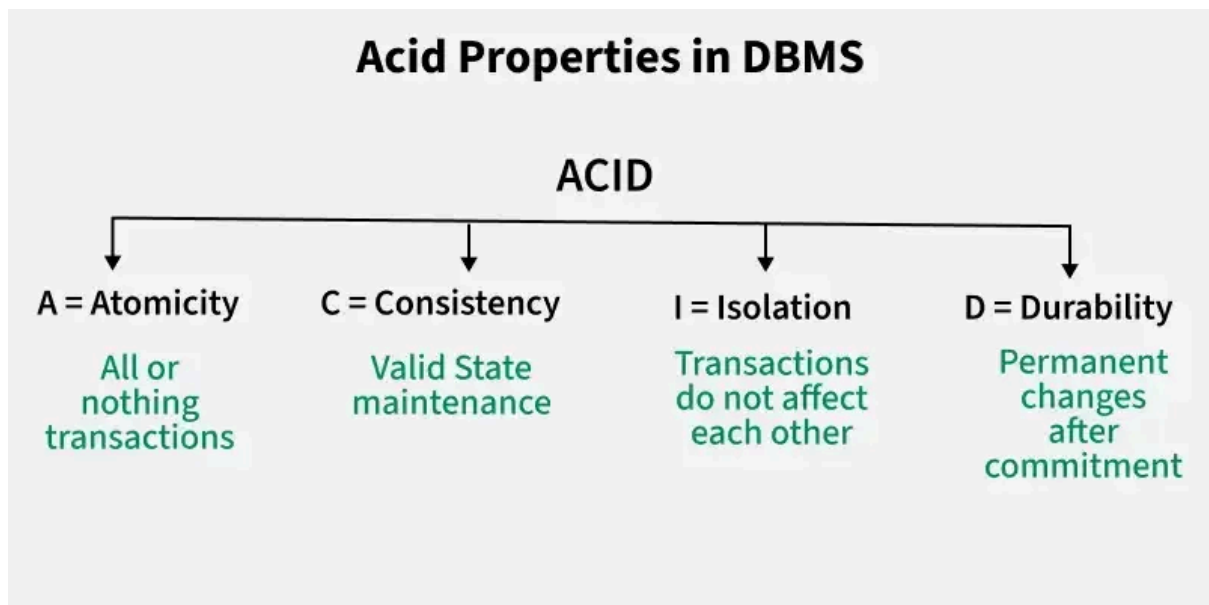


## تخيل كده...

بتحول فلوس من حسابك البنكي لحد تاني، وفجأة النت يقطع... هل ينفع الفلوس تختفي من عندك وما توصلش له  
أو إنك طلبت منتج من موقع أونلاين، والمخزن يبيع نفس القطعة لشخصين في نفس الوقت  
الحاجات دي بتحصل لو قواعد البيانات مش ماشية بقوانين صارمة. القوانين دي اسمها: **ACID Properties**

## إيه هي الـ ACID؟

هي مجموعة قواعد أساسية بتخلي المعاملات (Transactions) في قواعد البيانات آمنة وموثوقة حتى لو حصلت مشاكل أو أعطال



## 1. Atomicity

**Transaction** يا إما تحصل بالكامل يا إما ماتحصلش خالص.  
مثال: لو بتحول \$100 من حسابك لحساب تاني ماينفعش يتسحبوا منك وما يوصلوش للتاني. لو العملية وقفت في النص بيرجع كل شيء كأنه ما حصلش.

## 2. Consistency

ال data base لازم تفضل في حالة صحيحة قبل وبعد ال Transaction

### مثال بسيط:

في بنك عنده حسابين:

- الحساب الأول: \$500

- الحساب الثاني: \$200

- إجمالي الرصيد = \$700

لو عايز تحول \$100 من الحساب الأول للتاني:

بعد التحويل لازم يفضل الإجمالي = \$700 (يعني \$400 + \$300).

لو اتسحب المبلغ من حساب واحد وماوصلش للتاني يبقى كده فيه كسر لل **Consistency**.  
النظام هيعمل **rollback** علشان يرجع الوضع زي ما كان

## 3. Isolation (العزل)

**Isolation** يعني إن كل **transaction** بتنفذ كأنها لوحدها في النظام، من غير ما تتأثر أو تؤثر على أي **transaction** تانية شغالة في نفس الوقت.

### مثال:

عندنا حسابين:

X = 500

Y = 500

| X = 500 Rs                      Y = 500 Rs                                    |   |
|---|---|
| T   | T''   |
| Read (X)<br>X := X * 100<br>Write (X)<br>Read (Y)<br>Y := Y - 50<br>Write (Y) | Read (X)<br>Read (Y)<br>Z := X + Y<br>Write (Z) |

#### - Transaction T :

تسحب 50 من X يبقى  $X = 450$ .

تضيف 50 لـ Y يبقى  $Y = 550$ .

#### - في نفس الوقت، Transaction T' :

قرأت  $X = 500$  و  $Y = 500$  (قبل ما T يخلص).

حسبت المجموع:  $1000 = 500 + 500$ .

لكن بعد ما T يخلص:

$X = 450, Y = 550$

المجموع الصحيح  $= 1000$ .

**Isolation** يضمن إن "T" ما تقراش القيم الغلط (500 و 500) أثناء ما T لسه شغالة، وتستنى لحد ما T تعمل **commit**.

#### 4. Durability (الاستمرارية)

معناها إن بمجرد ما **transaction** تعمل **commit**، كل التغييرات اللي حصلت بتتبع وتُسجل بشكل دائم في ال **non-volatile storage** (زي ال **hard disk**).  
حتى لو النظام وقع فجأة (**crash** أو **power failure**)، البيانات اللي اتعملها **commit** ما تتفقدش.

مثال:

عندنا Transaction T بتنقل \$100 من Account A إلى Account B.

بعد ما T خلصت واتعمل **commit**:

$A = 400$

$B = 600$

لو السيرفر وقع بعدها بثانية، لما النظام يشتغل تاني، القيم هتفضل:

$A = 400$

$B = 600$

**Durability** بيبضمن إن العملية مش هترجع تاني كأنها ما حصلتش.

#### المصادر :

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