



# İŞLEM YÖNETİMİ VE EŞ ZAMANLILIK

# **İŞLEM YÖNETİMİ**

## **(TRANSACTION MANAGEMENT)**

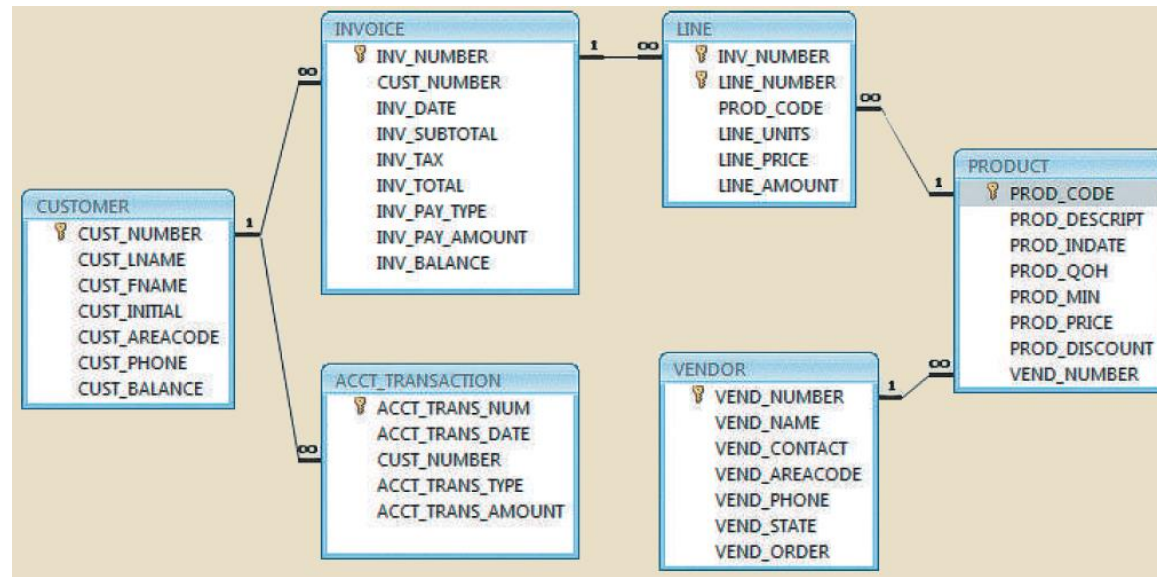
- İşlem (transaction), veri tabanındaki verilere erişen ve çoğunlukla bu veriler üzerinde değişiklikler yapan istekler dizisidir.
- Bir başka ifadeyle; mantıksal bir çalışma birimidir:
  - **Tamamen tamamlanmış veya durdurulmuş olmalıdır.**
  - **Hiçbir ara bitiş durumu kabul edilemez.**

### **İşlemler:**

- SELECT
- UPDATE
- INSERT
- SELECT, UPDATE ve INSERT ifadelerinin birleşimlerini içerirler.

# İŞLEM (TRANSACTION)-ÖRNEK

- İşlemlerin ne olduğunu ve nasıl çalıştıklarını göstermek için ilişkisel diyagram aşağıdaki şekilde gösterilmiştir.



Veritabanı tasarımında, örnek bir muhasebe programı verilmiştir.

- Ürün **PRODUCT** ve bu ürünün **tedarikçi bilgileri** **VENDOR**, **fatura** **INVOICE** ve **faturanın detayı** **LINE** tablolarında saklanmıştır.
- Müşterinin **borçlu olduğu toplam tutarı belirtmek** için (**CUST\_BALANCE**) değeri, **müşteri** (**CUSTOMER**) tablosunda saklanmıştır.
- CUST\_BALANCE** değeri, müşteri alışveriş yaptığında değişecektir.
- ACCT\_TRANSACTION** tablosu, müşteri hesabı etkinliğinin ayrıntılarını izlemek için tüm müşteri alımlarını ve ödemelerini kaydeder.

\* Müşteriye, ürün satıldığında **INVOICE** tablosunda yeni bir fatura kaydı eklenir. Ürün (**PRODUCT**) miktarında (envanter kaydında) azalma yapılır.

\* Müşteri hesap etkinliklerinde (**ACC\_TRANSACTION**) güncellemeler yapılır. Ayrıca müşterinin borç durumu/bakiyesi (**CUST\_BALANCE**) güncellenir.

# İŞLEM (TRANSACTION)

- Söz konusu veritabanında satış işleminin kısmen tamamlanmamış olmalıdır.

Örneğin yalnızca envanter kaydını veya yalnızca alacak hesapları güncellemek kabul edilemez. Bu durumda veritabanında tutarsızlıklar oluşacaktır.

- Başarılı bir işlem, veritabanının tutarlı olma durumuna göre değiştirir.
- **Veritabanının tutarlı olma durumu**, tüm veri bütünlüğü kısıtlamalarının karşılanması ile oluşur.

# İŞLEM (TRANSACTION)

- İşlemdeki tüm SQL ifadeleri başarıyla tamamlanmalıdır.

*SQL ifadelerinden herhangi biri başarısız olursa, tüm işlem, işlem başlamadan önce var olan orijinal veritabanı durumuna geri alınır.*

Gerçek dünya veritabanı işlemlerinin çoğu iki veya daha fazla veritabanı isteği ile oluşturulur.

# İŞLEM (TRANSACTION)

**Veri tabanı isteği**, bir uygulama programındaki veya işlemdeki tek bir SQL ifadesinin eşdeğeridir.

Örneğin, bir işlem iki UPDATE ifadesi ve bir INSERT ifadesinden oluşuyorsa, üç işlem veritabanı isteği kullanır.

Buna karşılık, her veritabanında fiziksel depolama ortamından okuyan veya bunlara yazılan birkaç giriş / çıkış (G / Ç) işlemi oluşturur.

# İşlem Sonuçlarının Değerlendirilmesi

- Tüm işlemler veritabanında değişikliklere yol açmaz.

Örneğin bir müşteri tablosunda bir kişiye ait bilgilerin listelenmesi.

Sorgu MÜŞTERİ tablosunda herhangi bir değişiklik yapmasa da, SQL kodu veritabanına eriştiği için bir işlemi temsil eder.

- Veritabanında tasarımında kurulan ilişkiler dikkate alınmalıdır. Kurulan ilişkilere göre yapılacak işlemler tablolara yansıtılmalıdır.

