Managing Undo Data

Objectives

After completing this lesson, you should be able to do the following:

- Explain DML and undo data generation
- Monitor and administer undo data
- Describe the difference between undo data and redo data
- Configure undo retention
- Guarantee undo retention
- Use the Undo Advisor

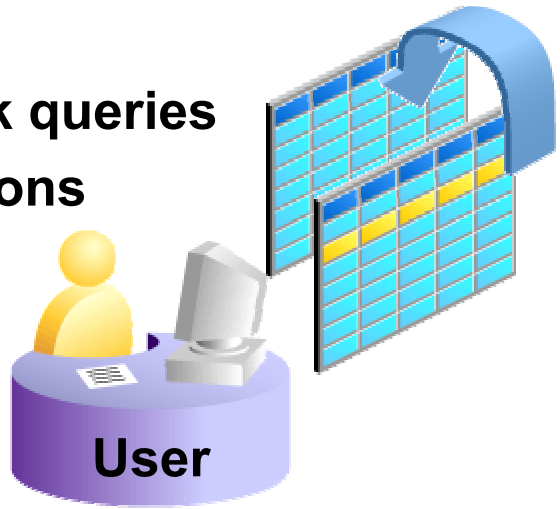
Data Manipulation

- Data manipulation language (DML) consists of the following SQL statements:
  - `INSERT`
  - `UPDATE`
  - `DELETE`
  - `MERGE`
- DML always executes as part of a transaction, which can be:
  - Rolled back, using the `ROLLBACK` command
  - Committed, using the `COMMIT` command

