Database Backup and Recovery

Watch video: https://youtu.be/3mZQFDBZ-d4?t=606

Agenda

- Backup and recovery
- Types of backups
- Types of database failure
- Database recovery
- Recovery facilities
  - Log files
  - Checkpoints
- Recovery techniques

Toy story 2 vanishes:

- https://www.youtube.com/watch?v=8dhp_20j0Ys

What can we learn from this?

- Backup, backup, backup…
- … and RESTORE!

Agenda

- Backup and recovery
- Types of backups
- Types of database failure
- Database recovery
- Recovery facilities
  - Log files
  - Checkpoints
- Recovery techniques

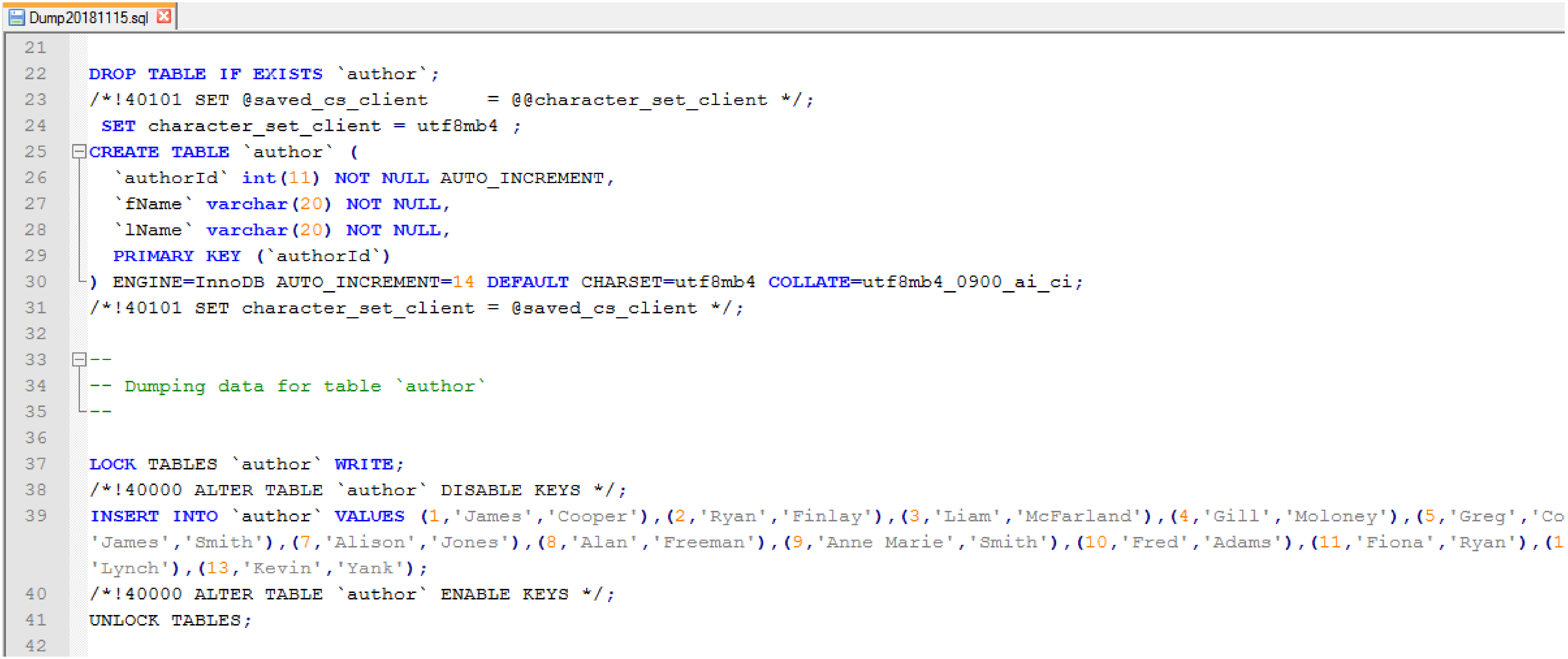
Types of backups

- Logical vs. Physical
- Offline vs. Online
- Full vs. Partial

Logical backups

- A logical backup of the database involves reading a set of database records and writing them to file. These records are read independently of their physical location.
- In MySQL this can be done using the mysqldump function.
- The file written from this function will have the commands necessary to re-create all of the database objects (e.g. tables, indexes) and the data contained in the database.
- The data that has been exported does not have to be imported into the same database, or the same schema. You can use the export file to create a duplicate set of objects in a separate database.

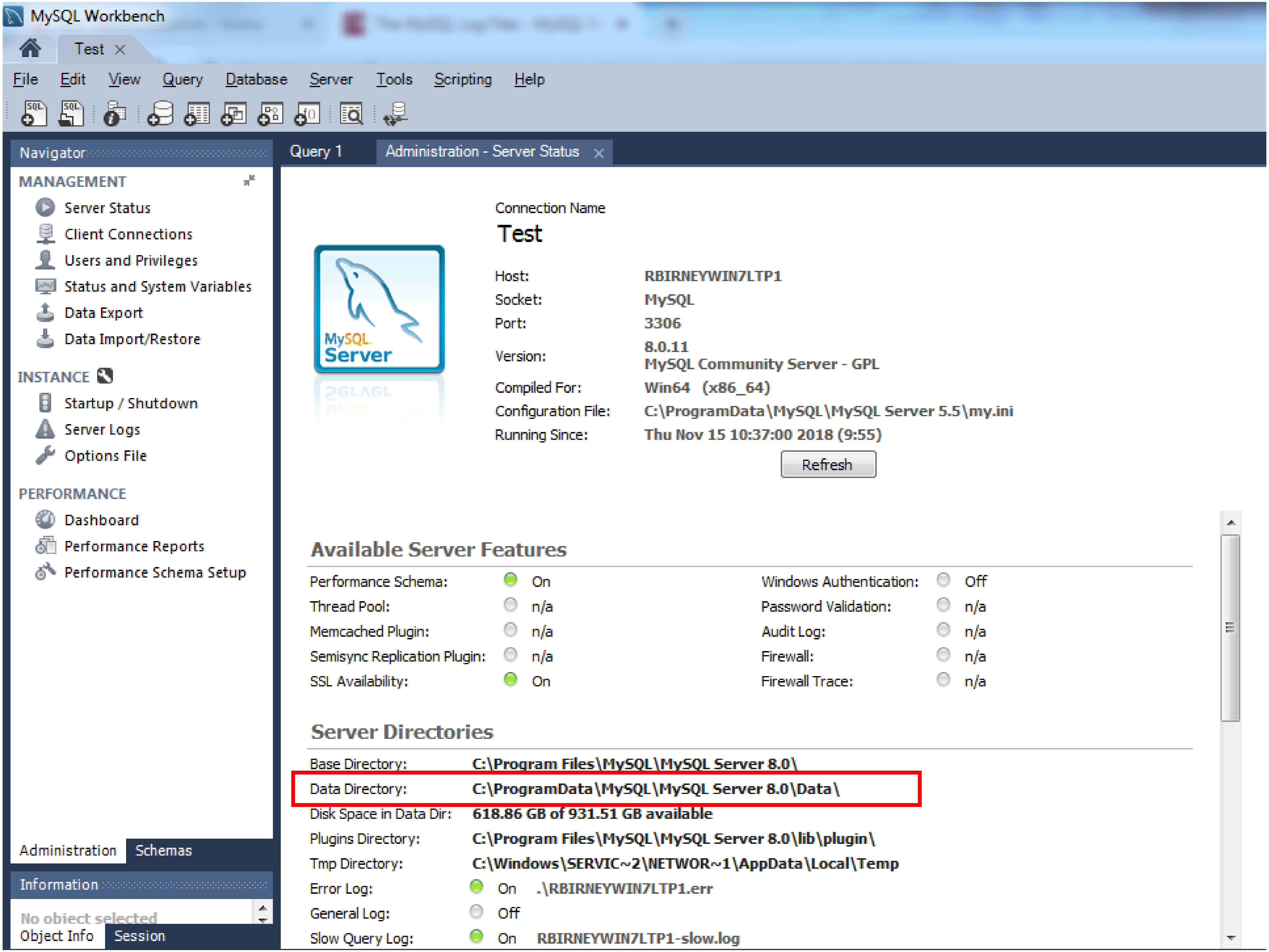
Logical backups



```

DROP TABLE IF EXISTS `author`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
 SET character_set_client = utf8mb4 ;
CREATE TABLE `author` (
  `authorId` int(11) NOT NULL AUTO_INCREMENT,
  `fName` varchar(20) NOT NULL,
  `lName` varchar(20) NOT NULL,
  PRIMARY KEY (`authorId`)
) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `author`
--

LOCK TABLES `author` WRITE;
/*!40000 ALTER TABLE `author` DISABLE KEYS */;
INSERT INTO `author` VALUES (1,'James','Cooper'),(2,'Ryan','Finlay'),(3,'Liam','McFarland'),(4,'Gill','Moloney'),(5,'Greg','Co
'James','Smith'),(7,'Alison','Jones'),(8,'Alan','Freeman'),(9,'Anne Marie','Smith'),(10,'Fred','Adams'),(11,'Fiona','Ryan'),(1
'Lynch'),(13,'Kevin','Yank');
/*!40000 ALTER TABLE `author` ENABLE KEYS */;
UNLOCK TABLES;

```

Physical backups

- Physical backups involve copying the files that constitute the database. These backups are also referred to as file system backups since they involve using operating system file backup commands.