Örneğin 18 Ocak 2018 tarihinde 89-WRE-Q kodlu üründen bir birim 277.55 dolara 10016 numaralı müşteriye satılsın. Bu durumda gerekli işlemler; INVOICE, LINE, PRODUCT, CUSTOMER ve ACCT\_TRANSACTION tablolarını etkiler.

Ürün tablousunda 89-WRE-Q kodlu ürününün eldeki miktarı (PROD\_QOH)azaltılır.

Müşteri bakiyesi (CUST\_BALANCE) mevcut bakiyeye 277,55 \$ eklenir.

Yeni bir fatura eklenir.

Table name: INVOICE

INV_NUMBER	CUST_NUMBER	INV_DATE	INV_SUBTOTAL	INV_TAX	INV_TOTAL	PAY_AMOUNT	INV_BALANCE
1001	10014	18-Jan-18	54.92	4.99	59.91	59.91	0.00
1002	10011	18-Jan-18	9.90	0.00	10.78	10.78	0.00
1003	10012	18-Jan-18	270.70	21.66	292.36	292.36	0.00
1004	10011	17-Jan-18	34.87	2.78	37.65	37.65	0.00
1005	10018	17-Jan-18	70.44	5.64	76.08	76.08	0.00
1006	10014	17-Jan-18	397.83	26.83	424.66	160.00	329.66
1007	10015	17-Jan-18	34.97	2.80	37.77	37.77	0.00
1008	10011	17-Jan-18	1033.00	82.85	1115.85	580.00	615.71
1009	10016	18-Jan-18	256.99	20.56	277.55	0.00	277.55

Table name: PRODUCT

PROD_CODE	PROD_DESCRIPTION	PROD_UNDATE	PROD_QOH	PROD_MN	PROD_PRICE	PROD_DISCOUNT	PROD_NUMBER
116R01	Power pointer, 15 psi., 3-estate	03-Nov-17	8	5	109.99	0.00	26695
13-Q2P2	7.25-in. pow. saw blade	13-Dec-17	32	15	14.99	0.05	26344
14-Q1L3	9.09-in. pow. saw blade	13-Nov-17	18	12	17.49	0.00	26344
1546-QG2	Hnd. cloth, 14-in., 2x50	15-Jan-18	15	8	39.95	0.00	29119
1558-QXV	Hnd. cloth, 12-in., 3x50	15-Jan-18	23	5	43.99	0.00	29119
2232QTY	BSD igresser, 12-in. blade	30-Dec-17	8	5	109.92	0.05	24289
2232QNE	BSD igresser, 8-in. blade	24-Dec-17	6	5	99.87	0.05	24289
2232QPD	BSD cordless dril, 12-in.	20-Jan-18	12	5	38.95	0.05	25595
23109-HB	Crew hammer	20-Jan-18	23	10	9.95	0.10	28225
23114-AA	Sledge hammer, 12 lb.	20-Jan-18	8	5	14.40	0.05	28225
54775-2T	Flat file, 18-in. file	15-Dec-17	43	20	4.99	0.00	26344
89-WRE-Q	Recut chain saw, 16 in.	07-Jan-18	11	5	256.99	0.05	24289
PVC230R3	PVC pipe, 3.5-in., 8-ft	06-Jan-18	188	75	5.87	0.00	28225
SM-18277	1.25-in. metal screw, 25	01-Jan-18	172	75	6.99	0.00	28225
SH-23116	2.5-in. wd. screw, 50	24-Feb-18	237	180	8.45	0.00	28231
VR32IT3	Bestimating, 4x8x16", 8" mesh	17-Jan-18	18	5	119.95	0.10	26695

Table name: CUSTOMER

CUST_NUMB	CUST_LNAME	CUST_FNAME	CUST_INITIAL	CUST_AREACODE	CUST_PHONE	CUST_BALANCE
10010	Ramirez	Alfred	A	615	844-2573	0.00
10011	Dennis	Leona	K	713	894-1238	615.73
10012	Smith	Kathy	W	615	894-2205	0.00
10013	Olowski	Paul	F	615	894-2180	0.00
10014	Orlando	Myron		615	222-1672	0.00
10015	O'Brien	Amy	B	713	442-3361	0.00
10016	Brown	James	G	615	297-1228	277.55
10017	Williams	George		615	290-2556	0.00
10018	Farriss	Anne	G	713	382-7865	0.00
10019	Smith	Olette	K	615	297-3009	0.00

Database name: Ch10\_SaleCo

Table name: LINE

INV_NUMBER	LINE_NUMBER	PROD_CODE	LINE_UNITS	LINE_PRICE	LINE_AMOUNT
1001	1	13-Q2P2	3	14.99	44.97
1001	2	23109-HB	1	9.95	9.95
1002	1	54775-2T	2	4.99	9.98
1003	1	1238-QPD	4	38.95	155.80
1003	2	1546-QG2	1	39.95	39.95
1003	3	13-Q2P2	5	14.99	74.95
1004	1	54775-2T	3	4.99	14.97
1004	2	23109-HB	2	9.95	19.90
1005	1	PVC230R3	12	5.87	70.44
1005	1	SM-18277	3	6.99	20.97
1005	2	2232QTY	1	109.92	109.92
1005	3	23109-HB	1	9.95	9.95
1005	4	89-WRE-Q	1	256.99	256.99
1007	1	13-Q2P2	2	14.99	29.98
1007	2	54775-2T	1	4.99	4.99
1008	1	PVC230R3	5	5.87	29.35
1008	2	VR32IT3	4	119.95	479.80
1008	3	23109-HB	1	9.95	9.95
1008	4	89-WRE-Q	2	256.99	513.98
1009	1	89-WRE-Q	1	256.99	256.99

Ürünün alım fiyatı (256.99 \$) eklenir.

Hesap hareketi (ACCT\_TRANSACTION) tablosuna, müşterinin yapmış olduğu alım işlemi, bir kayıt eklenerek yansıtılır.

Table name: ACCT\_TRANSACTION

ACCT_TRANS_NUM	ACCT_TRANS_DATE	CUST_NUMBER	ACCT_TRANS_TYPE	ACCT_TRANS_AMOUNT
10003	17-Jan-18	10014	charge	329.66
10004	17-Jan-18	10011	charge	615.73
10005	28-Jan-18	10014	payment	329.66
10007	18-Jan-18	10016	charge	277.55

Not: Veri bütünlüğünün sağlanması, tamamıyla veritabanı yönetim sistemi ile aracılığı ile gerçekleştirilemez. Uygulama yazılımlarını geliştiren kişilerin SQL sorgularını doğru bir şekilde oluşturması gerekir.



