IDE

ACID

ACID

- A: Atomicity
  - Transaction

- Bkash> -100 [500-100] =400
- Sim < +100 [0]
- Bkash> -100 [400+100] =500

- Either Full Trx
- Or No trx (Rollback)

ACID

- C: Consistency

- Bkash (X) = 500
- Sim (Y) = 1.20
- =========500+1.20 = 501.20

- Recharge (TRX)
- Bkash (X) = 500-100 =400
- Sim (Y) = 1.20 = 1.20+100= 101.20
- ======================400+101.20= 501.20

ACID

- I: Isolation

- Cover (Private)
- **U101 [2023-09-15 15:57]**
- Recharge (TRX1) ----**FAILED**
- Bkash (X) = 500-100 =400
- Sim (Y) = 1.20 = 1.20+100= 101.20

- **U102 [2023-09-15 15:57]**
- Recharge (TRX2)
- Bkash (A) = 1000000-500
- Sim (B) = 100.50 = 100.50+500

ACID

- D: Durability

- **U102 [2023-09-15 15:57]**
- Recharge (TRX2)
- Bkash (A) = 1000000-500
- Sim (B) = 100.50 = 100.50+500
- **SUCCESS**

- **Storm / Flood/ Fire / PC problem / Network Problem/ SYSTEM FAILURE**
- **RECOVER> GET ALL SUCCESS /COMMITED DATA**

Database Design

DB Architecture

- Normalization
- Denormalization

- Normalization: Transactional DB *(Row Oriented)*
  - Insert**, update, Delete**

Normalization

- Make database simple to keep data
- Make it optimize for insert/update
- Make it splitted as possible

- **Benifits**
- → Managing data easier / simpler/ Optimized
- → Data storage less used
- → Less redundant data

- **Drawbacks**
- Analytics is not recommended
- Projection/Selection less optimized

Denormalization

- Warehouse / Column oriented DB
- → From a normalize db
- → optimized for selection/projection
- → Used in analytics

- **Benefits:**
- Faster retrieval
- Optimized for select query
- Less join to use
- **Drawbacks:**
- Redundant data
- Stoarge
- Update/ delete unoptimized (not remmonded)

Q&A