بسم الله الرحمن الرحيم

طبيعي ان قبل ما اكلمك عن ال ACID وايه هم اصلا لازم نتكلم عن ال **transaction** :

- بيقولك ان ال transaction عباره عن وحده العمل اللي بتتعمل جوه ال DBMS ع ال DB وكمان بتغير ف ال DB بتاعتي بشكل عام .. ايه الكلام الغريب دا !

- ولا غريب ولا حاجه تعالي نشوف الكلام دا واحده واحده .. يعني ايه وحده عمل دي ؟ وحده عمل يعني معامله .. شوفت سهله ازاي .. معامله دي ممكن تكون واحده لوحدها سواء بتضيف او بتعدل او بتمسح حاجه من ال DB او شويه معاملات مع بعض كدا بيتنفذوا مع بعض .. كلهم شغالين مع بعض (كوحده واحده) شوفت سهله ازاي .

- مش بس كدا .. سواء هي عمليه واحده او اكتر ف لازم يجي شرط هنا .. وهو ان ال transaction دي ياما تتعمل كلها وdone والدنيا زي الفل لو مفيش errors خلاص احفظ كل حاجه بقي .. او لو فيه Error خلاص ارجع كما كنت ومش عايز اي تغيير يتم .

تعالي ناخد مثال بسيط كدا عشان نفهم اكتر :

والمثال دا هو الاشهر ف ال transaction وهو لمه نحب نبعت فلوس من account للتاني .. الكلام دا بيتم ع خطوات بسيطه جدا (طبعا الموضوع معقد اكتر من كدا لكن للتبسيط هم 3 بس) :

- عايزين نتاكد الاول هل ال account الاول عنده الملبلغ دا او اكتر (مثلا 100 دولار) لو الاجابه اه عنده هنروح للخطوه التانيه .
- وهي اننا عايزين نطرح ال 100 دولار دول من ال Account الاول .
- بعدها عايزين نضيف ال 100 دولار دول ف ال account التاني .

account_id	account_name	transaction
1	$ 1000	$ 100-
2	$ 200	$ 100+
1	$ 900	-
2	$ 300	-

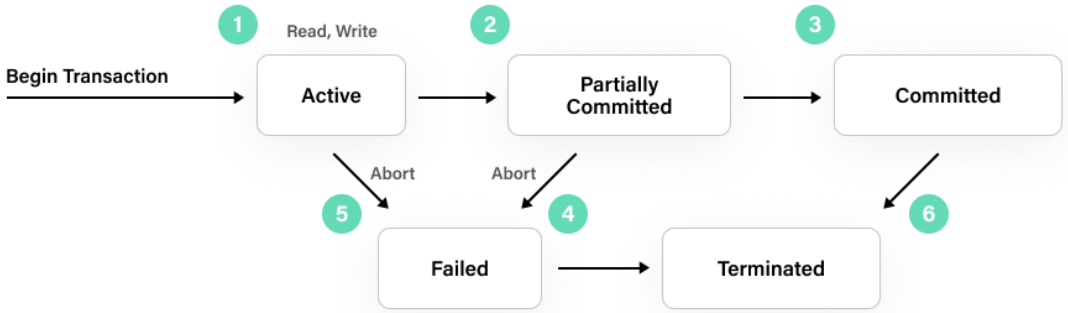
- طيب افرض ان حصل error بعد ما طرحت ال 100 دولار دول من ال account الاول ؟
  - كدا فيه 100 دولار اتبخرت من ال system بتاعك !
- طب افرض ان حصل عمليتين سحب ف نفس الوقت ع ال Account الاول واحده ب 1000 وواحده ب 100 ؟
  - كدا فيه عندك 100 دولار راحت عليك !

يبقي احنا بنقول ان ال transaction دي عباره عن لو فيه عمليه او اكتر بتتعمل ع ال DB عايزين نضمن ان :

- لو حصل اي error ف اي مرحله او حتي النور قطع عن ال DB او الانترنت او اي عامل ممكن ياثر ع ال transaction دي ان الوضع يبقي كما كان عليه قبل ال transaction بدون اي تغيير .
- او لو حصل اكتر من عمليه ع نفس ال value عايزين نعزل كل واحده لوحدها وبالترتيب عشان ميحصلش زي المشكله التانيه .

