

Өгөгдлийн сангийн програмчлал – Семинар 12 /транзакшин 2/

[1]. Хүснэгт Person(name, age) болон доорх транзакшин байв. Q1, Q2 асуулгууд ямагт нэг бүхэл (atomically) байдлаар ажилладаг.

--T

Begin Transaction;

Q1: Select Avg(age) From Person;

<read-only activity>

Q2: Select Avg(age) From Person;

Commit;



1 (a). Тухайн системийн бусад транзакшин бүгд Serializable ба Read-Only гэж зарлагдсан байг. Q1 Q2 ямагт адил утга буцаадаг байх нөхцөл дор T транзакшины ажиллаж болох хамгийн сул тусгаарлалтын түвшинг олж, дэлгэрэнгүй тайлбарлана уу. Сонголт: Read-Uncommitted, Read-Committed, Repeatable-Read, Serializable

Тайлбар: Эхлээд, T нь serializable тусгаарлалтын түвшинтэй байна гэж үзвэл бусад транзакшины serializable байдалд нь нөлөөлж чадахгүй бөгөөд бусад транзакшин ч гэсэн T транзакшины serializable байдалд нөлөөлж чадахгүй учир нь T болон бусад транзакшин бүгд зөвхөн уншилт хийдэг. Иймд T нь serializable тусгаарлалтын түвшинтэй байж болно. **Хоёрт нь,** T - г repeatable-read isolation level – тэй байна гэж үзвэл бусад транзакшин бүгд зөвхөн уншилт хийх буюу бичилт/insert, update/ хийхгүй тул T – ийн тусгаарлалтын түвшинд нөлөөлж чадахгүй. Мөн T нь зөвхөн уншилт хийдэг транзакшин тул бусад транзакшины serializable түвшинд нөлөөлөхгүй тул T нь repeatable-read тусгаарлалтын түвшинтэй байж болно. **Гуравт нь,** T – г read-committed тусгаарлалтын түвшинтэй буюу бохир уншилт хийдэггүй гэж үзвэл бусад транзакшин нь зөвхөн уншилт хийх тул T – д бохир уншилт үүсгэх ямар нэг нөхцөл бүрдэхгүй юм. Мөн T нь зөвхөн уншилт хийх тул бусад транзакшины serializable түвшинд нөлөөлж чадахгүй. Иймд T нь read-committed тусгаарлалтын түвшинтэй байж болно. **Дөрөвт нь,** T – г read-uncommitted тусгаарлалтын түвшинтэй гэж үзвэл мөн T нь бусад транзакшины serializable түвшинд, бусад транзакшин T – ийн тусгаарлалтын түвшинд нөлөөлж чадахгүй учир нь бүгд зөвхөн уншилт хийдэг. Иймд T нь read-uncommitted түвшинтэй байж болно. Иймд T болон бусад транзакшин бүгд зөвхөн уншилт хийх тул T нь ямар ч тусгаарлалтын түвшинтэй байсан T нь ижил утга буцаах юм. Эцэст нь байж болох хамгийн сул тусгаарлалтын түвшинг олно гэсэн тул serializable>repeatable-read>read-committed>read-uncommitted -> гэдгээс **read-uncommitted** тусгаарлалтын түвшин гэж олдох юм.

Өгөгдлийн сангийн програмчлал – Семинар 12 /транзакшин 2/

--T

Begin Transaction;

Q1: Select Avg(age) From Person;

<read-only activity>

Q2: Select Avg(age) From Person;

Commit;



1 (b). Тухайн системийн бусад транзакшин бүгд Serializable ба тэд зөвхөн select, update, delete үйлдлүүдийг агуулдаг байв. Q1, Q2 ямагт адил утга буцаадаг байх нөхцөл дор T транзакшины ажиллаж болох хамгийн сул тусгаарлалтын түвшинг олж, дэлгэрэнгүй тайлбарлана уу. Сонголт: Read-Uncommitted, Read-Committed, Repeatable-Read, Serializable

