Veritabanı Yönetim Sistemleri (335)

Dr. Öğr. Üyesi Ahmet Arif AYDIN

L22- Hareket Yönetimi, ACID Özellikleri, Stored Procedure

(Transaction Management, ACID Properties, Stored Procedure)

GÜZ -2022

Sorular

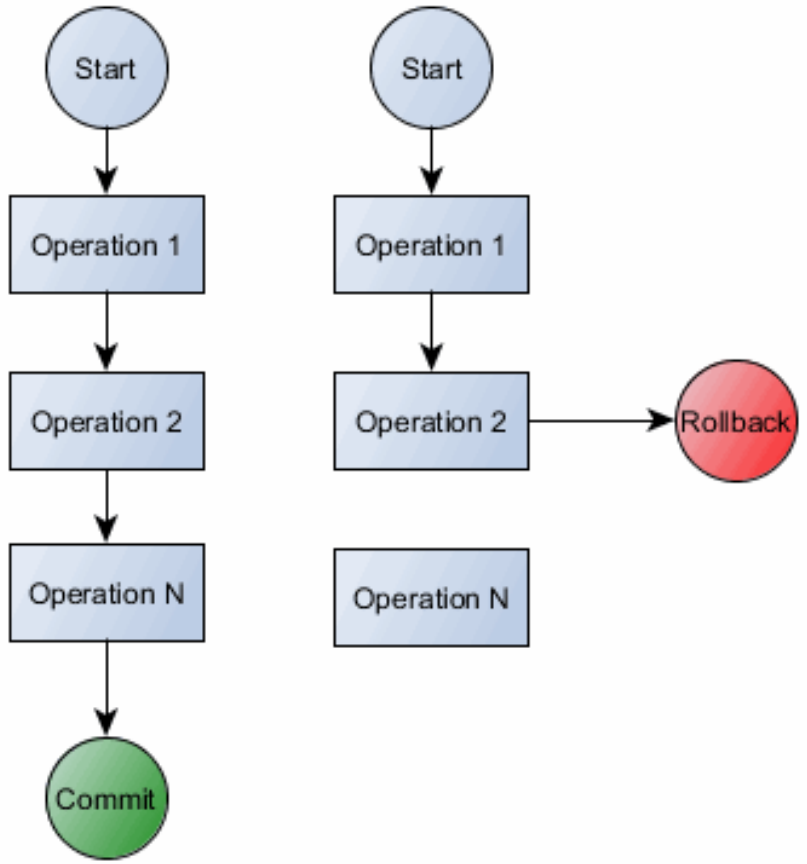
- Sorgu Optimizasyonu nedir?
- Bir Sorgunun değerlendirilmesinde (query evaluation) hangi işlemler gerçekleştirilir?
- Sorguların oluşturulmasında ***selection, projection, join*** işlemleri neden önemlidir?
- Partitioning, iteration, indexing kavramları nedir? Nerede kullanılır ?
- Tree ve hash tabanlı indexing
- **explain analyze** komutu neden kullanılır?

Hareket Yönetimi (Transaction Management)

- Kullanıcı uygulamalarının veritabanı içerisinde gerçekleştirmiş olduğu birden fazla **CRUD işlemi hareket (transaction)** olarak isimlendirilir.

- Bir hareket(transaction) birden fazla işlemden oluşur.

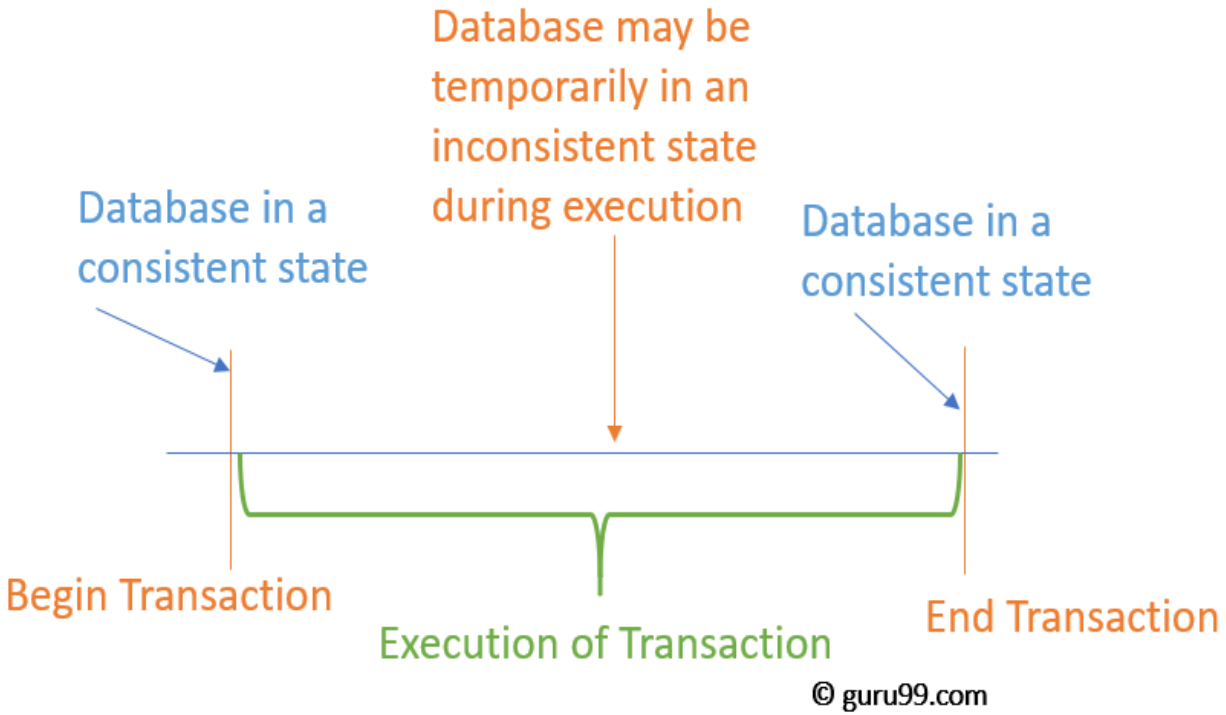
- Gerçekleştirilen hareket içinde bütün işlemler **tamamlanır(commit)** yada geri alınır **(rollback)**.



