

# **Performing Incremental Backups**

**By Ahmed Baraka**

# Objectives

---

In this lecture, you will learn how to perform the following:

- Describe the difference between differential and cumulative incremental backups
- Take incremental backup
- Implement the incrementally updated backups
- Configure the Block Change Tracking (BCT)

# Backup Terminology

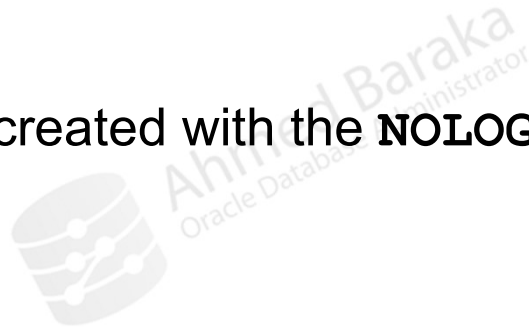
---

- Backup strategy may include:
  - **Whole**: entire database
  - **Partial**: portion of the database
- Backup type may indicate inclusion of:
  - **Full**: all data blocks within your chosen files
  - **Incremental**: only information that has changed since a previous backup
    - **Differential**: changes since last incremental
    - **Cumulative**: changes since last level 0
- Backup mode may be:
  - **Offline** (consistent, cold)
  - **Online** (inconsistent, hot)