Тайлбар: Эхлээд, T нь serializable тусгаарлалтын түвшинтэй байна гэж үзвэл T нь зөвхөн уншилт хийх тул бусад транзакшины serializable байдалд нь нөлөөлж чадахгүй. Харин бусад транзакшины талаас select – үүд нь T – ийн serializable түвшинд нөлөөлж чадахгүй харин update, delete -үүд нь нөлөөлөх магадлалтай юм. Гэхдээ энэ тохиолдолд T болон бусад транзакшины тусгаарлалтын түвшин serializable тул select, update, delete – үүд бүрэн ажиллаж дууссаны дараа эсвэл ажлахаас нь өмнө ажиллах боломжтой. Харин T болон бусад транзакшинууд нь өөр өөр хүний транзакшин тул T – ийн 2 уншилтийн голоор select, update, delete үйлдэл бүхий транзакшинууд бүхэл цогцоороо орж ажиллах боломжтой харагдах хэдий ч T – ийн 2 уншилт нэг хүснэгтийн ижил баганаас уншилт хийж байгаа учир голоор нь орж ижил багана дээр update хийгдэхэд T – д non-repeatable read үүсч, serializable түвшинг нь зөрчинө. Иймд T нь serializable – аар ажиллах үед T – ийн 2 уншилтийн голоор ямар нэг транзакшин орж ажиллах боломжгүй буюу сериалаар л ажиллах учраас T – ийн 2 select ижил үр дүн буцаах нь тодорхой юм. **Хоёрт нь,** T – г repeatable-read isolation level – тэй байна гэж үзвэл T нь зөвхөн уншилт хийх учраас бусад транзакшины serializable байдалд нөлөөлж чадахгүй. Харин бусад транзакшины талаас select – үүд нь T – ийн serializable түвшинд нөлөөлж чадахгүй харин update, delete – үүд нь нөлөөлөх магадлалтай юм. Гэхдээ энэ тохиолдолд T нь non-repeatable read – ийг зөвшөөрөхгүй тул T – ийн 2 уншилтийн голоор нь update хийгдэх боломжгүй болно. Мөн delete – ийг T – ийн 2 уншилтийн голоор ороход phantom tuple гэж үздэггүй тул delete хийгдэхэд алдаа үүснэ. Иймд T нь бусад транзакшинтай сериал байдлаар л ажиллах боломжтой болно гэдгээс энэ тохиолдолд мөн T – ийн 2 уншилт ижил утга буцаана. T нь үлдсэн 2 тусгаарлалтын түвшинд ямагт ижил утга буцаах боломжгүй нь илэрхий учраас хариулт **repeatable-read** болно.

Өгөгдлийн сангийн програмчлал – Семинар 12 /транзакшин 2/



--T

Begin Transaction;

Q1: Select Avg(age) From Person;

<read-only activity>

Q2: Select Avg(age) From Person;

Commit;

1 (с). Тухайн системийн бусад транзакшин бүгд Serializable ба тэдний талаар өөр нэмэлт мэдээллийг байхгүй байв. Q1, Q2 ямагт адил утга буцаадаг байх нөхцөл дор T транзакшины ажиллаж болох хамгийн сул тусгаарлалтын түвшинг олж, дэлгэрэнгүй тайлбарлана уу. Сонголт: Read-Uncommitted, Read-Committed, Repeatable-Read, Serializable

Тайлбар: Эхлээд, T нь serializable тусгаарлалтын түвшинтэй байна гэж үзвэл T нь зөвхөн уншилт хийх тул бусад транзакшины serializable байдалд нь нөлөөлж чадахгүй. Харин бусад транзакшины талаас select – тэй бол select – үүд нь T – ийн serializable түвшинд нөлөөлж чадахгүй харин insert, update, delete – үүдтэй бол эдгээр нь нөлөөлөх магадлалтай юм. Гэхдээ энэ тохиолдолд T болон бусад транзакшины тусгаарлалтын түвшин serializable тул select, insert, update, delete – үүд бүрэн ажиллаж дууссаны дараа эсвэл ажлахаас нь өмнө ажиллах боломжтой. Харин T болон бусад транзакшинууд нь өөр өөр хүний транзакшин тул T – ийн 2 уншилтийн голоор select, update, delete үйлдэл бүхий транзакшинууд бүхэл цогцоороо орж ажиллах боломжтой харагдах хэдий ч T – ийн 2 уншилт нэг хүснэгтийн ижил баганаас уншилт хийж байгаа учир голоор нь орж ижил багана дээр update хийгдэхэд T – д non-repeatable read үүсч, serializable түвшинг нь зөрчинө. Иймд T нь serializable – аар ажиллах үед T – ийн 2 уншилтийн голоор ямар нэг транзакшин орж ажиллах боломжгүй буюу сериалаар л ажиллах учраас T – ийн 2 select ижил үр дүн буцаана. **Хоёрт нь,** T – г repeatable-read isolation level – тэй байна гэж үзвэл T нь зөвхөн уншилт хийх учраас бусад транзакшины serializable байдалд нөлөөлж чадахгүй. Харин бусад транзакшины талаас select – тэй бол select – үүд нь T – ийн serializable түвшинд нөлөөлж чадахгүй харин insert, update, delete – үүдтэй бол эдгээр нь нөлөөлөх магадлалтай юм. Энэ тохиолдолд T нь мөн non-repeatable read – ийг зөвшөөрөхгүй тул T – ийн 2 уншилтийн голоор нь update хийгдэх боломжгүй болно. Эндээс энэ үед T – ийн 2 уншилт ямагт ижил утга буцаах боломжгүй юм. Мөн T нь үлдсэн 2 тусгаарлалтын түвшинд ямагт ижил утга буцаах боломжгүй нь илэрхий учраас T – ийн 2 уншилт нь зөвхөн serializable түвшинтэй үед л ямагт ижил утга буцаана. Иймд хариулт **serializable** болно.