https://vladmihalcea.com/a-beginners-guide-to-acid-and-database-transactions/

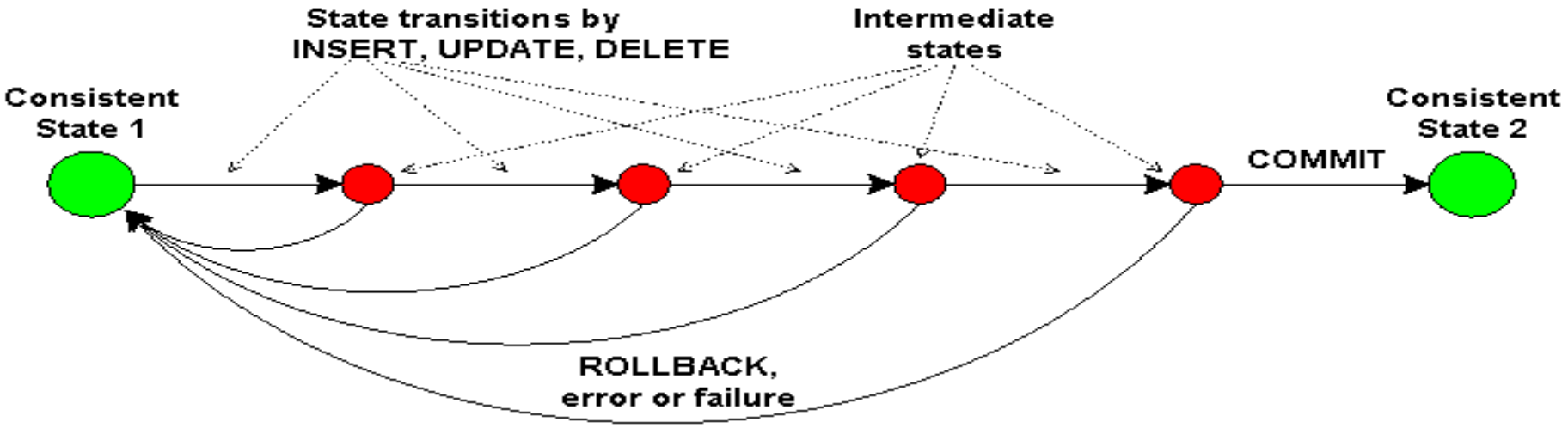
Hareket Yönetimi (Transaction Management)

- Hareketler gerçekleştirilirken veritabanın bütün kısıtlama şartları sağlanarak işlem tamamlanır.
- Hareket örnekleri
  - müşterinin bir ürünü satın alması
  - bir ögrencinin bir derse kayıt olması
  - bir hesaptan başka bir hesaba para transferinin gerçekleştirilmesi



https://www.guru99.com/dbms-transaction-management.html

Hareket Yönetimi (Transaction Management)



https://maxdb.sap.com/doc/7_7/44/d776a368113ee3e10000000a114a6b/content.htm

Bir hareket gerçekleştirildiğinde veritabanı tutarlı (consistent)

bir durumdan başka bir tutarlı duruma geçiş yapar

Most real-world database transactions are formed by ***two or more database requests***

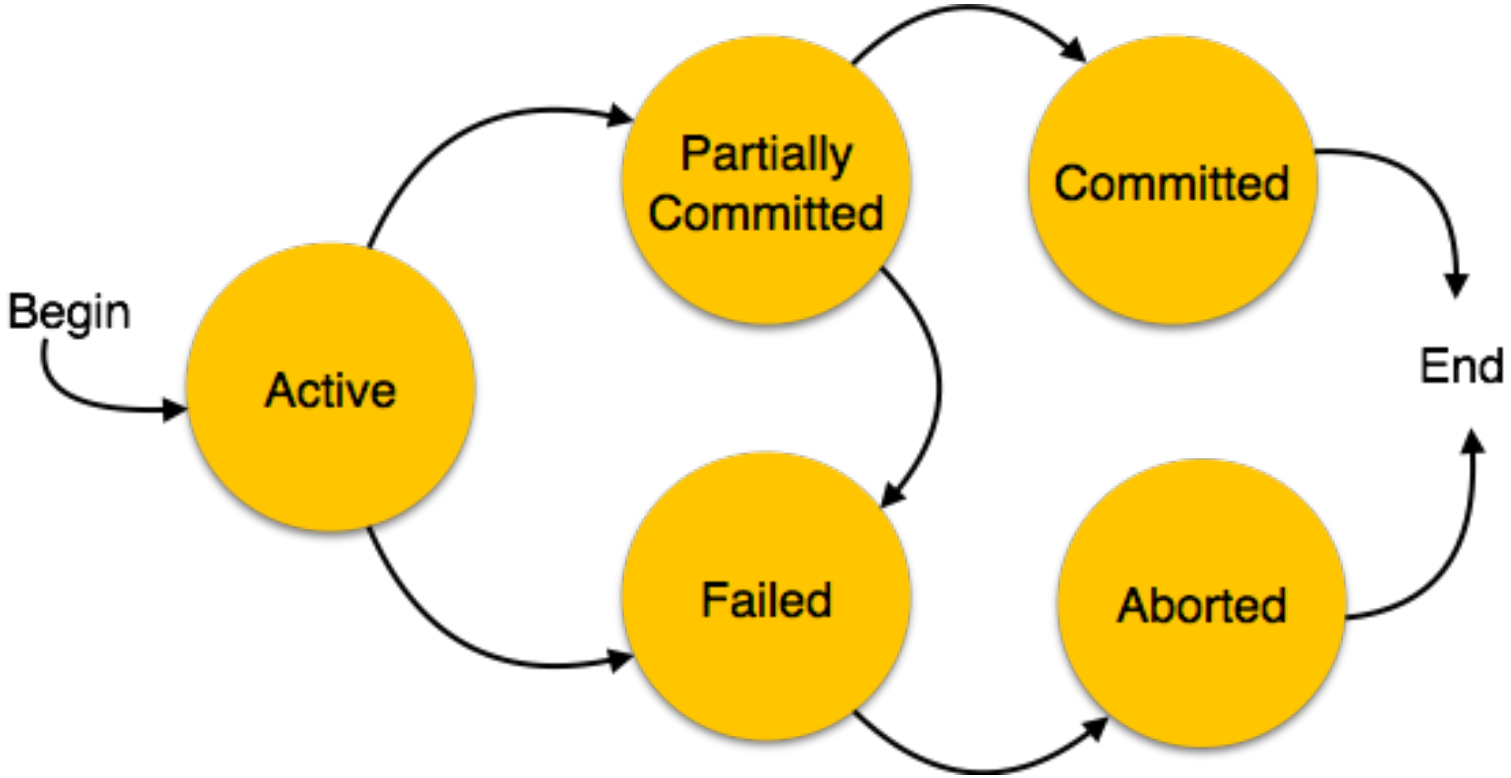
Hareket Yönetimi (Transaction Management)

