Recovery Management

1. **Device characteristics and failure types**
2. **Recovery tools**
   - *transaction log, checkpointing*
3. **Recovery processes**

Storage Device Basics

- แบบไม่ถาวร: RAM
  - ข้อมูลสูญหายเมื่อมี failures
- แบบถาวร:
  - Disk, tape, optical media
  - ข้อมูลคงสภาพเมื่อปิดเครื่อง
  - แต่ก็เชื่อถือไม่ได้ 100%

วิธีรักษาสภาพข้อมูล

- Redundant levels:
  - เก็บข้อมูลไว้หลายชุดในหลายดิสก์
  - Read/Write in parallel
- Multiple levels:
  - ใช้สื่อที่ราคาถูกในการจัดเก็บข้อมูลสำรอง

Failure Types

- Local
  - Program detected and abnormal termination(% by zero)
  - Affects only a single transaction
- Operating System
  - Affects all active transactions
  - Less common than local failures
- Device *(disk failure)*
  - Affects all active and past transactions
  - Least common

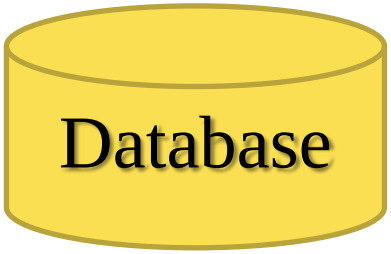
Transaction Log

- ตารางเก็บประวัติการแก้ไข/เปลี่ยนแปลงข้อมูลในฐาน
- ใช้พื้นที่เยอะในการเก็บ **log records**
- มีไว้เพื่อฟื้นสภาพฐานข้อมูล กรณีมีข้อขัดข้อง
  - **Undo T:** ย้อนกลับไปสถานะก่อนหน้าที่จะรัน T
  - **Redo T:** กระทำกับฐานข้อมูลอีกครั้ง ตาม **log** ของ T

Transaction Log Example

Log Sequence Number (LSN): unique id for each log row

LSN	TransNo	Action	Time	Table	Row	Column	Old	New
1	101001	START	10:29					
2	101001	UPDATE	10:30	Acct	10001	AcctBal	100	200
3	101001	UPDATE	10:30	Acct	15147	AcctBal	500	400
4	101001	INSERT	10:32	Hist	25045	* *all columns*		<1002, 500, …>
5	101001	COMMIT	10:33					



ถ้าเป็นการ Select data ไม่เก็บใน log นี้

Transaction Log Example

LSN	TransNo	Action	Time	Table	Row	Column	Old	New
1	101001	START	10:29					
2	101001	UPDATE	10:30	Acct	10001	AcctBal	100	200
3	101001	UPDATE	10:30	Acct	15147	AcctBal	500	400
4	101001	INSERT	10:32	Hist	25045	* *all columns*		<1002, 500, …>
5	101001	COMMIT	10:33					

Redo T: กระทำกับฐานข้อมูลอีกครั้ง ตาม log records ของ T จะไม่ได้ไปรัน T ใหม่ทุกคำสั่ง

Checkpointing

- ปกติระบบจะ write buffer to disk ในเวลาที่เหมาะสม
- แต่เราสามารถตั้งเวลาให้ write to disk ได้ (Checkpointing)
  - *ปกติ 5 to 10 minutes in high volume environments*
- เมื่อถึงเวลา checkpoint สิ่งที่ DBMS กระทำคือ
  - Stop all transaction processing
  - บันทึก Checkpoint record ลงในทุกๆ working transaction
  - Write log buffers and database buffers (to disk)

Other Recovery Tools

- Force writing to disk when
  1. Checkpoint time
  2. End of transaction
- Database backup 
  - เป็นการสำรองทั้งฐานข้อมูล
  - สะสมมากขึ้นเรื่อยๆ

Recovery from a Media Failure

ถ้าระบบขัดข้องจาก media เสียจะกระทบทุก T ที่ทำงานในขณะนั้น

- *Restore database from the most recent backup*
- **Redo** ทำซ้ำทุก transactions ที่เสร็จสิ้นไปแล้ว ตามที่บันทึกไว้ใน log ตั้งแต่ที่ backup ครั้งสุดท้าย
- **Restart** สำหรับ transactions ที่ยังทำงานไม่เสร็จ
  - แต่ต้อง Rollback เพื่อย้อนผลกระทบที่ทำค้างคาอยู่
  - จึงไปตั้งต้น restart คือรัน T ใหม่