# İşlem Özellikleri

## Bölünmezlik (Atomicity)

- Bir işlemin tüm işlemleri (SQL istekleri) tamamlanmalıdır.
- Tamamlanmamışsa, işlem iptal edilir

## Tutarlılık (Consistency)

- İşlem parçalarından herhangi biri bir bütünlük kısıtlamasını ihlal ederse, tüm işlem iptal edilir.

## Ayrılma/Yalıtım (Isolation)

- İşlem sırasında kullanılan veriler, ilk işlem tamamlanana kadar ikinci işlem tarafından kullanılamaz.

## Kalıcılık (Durability)

- İşlemler yapıldıktan sonra geri alınamaz veya kaybedilemez.

## Serileştirme (Serializability)

- Birkaç işlemin eşzamanlı yürütülmesi için programın tutarlı sonuçlar vermesini sağlar.

# İşlem Özellikleri

- Tek kullanıcılı bir veritabanı sistemi, bir seferde yalnızca bir işlem yürütüldüğünden, veritabanının **serileştirilmesini** ve **yalıtılmasını** otomatik olarak sağlar.
- İşlemlerin **bölünmezliği**, **tutarlılığı** ve **kalıcılığı** tek kullanıcılı veritabanı yönetim sistemlerinde garanti edilmelidir.
  - Tek kullanıcılı bir VYTS bile işletim sistemi kaynaklı kesintiler, güç kesintileri ve anormal uygulama sonlandırmaları veya çökmeler olabilir. Bu durumda oluşan hatalarda kurtarılma (recovery) durumları yönetilmelidir.

# İşlem Özellikleri

- **Çok kullanıcılı veritabanları**, genellikle birden çok eşzamanlı işleme tabidir. Bu nedenle, çok kullanıcılı VTYS, **veritabanının tutarlılığını** ve **bütünlüğünü** korumak için bölünmezlik ve dayanıklılığın yanı sıra serileştirilebilirliği ve işlemlerin yalıtılmasını sağlamak için kontroller uygulamalıdır.
- Örneğin, aynı veri kümesi üzerinden eşzamanlı birkaç işlem yürütülürse ve ikinci işlem ilk işlem tamamlanmadan veritabanını güncelleştirirse, yalıtım özelliği ihlal edilir ve veritabanı artık tutarlı olmaz.
- VTYS, istenmeyen durumlardan kaçınmak için eşzamanlılık kontrol teknikleri kullanarak işlemleri yönetmelidir.

# SQL ile İşlem Yönetimi

Amerikan Ulusal Standartlar Enstitüsü (ANSI), SQL veritabanı işlemlerini yöneten standartları tanımlamıştır.

İşlem desteği iki SQL ifadesi ile sağlanır:

- ❑ COMMIT

- ❑ ROLLBACK

# SQL ile İşlem Yönetimi

Bir işlem dizisi aşağıdaki dört olaydan biri gerçekleşinceye kadar sonraki tüm SQL deyimleri boyunca devam etmelidir:

1. Bir COMMIT deyimine ulaşılması
2. ROLLBACK deyimine ulaşılması
3. Programın sonuna erişilmesi (Bu durum COMMIT deyimi ile eşdeğerdir.)
4. Program anormal bir şekilde sonlandırılması (Bu durum ROLLBACK deyimi ile eşdeğerdir.)

# SQL ile İşlem Yönetimi

- Bir **COMMIT** deyimine ulaşıldığında tüm değişiklikler veritabanına kalıcı olarak kaydedilir.
- Bir **ROLLBACK** deyimine ulaşıldığında tüm değişiklikler iptal edilir ve veritabanı önceki tutarlı durumuna geri alınır.

- Örnek olarak 10011 numaralı müşteri, birim fiyatı 43.99 \$ olan 1558-QW1 kodlu üründen iki adet alsın. Bu durumda:
  - Ürün miktarında (PROD\_QOH) azalma ÜRÜN (PRODUCT) tablosuna yansıtılmalıdır.
  - Yapılan harcama müşterinin hesabına (CUST\_BALANCE) yansıtılmalıdır.

```
UPDATE      PRODUCT
SET          PROD_QOH = PROD_QOH - 2
WHERE        PROD_CODE = '1558-QW1';

UPDATE      CUSTOMER
SET          CUST_BALANCE = CUST_BALANCE + 87.98
WHERE        CUST_NUMBER = '10011';
COMMIT;
```

\* UPDATE deyimi uygulamanın son eylemi ise ve uygulama normal olarak sonlandırılırsa, örnekte kullanılan COMMIT deyimi gerekli değildir.

\* Ancak, iyi programlama uygulaması, bir işlem bildiriminin sonuna COMMIT deyimini dahil etmenizi gerektirir.

- Tüm SQL uygulamaları ANSI standardına uygun değildir.

Bu nedenle yeni bir işlemin başlangıcını belirtmek için sorgunun en başına aşağıdakiler gibi işlem yönetimi ifadeleri yazılır:

- BEGIN TRANSACTION → SQL SERVER
- START TRANSACTION → MYSQL SERVER



# İşlem Günlüğü (Transaction Log)

- VTYS, veritabanının güncelleştiren tüm işlemleri, takip etmek için bir işlem günlüğü kullanır.

VTYS, bu günlükte depolanan bilgileri,

- bir ROLLBACK deyimi,
- bir programın anormal sonlanması,
- ağ tutarsızlığı ,
- disk çökmesi gibi

bir sistem hatası tarafından tetiklenen bir kurtarma gereksinimi için kullanır.

## İşlem Günlüğü (Transaction Log)

- Örneğin bir sunucu hatasından sonra Oracle, sonlanmamış işlemleri otomatik olarak geri alır ve icra edilmiş ancak henüz fiziksel veritabanına yazılmayan işlemleri geri alır.
- Bir işlem günlüğü kullanmak bir DBMS'nin işlem yükünü artırsa da, bozuk bir veritabanını geri yükleme yeteneği fiyat değeridir.