Bir hesaptan başka bir hesaba para transfer işlemi :

1. Gönderici Hesabını control (READ)
2. Alıcı hesabını kontrol et (READ)
3. Gönderilmek istenen miktar mevcut ve işlem için yeterli ise Göndericinin hesaptan düş (WRITE)
4. Alıcı hesabına istenilen miktarı gönder (WRITE)
5. Alıcı hesabını güncelle

Hareket Yönetimi (Transaction Management)

Veritabanı yönetim sistemleri hareketlerin yönetimi gerçekleştirilirken bir hareket aşağıdaki durumlardan (states of transactions) birinde olabilir..



https://www.tutorialspoint.com/dbms/dbms_transaction.htm

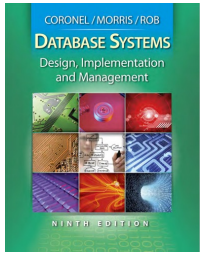
Hareket Yönetimi (Transaction Management)

```
INSERT INTO INVOICE
      VALUES (1009, 10016,'18-Jan-2010', 256.99, 20.56, 277.55, 'cred', 0.00, 277.55);
INSERT INTO LINE
      VALUES (1009, 1, '89-WRE-Q', 1, 256.99, 256.99);

UPDATE PRODUCT
SET PROD_QOH = PROD_QOH – 1
WHERE PROD_CODE = '89-WRE-Q';

UPDATE CUSTOMER
SET CUST_BALANCE = CUST_BALANCE + 277.55
WHERE CUST_NUMBER = 10016;

INSERT INTO ACCT_TRANSACTION
      VALUES (10007, '18-Jan-10', 10016, 'charge', 277.55);
COMMIT;
```



Hareket Yönetimi (Transaction Management)

```
INSERT INTO INVOICE
      VALUES (1009, 10016,'18-Jan-2010', 256.99, 20.56, 277.55, 'cred', 0.00, 277.55);
INSERT INTO LINE
      VALUES (1009, 1, '89-WRE-Q', 1, 256.99, 256.99);

UPDATE PRODUCT
SET PROD_QOH = PROD_QOH – 1
WHERE PROD_CODE = '89-WRE-Q';

UPDATE CUSTOMER
SET CUST_BALANCE = CUST_BALANCE + 277.55
WHERE CUST_NUMBER = 10016;

INSERT INTO ACCT_TRANSACTION
      VALUES (10007, '18-Jan-10', 10016, 'charge', 277.55);
COMMIT;
```

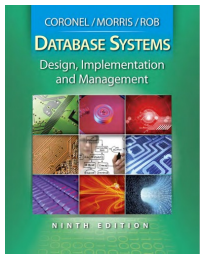


FIGURE 10.2 Tracing the transaction in the Ch10_SaleCo database

Database name: Ch10_SaleCo

Table name: INVOICE

INV_NUMBER	CUST_NUMBER	INV_DATE	INV_SUBTOTAL	INV_TAX	INV_TOTAL	INV_PAY_TYPE	INV_PAY_AMOUNT	INV_BALANCE
1001	10014	16-Jan-12	54.92	4.39	59.31	cc	59.31	0.00
1002	10011	16-Jan-12	9.98	0.80	10.78	cash	10.78	0.00
1003	10012	16-Jan-12	270.70	21.66	292.36	cc	292.36	0.00
1004	10011	17-Jan-12	34.87	2.79	37.66	cc	37.66	0.00
1005	10018	17-Jan-12	70.44	5.64	76.08	cc	76.08	0.00
1006	10014	17-Jan-12	397.83	31.83	429.66	cred	100.00	329.66
1007	10015	17-Jan-12	34.97	2.80	37.77	chk	37.77	0.00
1008	10011	17-Jan-12	1033.08	82.65	1115.73	cred	500.00	615.73
1009	10016	18-Jan-12	256.99	20.56	277.55	cred	0.00	277.55

Table name: LINE

INV_NUMBER	LINE_NUMBER	PROD_CODE	LINE_UNITS	LINE_PRICE	LINE_AMOUNT
1001	1	13-Q2/P2	3	14.99	44.97
1001	2	23109-HB	1	9.95	9.95
1002	1	54778-2T	2	4.99	9.98
1003	1	2238/QPD	4	38.95	155.80
1003	2	1546-QQ2	1	39.95	39.95
1003	3	13-Q2/P2	5	14.99	74.95
1004	1	54778-2T	3	4.99	14.97
1004	2	23109-HB	2	9.95	19.90
1005	1	PVC23DRT	12	5.87	70.44
1006	1	SM-18277	3	6.99	20.97
1006	2	2232/QTY	1	109.92	109.92
1006	3	23109-HB	1	9.95	9.95
1006	4	89-WRE-Q	1	256.99	256.99
1007	1	13-Q2/P2	2	14.99	29.98
1007	2	54778-2T	1	4.99	4.99
1008	1	PVC23DRT	5	5.87	29.35
1008	2	WR3/TT3	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

