

Veri Tabanı Yönetim Sistemleri HAREKET ÖRNEKLERİ

	Bağlantı A	Bağlantı B
1	<pre>use movie; start transaction; insert into person (id, name) values (10000, 'Cüneyt Arkın'); select count(*) from person; commit;</pre>	<pre>use movie; select count(*) from person; select count(*) from person; select count(*) from person;</pre>
	B bağlantısı, A bağlantısı onaylanana kadar (commit komutu yürütene kadar) yeni eklenen kaydı göremez.	
2	<pre>start transaction; insert into person (id, name) values (10001, 'Kemal Sunal'); select count(*) from person; rollback; select count(*) from person;</pre>	<pre>select count(*) from person; select count(*) from person; select count(*) from person;</pre>
	A bağlantısındaki işlemler, rollback ile geri sarıldığı için, geri sarma işleminden sonra hem A bağlantısında, hem de B bağlantısında yeni eklenen veya değiştirilen hiçbir kayıt görülmez.	

3	<p>start transaction;</p> <p>select * from person where id = 10000 lock in share mode;</p> <p>rollback;</p>	<p>start transaction;</p> <p>update person set id = 20000 where id = 10000;</p> <p>rollback;</p> <p>A bağlantısında select cümlecisi "lock in share mode" kilidi ile (S) kilitlendiği için, B bağlantısındaki update cümlecisi (X kilidi koyulmasını gerektirir) askıya alınır. Zaman aşımı dolana kadar ya da A bağlantısı rollback yapılana kadar askıda kalır.</p>
4	<p>start transaction;</p> <p>select * from person where id = 10000 lock in share mode;</p> <p>update person set id = 20000 where id = 10000;</p>	<p>start transaction;</p> <p>update person set id = 20000 where id = 10000;</p> <p>A bağlantısında select cümlecisi "lock in share mode" kilidi ile (S) kilitlendiği için, B bağlantısındaki update cümlecisi (X kilidi koyulmasını gerektirir) askıya alınır. Zaman aşımı dolana kadar ya da A bağlantısı rollback yapılana kadar askıda kalır. Ancak bu sırada A hareketi içerisinde de aynı satırı güncellemek için bir komut yürütürse bu hareket de bu kilit üzerindeki askı kuyruğuna alınır. Ancak bu kuyruktaki her iki hareket de birbirini beklediği için sonuçta bir "ölümcül kilitlenme" olur. MySQL bu durumu sezerek her iki komutu da hata vererek sonlandırır.</p>

5	<pre> use movie; SET GLOBAL TRANSACTION ISOLATION LEVEL READ UNCOMMITTED; start transaction; select count(*) from person; insert person (id, name) values (10002, 'Sener Şen'); select count(*) from person; commit; SELECT @@global.tx_isolation; SELECT @@tx_isolation; </pre>	<pre> use movie; select count(*) from person; select count(*) from person; select count(*) from person; </pre>
	<p>READ UNCOMMITTED yalıtım düzeyinde, A hareketinde yapılan değişiklikler henüz onaylanmamışken (commit) dahi B hareketi içerisinde görülebilir.</p>	

<p>6</p>	<pre> use movie; SET GLOBAL TRANSACTION ISOLATION LEVEL READ COMMITTED; start transaction; select avg(id) from person; update person set id = 20000 where id = 10000; select avg(id) from person; commit; SELECT @@global.tx_isolation; SELECT @@tx_isolation; </pre>	<pre> use movie; start transaction; select avg(id) from person; select avg(id) from person; select avg(id) from person; select avg(id) from person; rollback; </pre>
<p>READ COMMITTED yalıtım düzeyinde, A hareketinde yapılan değişiklikler henüz onaylanmamışken (commit) dahi B hareketi içerisinde kesinlikle görülmez. Bu değişiklikler, ancak A hareketi onaylandığı (commit) zaman B hareketi içerisinde görülebilir.</p>		

7	<p>SET GLOBAL TRANSACTION ISOLATION LEVEL REPEATABLE READ;</p> <p>start transaction;</p> <p>select avg(id) from person;</p> <p>update person set id = 10000 where id = 20000;</p> <p>select avg(id) from person;</p> <p>commit;</p>	<p>use movie;</p> <p>start transaction;</p> <p>select avg(id) from person;</p> <p>select avg(id) from person;</p> <p>select avg(id) from person;</p> <p>select avg(id) from person;</p> <p>rollback;</p> <p>select avg(id) from person;</p>
	<p>REPEATABLE READ yalıtım düzeyinde, A hareketinde yapılan değişiklikler henüz onaylanmamışken (commit) dahi B hareketi içerisinde kesinlikle görülmez. Bu değişiklikler, A hareketi onaylandığı (commit) zaman dahi B hareketi içerisinde görülemez. Bu yalıtım seviyesindeki mantık, B hareketinin başladığı andan itibaren diğer hareketlerden yalıtılmış olmasını sağlamak ve her okuma (select) işleminde aynı sonuçları elde etmektir (diğer hareketlerin onaylanmış değişikliklerini dahi görmemek).</p>	

8	<pre> use movie; SET GLOBAL TRANSACTION ISOLATION LEVEL SERIALIZABLE; start transaction; select * from person where name like 'Cüneyt Arkin'; update person set id=10000 where id=20000; rollback; </pre>	<pre> use movie; start transaction; select * from person where name like 'Cüneyt Arkin'; rollback; </pre>
	<p>SERIALIZABLE yalıtım düzeyi, REPEATABLE READ yalıtım düzeyi ile aynıdır. Tek fark, özellikle belirtilmese dahi, bütün SELECT cümleciklerinin otomatik olarak okuma kilidi (S) oluşturacak (SELECT ... LOCK IN SHARE MODE) şekilde değiştirilmeleridir. B hareketindeki seçim (select) cümlecği nedeni ile bazı kayıtlar üzerinde okuma (S) kilidi oluşturulur. Dolayısı ile A hareketindeki güncelleme (update) cümlecği de yazma kilidi (X) gerektirdiği için askıya alınır. Ancak B hareketi sonlanınca, A hareketindeki güncelleme cümlecği askıdan indirilir ve yürütülür.</p>	