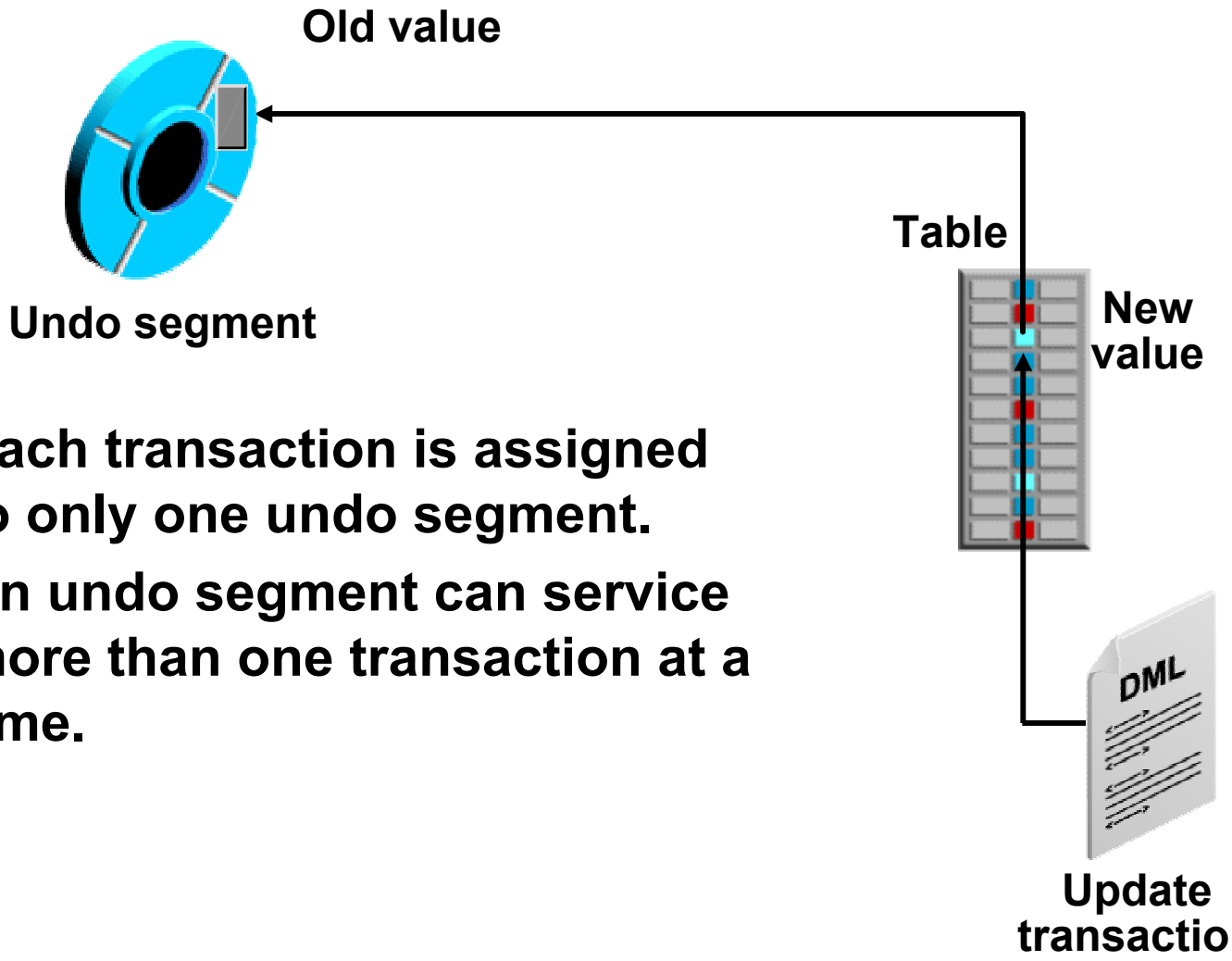
Undo Data

Undo data is:

- A copy of original, premodified data
- Captured for *every* transaction that changes data
- Retained at least until the transaction is ended
- Used to support:
  - Rollback operations
  - Read-consistent and flashback queries
  - Recovery from failed transactions



Transactions and Undo Data

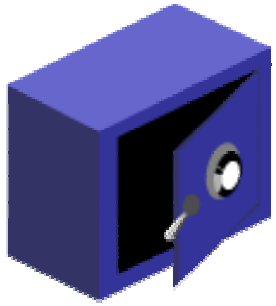


- Each transaction is assigned to only one undo segment.
- An undo segment can service more than one transaction at a time.

Storing Undo Information

Undo information is stored in undo segments, which are, in turn, stored in an undo tablespace. Undo tablespaces:

- Are used only for undo segments
- Have special recovery considerations
- May be associated with only a single instance
- Require that only one of them be the current writable undo tablespace for a given instance at any given time



Undo Data Versus Redo Data

	Undo	Redo
Record of	How to undo a change	How to reproduce a change
Used for	Rollback, read-consistency, flashback	Rolling forward database changes
Stored in	Undo segments	Redo log files
Protects against	Inconsistent reads in multiuser systems	Data loss

Monitoring Undo

Undo usually requires little management. The areas to monitor include:

- **Free space in an undo tablespace**
- **“Snapshot too old” errors**

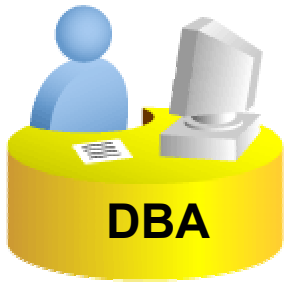
Administering Undo

Administration of undo should include preventing:

- Space errors in an undo tablespace:
  - Size the undo tablespace properly.
  - Ensure that large transactions commit periodically.
- "Snapshot too old" errors:
  - Configure an appropriate undo retention interval.
  - Size the undo tablespace properly.
  - Consider guaranteeing undo retention.