Connection Name

Test

Host:	RBIRNEYWIN7LTP1
Socket:	MySQL
Port:	3306
Version:	8.0.11 MySQL Community Server - GPL
Compiled For:	Win64 (x86_64)
Configuration File:	C:\ProgramData\MySQL\MySQL Server 5.5\my.ini
Running Since:	Thu Nov 15 10:37:00 2018 (9:55)

Available Server Features

Performance Schema:	● On	Windows Authentication:	○ Off
Thread Pool:	○ n/a	Password Validation:	○ n/a
Memcached Plugin:	○ n/a	Audit Log:	○ n/a
Semisync Replication Plugin:	○ n/a	Firewall:	○ n/a
SSL Availability:	● On	Firewall Trace:	○ n/a

Server Directories

Base Directory:	C:\Program Files\MySQL\MySQL Server 8.0\
Data Directory:	C:\ProgramData\MySQL\MySQL Server 8.0\Data\
Disk Space in Data Dir:	618.86 GB of 931.51 GB available
Plugins Directory:	C:\Program Files\MySQL\MySQL Server 8.0\lib\plugin\
Tmp Directory:	C:\Windows\SERVIC~2\NETWOR~1\AppData\Local\Temp
Error Log:	● On .\RBIRNEYWIN7LTP1.err
General Log:	○ Off
Slow Query Log:	● On RBIRNEYWIN7LTP1-slow.log

Offline backups (aka “cold” or “consistent” backups)

- Consistent offline backups occur when the database has been shut down normally. While the database is offline, the following files should be backed up:
  - All datafiles, all control files, and all online redo log files
- Having all these files backed up while the database is closed provides a complete image of the database as it existed at the time it was closed. The full set of these files could be retrieved from the backups at a later date and the database would be able to function.

Online backup (aka “hot” or “inconsistent” backups)

- Online backups do not require the database to be shut down

- Increases availability of the database

- However, inconsistencies may occur as areas of the database could be in use while being backed up; it is important to lock tables as they are backed up to ensure that conflicts do not occur

Which backup method is appropriate to use as the primary backup?

Method	**Type**	**Recovery Characteristics**
Export (Dump)	Logical	Can recover any database object to its status as of the moment it was exported.
Offline backups	Physical	Can recover the database to its status as of the moment it was shut down; if archived redo logs are maintained, you can recover the database to its status at any point in time.
Online backups	Physical	Can recover the database to its status at any point in time.

Full vs. Partial Backups

- A full backup backs up the whole database; this is also known as a Level 0 backup

- A partial or incremental backup only backs up the data blocks that have changed since the last full backup. A partial backup is also known as a Level 1 backup

Agenda

- Backup and recovery
- Types of backups
- Types of database failure
- Database recovery
- Recovery facilities
  - Log files
  - Checkpoints
- Recovery techniques

Types of database failure

- System crashes, resulting in loss of main memory.
- Media failures, resulting in loss of parts of secondary storage.
- Application software errors.
- Natural physical disasters.
- Carelessness or unintentional destruction of data or facilities.
- Sabotage.

Agenda

- Backup and recovery
- Types of backups
- Types of database failure
- Database recovery
- Recovery facilities
  - Log files
  - Checkpoints
- Recovery techniques

Database Recovery

- Process of restoring database to a correct state in the event of a failure.

- Need for Recovery Control

    - Two types of storage: volatile (main memory) and nonvolatile.

    - Volatile storage does not survive system crashes.

    - Stable storage represents information that has been replicated in several nonvolatile storage media with independent failure modes.

Transactions and Recovery

- Transactions represent basic unit of recovery.
- Recovery manager responsible for atomicity and durability.
- If failure occurs between commit and database buffers being flushed to secondary storage then, to ensure durability, recovery manager has to redo (rollforward) transaction’s updates.

Transactions and Recovery

- If transaction had not committed at failure time, recovery manager has to undo (rollback) any effects of that transaction for atomicity.

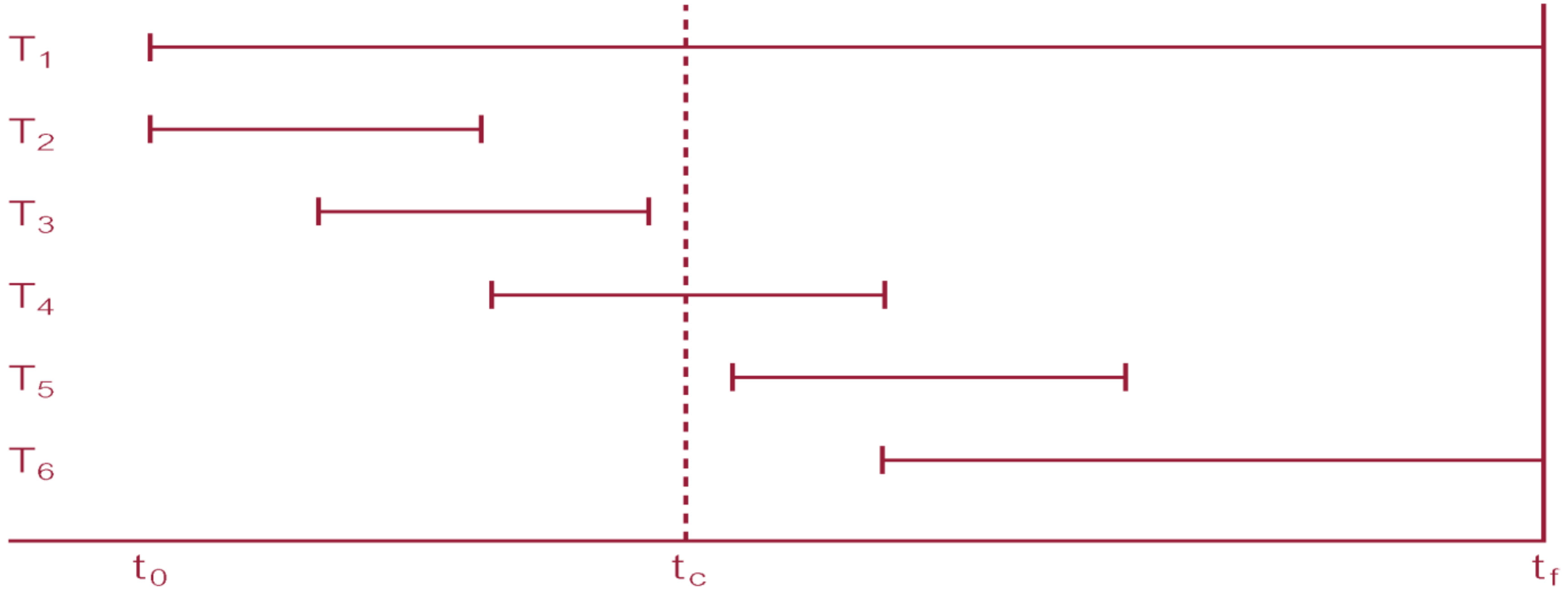
- Partial undo - only one transaction has to be undone.

- Global undo - all transactions have to be undone.

Example

- DBMS starts at time t0, but fails at time tf. Assume data for transactions T2 and T3 have been written to secondary storage.

- T1 and T6 have to be undone. In absence of any other information, recovery manager has to redo T2, T3, T4, and T5.



Agenda

- Backup and recovery
- Types of backups
- Types of database failure
- Database recovery
- Recovery facilities
  - Log files
  - Checkpoints
- Recovery techniques

Recovery Facilities

- DBMS should provide following facilities to assist with recovery:

    - Backup mechanism, which makes periodic backup copies of database.

    - Logging facilities, which keep track of current state of transactions and database changes.

    - Checkpoint facility, which enables updates to database in progress to be made permanent.

    - Recovery manager, which allows DBMS to restore database to consistent state following a failure.

Agenda

- Backup and recovery
- Types of backups
- Types of database failure
- Database recovery
- Recovery facilities
    - Log files
    - Checkpoints
- Recovery techniques