# Eşzamanlılık Kontrolü (Concurrency Control)

- ❑ Çok kullanıcılı bir veritabanı sisteminde işlemlerin eşzamanlı yürütülmesini koordine etmek eşzamanlılık kontrolü olarak bilinir.
- ❑ Eşzamanlılık kontrolünün amacı, çok kullanıcılı bir veritabanı ortamında işlemlerin serileştirilmesini sağlamaktır.
- ❑ Bu hedefe ulaşmak için, eşzamanlılık kontrol tekniklerinin çoğu, eşzamanlı olarak işlemlerin yürütülmesi ayrılma/yalıtım özelliğini korumaya yöneliktir.

# Eşzamanlılık Kontrolündeki Sorunlar

## Kaybolmuş Güncellemeler

### İki eşzamanlı işlem:

- Aynı veri ögesi günceller.
- Güncellemelerden biri kaybolur (diğer işlem tarafından üzerine yazılmalar olur.)

## Kaydedilmemiş Veri

- İki işlem eş zamanlı çalıştırılır.
- İkinci işlem kaydedilmemiş veriye eriştikten sonra birinci işlem geri alınır.

## Tutarsız Bilgi Elde Etme

- Bir işlemin veriye erişmesi, diğer başka işlem(ler)in bu veri ile çalışmasını bitirmeden önce veya sonraki bir sürece denk gelmesi sonucu oluşur.

# Zamanlayıcı (Scheduler)

- Eşzamanlı işlemler içinde işlemlerin gerçekleştirilme sırasını belirler.
  - Zamanlayıcı, serileştirilmeyi sağlamak için veritabanı işlemlerinin belirli bir sırada yürütülmesini aralıklı gerçekleştirir.
- Uygun sırayı belirlemek için eşzamanlı kontrol algoritmalarını temel alır.
- Serileştirme programı oluşturur.
  - **Serileştirilebilir program:** İşlemlerin aralıklı yürütülmesi, işlemlerin seri yürütülmesi ile aynı sonuçları verir.

# Kilitleme Yöntemleri ile Eşzamanlılık Kontrolü

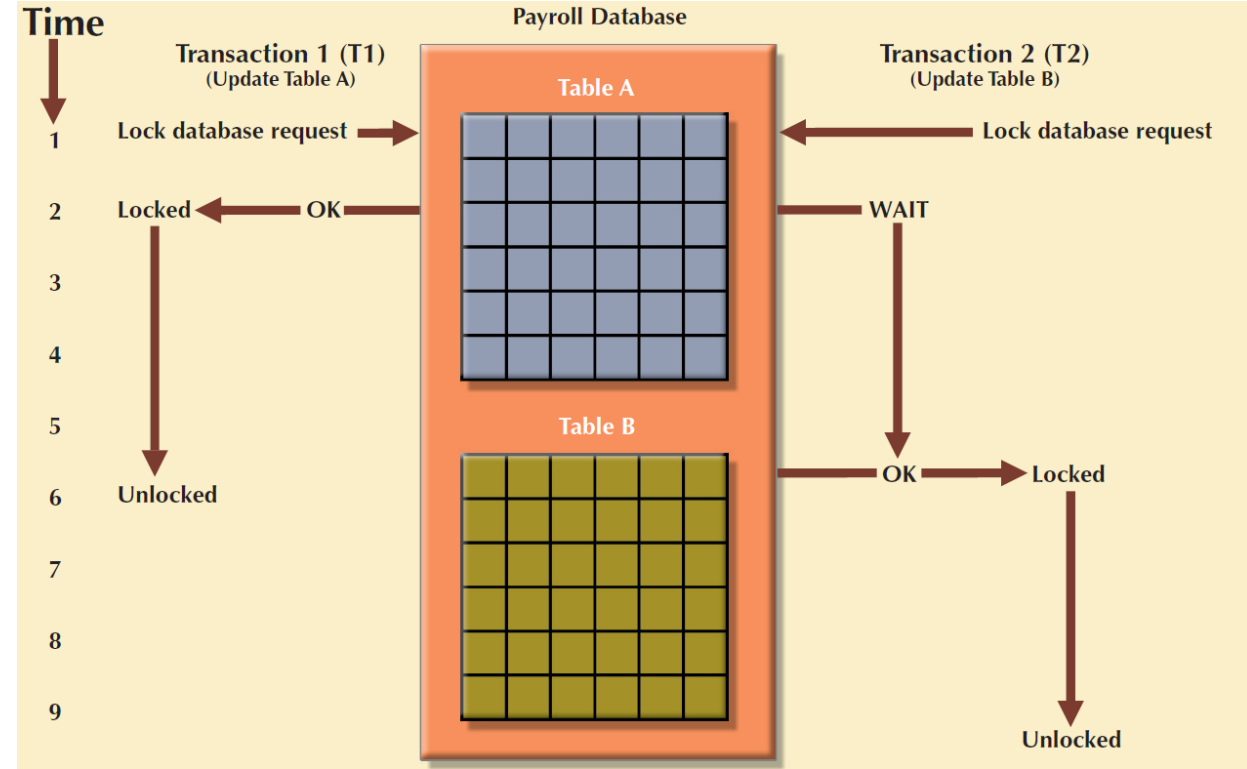
- Kilitleme yöntemleri - Eşzamanlı olarak işlem yaparken kullanılan veri öğelerinin izolasyonuna imkan sağlar.
- **Kilit:** Bir veri ögesinin geçerli bir işleme özel olarak kullanımını garanti eder.
- **Kötümser kilitleme:** İşlemler arasında çatışmanın muhtemel olduğu varsayımına dayanarak kilitlerin kullanılmasıdır.
- **Kilit yöneticisi:** İşlemler tarafından kullanılan kilitlerin atanması ve denetlenmesinden sorumludur.