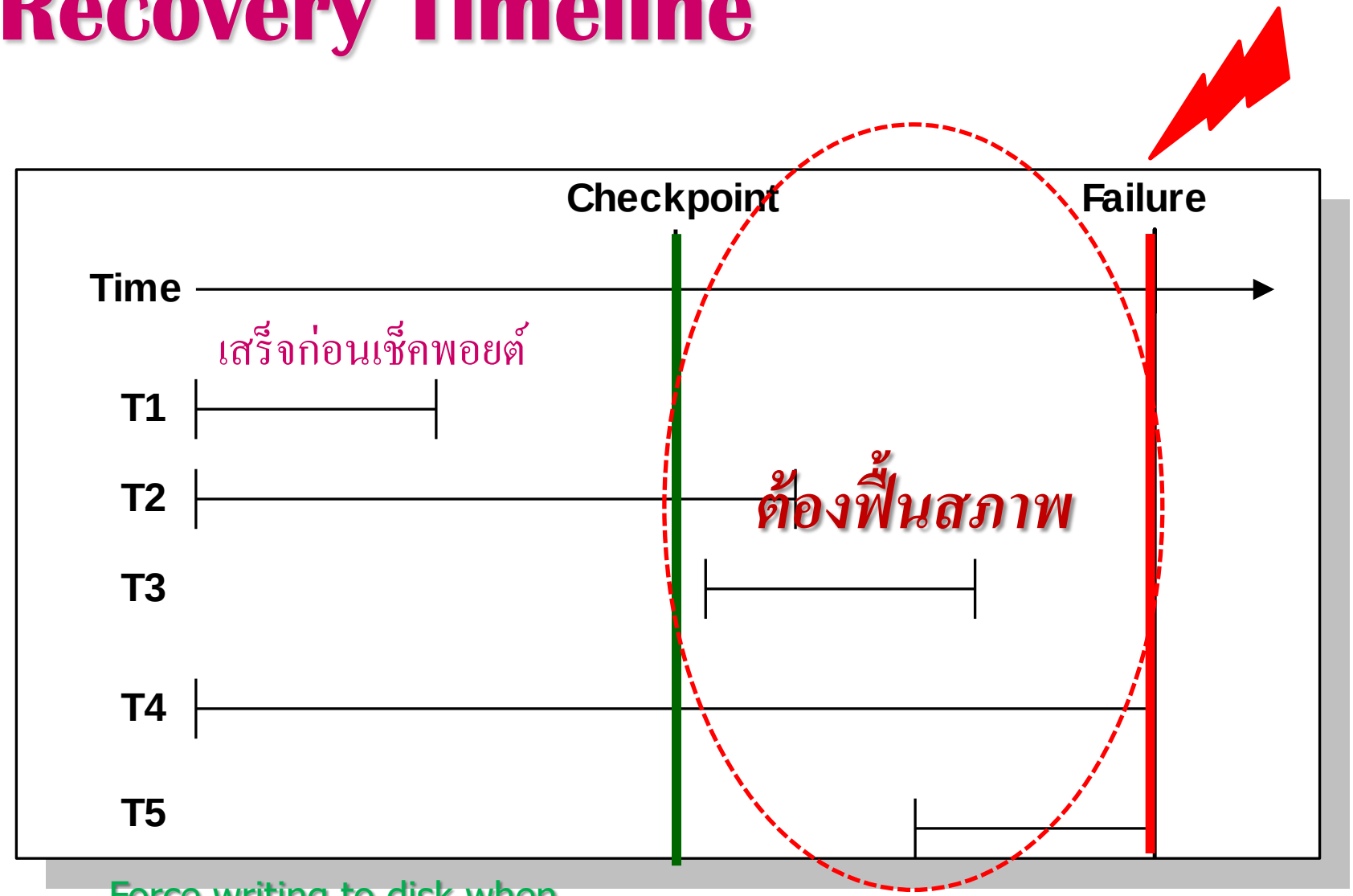
Assumes not log failure: media failure affects the database only, not the log

Recovery from a Media Failure

- Restore database from the most recent backup
- **Redo** all committed transactions since the most recent backup
- **Restart** active transactions
  - Rollback and then restart

•*Assumes not log failure: media failure affects the database only, not the log.*

Recovery Timeline



Force writing to disk when

1. Checkpoint time
2. End of transaction

Recovery Processes

การ recovery ขึ้นกับ timing of database writes:

