explain again easily according to
credit card transaction to maintain
ACID by dynamo db

In simple terms: DynamoDB transactions make sure that when you do something like a **credit card payment**, all the steps succeed together or all fail together. This is how DynamoDB maintains **ACID guarantees** (Atomicity, Consistency, Isolation, Durability) even though it's a NoSQL database.

💳 Credit Card Example with DynamoDB Transactions

Imagine you're paying for something with your credit card. Behind the scenes, several things must happen:

1. **Check balance** in your account.

2. **Deduct money** from your account.

3. **Add money** to the merchant's account.

4. **Record the transaction** in a ledger table.

If any one of these steps fails (say, the merchant's account update fails), you don't want the money deducted from your account without being credited to the merchant.

🔑 How DynamoDB Ensures ACID Here

- **Atomicity:**
  All four operations (check, deduct, add, record) are grouped in a single DynamoDB transaction. Either *all succeed* or *none happen*.

- **Consistency:**
  DynamoDB enforces rules so that balances never go negative and the ledger always matches the accounts. After the transaction, the system is in a valid state.

- **Isolation:**
  While your payment is being processed, no other transaction can "see" a half-finished state. Other users won't see your balance deducted until the whole transaction is complete.

- **Durability:**
  Once DynamoDB confirms the transaction, it's safely written to storage and replicated across multiple servers. Even if a server crashes, the transaction is not lost.

⚡ DynamoDB APIs for This

- `TransactWriteItems`: Used to perform multiple writes (deduct from your account, add to merchant, record transaction).

- `TransactGetItems`: Used if you need to read multiple items atomically (e.g., check balances before approving).

📌 Takeaway

DynamoDB transactions let you treat **multiple operations across multiple tables as one reliable unit**, just like a bank or credit card system must do. This is how DynamoDB achieves **ACID compliance** in a distributed NoSQL environment.