Table name: PRODUCT

PROD_CODE	PROD_DESCRIPT	PROD_INDATE	PROD_QOH	PROD_MIN	PROD_PRICE	PROD_DISCOUNT	VEND_NUMBER
11QER/31	Power painter, 15 psi., 3-nozzle	03-Nov-11	8	5	109.99	0.00	25595
13-Q2/P2	7.25-in. pwr. saw blade	13-Dec-11	32	15	14.99	0.05	21344
14-Q1/L3	9.00-in. pwr. saw blade	13-Nov-11	18	12	17.49	0.00	21344
1546-QQ2	Hrd. cloth, 1/4-in., 2x50	15-Jan-12	15	8	39.95	0.00	23119
1558-QW1	Hrd. cloth, 1/2-in., 3x50	15-Jan-12	23	5	43.99	0.00	23119
2232/QTY	B&D jigsaw, 12-in. blade	30-Dec-11	8	5	109.92	0.05	24288
2232/QWE	B&D jigsaw, 8-in. blade	24-Dec-11	6	5	99.87	0.05	24288
2238/QPD	B&D cordless drill, 1/2-in.	20-Jan-12	12	5	38.95	0.05	25595
23109-HB	Claw hammer	20-Jan-12	23	10	9.95	0.10	21225
23114-AA	Sledge hammer, 12 lb.	02-Jan-12	8	5	14.40	0.05	
54778-2T	Rat-tail file, 1/8-in. fine	15-Dec-11	43	20	4.99	0.00	21344
89-WRE-Q	Hicut chain saw, 16 in.	07-Jan-12	11	5	256.99	0.05	24288
PVC23DRT	PVC pipe, 3.5-in., 8-ft	06-Jan-12	188	75	5.87	0.00	
SM-18277	1.25-in. metal screw, 25	01-Mar-12	172	75	6.99	0.00	21225
SW-23116	2.5-in. wd. screw, 50	24-Feb-12	237	100	8.45	0.00	21231
WR3/TT3	Steel matting, 4'x8'x1/8", .5" mesh	17-Jan-12	18	5	119.95	0.10	25595

Table name: CUSTOMER

CUST_NUME	CUST_LNAME	CUST_FNAME	CUST_INITIAL	CUST_AREACODE	CUST_PHONE	CUST_BALANCE
10010	Ramas	Alfred	A	615	844-2573	0.00
10011	Dunne	Leona	K	713	894-1238	615.73
10012	Smith	Kathy	W	615	894-2285	0.00
10013	Olowski	Paul	F	615	894-2180	0.00
10014	Orlando	Myron		615	222-1672	0.00
10015	O'Brian	Amy	B	713	442-3381	0.00
10016	Brown	James	G	615	297-1228	277.55
10017	Williams	George		615	290-2556	0.00
10018	Farriss	Anne	G	713	382-7185	0.00
10019	Smith	Olette	K	615	297-3809	0.00

Table name: ACCT_TRANSACTION

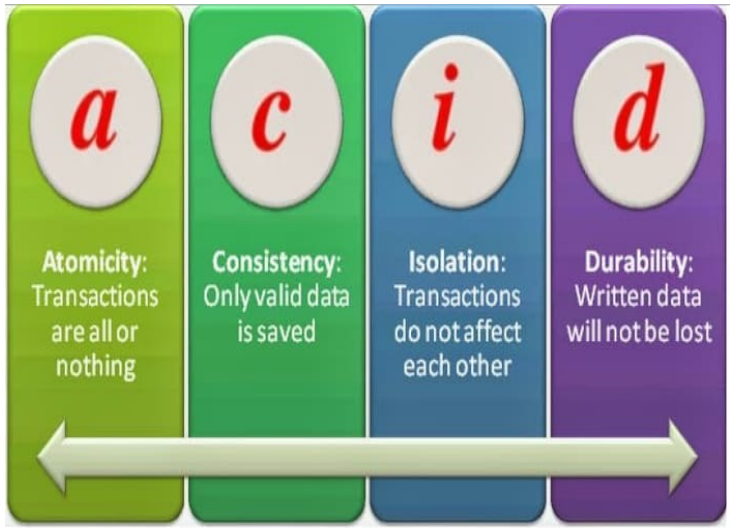
ACCT_TRANS_NUM	ACCT_TRANS_DATE	CUST_NUMBER	ACCT_TRANS_TYPE	ACCT_TRANS_AMOUNT
10003	17-Jan-12	10014	charge	329.66
10004	17-Jan-12	10011	charge	615.73
10006	29-Jan-12	10014	payment	329.66
10007	18-Jan-12	10016	charge	277.55

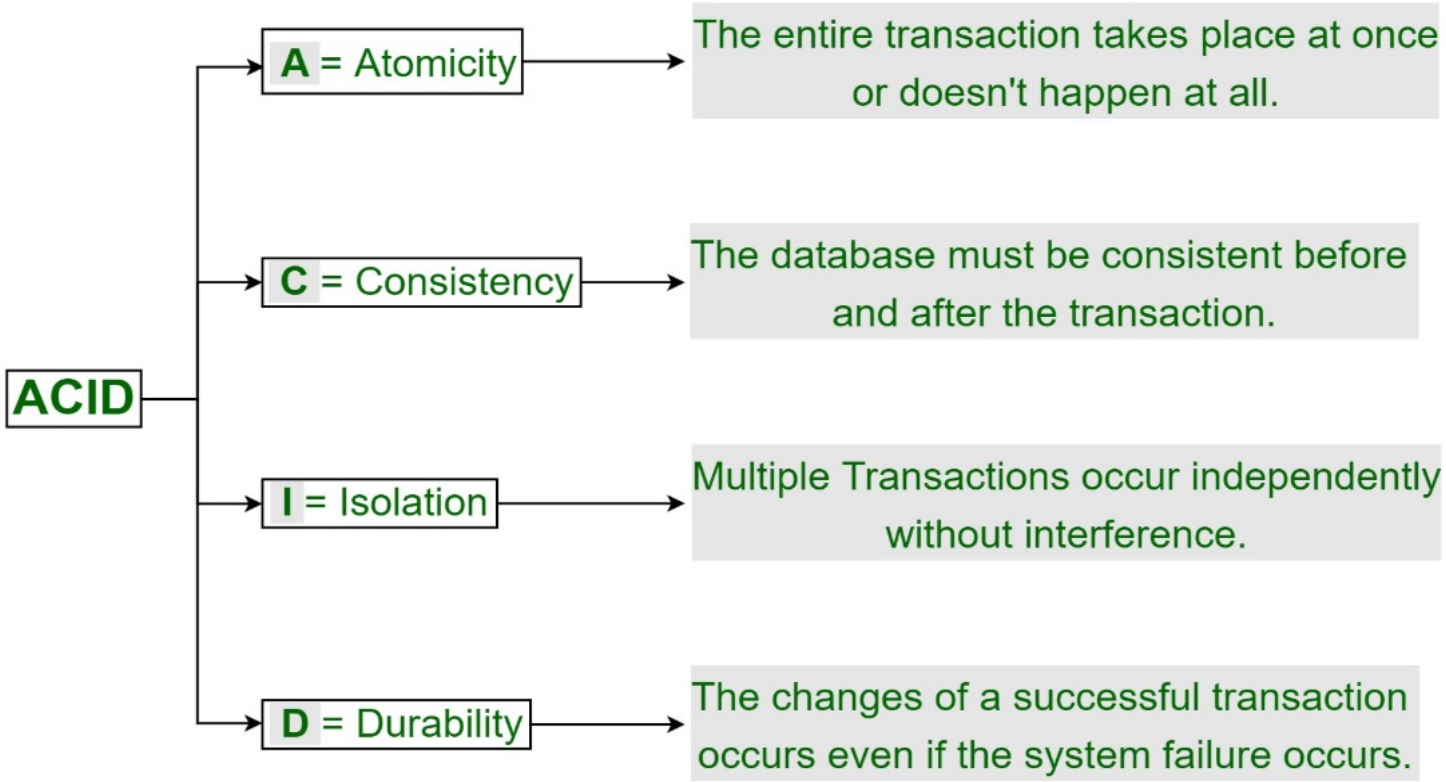
SOURCE: Course Technology/Cengage Learning

