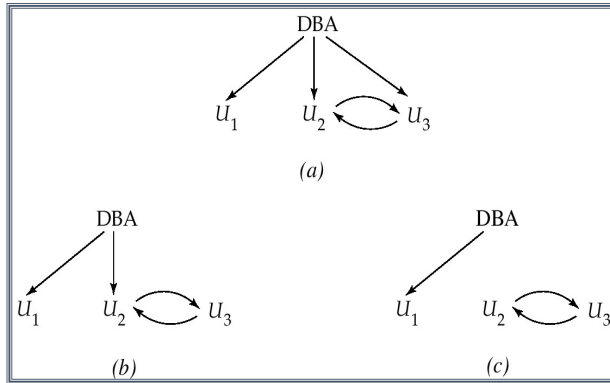
Yetkilendirme Grafiği

- Yetkilerin kullanıcıdan kullanıcıya geçişini göstermek için kullanılır.
- Grafiğin düğümleri kullanıcıları gösterir.
- Grafiğin kökü veri tabanı yöneticisini (DBA) gösterir.
- **Koşul:** Grafikteki tüm kenarlar DBA tarafından başlatılmış bir yolun parçası olmalıdır.
 - $U_i \rightarrow U_j$ gösterimi U_i isimli kullanıcının *sube* tablosu için *update* yetkisini U_j isimli kullanıcıya verdiğini ifade eder.



Kaynak: Database System Concepts, Silberschatz, Korth and Sudarshan, 9. Baskı

Yetkilendirme Grafiği



Kaynak: Database System Concepts, Silberschatz, Korth and Sudarshan, 9. Baskı

Yetkilendirme Grafiği

- Bir önceki yansıda yer alan örnekte;
- Kullanıcılar veri tabanı yöneticisinden bağımsız olan **çevrimler** oluşturmaya çalışabilirler;
 - DBA U_1 , U_2 , U_3 isimli kullanıcıları yetkilendirir.
 - U_2 , U_3 isimli kullanıcıyı yetkilendir.
 - U_3 tekrardan U_2 isimli kullanıcıyı yetkilendir.
 - DBA U_2 isimli kullanıcının yetkisini iptal eder.
- Bu durumda U_2 den U_3 'e ve U_3 den U_2 'ye olan izinlerin iptal edilmesi gerekir çünkü DBA dan U_2 'ye veya U_3 'e bağlı olan bir yol kalmamıştır.
- Veri tabanı Yöneticisi bu tip **çevrim** oluşturabilecek yetkilendirmeleri takip etmeli ve oluşumlarını engellemelidir.

Kaynak: Database System Concepts, Silberschatz, Korth and Sudarshan, 9. Baskı

DENY KOMUTU

GRANT komutunun tersidir, aynı şekilde kullanılır.

DENY (ALL | izinler) TO (izinVerilenler)

1-) MERYEM kullanıcısına tablo yaratmayı yasakladık.

```
DENY CREATE TABLE TO MERYEM
```

2-) MERYEM kullanıcısının öğrenciler tablosunda INSERT ve SELECT kullanmasını engelledik.

```
DENY INSERT, SELECT ON ogrenciler TO MERYEM
```

Kaynak: <https://webodasi.com/sql-grant-deny-revoke-komutlari/#>

İŞLEM KONTROL DİLİ (TCL) Commit, Rollback, Savepoint

COMMIT

Bu komut tüm işlemleri veritabanına kaydetmek için kullanılır.

Örnek:

```
DELETE FROM Students  
WHERE RollNo =25;  
COMMIT;
```

Kaynak: <https://www.guru99.com/tr/sql-commands-dbms-query.html>

ROLLBACK

Geri alma komutu, henüz veritabanına kaydedilmemiş işlemleri geri almanıza olanak tanır.

Kaynak: <https://www.guru99.com/tr/sql-commands-dbms-query.html>

SAVEPOINT

Bu komut, bir işlem içinde bir kayıt noktası ayarlamanıza yardımcı olur.

Örnek:

SAVEPOINT *kontrol5;*

Kaynak: <https://www.guru99.com/tr/sql-commands-dbms-query.html>