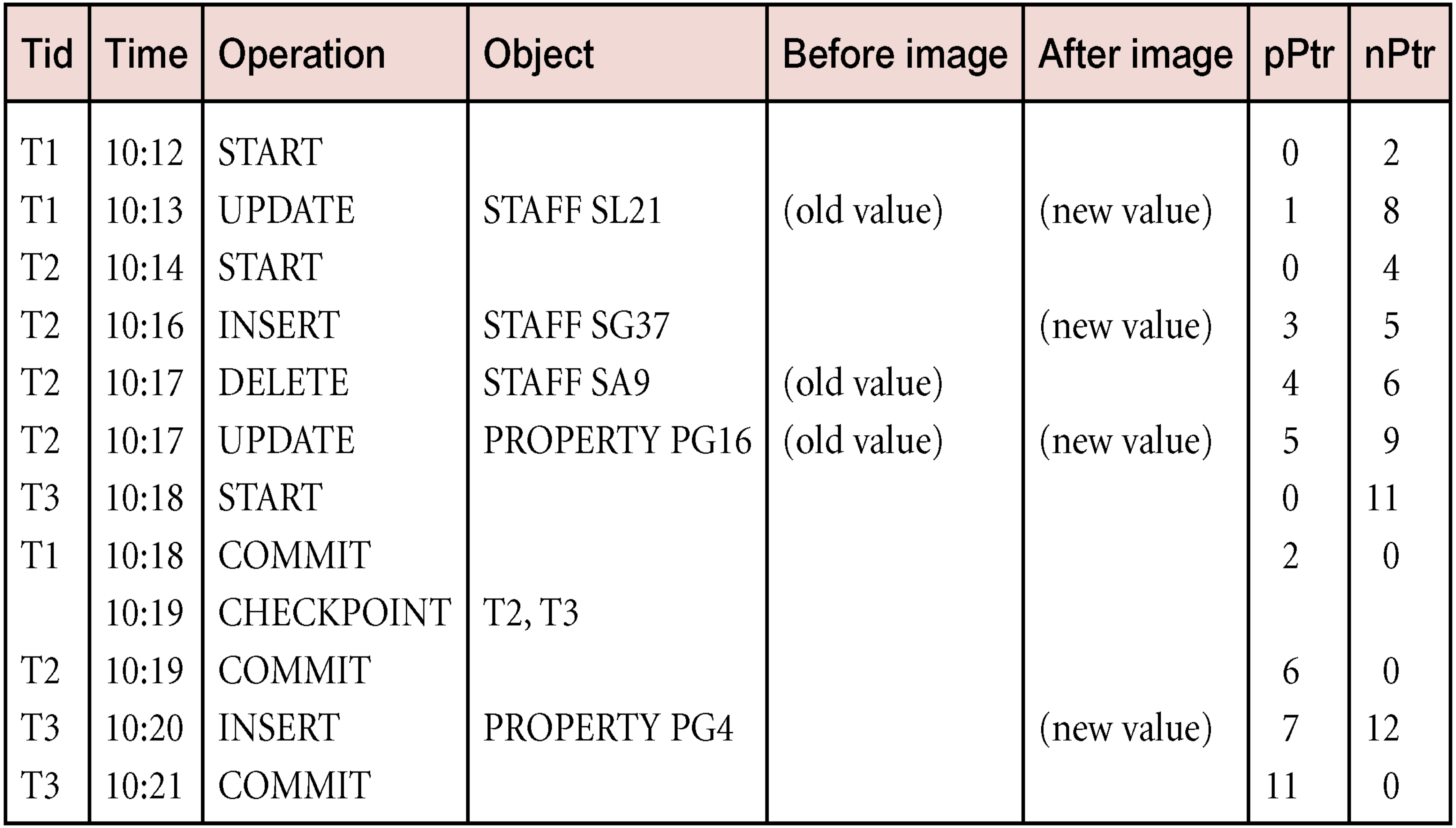
Log File

- Contains information about all updates to database:
  - Transaction records.
  - Checkpoint records.
- Often used for other purposes (for example, auditing).

Log File

- Transaction records contain:
  - Transaction identifier.
  - Type of log record, (transaction start, insert, update, delete, abort, commit).
  - Identifier of data item affected by database action (insert, delete, and update operations).
  - Before-image of data item.
  - After-image of data item.
  - Log management information.

Sample Log File

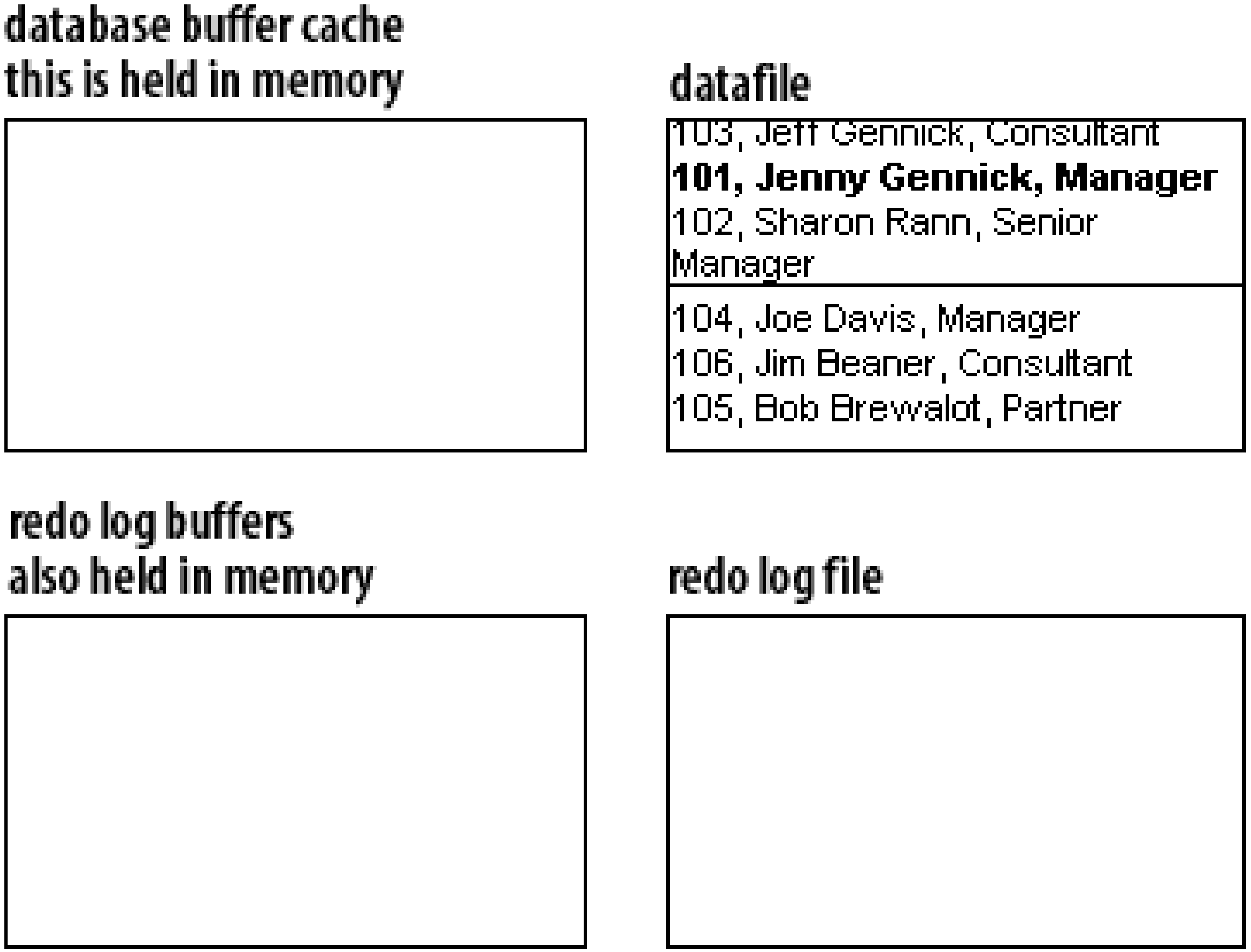


Tid	Time	Operation	Object	Before image	After image	pPtr	nPtr
T1	10:12	START				0	2
T1	10:13	UPDATE	STAFF SL21	(old value)	(new value)	1	8
T2	10:14	START				0	4
T2	10:16	INSERT	STAFF SG37		(new value)	3	5
T2	10:17	DELETE	STAFF SA9	(old value)		4	6
T2	10:17	UPDATE	PROPERTY PG16	(old value)	(new value)	5	9
T3	10:18	START				0	11
T1	10:18	COMMIT				2	0
	10:19	CHECKPOINT	T2, T3				
T2	10:19	COMMIT				6	0
T3	10:20	INSERT	PROPERTY PG4		(new value)	7	12
T3	10:21	COMMIT				11	0

Log File

- Log file may be duplexed or triplexed.
- Log file sometimes split into two separate random-access files.
- Potential bottleneck; critical in determining overall performance.

Log files example

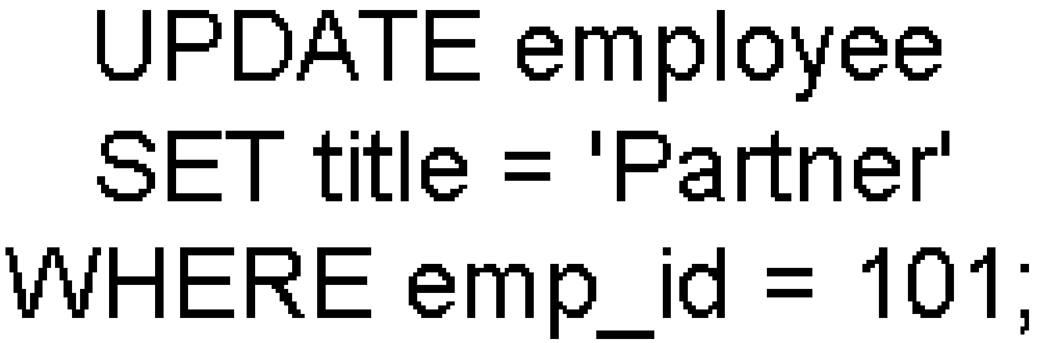


103, Jeff Gennick, Consultant
101, Jenny Gennick, Manager
102, Sharon Rann, Senior Manager

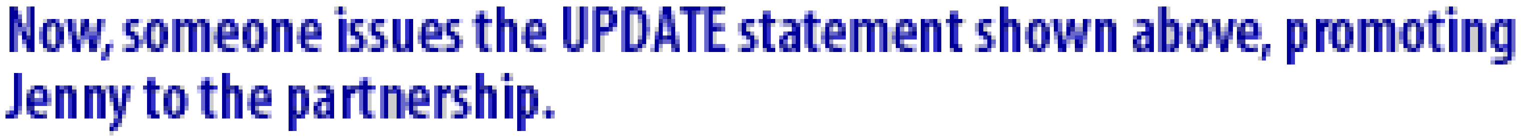
104, Joe Davis, Manager
106, Jim Beaner, Consultant
105, Bob Brewalot, Partner

Here is the initial state. You can see that Jenny Gennick holds the title of Manager.