ACID Properties

İlişkisel VTYS'nde hareketlerin dört temel özelliği bulunmaktadır.

Atomicity (bölünemezlik) , Consistency (tutarlılık), Isolation (yalıtım) , Durability (dayanıklılık)



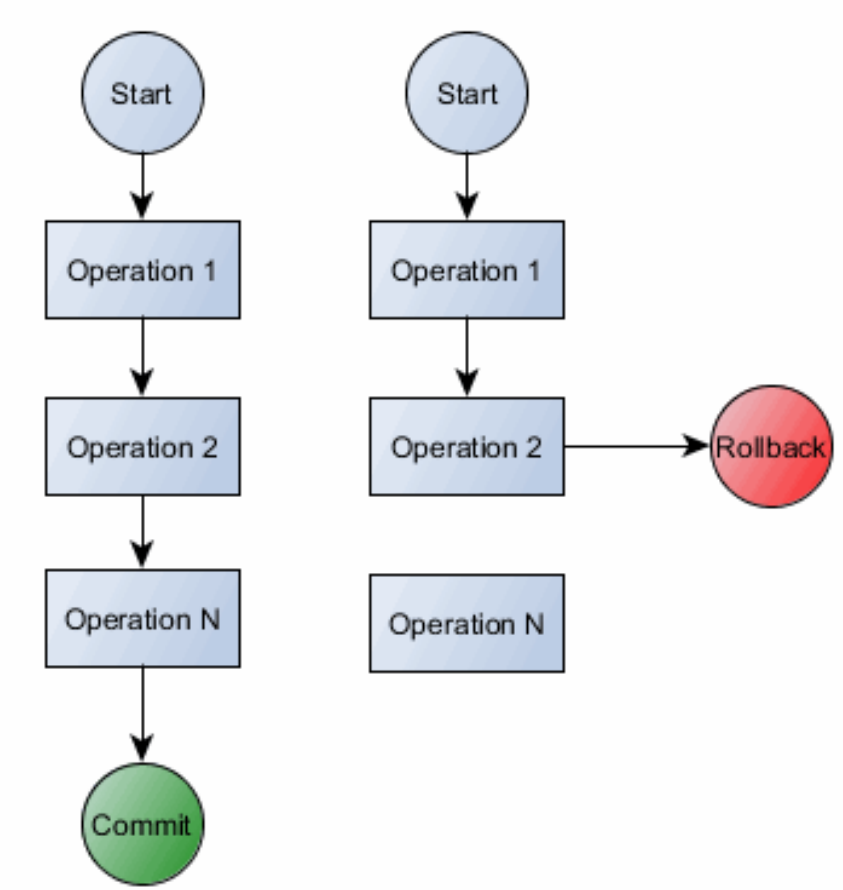


https://dev.to/princessanjana1996/acid-properties-in-databases-43aa

https://www.geeksforgeeks.org/acid-properties-in-dbms/

ACID: Atomicity (bölünemezlik)

- Bir hareketin bütün bölünmezliğidir yani bütün SQL sorgularının ayrılmadan bir bütün olarak değerlendirilip hepsinin başarılı olarak çalışmasının sağlanmasıdır.
- İşlemlerin kısmı gerçekleştirilmez yani hepsi başlar ve başarılı olarak biter yada işlem iptal edilir.
- Tamamlanamayan veya yarıda kalan hareketlerin işlemleri veritabanı tarafından geri alınır.

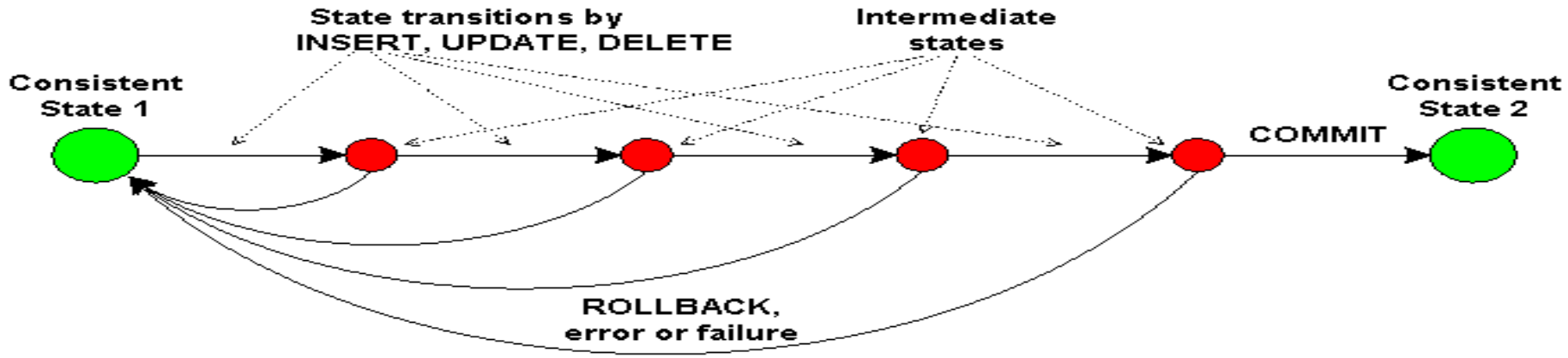


https://vladmihalcea.com/a-beginners-guide-to-acid-and-database-transactions/

ACID: Atomicity (bölünemezlik)