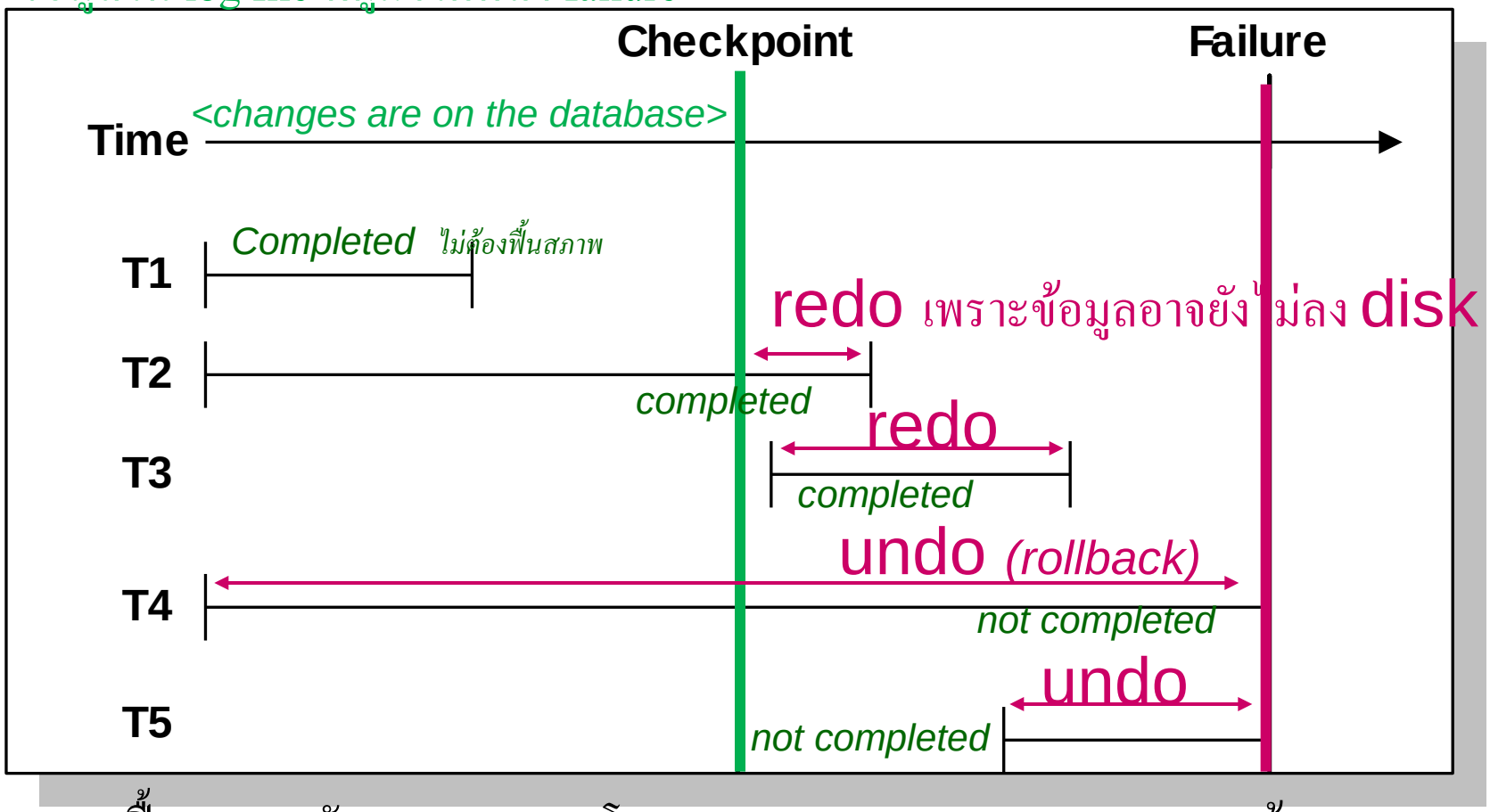
- Immediate update approach:
  - คำสั่ง update database ในทรานแซคชั่น T จะมีผลทันทีต่อฐานข้อมูล แม้ยังไม่จบ T
  - Need both Redo & undo
- Deferred update approach
  - คำสั่ง update database ในทรานแซคชั่น T จะมีผลต่อฐานข้อมูล ก็ต่อเมื่อ T committed แล้ว
    - ดังนั้น no database writes of uncommitted transactions at checkpoint
  - Undo operations not needed

Recovery process

- **Redo process**
  - read log file and redo operations recorded in log file
- **Undo process**
  - send failure message to the computer that running the transaction
  - **rollback**

Immediate Update Recovery

ข้อมูลใน log file ที่กู้มาได้หลัง failure



ระบบจะฟื้นสภาพหลังจุด checkpoint โดยจะ redo T2&T3 เพราะ committed แล้ว.