Log files example

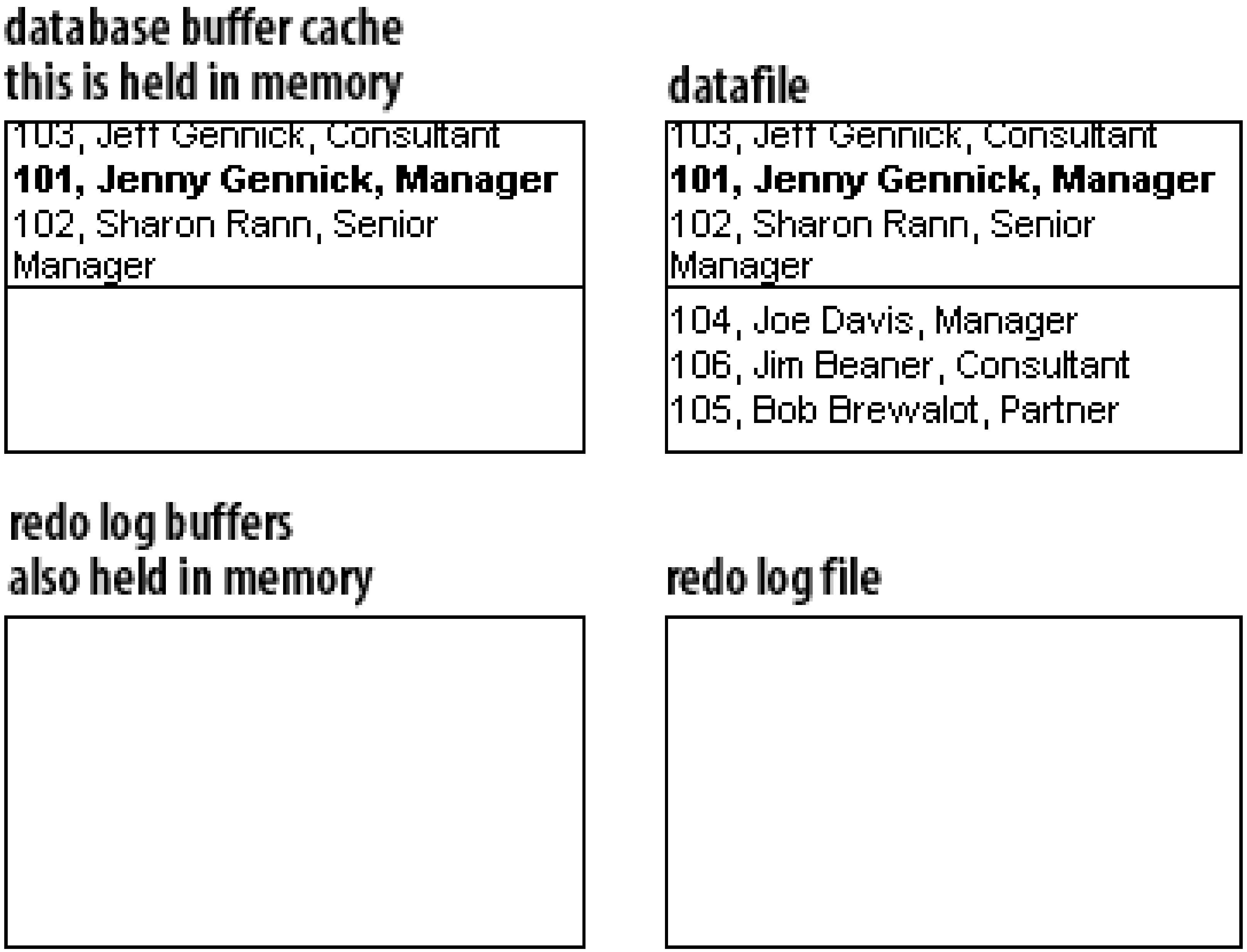


```
UPDATE employee
SET title = 'Partner'
WHERE emp_id = 101;
```



Now, someone issues the UPDATE statement shown above, promoting Jenny to the partnership.

Log files example



database buffer cache
this is held in memory

103, Jeff Gennick, Consultant
101, Jenny Gennick, Manager
102, Sharon Rann, Senior Manager

datafile

103, Jeff Gennick, Consultant
101, Jenny Gennick, Manager
102, Sharon Rann, Senior Manager

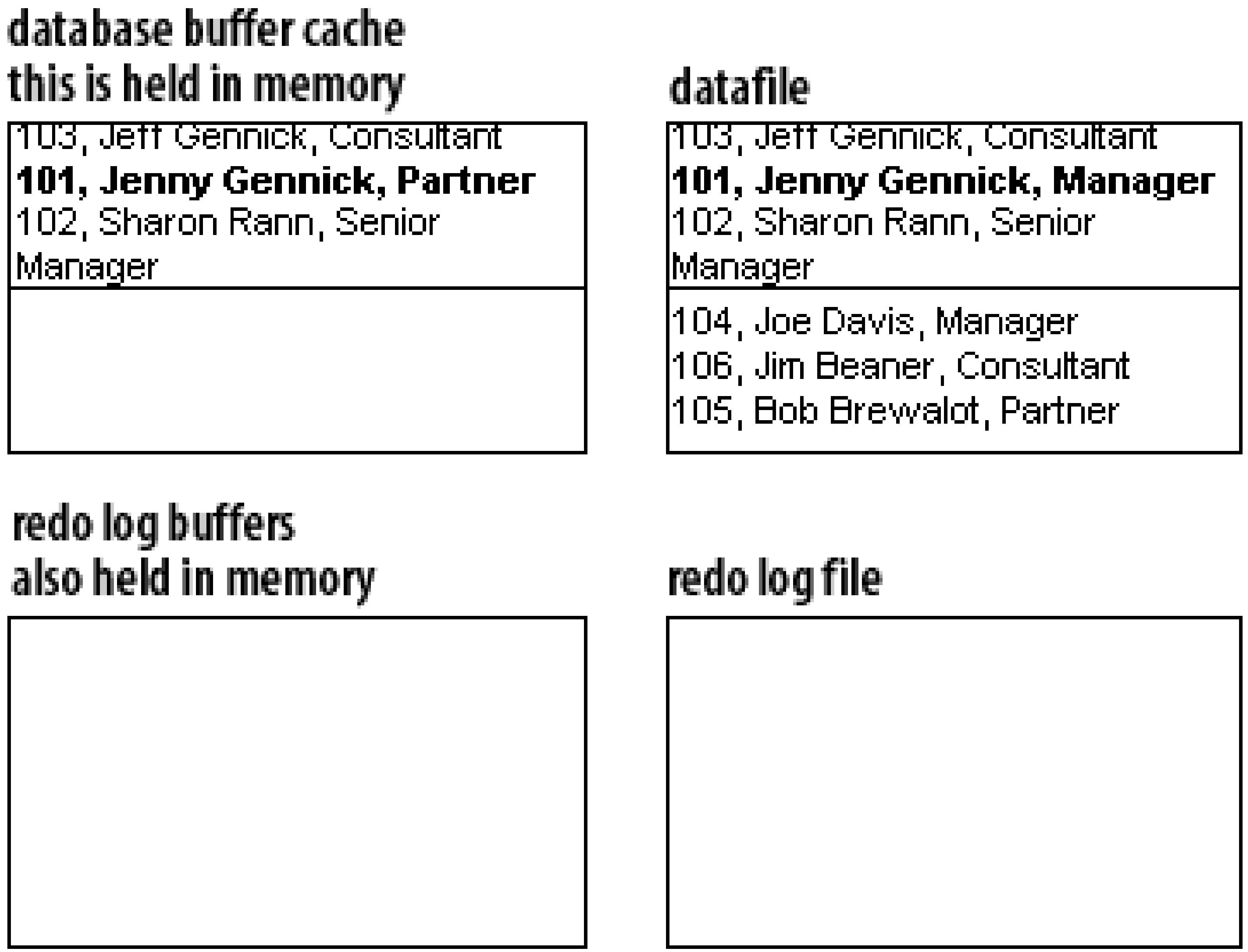
104, Joe Davis, Manager
106, Jim Beaner, Consultant
105, Bob Brewalot, Partner

redo log buffers
also held in memory

redo log file

The first thing that happens is that Oracle reads the data block containing Jenny's record into memory.

Log files example



database buffer cache
this is held in memory

103, Jeff Gennick, Consultant
101, Jenny Gennick, Partner
102, Sharon Rann, Senior Manager

datafile

103, Jeff Gennick, Consultant
101, Jenny Gennick, Manager
102, Sharon Rann, Senior Manager