Өгөгдлийн сангийн програмчлал – Семинар 12 /транзакшин 2/

T1:

Begin Transaction;

Q1: Select Avg(age) From Person;

<read-only activity>

Commit;

T2:

Begin Transaction;

Q2: Select Avg(age) From Person;

<read-only activity>

Commit;

1 (d). T1, T2 транзакшины тусгаарлалтын түвшин Serializable байв. Дээрх (a), (b), (c) тохиолдлуудын хувьд бодлогыг дахин бодно уу. Дээрх (a), (b), (c) тохиолдлуудын хувьд Q1, Q2 ямагт адил үр дүн буцаах эсэхийг шалга, хариултаа нарийн тайлбарла.

Тайлбар: (a) – ийн T1, T2 нь ямагт ижил утга буцаах боломжтой. Учир нь бусад транзакшин нь бүгд уншилт хийдэг учраас T1, T2 – ийн унших өгөгдөлд ямар нэг байдлаар өөрчлөлт оруулж чадахгүй юм. Мөн T1, T2 болон бусад транзакшин бүгд зөвхөн уншилт хийх тул нэгнийхээ serializable түвшинд нөлөөлж чадахгүй. Харин (b), (c) – ийн хувьд T1, T2 нь бусад транзакшинтай сериал байдлаар ажиллах үед л ижил биш утга буцаах боломжтой бөгөөд энэ нь ямагт ижил утгатай үр дүн буцаана гэдэгт харших тул **(a) бодлогын хувьд л T1, T2 нь ямагт ижил утга буцаах боломжтой юм.**



Өгөгдлийн сангийн програмчлал – Семинар 12 /транзакшин 2/

[2]. Хүснэгт Worker(name, pay), энд name багана primary key байв. Доорх 2 транзакшин зэрэг ажилладаг. S1, S2, S3, S4 стетментүүд нэг бүхэл (atomically) байдлаар ажилладаг. Эхлэхдээ worker хүснэгт (Amy, 50) гэсэн мөр бичлэгтэй байжээ.

T1:
Begin Transaction
S1: update Worker set pay = 2*pay where name = 'Amy'
S2: update Worker set pay = 3*pay where name = 'Amy'
Commit

T2:
Begin Transaction
S3: update Worker set pay = pay-20 where name = 'Amy'
S4: update Worker set pay = pay-10 where name = 'Amy'
Commit

2 (a). T1, T2 транзакшинууд Serializable тусгаарлалтын түвшинтэй бөгөөд бүрэн ажиллаж дуусдаг байг. Amy –гийн pay хэд болох боломжтой вэ? Яагаад?

Тайлбар: T1, T2 – ийн тусгаарлалтын түвшингээс үл хамааран сериалаар буюу T1;T2, T2;T1 гэж ажиллах боломжтой. Мөн T1, T2 нь serializable тусгаарлалтын түвшинтэй хэдий ч өөр өөр хүний транзакшин тул T1:T2, T2:T1 гэж ажиллах боломж байх бөгөөд эдгээр тохиолдлуудыг шалгаж үзье. S1;T2;S2 тохиолдлыг авч үзвэл update нь уншаад бичдэг учраас S1, S2 – ийг ижил баганаас уншилт хийнэ гэж үзэх бөгөөд T2 нь тэдгээрийн голоор орж бүрэн ажиллаж дуусахад T2 нь тухайн S1, S2 уншиж байгаа багана дээр өөрчлөлт хийж байгаа учир T1 – д non-repeatable read үүсч serializable түвшинд нь харшлах юм. S3;T1;S4 тохиолдолд мөн өмнөхтэй адил байдлаар T2 – д non-repeatable read үүсэх тул T2 – ийн serializable түвшинд харшлах тул Amy – ийн pay – ийн боломжтой утгууд нь **T1;T2 => 270, T2;T1 => 120** болно.