# Kilit Ayrıntı Düzeyi

- Kilit kullanım seviyesini gösterir
- Kilitleme seviyeleri
  - **Veritabanı düzeyinde kilit**
  - **Tablo düzeyinde kilit**
  - **Sayfa düzeyinde kilit**
    - **Sayfa veya disk sayfası:** Bir diskin doğrudan adreslenebilir bölümü
  - **Satır düzeyinde kilit**
  - **Alan seviyesinde kilit**

# Veritabanı düzeyinde kilit

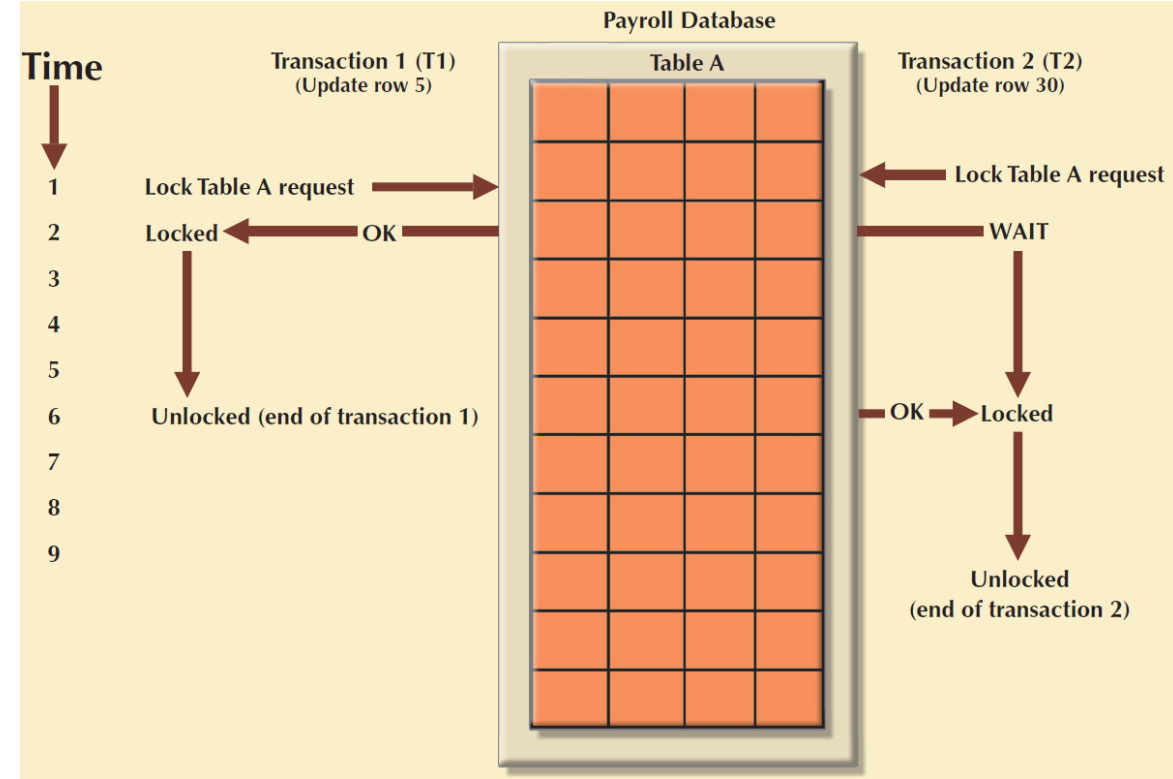
- Tüm veritabanı kilitlenir, böylece T1 işlemi yürütülürken veritabanındaki herhangi bir tablonun T2 işlemi ile kullanılmasını önlenir.
- Bu kilitleme düzeyi, toplu işlemler için iyidir, ancak çok kullanıcılı VTYS'ler için uygun değildir.
- T1 ve T2 işlemleri, farklı tablolar kullandıklarında bile aynı veritabanına aynı anda erişemez.





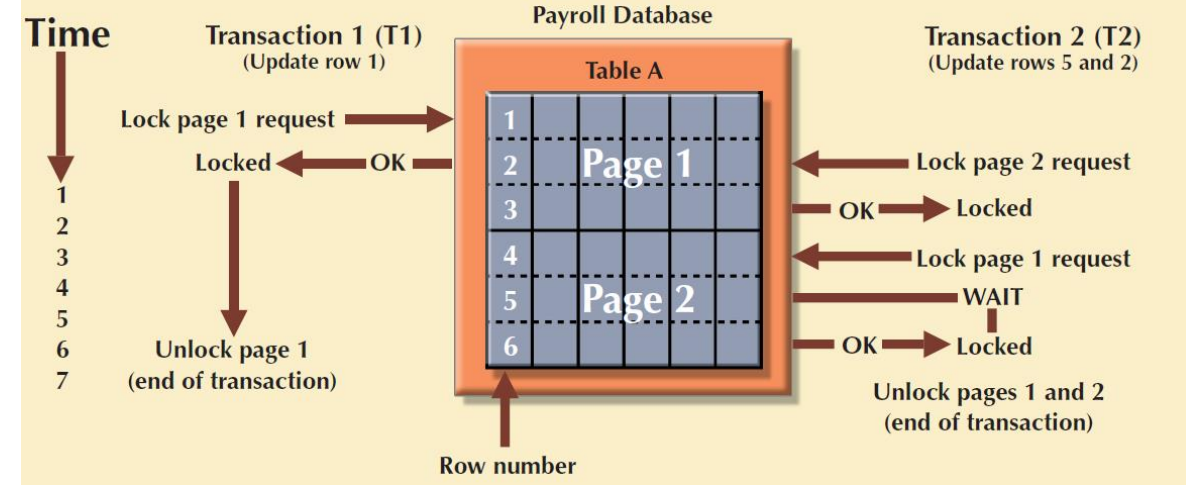
# Tablo düzeyinde kilit

- Tablo düzeyindeki bir kilitte, tüm tablo kilitlenir.
  - **T1 işlemi**, tabloyu kullanılırken T2 işlemi ile herhangi bir satıra erişim engellenir.
- Birçok işlem aynı tabloya erişmeyi beklerken trafik sıkışıklıklarına neden olur.
  - Örneğin bir durum, farklı işlemler aynı tablonun farklı bölümlerine erişim gerektirdiğinde gecikmeye yol açar.
  - Sonuç olarak, tablo düzeyi kilitler çok kullanıcılı VTYS'ler için uygun değildir.



