SQL

CONCEPTS

Mahfuzatul *Bushra*

ACID Properties in SQL

ENSURING RELIABLE TRANSACTIONS

Atomicity (A)

“All or nothing” – transaction fully succeeds or fails

Example

Transfer $100 from Account A → B. If one step fails, nothing changes

Consistency (C)

Database remains valid after transaction

Example

Total balance before = total balance after
[no money was created or lost during the transaction]

Isolation (I)

Transactions do not interfere

Example

Two users updating same table → no conflict

Durability (D)

Committed data cannot be lost

Example

After bank transfer, data remains even if system crashes

Key Interview Questions

What does ACID stand for?

Answer

Atomicity, Consistency, Isolation, Durability

Why are ACID properties important?

Answer

Ensure data integrity, reliability, and safe transactions

Example of Atomicity?

Answer

Bank transfer: either all steps succeed or none

Difference: Isolation vs Consistency?

Answer

- Consistency = database remains valid
- Isolation = transactions don’t affect each other

What if Durability fails?

Answer

Committed data may be lost during a crash

Which SQL commands are affected by ACID?

Answer

INSERT, UPDATE, DELETE, and transactions

Can ACID be achieved in NoSQL databases?

Answer

Some NoSQL databases (like MongoDB) support ACID transactions at the document or collection level, but not always across multiple collections.

Simple transaction example for context

```
BEGIN TRANSACTION;
UPDATE Account SET balance = balance - 100 WHERE id = 1;
UPDATE Account SET balance = balance + 100 WHERE id = 2;
COMMIT;
```

- **Label keywords: BEGIN TRANSACTION, COMMIT → relates to ACID properties.**

Thank

YOU