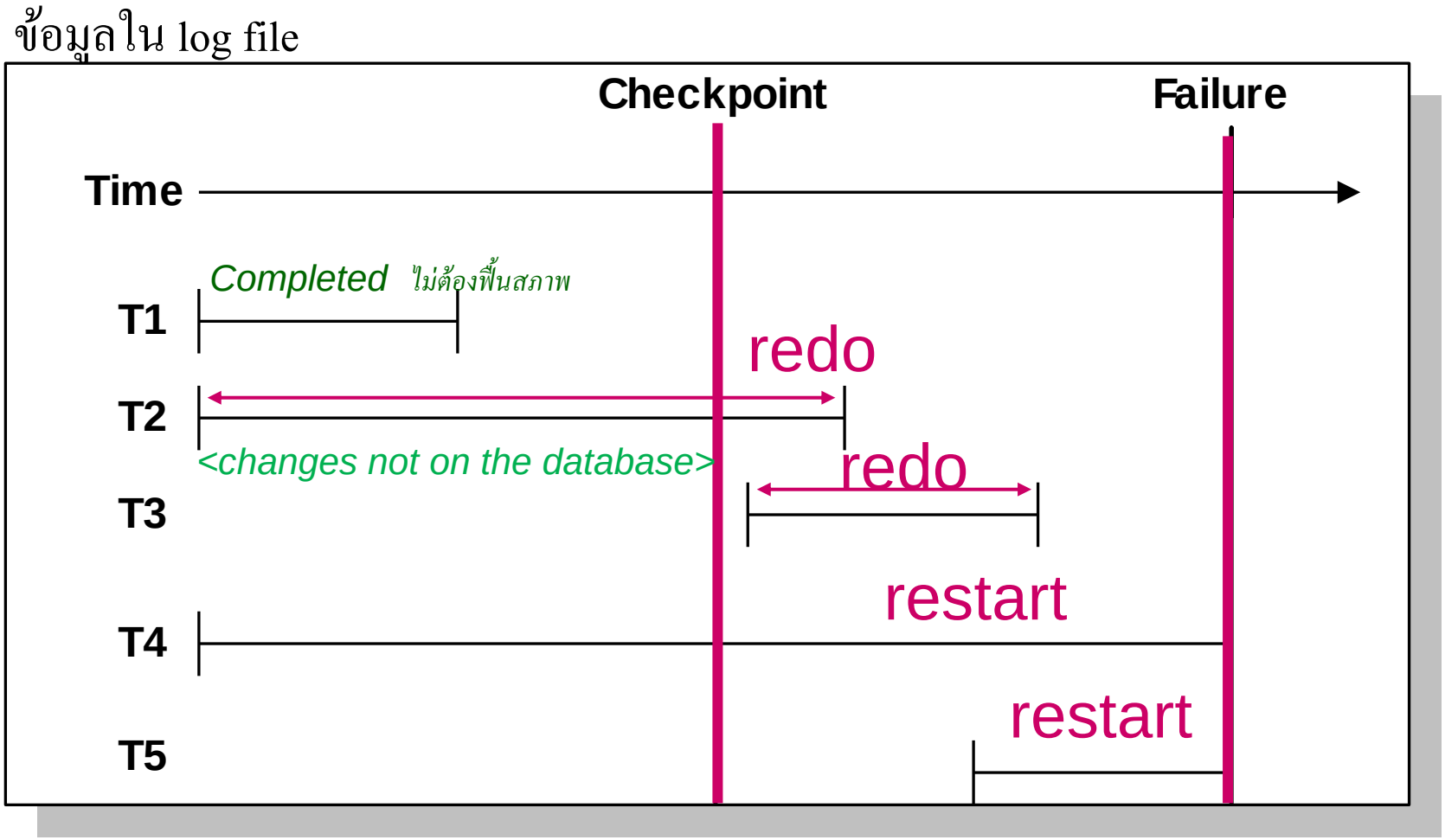
Undo T4 & T5 เพราะยังรันไม่เสร็จสิ้นเมื่อระบบ fail, ต้องเริ่มต้นใหม่

