

Database Transactions Notes

1 Database Transaction কি?

- Database Transaction হলো একটি **Atomic কাজের ইউনিট**, যেখানে একাধিক SQL operation (INSERT, UPDATE, DELETE, SELECT) একসাথে করা হয়।
- Atomic মানে: সব কাজ সফল হলে commit হবে, না হলে সব বাতিল (rollback) হবে।

সরাসরি সংজ্ঞা:

Transaction হলো একসাথে অনেক কাজ করার একটি পদ্ধতি যেখানে সব কাজ একসাথে সম্পন্ন হবে বা একসাথে বাতিল হবে।

2 ACID Properties

1. **Atomicity (পরিপূর্ণতা)**
 2. সব কাজ একসাথে হবে বা একসাথে বাতিল হবে।
 3. উদাহরণ: ব্যাংক transfer – টাকা পাঠানো এবং গ্রহণ করা একসাথে সফল হতে হবে।
4. **Consistency (সঙ্গতি)**
 5. Transaction এর আগে এবং পরে ডাটাবেস সবসময় consistent থাকবে।
6. **Isolation (নিরোধক)**
 7. এক transaction অন্য transaction থেকে আলাদা।
 8. একাধিক transaction একই সময়ে হলেও তারা একে অপরকে প্রভাবিত করবে না।
9. **Durability (টেকসইতা)**
 10. একবার commit হলে, ডাটা স্থায়ীভাবে থাকবে।

3 উদাহরণ (SQL)

```
START TRANSACTION;
```

```
UPDATE accounts SET balance = balance - 500 WHERE id = 1;
```

```
UPDATE accounts SET balance = balance + 500 WHERE id = 2;
```

```
COMMIT;
```

- দুইটি update একসাথে হবে। - যদি দ্বিতীয়টি fail হয়, সব কাজ rollback হবে।

4 Django তে Transactions

```
from django.db import transaction

try:
    with transaction.atomic():
        user1.balance -= 500
        user1.save()

        user2.balance += 500
        user2.save()
except:
    print("Transaction failed, everything rolled back")
```

- `transaction.atomic()` ব্যবহার করলে সব কাজ একসাথে commit হবে। - Exception হলে সব কাজ rollback হবে।

5 Key Tip

- Django তে প্রতিটি request auto-commit mode এ থাকে।
- বড় multi-step operations বা critical updates এর জন্য `transaction.atomic()` ব্যবহার করা উত্তম।

Optional Advanced: - Savepoints তৈরি করা যায় বড় transaction এ, যাতে partial rollback করা যায়। - Nested transactions ও handled করা যায় `atomic()` এর ভিতরে।