Use automatic undo management:

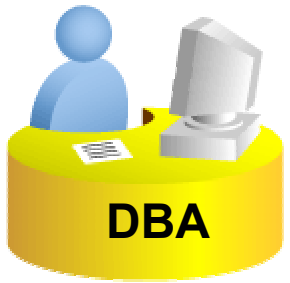
```
UNDO_MANAGEMENT=AUTO
UNDO_TABLESPACE=UNDOTBS1
```



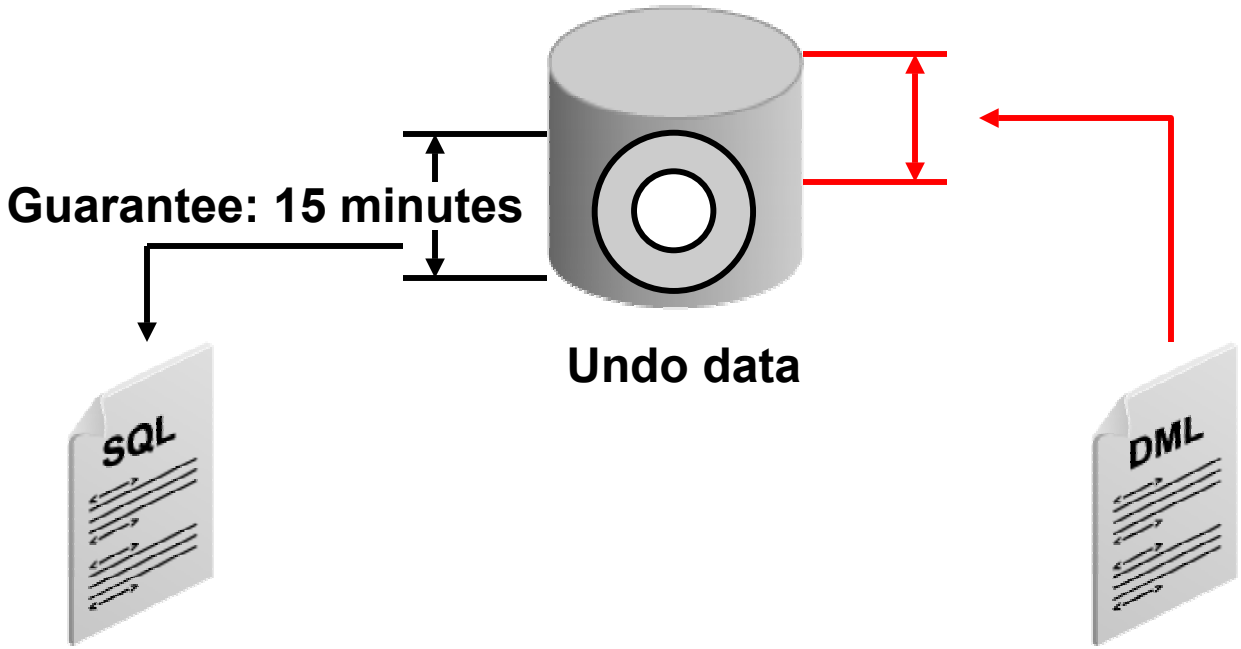
Configuring Undo Retention

`UNDO_RETENTION` specifies (in seconds) how long already committed undo information is to be retained. The only time you must set this parameter is when:

- The undo tablespace has the `AUTOEXTEND` option enabled
- You want to set undo retention for LOBs
- You want to guarantee retention