การ undo หรือ rollback อาจมีผลกระทบเช่น dirty read จึงอาจเกิด cascading rollback ได้

Immediate Update Recovery

Class	Description	Restart Work	
T1	Finished before CP	None	
T2	Started before CP; finished before failure	Redo forward from checkpoint	redo
T3	Started after CP; finished before failure	Redo forward from checkpoint	redo
T4	Started before CP; not yet finished	Undo backwards from most recent log record	undo
T5	Started after CP; not yet finished	Undo backwards from most recent log record	undo

Deferred Update Recovery



- Restart T4 ,T5 (ไม่ต้องundo) เพราะยังไม่มีการ update ข้อมูลลงในฐานข้อมูล
- Redo T2 ตั้งแต่ต้นเพราะ*ก่อน checkpoint* ผลของคำสั่งยังไม่ได้บันทึกลง database ถูกบันทึกลง log เท่านั้น

Deferred Update Recovery

Class	Description	Restart Work
T1	Finished before CP	None
T2	Started before CP; finished before failure	Redo forward from first log record
T3	Started after CP; finished before failure	Redo forward from first log record
T4	Started before CP; not yet finished	None
T5	Started after CP; not yet finished	None

redo

redo

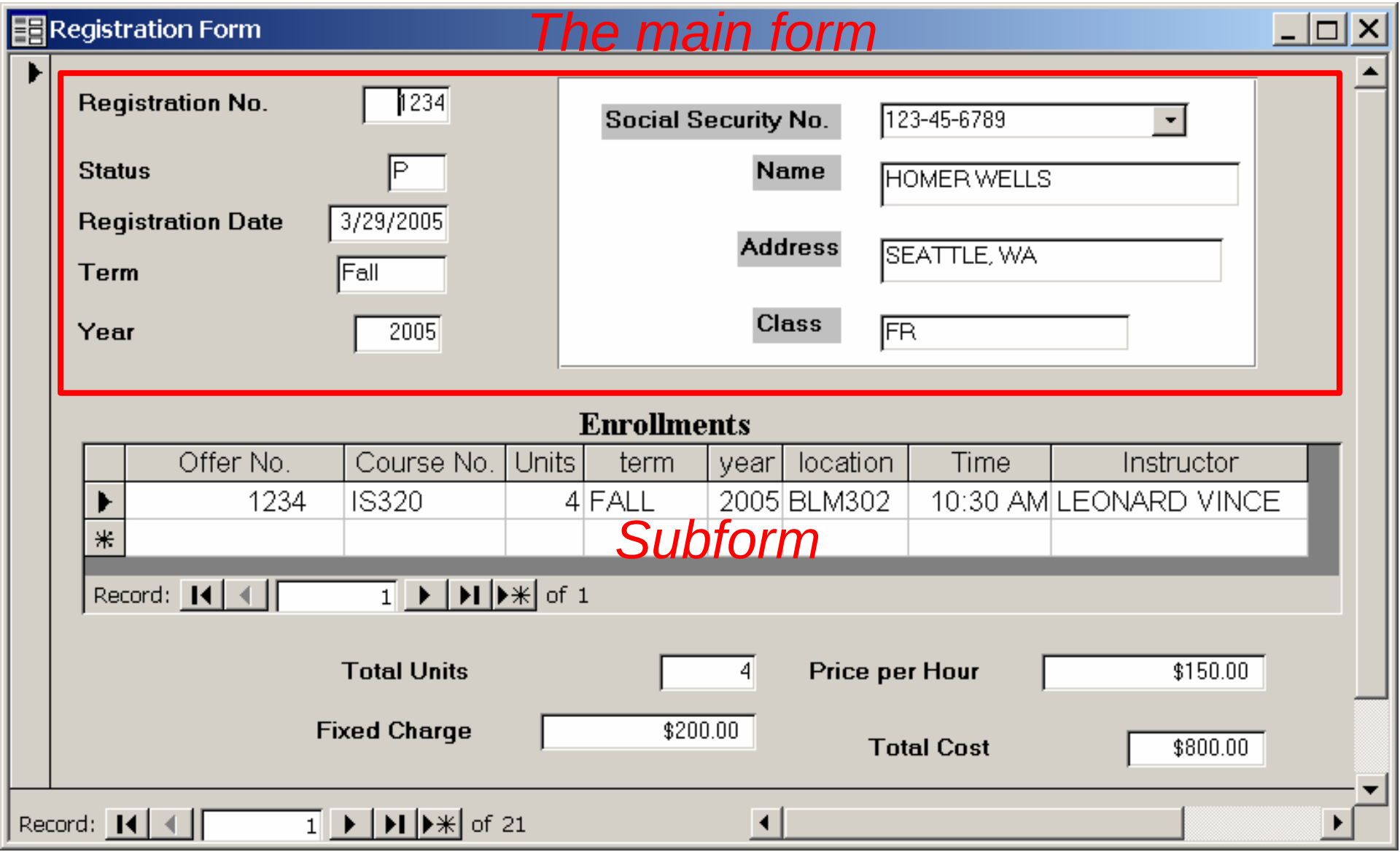
Transaction Design Issues

- Transaction boundary
- Isolation levels
- Deferred constraint checking
- Savepoints

Transaction Boundary Decisions

- Division of work into transactions
- Objective: minimize transaction duration
- Constraint: enforcement of important integrity constraints
- Transaction boundary decision can affect hot spots

Registration Form Example



Transaction Boundary Choices