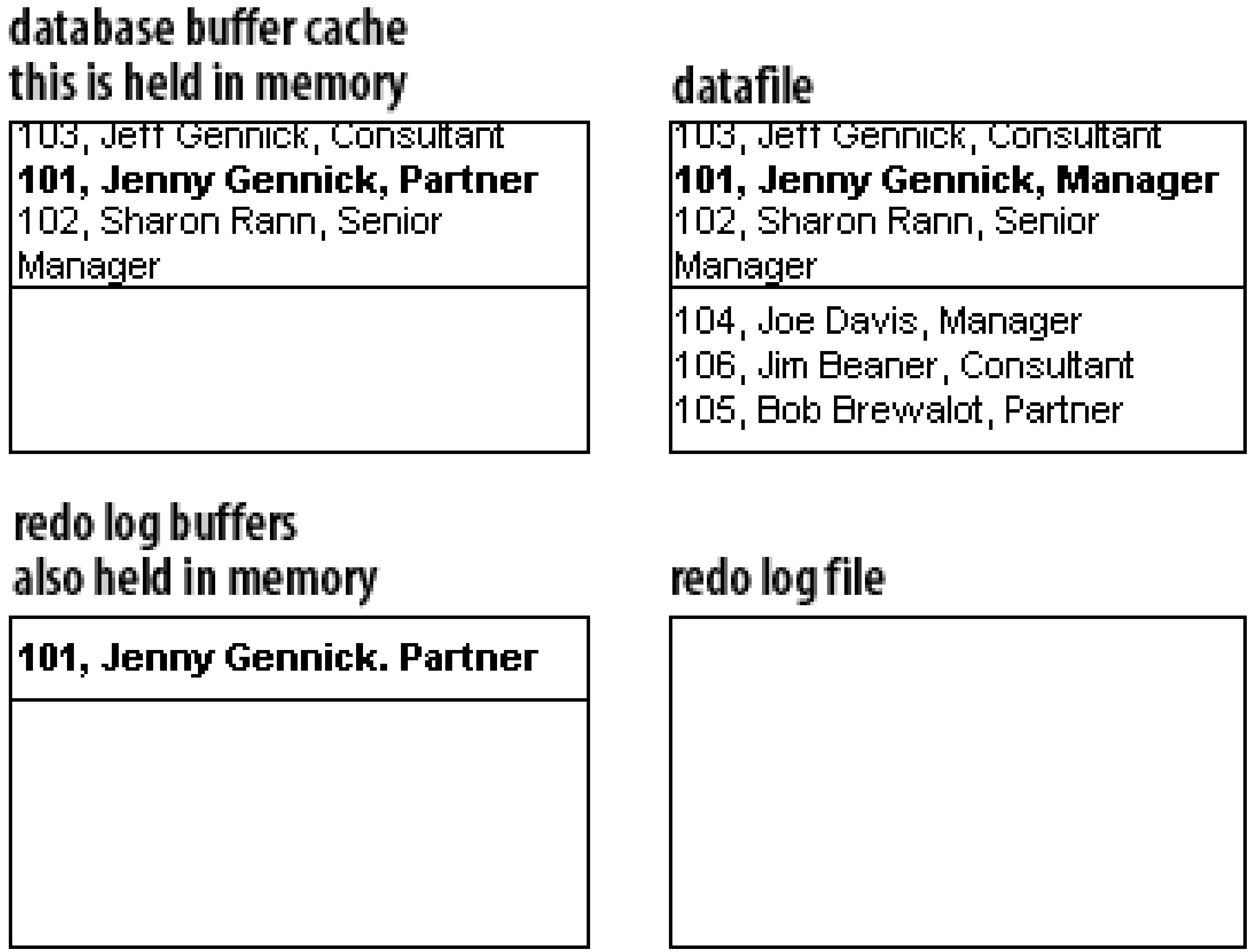
104, Joe Davis, Manager
106, Jim Beaner, Consultant
105, Bob Brewalot, Partner

redo log buffers
also held in memory

redo log file

Next, Oracle updates the block of memory.

Log files example



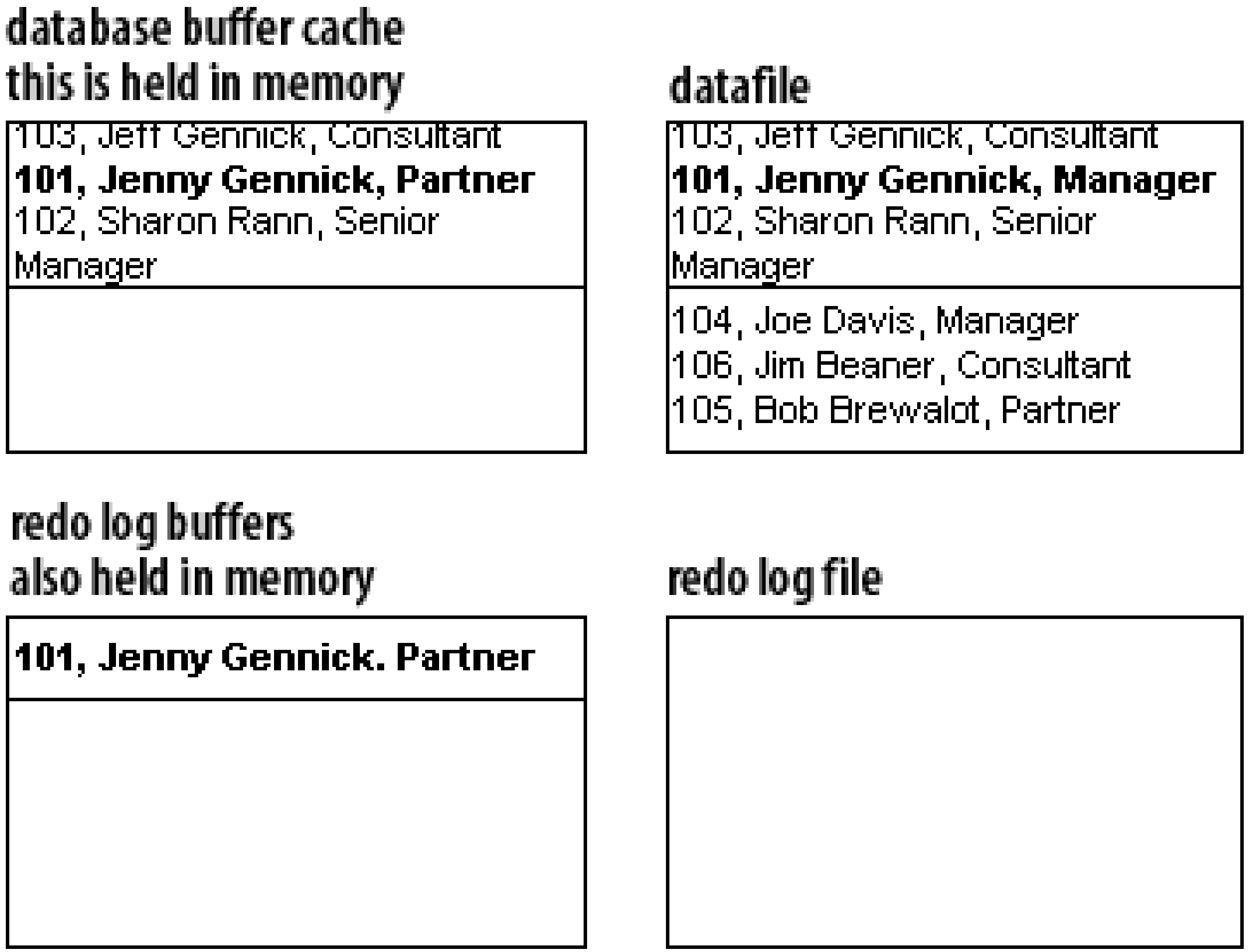
103, Jeff Gennick, Consultant
101, Jenny Gennick, Partner
102, Sharon Rann, Senior Manager

103, Jeff Gennick, Consultant
101, Jenny Gennick, Manager
102, Sharon Rann, Senior Manager
104, Joe Davis, Manager
106, Jim Beaner, Consultant
105, Bob Brewalot, Partner

101, Jenny Gennick. Partner

Oracle places information about the change made into the redo log buffer. From here it will be written to disk.

Log files example



database buffer cache
this is held in memory

103, Jeff Gennick, Consultant
101, Jenny Gennick, Partner
102, Sharon Rann, Senior Manager

datafile

103, Jeff Gennick, Consultant
101, Jenny Gennick, Manager
102, Sharon Rann, Senior Manager

104, Joe Davis, Manager
106, Jim Beaner, Consultant
105, Bob Brewalot, Partner

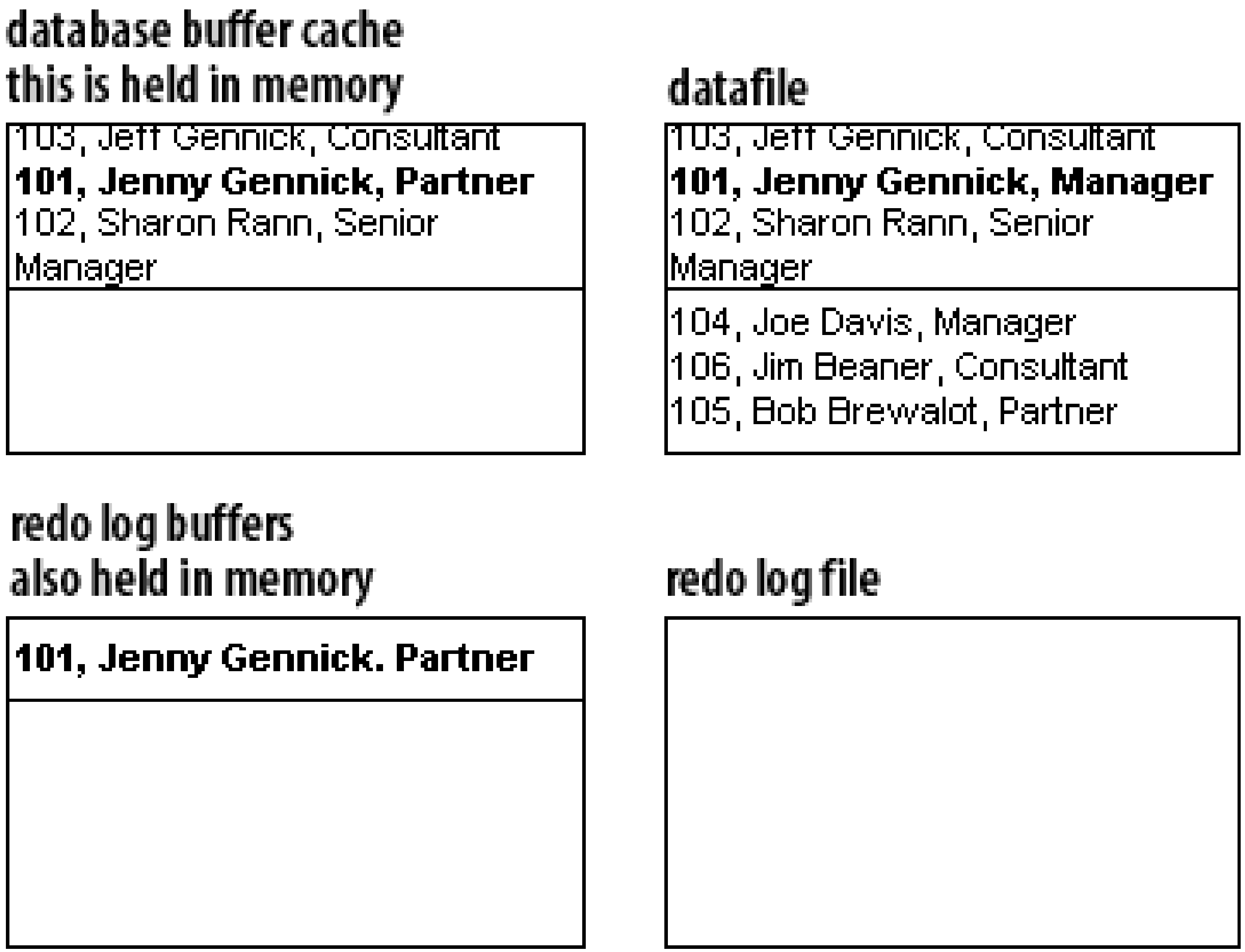
redo log buffers
also held in memory

101, Jenny Gennick. Partner

redo log file

If the server crashes now, this change will be lost. That's OK, because the transaction hasn't been committed yet.