Before: X : 500	Y: 200
Transaction T	
T1	**T2**
Read (X) X: = X − 100 Write (X)	Read (Y) Y: = Y + 100 Write (Y)
After: X : 400	Y : 300

ACID: Consistency (tutarlılık)



- Gerçekleştirilen hareketlerde kayıtlı bulunan verinin tutarlılığının korunmasına **consistency** denir.
- Bir hareket gerçekleştirildiğinde veritabanı tutarlı (consistent) bir durumdan başka bir tutarlı duruma geçiş yapar
- Herhangi bir hareket İlişkisel bütünlük kriterini (relational integrity) ihlal ederse işlem geri alınır.

ACID: Consistency (tutarlılık)

Mehmet'in hesabında 700tl bulunmaktadır

Kamil'in hesabında 400tl bulunmaktadır.

- Kamil'in Mehmet'e 200 tl gönderilmesi gerekmektedir.
- Bu transfer iki aşamada gerçekleşmektedir.
  1. Kamil'in hesabından 200tl düşülecek
  2. Mehmet'in hesabına 200 tl eklenecek

ACID: Consistency (tutarlılık)

Mehmet'in hesabında 700tl bulunmaktadır

Kamil'in hesabında 400tl bulunmaktadır.

- Kamil'in Mehmet'e 200 tl gönderilmesi gerekmektedir.
- Bu transfer iki aşamada gerçekleşmektedir.
  1. Kamil'in hesabından 200tl düşülecek
  2. Mehmet'in hesabına 200 tl eklenecek

Hesaplar arasında transfer yapılırken toplam miktar aynı kalmalıdır

Hareketten önce Kamil (400)+ Mehmet (700) = 1100

Hareket başarılı olarak tamamlandıktan sonra Kamil (200)+ Mehmet (900) = 1100

ACID: Isolation (yalıtım)

Veritabanı yönetim sistemlerinde birden fazla hareket yönetilirken bir hareketin kullandığı verinin diğer hareketler tarafından kullanımasının engellenmesi ve diğer hareketlerden yalıtılmasına isolation denir.

ACID: Isolation (yalıtım)

Veritabanı yönetim sistemlerinde birden fazla hareket yönetilirken bir hareketin kullandığı verinin diğer hareketler tarafından kullanımasının engellenmesi ve diğer hareketlerden yalıtılmasına isolation denir.

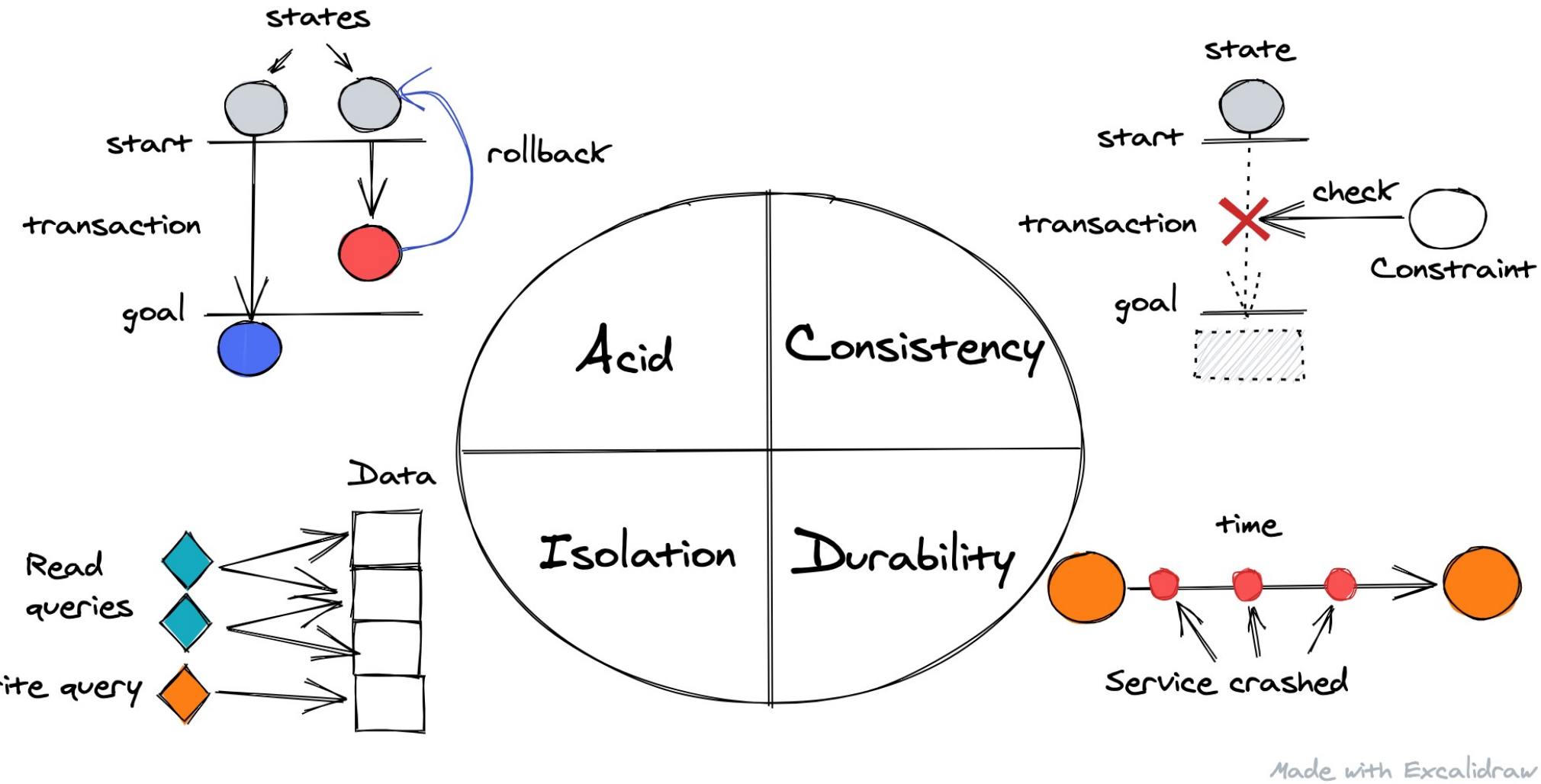
T	T''
Read (X) X: = X*100 Write (X) Read (Y) Y: = Y – 50 Write	Read (X) Read (Y) Z: = X + Y Write (Z)

