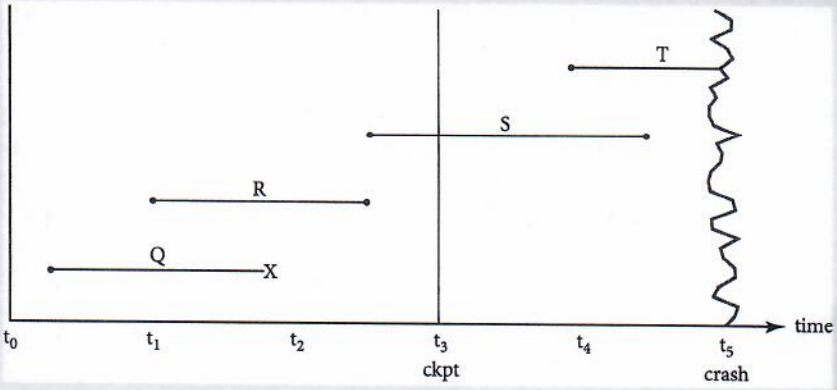
is written. The log can be used to **undo** as well as to **redo** transactions event of failure. **Shadow paging** is a recovery technique that uses no Instead, updates are made to a new copy of each page. A **current page** points to the new page. A **shadow page table** continues to point to the until the transaction commits, at which time the current page table es the shadow page table. The **ARIES** recovery algorithm is a highly e and flexible algorithm that does recovery by attempting to recreate ct state the database was in at the time of failure, and then applying and redo operations as needed.

Oracle handles concurrency control by using a variety of locking and mping techniques, including multiversioning and logging but no ocks. It uses undo data segments for both concurrency and recovery. action **IDs** are used to identify transactions, and **system change** ers (SCNs) are used to identify and give the order of consistent states database. The Oracle recovery manager maintains control files, rollback nts, redo logs, and archived redo logs for recovery.

Exercises

1. Assume a DBMS that uses immediate updates has the following log entries. FIGURE 9.18 shows a timeline depicting each transaction's starting and ending time, along with the checkpoint. A start and

URE 9.18
saction Timeline for Exercise 9.1



a commit are indicated by a large dot, while an abort is indicated by an X.

```
<Q starts>
<R starts>
<Q,w,0,20>
<R,x,1,5>
<Q aborts>
<R,y, - 1,0>
<R commits>
<S starts>
<S,z,8,12>
<checkpoint record>
<S,x,5,10>
<T starts>
<T,y,0,15>
<S commits>
——system crash———
```

Assuming a system crash occurs as indicated immediately after the <S commits> log record:

a. Which transactions, if any, need to be redone?
b. Which transactions, if any, need to be undone?
c. Which transactions, if any, are not affected by the crash?
d. Assuming the variables in different transactions refer to database items with the same names, what are the final values of w, x, y, and z?
e. If a second system failure occurs while the first recovery is in progress, what needs to be done after the system recovers for the second time?

9.2 Assume that the same transactions and operations as shown in Exercise 9.1 are being done by a system that uses the deferred update protocol.

a. Rewrite the log entries for the transactions in Exercise 9.1 for this logging method.
b. Which transactions, if any, need to be redone?

9.5

9.6

c. Which transactions, if any, need to be undone?
d. Which transactions, if any, are not affected by the crash?
e. What will be the final values of w, x, y, and z?
f. If a second system failure occurs while the first recovery is in progress, what needs to be done after the system recovers for the second time?

9.3 Suppose the log in Exercise 9.1 contained the entry <S aborts> in place of the entry <S commits>.

a. If the system is using immediate updates and the crash occurred before the rollback took place, what changes, if any, would you make in the recovery process?
b. If the system were using deferred updates, what changes would you make in the recovery process of Exercise 9.2?

9.4 Assume the following transactions are to be performed.

Transaction S:

read(a);

$a = a + 10$;

write(a);

read(b);

$b = b*5$;

write(b);

Transaction T:

read(a);

$a = a*2$;

write a;

a. If the initial value of a is 10 and the initial value of b is 20, what are their final values if we perform the transactions serially, using order S,T?
b. Using the same initial values, what are the final values of a and b if the order of execution is T,S?
c. Does this result have any implications for serializability?

9.5 Write a concurrent schedule for transactions S and T in Exercise 9.4 that illustrates the lost update problem.

9.6 Apply the standard two-phase locking protocol to the schedule you devised in Exercise 9.5. Will the protocol allow the execution of that schedule? Does deadlock occur?