Log files example



database buffer cache
this is held in memory

103, Jeff Gennick, Consultant
101, Jenny Gennick, Partner
102, Sharon Rann, Senior Manager

datafile

103, Jeff Gennick, Consultant
101, Jenny Gennick, Manager
102, Sharon Rann, Senior Manager

104, Joe Davis, Manager
106, Jim Beaner, Consultant
105, Bob Brewalot, Partner

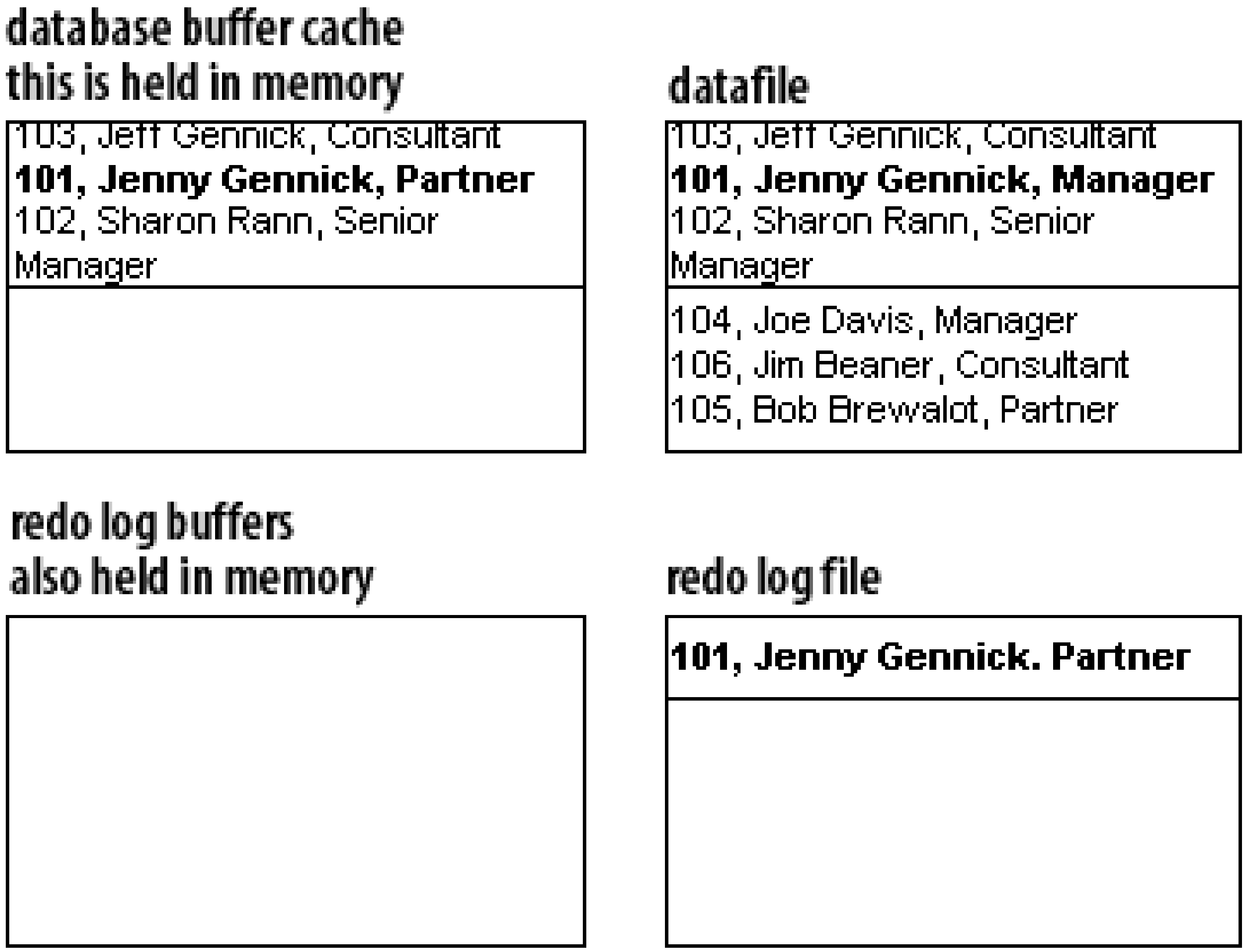
redo log buffers
also held in memory

101, Jenny Gennick. Partner

redo log file

The application that issued the UPDATE now issues a COMMIT.

Log files example



database buffer cache
this is held in memory

103, Jeff Gennick, Consultant
101, Jenny Gennick, Partner
102, Sharon Rann, Senior Manager

datafile

103, Jeff Gennick, Consultant
101, Jenny Gennick, Manager
102, Sharon Rann, Senior Manager

104, Joe Davis, Manager
106, Jim Beaner, Consultant
105, Bob Brewalot, Partner

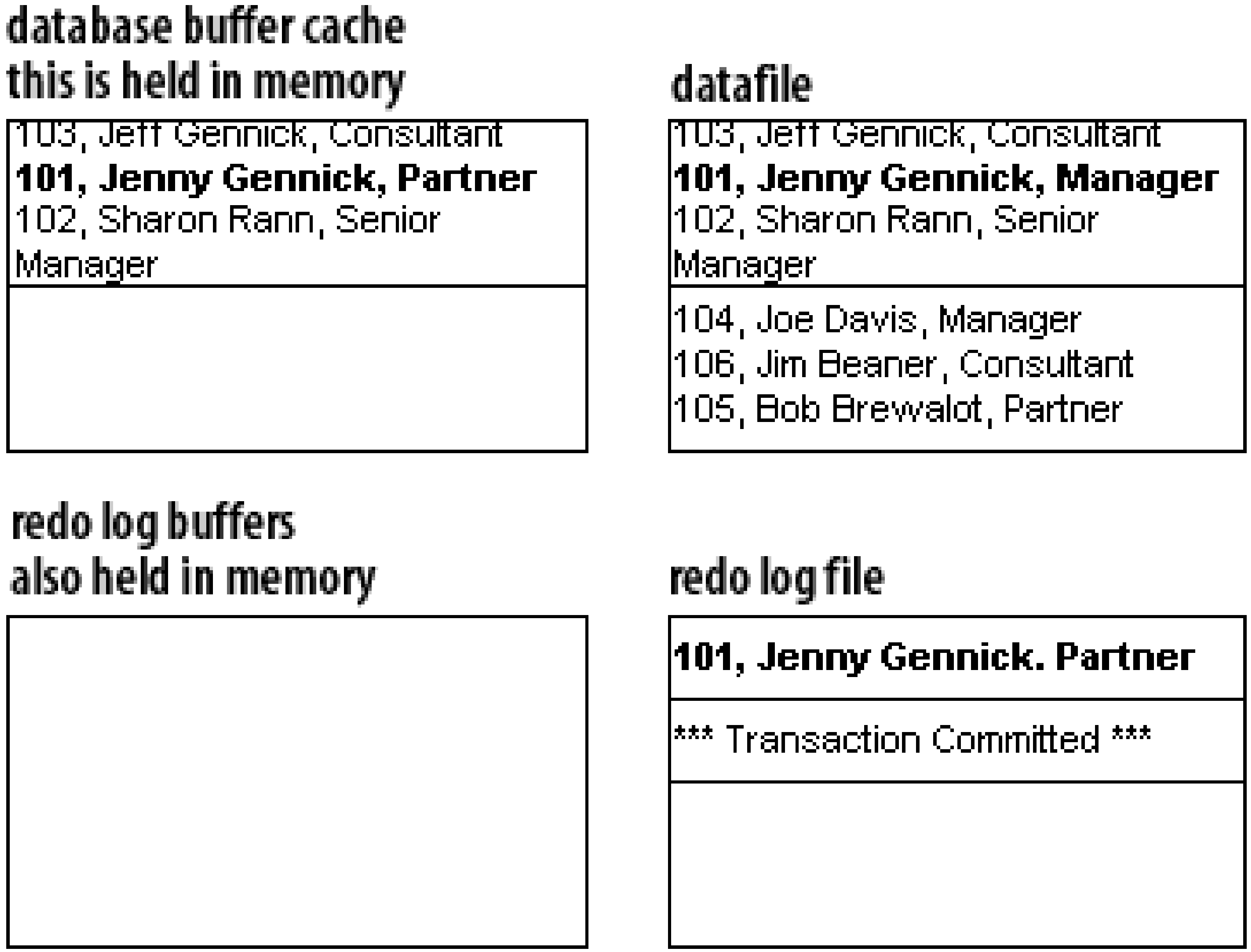
redo log buffers
also held in memory

redo log file

101, Jenny Gennick. Partner

Oracle ensures that the associated redo information is physically written to the redo log file.