# Sayfa düzeyinde kilit

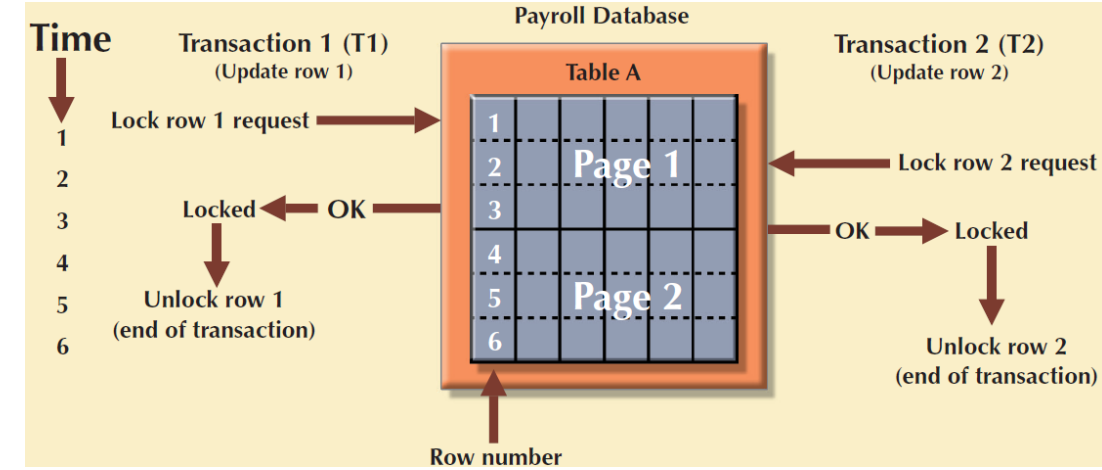
- Bir disk sayfası veya sayfa, bir diskin doğrudan adreslenebilir bir bölümü olarak tanımlanabilecek bir disk bloğuna eşdeğerdir.
- Bir sayfanın 4K, 8K veya 16K gibi sabit bir boyutu vardır.
- Bir tablo birkaç sayfaya yayılabilir ve bir sayfa bir veya daha fazla tablonun birkaç satırını içerebilir.
- **Sayfa düzeyi kilitler şu anda çok kullanıcı VTYS'ler için en sık kullanılan kilitleme yöntemidir.**



- \* T1 ve T2'nin farklı disk sayfalarını kilitlerken aynı tabloya erişebilmektedir.
- \* T2, T1 tarafından kilitlenmiş bir sayfada bulunan bir satırın kullanılmasını gerektiriyorsa, T1 sayfanın kilidini açana kadar beklemelidir.

# Satır düzeyinde kilit

- Satırlar aynı sayfada olsa bile aynı tablonun farklı satırlarına erişilmesine izin verilir.
- Verilerin kullanılabilirliğini geliştirmesine rağmen, her satır için bir kilit bulunması veritabanı yönetimine yüksek yük gerektirir.
- Modern VTYS'lerde, uygulama oturumu aynı sayfada birden çok kilit talep etmeyi gerektirdiğinde kilit seviyesi satır düzeyinden sayfa düzeyine yükseltilir.



- \* İstenen satırlar, aynı sayfada olsa bile her iki işlem aynı anda gerçekleştirilebilir.
- \* T2 yalnızca T1 ile aynı satırı istediğinde bekleme durumuna geçmektedir.

## Alan seviyesinde kilit

- Eşzamanlı işlemlerin o satırda farklı alanların (niteliklerin) kullanılmasını gerektirdikleri sürece aynı satıra erişmesine izin verir.
- Son derece yüksek düzeyde yük getirdiği için VTYS'lerde nadiren kullanılır.
- Satır düzeyinde kilit alan düzeyine göre çok daha kullanışlıdır.

# Kilit Çeşitleri

## İkili kilit

Kilitli (1) ve kilidi açılmış (0) olarak temsil edilir.

- Bir nesne, bir işlem tarafından kilitlenmişse, başka hiçbir işlem bu nesneyi kullanamaz.
- Bir nesnenin kilidi açılırsa, herhangi bir işlem nesneyi kullanım için kilitleyebilir.

## Özel kilit

- Bir işlem bir veri ögesini güncellemek için izin istediğinde ve bu veri ögesi, başka bir işlem tarafından kilitli tutulmadığı durumlarda özel bir kilit verilir.
- Özel bir kilit, diğer işlemlerin veritabanına erişmesine izin vermez.

## Paylaşılan kilit

- Bir işlem bir veritabanından veri okumak için izin istediğinde ve başka bir işlem tarafından veriler üzerinde özel kilitler tutulmadığı durumlarda paylaşılan kilit verilir.
- Paylaşılan kilit, diğer salt okunur işlemlerin veritabanına erişmesini sağlar.

# Kilitlerin Kullanımındaki Sorunlar

- Ortaya çıkan işlem programı serileştirilemeyebilir
- Zamanlayıcı (Scheduler) kilitlenme/kördüğüm (deadlock) oluşturabilir.

# İki Aşamalı Kitleme (2PL)

- İşlemlerin kilitleri nasıl elde edip bıraktığını tanımlar
- Serileştirilebilirliği garanti eder, ancak kilitlenmeleri/kördüğümüleri (deadlock) engellemez.
- **Aşamalar**
  - Büyüme aşaması - İşlem, tüm kilitleri veri kilidini açmadan alır
  - Daralan aşama - İşlem tüm kilitleri serbest bırakır ve yeni bir kilit elde edemez

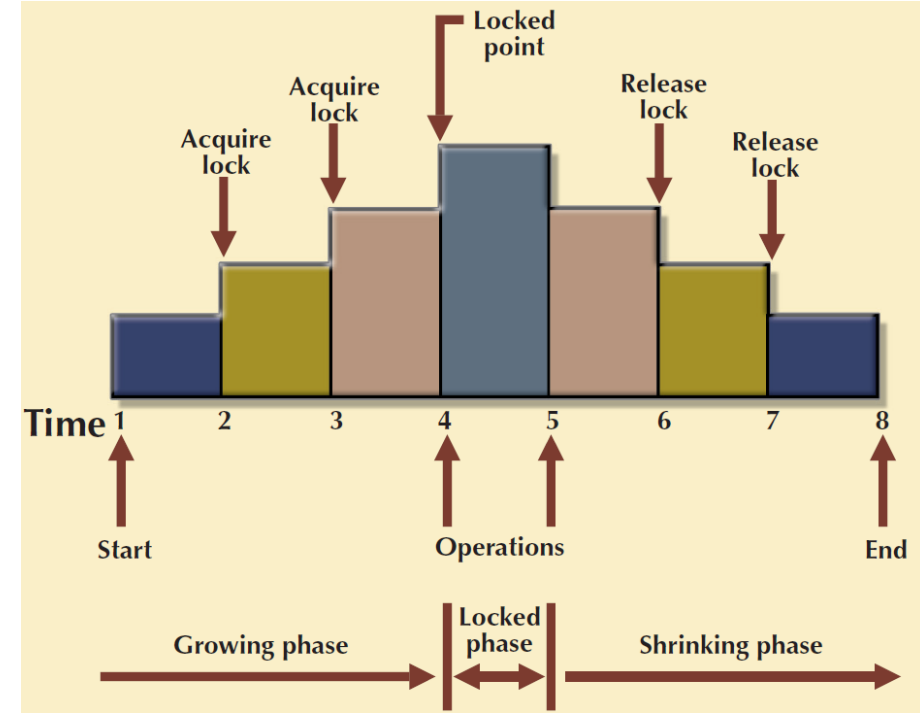
# İki Aşamalı Kilitleme (2PL)

## Geçerli kurallar

- İki işlemin çakışan kilitleri olamaz
- Hiçbir kilit açma işlemi aynı işlemde kilitleme işleminden önce olamaz.
- Tüm kilitler elde edilene kadar hiçbir veri etkilenmez.