تعالي نشوف ال DB بتاعتنا وقت ال transaction دا بيكون شكلها عامل ازاي .. او بمعني اصح .. عايزين نشوف الحالات states اللي ممكن ال DB بتاعتنا تكون بتعملها وقت ال transaction :

- اول حاله وهي انها تكون active : وال state دي بتكون ف بدايه ال transaction ودي معناها ان فيه عمليه حاليا بتم ع ال DB ممكن read, update اي عمليه .
- الحاله التانيه وهي ال Partially committed : وف الحاله دي العمليه اللي كانت بتم ف الخطوه الاولي فعلا تمت بس لسه متعملهاش save .. هي اتعملت ف ال memory لكن مش ف ال disc .
- الحاله الثالثه وهي ال Committed : وهنا خلاص ال transaction اتعملها save ف ال DB بتاعتك ومفيش عوده فيها.
- Failed : وهنا حصل عندي مشكله ف ال active او ال partially committed states .
- اخر حاله وهي ال Terminated state : ودي اخر حاله بتكون عليها ال DB .



تعالي بقي نشوف ايه حوار ال ACID دا ؟

دول شويه خصائص كدا بنستعملهم عشان نتاكد ان ال transaction بتاعتنا هتم ان شاء الله بالشكل المطلوب منها .

وهم :

- (Atomic (Atomicity
- Consistency
- Isolation
- Durability

تعالي نتكلم عليهم واحده واحده ونشوف كل واحده معناها ايه وبتحل مشكله ايه ...

- **Atomicity :** ودا عباره عن مصطلح بيعبر عن (الكل او ولا حاجه) يعني ال DB لمه توصل لمرحله ال committed اما انها تحفظ العمليه دي خلاص ع اساس انها عمليه ناجحه او انها ترجع لحالتها الاولي قبل ال transaction دي .

مثال ع دا :
لو انا عندي فندق والمفروض اني احجز اوضه online .. عمليه الحجز دي مثلا هتتقسم ع عمليتين :
  - الاولي وهي اني احجز مكان الاوضه نفسها .
  - والتانيه اني ادفع ثمن الحجز دا .

ف لو حصل اني فعلا حجزت الاوضه فعلا خلاص لكن حصلت مشكله وقت الدفع ف انا خلاص عايز اشيل الحجز دا من ع الاوضه وتبقي الاوضه متاحه انها تتحجز من عميل تاني وهكذا .

- **Consistency** : ودي بتساعد ع تحقيق ال data integrity .. سواء ال transaction تمت او لا كمان لو فيه قيود زي ان ال column دا مينفعش نحط فيه قيم بالسالب او ال age دا مينفعش يبقي اكتر من 120 سنه !
مثال كمان ع دا :
لو ف بنك .. وحبيت اعمل ايداع ب 1234 دولار .. ف انا ك عميل انا عايز ف لحظتها اشوف فعلا ان المبلغ دا تم ايداعه ورصيدي الحالي بالمملي كام .
  - مينفعش استني 5د او حتي دقيقه ! ودا الـ consistency in read ومش بس الوقت .. انا عايز كل ما اعمل استعلام يقولي فعلا رصيدي الحقيقي (ودا ف حاله ان فيه معاملات كتيره بتم ع الحساب دا) .
  - ومينفعش يقولي تم ايداع 1230 بس ! ودا ال consistency in data .

مفهوم ال consistency بيهتم ان ال data بتاعتي تبقي بتمثل الواقع فعلا بدون اي تغيير او نقص ف ال Data دي وكمان سرعه تحديثها .