Log files example



database buffer cache
this is held in memory

103, Jeff Gennick, Consultant
101, Jenny Gennick, Partner
102, Sharon Rann, Senior Manager

datafile

103, Jeff Gennick, Consultant
101, Jenny Gennick, Manager
102, Sharon Rann, Senior Manager

104, Joe Davis, Manager
106, Jim Beaner, Consultant
105, Bob Brewalot, Partner

redo log buffers
also held in memory

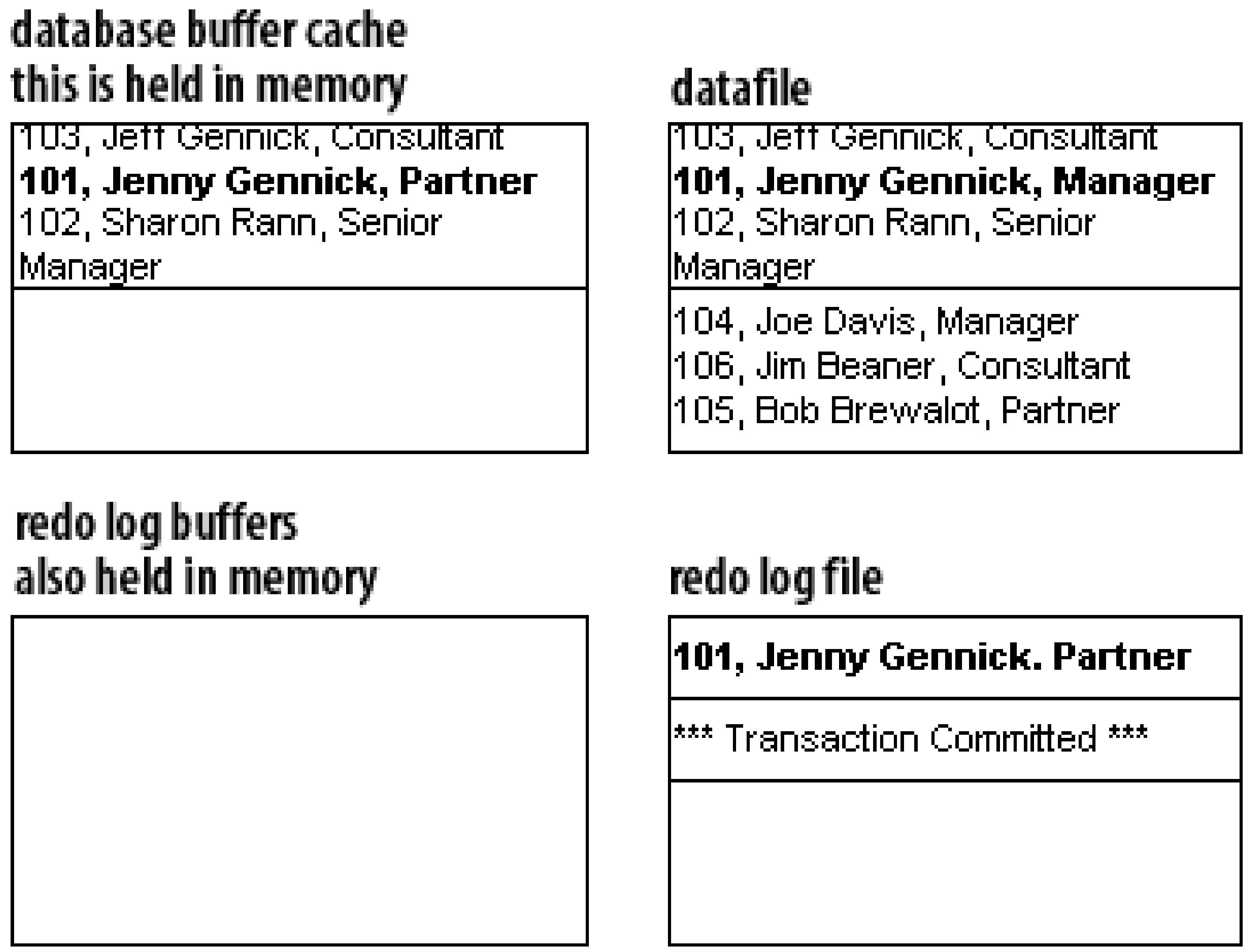
redo log file

101, Jenny Gennick. Partner

*** Transaction Committed ***

Oracle also writes a commit record to the redo log file for this transaction.

Log files example



database buffer cache
this is held in memory

103, Jeff Gennick, Consultant
101, Jenny Gennick, Partner
102, Sharon Rann, Senior Manager

datafile

103, Jeff Gennick, Consultant
101, Jenny Gennick, Manager
102, Sharon Rann, Senior Manager

104, Joe Davis, Manager
106, Jim Beaner, Consultant
105, Bob Brewalot, Partner

redo log buffers
also held in memory

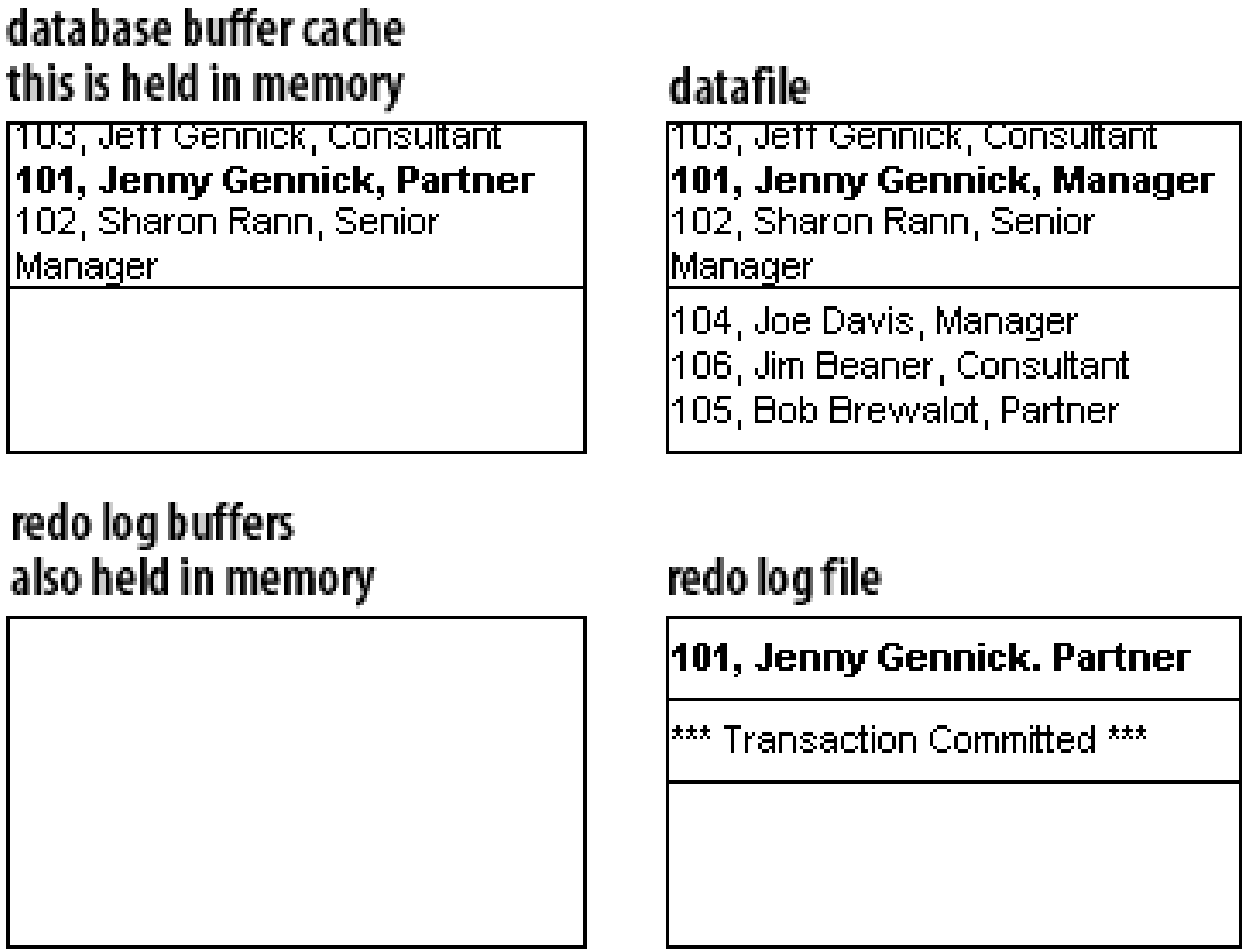
redo log file

101, Jenny Gennick. Partner

*** Transaction Committed ***

Only now does Oracle return control back to the application, and indicate that the commit was successful. Jenny is now a partner.