Өгөгдлийн сангийн програмчлал – Семинар 12 /транзакшин 2/

<p>T1:</p> <p>Begin Transaction</p> <p>S1: update Worker set pay = 2*pay where name = 'Amy'</p> <p>S2: update Worker set pay = 3*pay where name = 'Amy'</p> <p>Commit</p>	<p>T2:</p> <p>Begin Transaction</p> <p>S3: update Worker set pay = pay-20 where name = 'Amy'</p> <p>S4: update Worker set pay = pay-10 where name = 'Amy'</p> <p>Commit</p>
<p>2 (b). T1, T2 транзакшинууд Read-Committed тусгаарлалтын түвшинтэй бөгөөд бүрэн ажиллаж дуусдаг байг. Amy –гийн pay хэд болох боломжтой вэ? Яагаад?</p>	
<p>Тайлбар: T1, T2 – ийн тусгаарлалтын түвшингээс үл хамаарч сериалаар буюу T1;T2, T2;T1 гэж ажиллах боломжтой. Мөн T1, T2 нь read-committed тусгаарлалтын түвшинтэй тул T1:T2, T2:T1 гэж ажиллах боломж байх бөгөөд эдгээр тохиолдлуудыг шалгаж үзье. Read-committed тусгаарлалтын түвшин нь бохир уншилтаас бусдыг нь зөвшөөрөх тул зэрэг ажиллах үед T1 нь T2 – д, T2 нь T1 – д бохир уншилт үүсэх эсэхийг шалгана. S1;T2;S2 тохиолдлыг авч үзвэл update нь уншаад бичдэг учраас S1, S2 – ийг ижил баганаас уншилт хийж байна гэж үзэх бөгөөд S1 нь T2 – ийн ажиллахаас өмнө унших тул бохир уншилт хийхгүй, S2 нь T2 – ийг ажиллаад баталгаажсаны дараа унших тул мөн бохир уншилт хийхгүй тиймээс T1 – д бохир уншилт үүсэхгүй юм. Харин T2 нь T1 – ийг бүрэн баталгаажаагүй байхад буюу S1 ажилласны дараа шууд ажиллах тул S3, S4 нь баталгаажаагүй өгөгдөл уншиж, T2 – д бохир уншилт үүснэ. S3;T1;S4 тохиолдолд мөн T1 нь дээрхтэй адилханаар бохир уншилт хийх тул энэ бодлогын хувьд T1, T2 зөвхөн сериалаар нь л ажиллах боломжтой болох юм. Иймд Amy – ийн pay – ийн боломжтой утгууд нь T1;T2 => 270, T2;T1 => 120 болно.</p>	