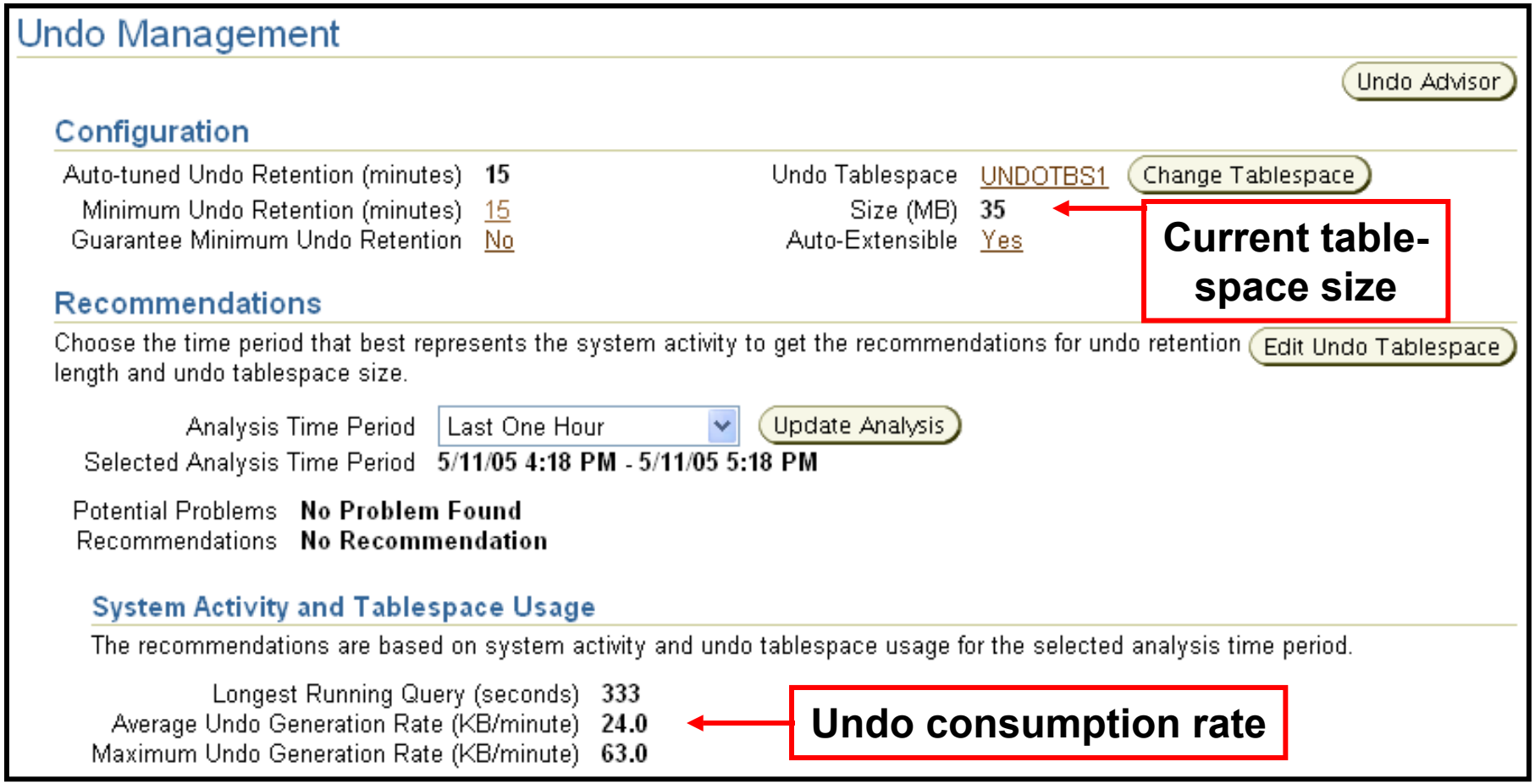
Guaranteeing Undo Retention



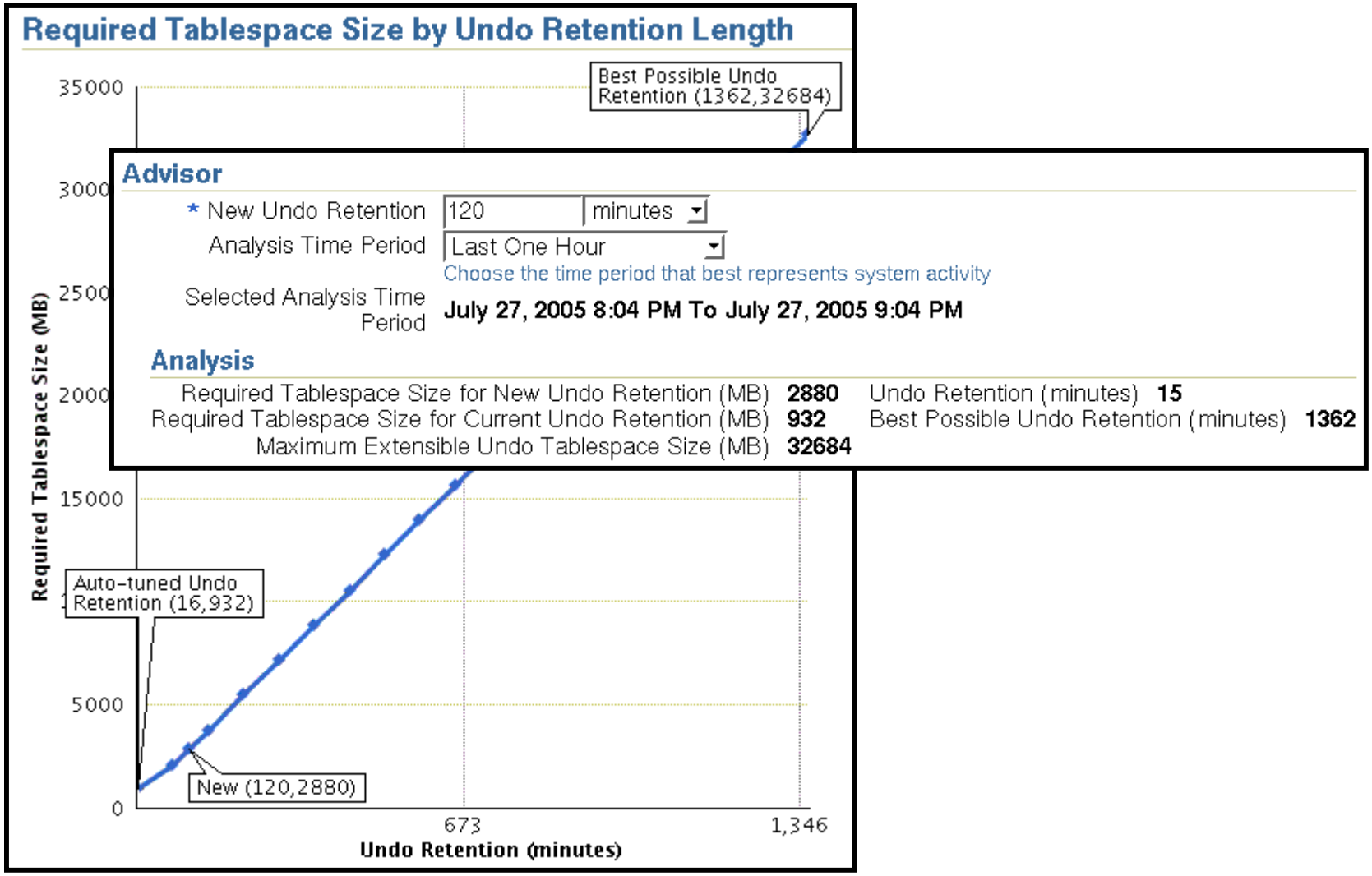
`SELECT` statements running 15 minutes or less are always satisfied.

A transaction that generates more undo than what there is space for will fail.

Sizing the Undo Tablespace



Using the Undo Advisor



Summary

In this lesson, you should have learned how to:

- Explain DML and undo data generation
- Monitor and administer undo segments
- Describe the difference between undo data and redo data
- Configure undo retention
- Guarantee undo retention
- Use the Undo Advisor

Practice Overview: Managing Undo Segments

This practice covers the following topics:

- Calculating undo tablespace sizing to support a 48-hour retention interval
- Modifying an undo tablespace to support a 48-hour retention interval