Recovery

During Normal Processing

State of Transaction

Active
- begin
- Read, Write
- Ask commit
- abort

Partially Committed
- Grant commit
- abort Dirty read

Committed

Failed

Done

Own choice to quit, dies because of deadlock policy or serialization policy
Deferred Update Abort

- Deferred Update Policy:
  - Data changed by transaction not available to others until commit
  - Strict 2PL (hold locks till end of transaction)
  - Data versioning (other transactions read earlier data)
- No UNDO needed
  - Changed data never available.

Immediate Update Abort

- Immediate Update Policy
  - Data changed by transaction available to others before commit
  - Ordinary 2PL
  - Optimistic protocols
- UNDO needed to restore original data
- Dirty reads may abort other transactions: Cascading Rollback
Immediate Update Example

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
</tr>
<tr>
<td>30</td>
<td>15</td>
<td>40</td>
<td>20</td>
</tr>
</tbody>
</table>

**start**, T3
**read**, T3, C
**write**, T3, B, 15, 12 12 **Available before commit**

**start**, T2
**read**, T2, B **Dirty Read**
**write**, T2, B, 12, 18 18

**start**, T1
**read**, T1, A
**read**, T1, D
**write**, T1, D, 20, 25 25

T1 asks and gets commit here--no one else wrote A or D
**read**, T2, D
**write**, T2, D, 25, 26 26

T2 asks but must wait because of dirty read
**read**, T3, A
**write**, T3, A, 30, 35 35

If T3 commits, dirty read is ok & T2 can then commit

---

Cascading Rollback

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
</tr>
<tr>
<td>30</td>
<td>15</td>
<td>40</td>
<td>20</td>
</tr>
</tbody>
</table>

**start**, T3
**read**, T3, C
**write**, T3, B, 15, 12 12 **UNDO**

**start**, T2
**read**, T2, B
**write**, T2, B, 12, 18 18 **UNDO**

**start**, T1
**read**, T1, A
**read**, T1, D
**write**, T1, D, 20, 25 25 **UNDO**

T1 asks and gets commit here
**read**, T2, D
**write**, T2, D, 25, 26 26 **UNDO**

T2 asks but must wait because of dirty read
**read**, T3, A
**write**, T3, A, 30, 35 35 **UNDO**

T3 ABORTS! **T2 ABORTS -- dirty read**
Summary

- Rollback needed in case of abort with immediate updates
- Cascading rollback needed if dirty read happened
  - Depends on update policy of the database.
- Log used for UNDO only
- No REDO for normal processing