Bir hesaptan aynı anda iki hareket (T1 ve T2) işlem gerçekleştirmek istediğinde önce başlanılan işlem (T1) tamamlanmadan diğerinin (T2) başlamaz beklenir.

ACID: Durability (dayanıklılık)

Sistem hataları ortaya çıktığında veya hareketlerde problemler ile karşılaşıldığında veritabanın hareket bilgilerini ve veritabanında kayıtlı olan veriyi kaybetmeden işleme devam edebilmesi **durability** olarak tanımlanır.

ACID



Yuji Tabata https://yuji.page/acid/

Hareket Yönetimi (Transaction Management)

- Bir banka hesabından gerçekleştirilmek istenen transfer tamamlanmadan ortaya çıkan sistem hatasını **veritabanı yönetim sistemleri düzeltmek zorundadır.**

https://www.interdb.jp/pg/pgsql09.html

Hareket Yönetimi (Transaction Management)

- Bir banka hesabından gerçekleştirilmek istenen transfer tamamlanmadan ortaya çıkan sistem hatasını **veritabanı yönetim sistemleri düzeltmek zorundadır**.

- VTYS gerçekleştirilen her işlemi log dosyalarına (hard diske)

  WAL (Write-Ahead Log) prensibini kullanarak yazmaktadır.

https://www.interdb.jp/pg/pgsql09.html

Hareket Yönetimi (Transaction Management)

- Bir banka hesabından gerçekleştirilmek istenen transfer tamamlanmadan ortaya çıkan sistem hatasını **veritabanı yönetim sistemleri düzeltmek zorundadır**.

- VTYS gerçekleştirilen her işlemi log dosyalarına (hard diske)

  WAL (Write-Ahead Log) prensibini kullanarak yazmaktadır.

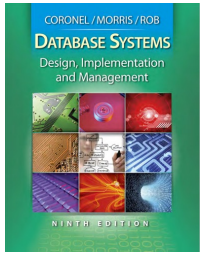
- Hataların düzeltilmesi işlemi VTYS tarafından periyodik olarak disk üzerindeki log dosyaları okunarak yapılır. Bu işlem **kontrol noktası (checkpoint)** olarak adlandırılır

https://www.interdb.jp/pg/pgsql09.html

SQL ile Hareket Yönetimi (Transaction Management with SQL)

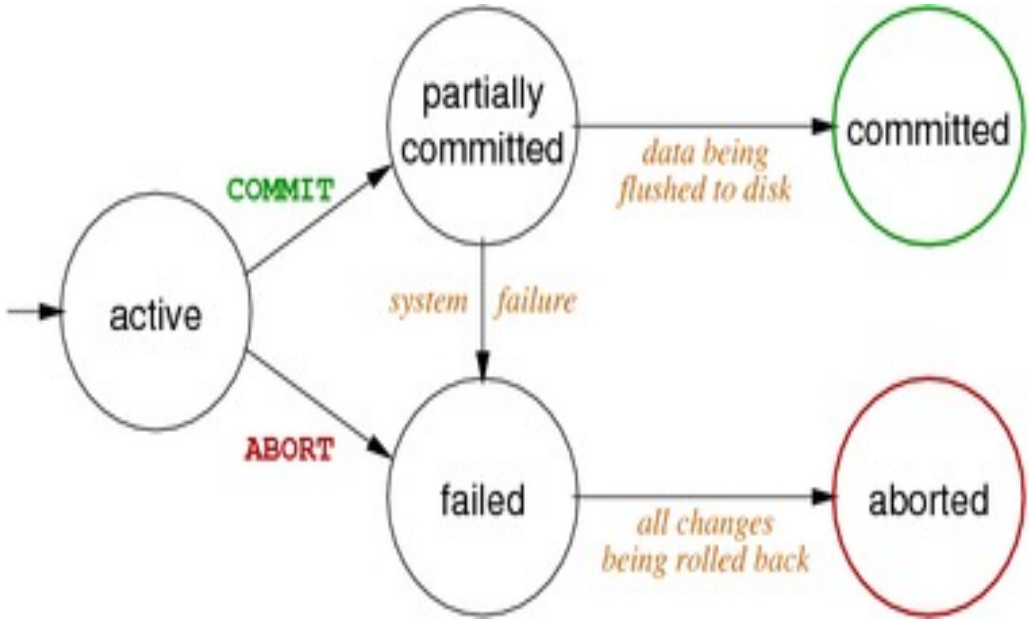
- COMMIT
- ROLLBACK

```
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;
```



Hareket Yönetimi (Transaction Management)