- İşlem, önce ihtiyacı olan **iki kilidi elde eder**. İşlem, kilitli noktaya ulaşır.
- Ardından, veriler işlemin gereksinimlerine uyacak şekilde değiştirilir.
- Son olarak, işlem ilk aşamada elde edilen **tüm kilitleri serbest bırakılır** ve işlem tamamlanır.



İki fazlı kilitleme, işlem maliyetini artırır ve kördüğüm olma gibi istenmeyen etkilere neden olabilir.

# Kilitlenme/Kördüğüm (Deadlock)

- ❑ İki işlem, verilerin kilidini açmak için birbirini süresiz olarak beklediğinde oluşur
  - Ölümcül kucaklama olarak bilinir.
- ❑ Kontrol teknikleri
  - Kilitlenme önleme
  - Kilitlenme tespiti
  - Kilitlenme önleme
- ❑ Kilitlenme kontrol yönteminin seçimi, veritabanı ortamına bağlıdır

# Bir Kilitlenme Durumu Nasıl Oluşur

TIME	TRANSACTION	REPLY	LOCK STATUS	
			DATA X	DATA Y
0			Unlocked	Unlocked
1	T1:LOCK(X)	OK	Locked	Unlocked
2	T2:LOCK(Y)	OK	Locked	Locked
3	T1:LOCK(Y)	WAIT	Locked	Locked
4	T2:LOCK(X)	WAIT	Locked	Locked
5	T1:LOCK(Y)	WAIT	Locked	Locked
6	T2:LOCK(X)	WAIT	Locked	Locked
7	T1:LOCK(Y)	WAIT	Locked	Locked
8	T2:LOCK(X)	WAIT	Locked	Locked
9	T1:LOCK(Y)	WAIT	Locked	Locked
...	.....	.....	.....	.....
...	.....	.....	.....	.....
...	.....	.....	.....	.....
...	.....	.....	.....	.....



**T1** = X ve Y  
**T2** = Y ve X  
veri öğelerine erişimi

**T1**, Y veri öğesinin kilidini açmadıysa, T2 başlatılamaz.

**T2**, X veri öğesinin kilidini açmadıysa, T1 devam edemez. Böylece ölümcül kucaklama olur.

# Zaman damgası (Time Stamping)

- Eşzamanlı işlemlerin zamanlanmasına zaman damgası yaklaşımı, her işleme genel ve benzersiz bir zaman damgası atar.
  - Zaman damgası değeri, işlemlerin VTYS'ye gönderildiği açık bir düzen oluşturur.

## Özellikleri

- **Benzersizlik:** Eşit zaman damgası değerlerinin bulunmamasını sağlar
- **Tekdüzelik:** Zaman damgası değerlerinin her zaman artmasını sağlar.

# Zaman damgası (Time Stamping)

## **Dezavantajları**

- Veritabanında saklanan her değer iki ek damga alanı gerektirir.
- Bellek ihtiyaçlarını artırır.
- Veritabanının işlem yükünü artırır.
- Bir çok sistem kaynağını talep gerektirir.

# Wait-Die ve Wound-Wait yöntemlerinin Kontrol Şeması

TRANSACTION REQUESTING LOCK	TRANSACTION OWNING LOCK	WAIT/DIE SCHEME	WOUND/WAIT SCHEME
T1 (11548789)	T2 (19562545)	<ul style="list-style-type: none"><li>T1 waits until T2 is completed and T2 releases its locks.</li></ul>	<ul style="list-style-type: none"><li>T1 preempts (rolls back) T2.</li><li>T2 is rescheduled using the same time stamp.</li></ul>
T2 (19562545)	T1 (11548789)	<ul style="list-style-type: none"><li>T2 dies (rolls back).</li><li>T2 is rescheduled using the same time stamp.</li></ul>	<ul style="list-style-type: none"><li>T2 waits until T1 is completed and T1 releases its locks.</li></ul>

- \* Bir işlem birden çok kilit ister. Bu durum bazı işlemlerin süresiz olarak beklemesine ve bir kilitlenmeye neden olabilir.
- \* Bir kördüğümü önlemek için, her kilit isteğinin ilişkili bir zaman aşımı değeri vardır.
- \* Zaman aşımı süresi dolmadan kilitleme izni verilmezse işlem geri alınır.

- Zaman damgalarından T1'in daha eski işlem (daha düşük zaman damgası değeri) ve T2 daha yeni işlemidir.
- **Wait/Die:** Eski işlem, genç olanın kilitlerini tamamlamasını ve serbest bırakmasını bekler.
- **Wound-Wait:** Eski işlem, genç işlemi geri alır (roll back) ve yeniden planlar.

# İyimser Yöntemlerle Eşzamanlılık Kontrolü

**İyimser yaklaşım:** Veritabanı işlemlerinin çoğunun çatışmayacağı varsayımına dayanır.

- Kilitleme veya zaman damgası teknikleri gerektirmez.
- İşlem, taahhüt edilene kadar kısıtlama olmadan gerçekleştirilir.

# İyimser Yöntemlerle Eşzamanlılık Kontrolü

- İyimser yaklaşım, birkaç güncelleme işlemi gerektiren çoğu okuma veya sorgu veritabanı sistemi için kabul edilebilir.
- Çok kullanılan bir VTYS ortamında, kilitlenmelerin yönetimi - önlenmesi ve algılanması - önemli bir VTYS işlevi oluşturur.
- VTYS, burada tartışılan tekniklerden bir veya daha fazlasını ve bu tekniklerdeki varyasyonları kullanacaktır.
- İşlem yönetiminin bir veritabanında nasıl uygulandığını daha iyi anlamak için, işlem yalıtım düzeylerinin öğrenilmesi önemlidir.