- One transaction for
  - the entire form
  - the main form and
    - one transaction for all subform records
  - the main form and
    - separate transactions for each subform record

Avoiding User Interaction Time

- Avoid to increase throughput
- Possible side effects: user confusion due to database changes
- Balance increase in throughput with occurrences of side effects
- Most situations increase in throughput more important than possible user confusion

Isolation Levels

- Degree to which a transaction is separated from the actions of other transactions
- Balance concurrency control overhead with interference problems
- Some transactions can tolerate uncommitted dependency and inconsistent retrieval problems
- Specify using the SET TRANSACTION statement

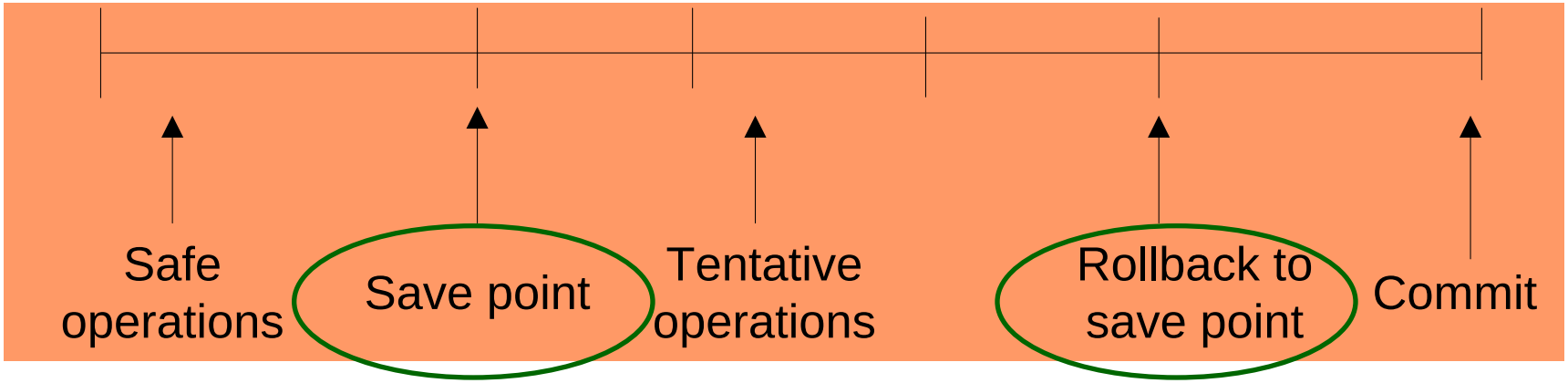
Integrity Constraint Timing

- Most constraints checked immediately
- Can defer constraint checking to EOT
- SQL
  - Constraint timing clause for constraints in a CREATE TABLE statement
  - SET CONSTRAINTS statement

Example:
- Primary key
- Data type
- Foreign key

Save Points

- Some transactions have tentative actions
- SAVEPOINT statement determines intermediate points
- ROLLBACK to specified save points



Summary

- Transaction: user-defined collection of work
- DBMSs support ACID properties
- Knowledge of concurrency control and recovery important for managing databases
- Transaction design issues are important
- Transaction processing is an important part of workflow management