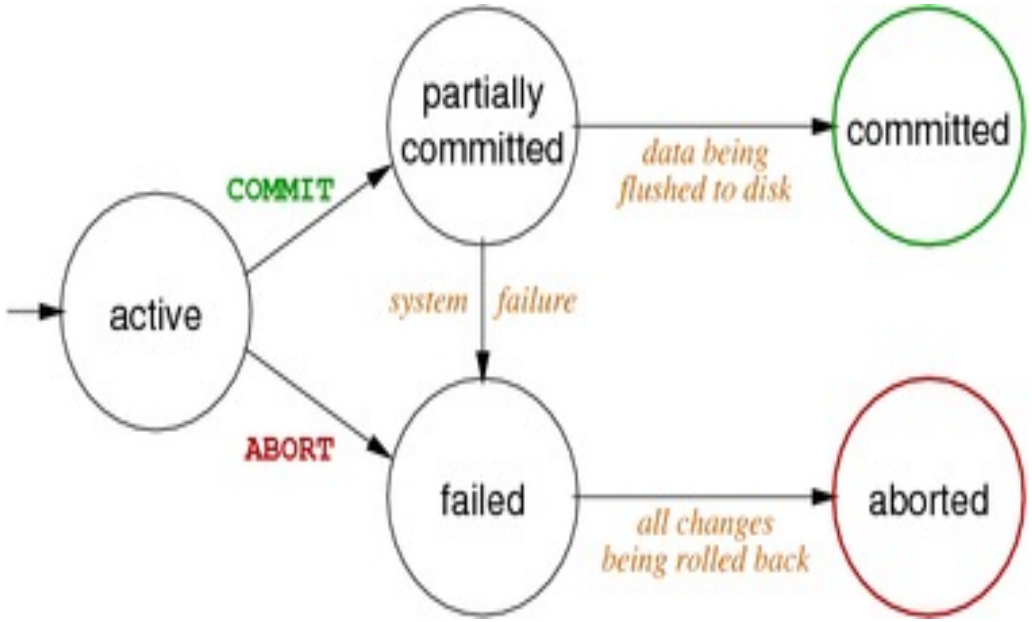
Bir hareket içerisindeki bütün işlemler başarıyla tamamlanmışsa bu işlem **commited (teslim etmek)** olarak adlandırılır ve yapılan bütün değişiklikler diske kaydedilir.



https://webcms3.cse.unsw.edu.au/COMP9315/16s1/resources/2352

Hareket Yönetimi (Transaction Management)

Bir hareket içerisindeki bütün işlemler başarıyla tamamlanmışsa bu işlem **commited (teslim etmek)** olarak adlandırılır ve yapılan bütün değişiklikler diske kaydedilir.



Bir hareket içerisinde herhangi bir hata oluştuğunda yapılan bütün değişiklikler geri alınarak hareket başlamadan önceki duruma geri dönülmesine **rollback (geri dönüş)** denir.

https://webcms3.cse.unsw.edu.au/COMP9315/16s1/resources/2352

Hareket Kayıtları (Transaction Log)

- Veritabanı yönetim sistemleri hareket kayıtları (transaction log) yardımıyla gerçekleştirilen haraketler ile ilgili işlem kayıtlarını saklar.
- ROLLBACK gerektiğinde (*programın normal olmayan biçimde kapatılması, server hataları, disk problemleri* vb.) hareket kayıtları kullanılarak veritabanı tutarlı duruma (consistent) geçiş yapar.

Hareket Kayıtları (Transaction Log)

Transaction log aşağıdaki bilgileri içerir:

1. Hareket başlangıç bilgisi
2. SQL (Ürün güncelle)
3. SQL (Müşteri güncellenmesi)
4. Hareketin tamamlanması (commit)

Hareket Kayıtları (Transaction Log)

Transaction log aşağıdaki bilgileri içerir:

1. Hareket başlangıç bilgisi
2. SQL (Ürün güncelle)
3. SQL (Müşteri güncellenmesi)
4. Hareketin tamamlanması (commit)

TABLE 10.1 A Transaction Log

TRL ID	TRX NUM	PREV PTR	NEXT PTR	OPERATION	TABLE	ROW ID	ATTRIBUTE	BEFORE VALUE	AFTER VALUE
341	101	Null	352	START	****Start Transaction	1			
352	101	341	363	UPDATE	PRODUCT	1558-QW1	PROD_QOH	25	23
363	101	352	365	UPDATE	CUSTOMER	10011	CUST_ BALANCE	525.75	615.73
365	101	363	Null	COMMIT	**** End of Transaction	4			

TRL_ID = Transaction log record ID
TRX_NUM = Transaction number
PTR = Pointer to a transaction log record ID
(Note: The transaction number is automatically assigned by the DBMS.)

PostgreSQL Transaction

```
BEGIN;

UPDATE accounts SET balance = balance - 100.00 WHERE name = 'Alice';

SAVEPOINT my_savepoint;

UPDATE accounts SET balance = balance + 100.00 WHERE name = 'Bob';
-- Yanlış hesaba gonderildi

ROLLBACK TO my_savepoint;  -- bir önceki tutarlı duruma geçiş

UPDATE accounts SET balance = balance + 100.00 WHERE name = 'Wally';

COMMIT;
```

PostgreSQL Transaction (Stored Procedures)

- Kullanıcı tanımlı fonksiyonlar (user-defined functions) hareket işlemlerini gereçekleştiremezler.
- Dolayısıyla bir hareketi başlatıp , **commit**, **rollback** işlemleri için stored procedure kullanılır.

```
create [or replace] procedure procedure_name(parameter_list)
language plpgsql
as $$
declare
-- variable declaration
begin
-- stored procedure body
end; $$
```

https://www.postgresqltutorial.com/postgresql-create-procedure/

PostgreSQL Transaction (Stored Procedures)

```
create or replace procedure paratransferi(gönderici int, alıcı int, miktar dec)
language plpgsql
as $$
begin
    -- Gödericinin hesabından gönderilen miktar düşecek
    update accounts
    set balance = balance - miktar
    where id = gönderici;

    -- Alıcının hesabına gönderilen miktar eklenecek
    update accounts
    set balance = balance + miktar
    where id = alıcı;

    commit;
end;
$$
```

https://www.postgresqltutorial.com/postgresql-create-procedure/

PostgreSQL Transaction (Stored Procedures)

```
create or replace procedure paratransferi(gönderici int, alıcı int, miktar dec)
language plpgsql
as $$
begin
    -- Gödericinin hesabından gönderilen miktar düşecek
    update accounts
    set balance = balance - miktar
    where id = gönderici;

    -- Alıcının hesabına gönderilen miktar eklenecek
    update accounts
    set balance = balance + miktar
    where id = alıcı;

    commit;
end;
$$
```

```
call paratransferi(1,2,1000)
```

```
drop paratransferi;
```

https://www.postgresqltutorial.com/postgresql-create-procedure/

Dinlediğiniz için

Teşekkürler...

İyi çalışmalar...

33