# İyimser Yaklaşımın Aşamaları

## Okuma

### İşlem:

- Veritabanını okur.
- Gerekli hesaplamaları yürütür.
- Veritabanı değerlerinin özel bir kopyasında güncellemeler yapar.

## Onaylama

- İşlem, yapılan değişikliklerin veritabanının bütünlüğünü ve tutarlılığını etkilemeyeceğinden emin olmak için onaylanmıştır

## Yazma

- Değişiklikler veritabanına kalıcı olarak uygulanır

# ANSI İşlem Yalıtım Düzeyleri

- İşlem izolasyon seviyeleri, işlem verilerinin diğer eşzamanlı işlemlerden ne ölçüde "korunduğu veya izole edildiği" anlamına gelir.
- Yalıtım seviyeleri, diğer işlemlerin yürütme sırasında görebileceği (okuyabileceği) verilerle tanımlanır.
- Daha kesin olarak, işlem yalıtım seviyeleri, bir işlemin izin verdiği veya vermediği "okuma" türüyle tanımlanır.

# ANSI İşlem Yalıtım Düzeyleri

## Okuma işlemlerinin türleri:

- **Kirli okuma (dirty read):** Bir işlem henüz tamamlanmamış verileri okuyabilir.
- **Tekrarlanamayan okuma (Nonrepeatable read) :** Bir işlem  $t1$  zamanında belirli bir satırı okur ve ardından  $t2$  zamanında aynı satırı okur ve farklı sonuçlar verir.
- **Hayali okuma (Phantom read):** Bir işlem  $t1$  zamanında bir sorgu yürütür ve ardından  $t2$  zamanında aynı sorguyu çalıştırır ve sorguyu tatmin eden ek satırlar verir.

# ANSI İşlem Yalıtım Düzeyleri

**ANSI dört işlem yalıtım düzeyi tanımlamıştır:**

## **1. İşlenmemiş/kesinleşmemiş Okuma (Read Uncommitted) :**

Diğer işlemlerden alınmamış verileri okuyacaktır. Bu yalıtım düzeyinde, veritabanı verilere herhangi bir kilit yerleştirmez.

## **2. İşlenmiş/kesinleşmiş Okuma (Read Committed) :**

İşlemleri yalnızca kesinleşmiş/kaydedilmiş verileri okumaya zorlar.

Bu, çoğu veritabanının (Oracle ve SQL Server dahil) varsayılan çalışma modudur.

Bu düzeyde, veritabanı veriler üzerinde özel kilitler kullanır ve diğer işlemlerin orijinal işlem tamamlanana kadar beklemesine neden olur.

# ANSI İşlem Yalıtım Düzeyleri

## **3. Tekrarlanabilir Okuma (Repeatable Read) :**

Sorguların tutarlı sonuçlar döndürmesini sağlar.

Bu tür yalıtım düzeyi, özgün sorgu okuduktan sonra diğer işlemlerin bir satırı güncellemediğinden emin olmak için paylaşılan kilitleri kullanır.

## **4. Serileştirilebilir yalıtım (The Serializable isolation) :**

En kısıtlayıcı seviyedir. Bununla birlikte, Seri hale getirilebilir bir izolasyon seviyesinde bile, kilitlenmelerin her zaman mümkün olduğunu belirtmek önemlidir.

# Veritabanı Kurtarma Yönetimi

- ❑ **Veritabanı kurtarma:** Veritabanının belirli bir durumdan daha önce tutarlı bir duruma geri yükler.
- ❑ **Kurtarma işlemleri atomik işlem özelliğine dayanır**
- ❑ **Atomik işlem özelliği:** Bir işlemin tüm bölümleri tek bir mantıksal iş birimi olarak ele alınmalıdır.
  - İşlem işlemi tamamlanamazsa:
    - İşlem iptal edilmelidir.
    - Veritabanındaki değişiklikler geri alınmalıdır.

# İşlem Kurtarmayı Etkileyen Kavramlar

## Ertelenmiş yazma tekniği veya ertelenmiş güncelleme

- Veriler güncellenmeden önce işlem günlüklerinin her zaman yazılmasını sağlar.

## Gereksiz işlem günlükleri

- (İşlem günlüğünün birkaç kopyası), fiziksel disk arızasının VTYS'nin veri kurtarma yeteneğini olumsuz etkilememesini sağlar.

## Tampon/Ara (Buffers)

- Birincil bellekteki geçici depolama alanlarıdır.

## Kontrol noktaları

- VTYS'nin tüm güncellenmiş arabelleklerini diske yazmasına izin verir

# İşlem Kurtarma Prosedürlerinde Kullanılan Teknikler

## Ertelenmiş yazma tekniği veya ertelenmiş güncelleme

- Yalnızca işlem günlüğü güncellenir.

## Yazma tekniği veya anında güncelleme

- Veritabanı, işlemin yürütülmesi sırasında işlem operasyonları ile hemen güncellenir



# Ertelenmiş Yazma Tekniğinde Kurtarma Süreci

- İşlem günlüğündeki son kontrol noktasını tanımlanır.
- İşlem son kontrol noktasından önce yapılmışsa
  - Hiçbir şey yapılmasına gerek yok
- İşlem son kontrol noktasından sonra yapılmışsa
  - İşlem günlüğü işlemi yeniden yapmak için kullanılır
- İşlemin son kontrol noktasından sonra GERİ ALMA (ROLLBACK) işlemi varsa
  - Hiçbir şey yapılmasına gerek yok

# Yazma Tekniğinde Kurtarma Süreci

- İşlem günlüğündeki son kontrol noktası tanımlanır.
- İşlem son kontrol noktasından önce yapılmışsa
  - Hiçbir şey yapılmasına gerek yok
- İşlem son kontrol noktasından sonra yapılmışsa
  - İşlem yeniden yapılmalı
- İşlemin son kontrol noktasından sonra GERİ ALMA (ROLLBACK) işlemi varsa
  - İşlem günlüğü işlemleri, GERİ ALMA için kullanılır

# Kaynakça

- <https://dev.mysql.com/doc/refman/8.0/en/sql-data-manipulation-statements.html>