Өгөгдлийн сангийн програмчлал – Семинар 12 /транзакшин 2/

<p>T1:</p> <p>Begin Transaction</p> <p>S1: update Worker set pay = 2*pay where name = 'Amy'</p> <p>S2: update Worker set pay = 3*pay where name = 'Amy'</p> <p>Commit</p>	<p>T2:</p> <p>Begin Transaction</p> <p>S3: update Worker set pay = pay-20 where name = 'Amy'</p> <p>S4: update Worker set pay = pay-10 where name = 'Amy'</p> <p>Commit</p>
<p>2 (c). T1 транзакшин Read-Committed тусгаарлалтын түвшинтэй, T2 транзакшин Read-Uncommitted тусгаарлалтын түвшинтэй бөгөөд 2 транзакшин бүрэн ажиллаж дуусдаг байг. Amy –гийн pay хэд болох боломжтой вэ? Яагаад?</p>	
<p>Тайлбар: T1, T2 – ийн тусгаарлалтын түвшингээс үл хамааран сериалаар буюу T1;T2, T2;T1 гэж ажиллах боломжтой. Мөн T1 нь read-committed, T2 нь Read-Uncommitted тусгаарлалтын түвшинтэй тул T1:T2, T2:T1 гэж ажиллах боломж байх бөгөөд эдгээр тохиолдлуудыг шалгаж үзье. Read-committed тусгаарлалтын түвшин нь бохир уншилтаас бусдыг нь зөвшөөрөх тул зэрэг ажиллах үед T2 нь T1 – д бохир уншилт үүсэх эсэхийг шалгана, харин Read-Uncommitted тусгаарлалтын түвшин нь бохир уншилт, non-repeatable read, phantom – уудыг бүгдийг нь зөвшөөрдөг. S1;T2;S2 тохиолдлыг авч үзвэл update нь уншаад бичдэг учраас S1, S2 – ийг ижил баганаас уншилт хийж байна гэж үзэх бөгөөд S1 нь T2 – ийн ажиллахаас өмнө унших тул бохир уншилт хийхгүй, S2 нь T2 – ийг ажиллаад баталгаажсаны дараа унших тул мөн бохир уншилт хийхгүй тиймээс T1 – д бохир уншилт үүсэхгүй юм. Харин T2 нь T1 – ийг бүрэн баталгаажаагүй байхад буюу S1 ажилласны дараа шууд ажиллах тул S3, S4 нь баталгаажаагүй өгөгдөл уншиж, T2 – д бохир уншилт үүснэ. Гэхдээ T2 – ийн тусгаарлалтын түвшин нь бохир уншилтийг зөвшөөрөх тул энэ тохиолдол боломжтой болно. Харин S3;T1;S4 тохиолдолд мөн T1 нь дээрхтэй адилханаар бохир уншилт хийх бөгөөд T1 – ийн тусгаарлалтын түвшин бохир уншилтыг зөвшөөрөхгүй тул энэ бодлогын хувьд T1, T2 зөвхөн сериалаар болон S1:T2:S2 - аар ажиллах боломжтой болох юм. Иймд Amy – ийн pay – ийн боломжтой утгууд нь T1;T2 => 270, T2;T1 => 120, S1:T2:S2 => 210 болно.</p>	



Өгөгдлийн сангийн програмчлал – Семинар 12 /транзакшин 2/

T1: Begin Transaction S1: update Worker set pay = 2*pay where name = 'Amy' S2: update Worker set pay = 3*pay where name = 'Amy' Commit	T2: Begin Transaction S3: update Worker set pay = pay-20 where name = 'Amy' S4: update Worker set pay = pay-10 where name = 'Amy' Commit
2 (d). T1, T2 транзакшинууд Read-Uncommitted тусгаарлалтын түвшинтэй бөгөөд бүрэн ажиллаж дуусдаг байг. Amy –гийн pay хэд болох боломжтой вэ? Яагаад?	
Тайлбар: T1, T2 – ийн тусгаарлалтын түвшингээс үл хамааран сериалаар буюу T1;T2, T2;T1 гэж ажиллах боломжтой. Мөн T1, T2 нь Read-Uncommitted тусгаарлалтын түвшинтэй тул T1:T2, T2:T1 гэж ажиллах боломжтой. Read-Uncommitted тусгаарлалтын түвшин нь бохир уншилт, non-repeatable read, phantom – уудыг бүгдийг нь зөвшөөрдөг. Иймд 2 транзакшины statement – үүдийг дэс дарааллыг алдагдуулахгүйгээр зохиосон бүх хослуудыг энэ тохиолдолд гаргаж авах боломжтой болох юм. Үүнээс буцаах боломжтой утгууд: T1;T2 => 270 T2;T1 => 120 S1;T2;S2 => 210 /T2 – д бохир уншилт , T1 non-repeatable read үүснэ. Гэхдээ Read-Uncommitted зөвшөөрнө. / S3;T1;S4 => 170 /T1 – д бохир уншилт , T2 non-repeatable read үүснэ. Гэхдээ Read-Uncommitted зөвшөөрнө. / S1;S3;S2;S4 => 230 /T2, T1 – д бохир уншилт , T1, T2 non-repeatable read үүснэ. Гэхдээ Read-Uncommitted зөвшөөрнө. / S3;S1;S4;S2 => 150 / T1, T2 – д бохир уншилт , T2, T1 non-repeatable read үүснэ. Гэхдээ Read-Uncommitted зөвшөөрнө. / болно.	