Log files example



database buffer cache
this is held in memory

103, Jeff Gennick, Consultant
101, Jenny Gennick, Partner
102, Sharon Rann, Senior Manager

datafile

103, Jeff Gennick, Consultant
101, Jenny Gennick, Manager
102, Sharon Rann, Senior Manager

104, Joe Davis, Manager
106, Jim Beaner, Consultant
105, Bob Brewalot, Partner

redo log buffers
also held in memory

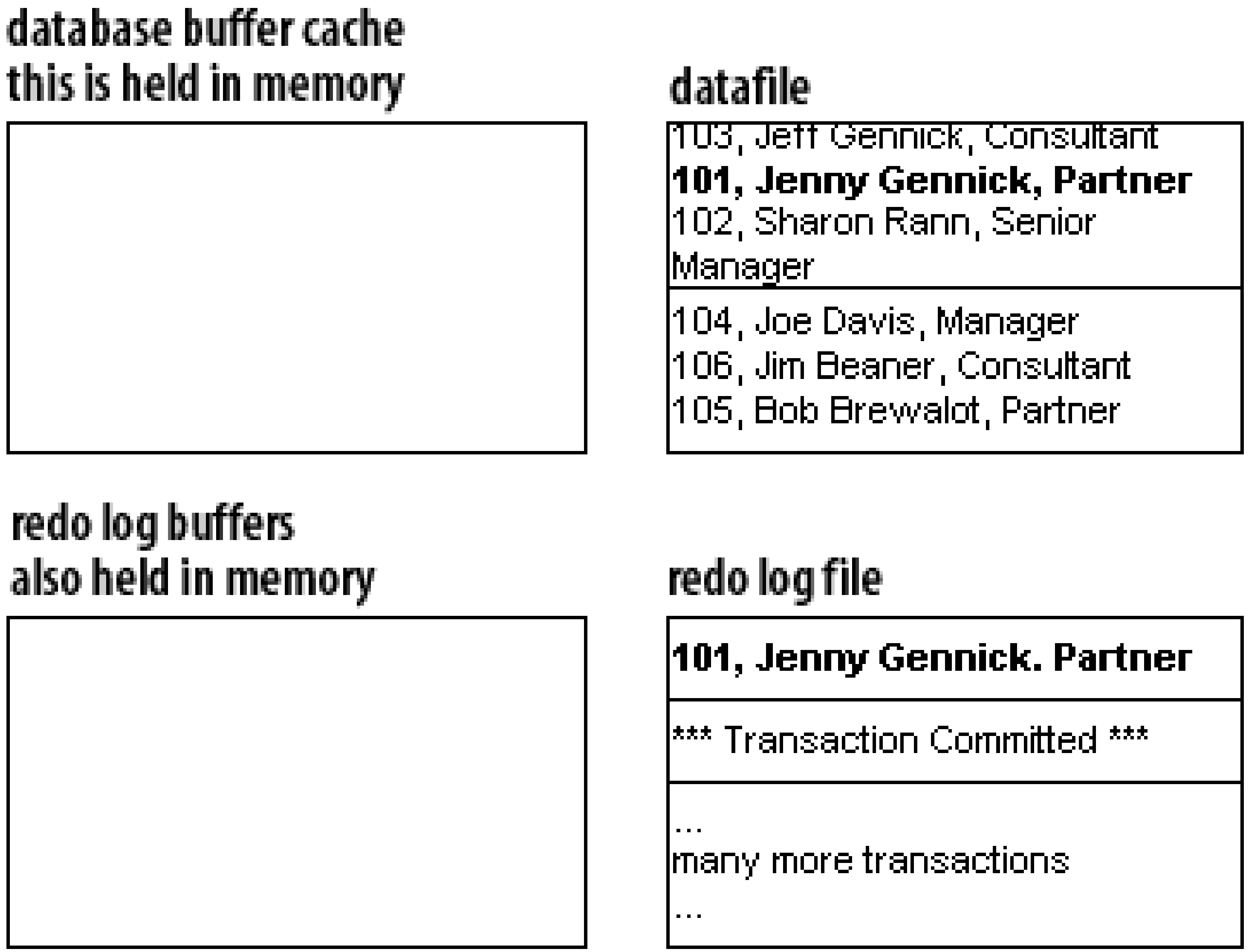
redo log file

101, Jenny Gennick. Partner

*** Transaction Committed ***

At this point, if the server crashes, Oracle can redo the change based on the information contained in the redo log file.

Log files example



database buffer cache
this is held in memory

datafile

103, Jeff Gennick, Consultant
101, Jenny Gennick, Partner
102, Sharon Rann, Senior Manager

104, Joe Davis, Manager
106, Jim Beaner, Consultant
105, Bob Brewalot, Partner

redo log buffers
also held in memory

redo log file

101, Jenny Gennick. Partner

*** Transaction Committed ***

...
many more transactions
...

Later, and this could be much later, the updated data block will be written back to the datafile.

Log files in MySQL

- MySQL has the same elements (redo logs, log buffer, etc.)

- Logs are stored in the program data folder

Agenda

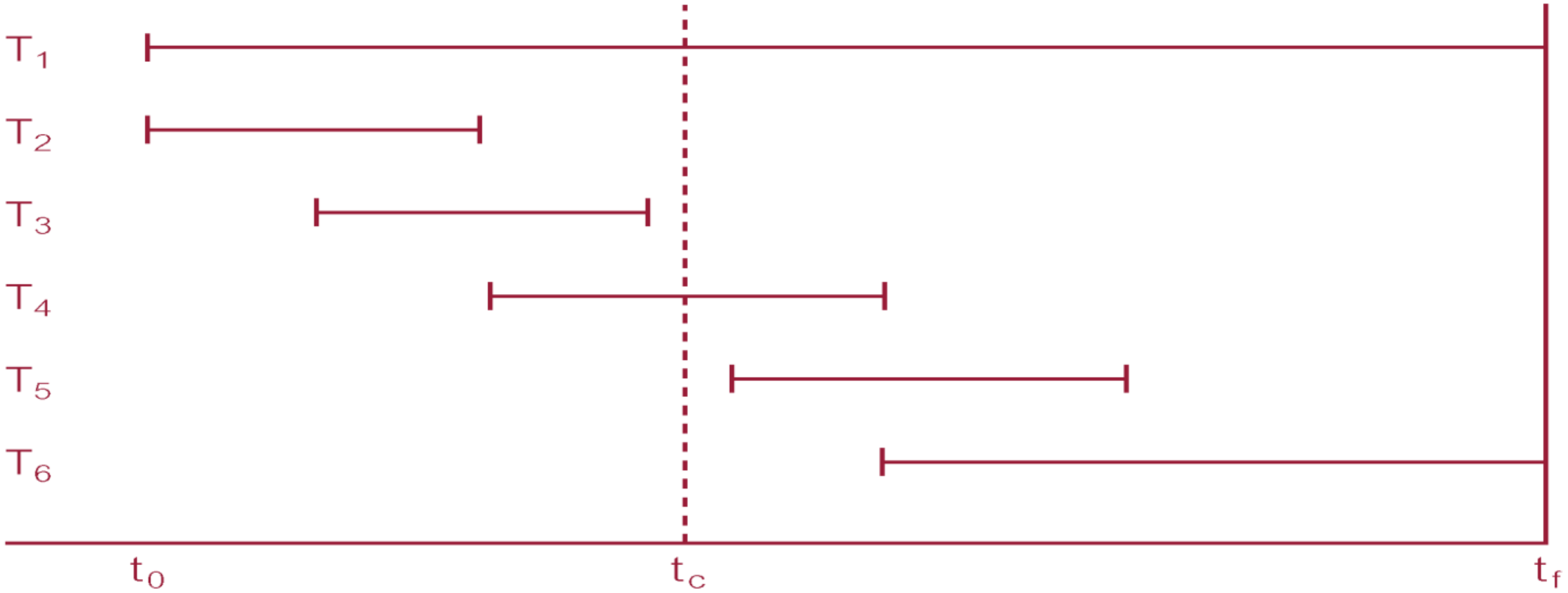
- Backup and recovery
- Types of backups
- Types of database failure
- Database recovery
- Recovery facilities
  - Log files
  - Checkpoints
- Recovery techniques

Checkpointing

- Checkpoint
  - Point of synchronization between database and log file. All buffers are force-written to secondary storage.
- Checkpoint record is created containing identifiers of all active transactions.
- When failure occurs, redo all transactions that committed since the checkpoint and undo all transactions active at time of crash.

Checkpointing

- In previous example, with checkpoint at time tc, changes made by T2 and T3 have been written to secondary storage.
- Thus:
  - only redo T4 and T5,
  - undo transactions T1 and T6.



Agenda

- Backup and recovery
- Types of backups
- Types of database failure
- Database recovery
- Recovery facilities
  - Log files
  - Checkpoints
- Recovery techniques

Recovery Techniques

- If database has been damaged:
  - Need to restore last backup copy of database and reapply updates of committed transactions using log file.
- If database is only inconsistent:
  - Need to undo changes that caused inconsistency. May also need to redo some transactions to ensure updates reach secondary storage.
  - Do not need backup, but can restore database using before- and after-images in the log file.

Recovery Techniques

- Three main recovery techniques:
  - Deferred Update
  - Immediate Update
  - Shadow Paging

Deferred Update

- Updates are not written to the database until after a transaction has reached its commit point.

- If transaction fails before commit, it will not have modified database and so no undoing of changes required.

- May be necessary to redo updates of committed transactions as their effect may not have reached database.

Immediate Update

- Updates are applied to database as they occur.
- Need to redo updates of committed transactions following a failure.
- May need to undo effects of transactions that had not committed at time of failure.
- Essential that log records are written before write to database. Write-ahead log protocol.

Immediate Update

- If no “transaction commit” record in log, then that transaction was active at failure and must be undone.

- Undo operations are performed in reverse order in which they were written to log.

Shadow Paging

- Maintain two page tables during life of a transaction: current page and shadow page table.

- When transaction starts, two pages are the same.

- Shadow page table is never changed thereafter and is used to restore database in event of failure.

- During transaction, current page table records all updates to database.

- When transaction completes, current page table becomes shadow page table.