- **Isolation** : ودا بيحل المشكله التانيه اللي احنا قولناها لمه حاولنا نسحب مبلغ ميعن من نفس الحساب ف نفس الوقت .. وهنا احنا عايزين ان كل transaction يتم يكون معزول عن التاني او يتم بعد ما التاني يخلص .. مينفعش 2 يتموا ف نفس الوقت .
ال isolation له levels :

  - Read Uncommitted : ودا اقل نوع من انواع ال isolation وفيه ال user يقدر يقرا ال data اللي لسه متعلمهاش commit ولسه ف ال Active state .

  - Read Committed : ودا تقريبا اكتر حاجه مستخدمه واللي فيه احنا بنقرا ال data اللي اتعملها commit بس .. لكن اللي لسه ف ال active state ملناش دعوه بيها .

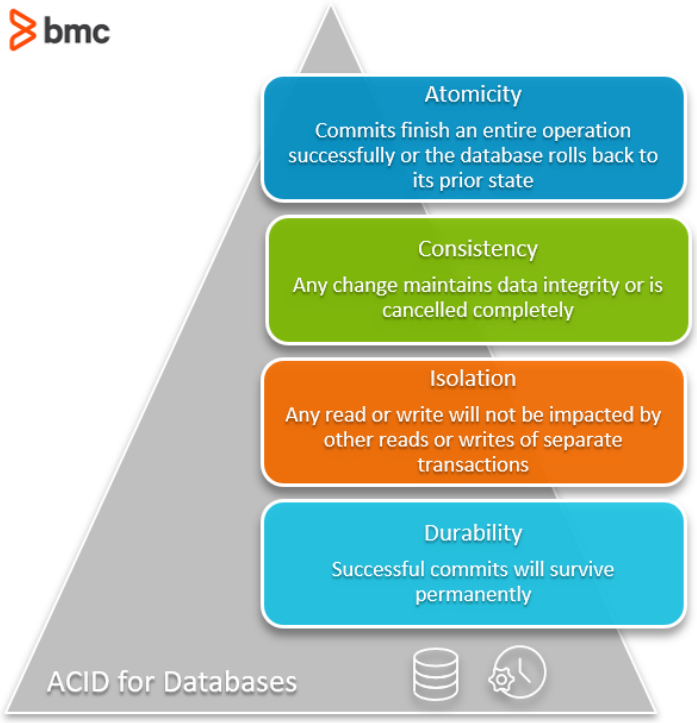
  - Repeatable Read : ودا بيحل مشكله موجوه عند ال read commit وهي اني مع كل استعلام جديد بيجيبلي ال commit فعلا ف ال DB بتاعتي .. يعني لو انا بعمل تقرير عن المبيعات وعايز اعمل استعلام عن المبيعات مرتين ورا بعض (ممكن يكون فيه فرق زمني بينهم وممكن يكونوا ورا بعض) لكن بين اول مره وتاني مره حصل ان فيه مبيعات تمت .. ف كدا انا عندي نسختين غير متطابقتين من المبيعات واللي ممكن يعملي مشاكل بعد كدا لو ال business بتاعي حساس (ولو مش حساس ف الموضوع دا مش هيفرق كتير)

  - Serializable : ودا اعلي حاجه ف ال isolation واللي بيحل مشكله بتظهر ف ال repeatable read وهي انها كانت بتمنع ان ال row يتعمله update لكنها مبتمنعش ان يكون فيه Row جديد اتعمل !

طيب انت هتختار انهي نوع فيهم ؟ والله كل حاجه وليها تمنها .. ال serializable اه احسن حاجه لكنه ابطئ .. ف انت بتقارن بين المميزات والعيوب وبتختار ال isolation المناسب لل business اللي انت شغال فيه .

Isolation Level	Dirty reads	Non-repeatable reads	Phantoms
Read Uncommitted	May occur	May occur	May occur
Read Committed	Don't occur	May occur	May occur
Repeatable Read	Don't occur	Don't occur	May occur
Serializable	Don't occur	Don't occur	Don't occur

- Durability : ودي معناها ان اي transaction اتعمله commit خلاص ف هو موجود ديما مش هينفع يرجع تاني .



ال ACID مش بس بيحقق مفهوم ال data integrity .. لا دا كمان بيزود ال security بتاعه ال DB بتاعتي .