Өгөгдлийн сангийн програмчлал – Семинар 12 /транзакшин 2/

<p>T1:</p> <p>Begin Transaction</p> <p>S1: update Worker set pay = 2*pay where name = 'Amy'</p> <p>S2: update Worker set pay = 3*pay where name = 'Amy'</p> <p>Commit</p>	<p>T2:</p> <p>Begin Transaction</p> <p>S3: update Worker set pay = pay-20 where name = 'Amy'</p> <p>S4: update Worker set pay = pay-10 where name = 'Amy'</p> <p>Commit</p>
<p>2 (e). T1, T2 транзакшинууд Serializable тусгаарлалтын түвшинтэй бөгөөд T1 бүрэн ажиллаж дууссан. Харин T2 транзакшин S3-н дараа Rollback хийгдээд дахин ажиллаагүй байг. Amy –гийн pay хэд болох боломжтой вэ? Яагаад?</p>	
<p>Тайлбар: T1, T2 – ийн тусгаарлалтын түвшингээс үл хамааран сериалаар буюу T1;T2, T2;T1 гэж ажиллах боломжтой. Гэхдээ T1, T2 – ыг хоёулаа бүрэн ажиллаж дуусна гэж хэлээгүй учир заавал T1;T2, T2;T1 байдлаар бүрэн ажиллах албагүй юм. Бодлогын нөхцөл ёсоор T1 – ийг бүрэн ажиллаж дуусан гэдгээс Amy –гийн pay нь 300 болж баталгаажсан байх юм. Иймд T2 транзакшин S3-н дараа Rollback хийгдээд дахин ажиллаагүй гэдгээс Amy –гийн pay нь 280 болох бөгөөд энэ нь commit хийгдээгүй өгөгдөл rollback хийгдэхэд урьдын хэвэндээ буюу хамгийн сүүлийн баталгаажсан өгөгдөлд очих юм. Эцэст нь Amy –гийн pay нь 300 болж баталгаажсан хэвээрээ байна. Эндээс хариу нь Amy –гийн pay нь 300 гэж гарна.</p>	



Өгөгдлийн сангийн програмчлал – Семинар 12 /транзакшин 2/

[3]. Бүхэл тоон хүснэгт R(x) байв. R дах нийт тооны нийлбэр 1000, бөгөөд 10, 20, 30 гэсэн утгууд R байдгүй байг. Доорх 3 тран. зэрэг ажиллав. Бүгд READ COMMITTED тусгаарлалтын түвшинтэй байв. Амарийн тран-ы буцааж болох бүх утгуудыг ол.

Амарын Транзакшин:

```
select sum(x) from R;  
commit;
```

Билгүүний
Транзакшин:

```
insert into R values (10);  
insert into R values (20);  
insert into R values (30);  
commit;
```

Заяагийн Транзакшин:

```
delete from R where x=30;  
delete from R where x=20;  
commit;
```

Тайлбар: Эхний ээлжинд сериал байдлаар ажиллах үед Амарийн тран-ы буцах утгуудыг олъё. Сериалаар ажиллах үед ямар нэг бохир уншилт үүсэх боломжгүй юм. Үүнд:

Амар; Билгүүн; Заяа == Заяа; Амар; Билгүүн => 1000

Билгүүн; Амар; Заяа == Заяа; Билгүүн; амар => 1060

Билгүүн; Заяа; Амар == 1010

READ COMMITTED тусгаарлалтын түвшин нь бохир уншилтыг зөвшөөрдөггүй. Амарын тран нь зөвхөн уншилт хийж байгаа учраас Билгүүн болон Заяа тран-д бохир уншилт үүсгэх боломжгүй. Харин Билгүүн болон Заяагийн тран нь Амарын тран – д бохир уншилт үүсгэх магадлалтай юм. Мөн билгүүний тран нь зөвхөн бичилт хийнэ, харин Заяагийн тран нь зөвхөн устгал хийх юм. Delete нь эхлээд уншаад, дараа нь устгадаг. Тийм учраас Билгүүний тран Заяагийн тран – д мөн бохир уншилт үүсгэх магадлалтай. Амарын тран Билгүүний эсвэл Заяагийн тран-ы statement – үүдийн голоор орж ажиллах боломжгүй учир нь тэгх үед Амарын тран-д бохир уншилт үүснэ. Дараагийн нэг тохиолдол нь Заяагийн 2 delete – ийн голоор Амарын тран бүхлээрээ орж ажиллах юм. Энэ тохиолдолд Заяагийн тран нь бохир уншилт хийхгүй тул ингэж ажиллах боломжтой юм.

Заяа: Билгүүн; Амар => 1040 олдоно. Эндээс Амарын тран-ы буцаах болох утгууд: **1000, 1010, 1040, 1060** болно.