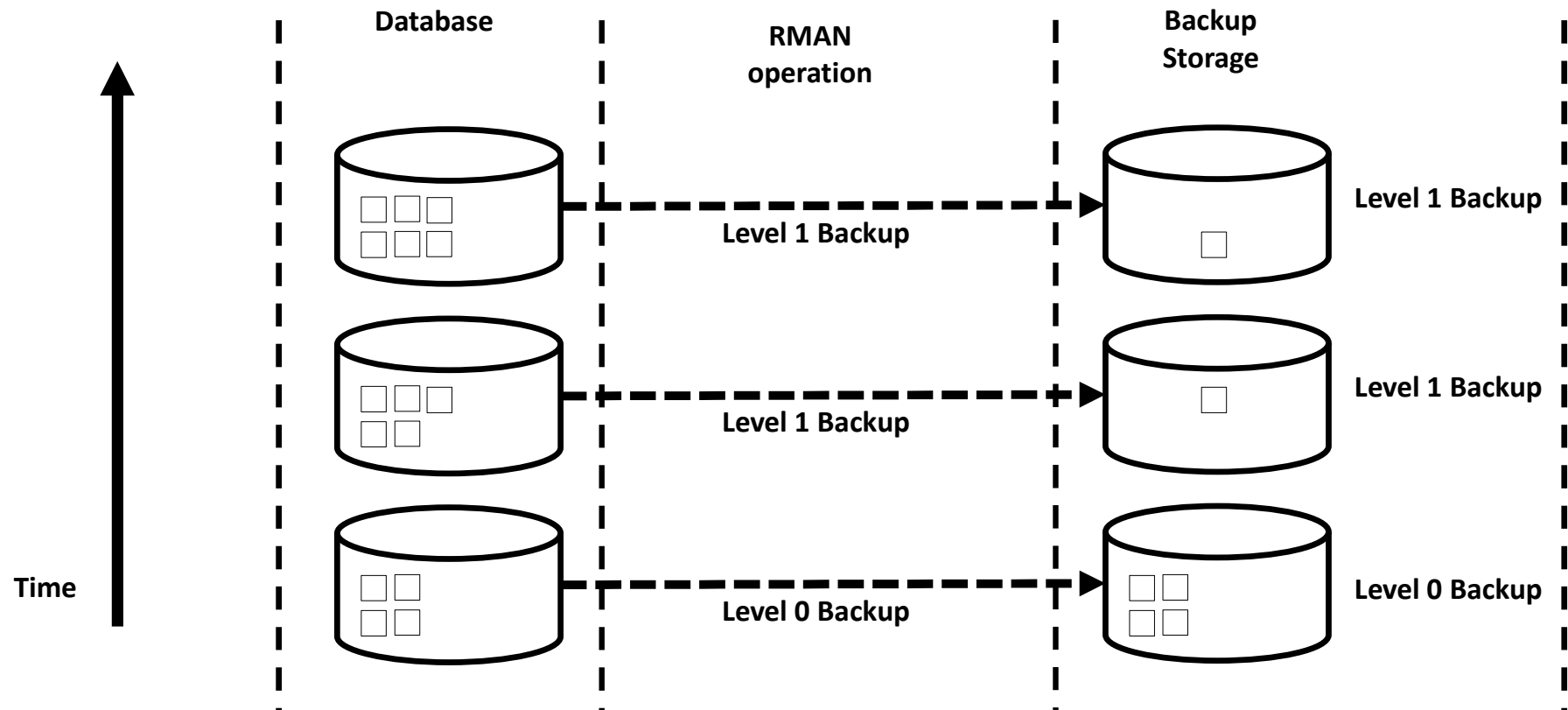
# RMAN Incremental Backup Benefits

---

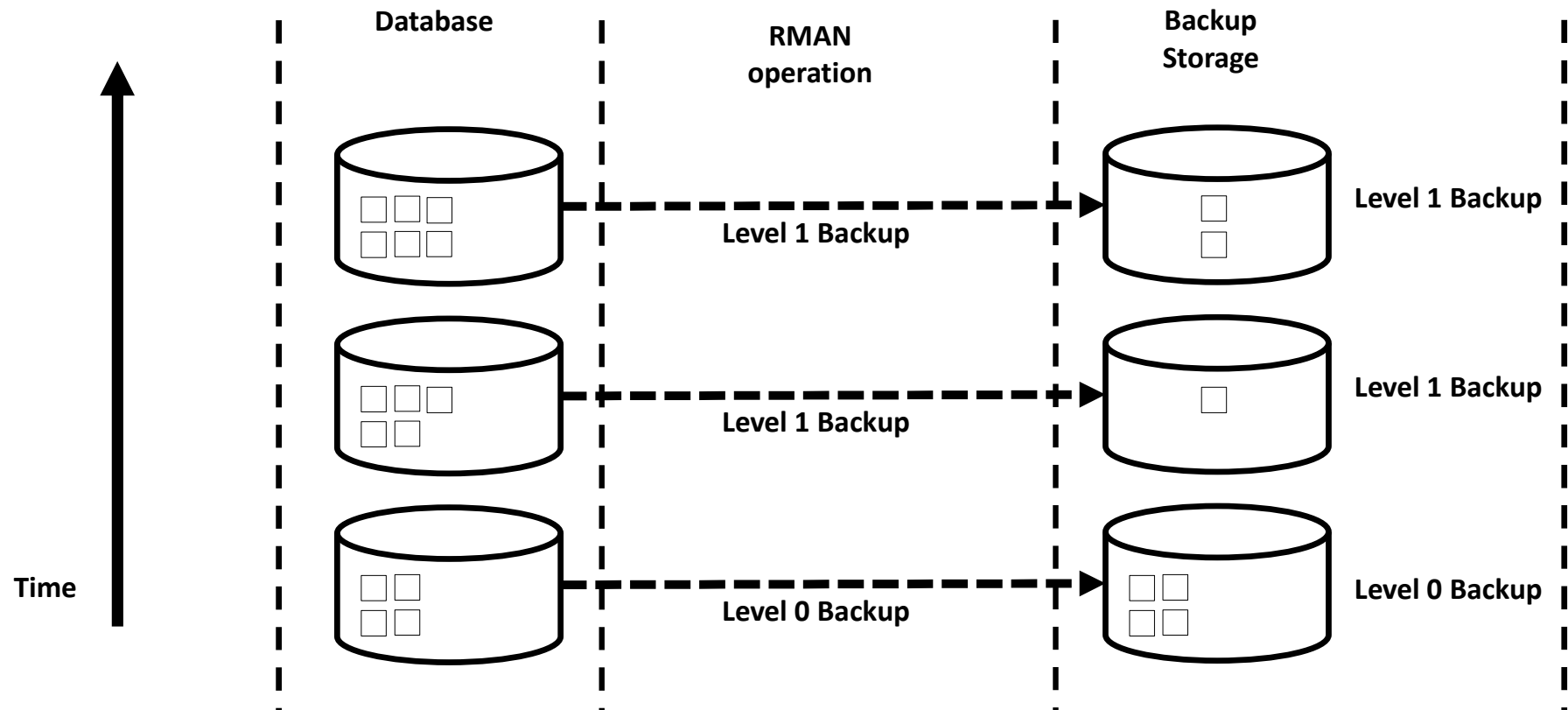
- Faster than full backup
- Less disk space consumption
- Less I/O bandwidth
- Could recover objects created with the **NOLOGGING** option



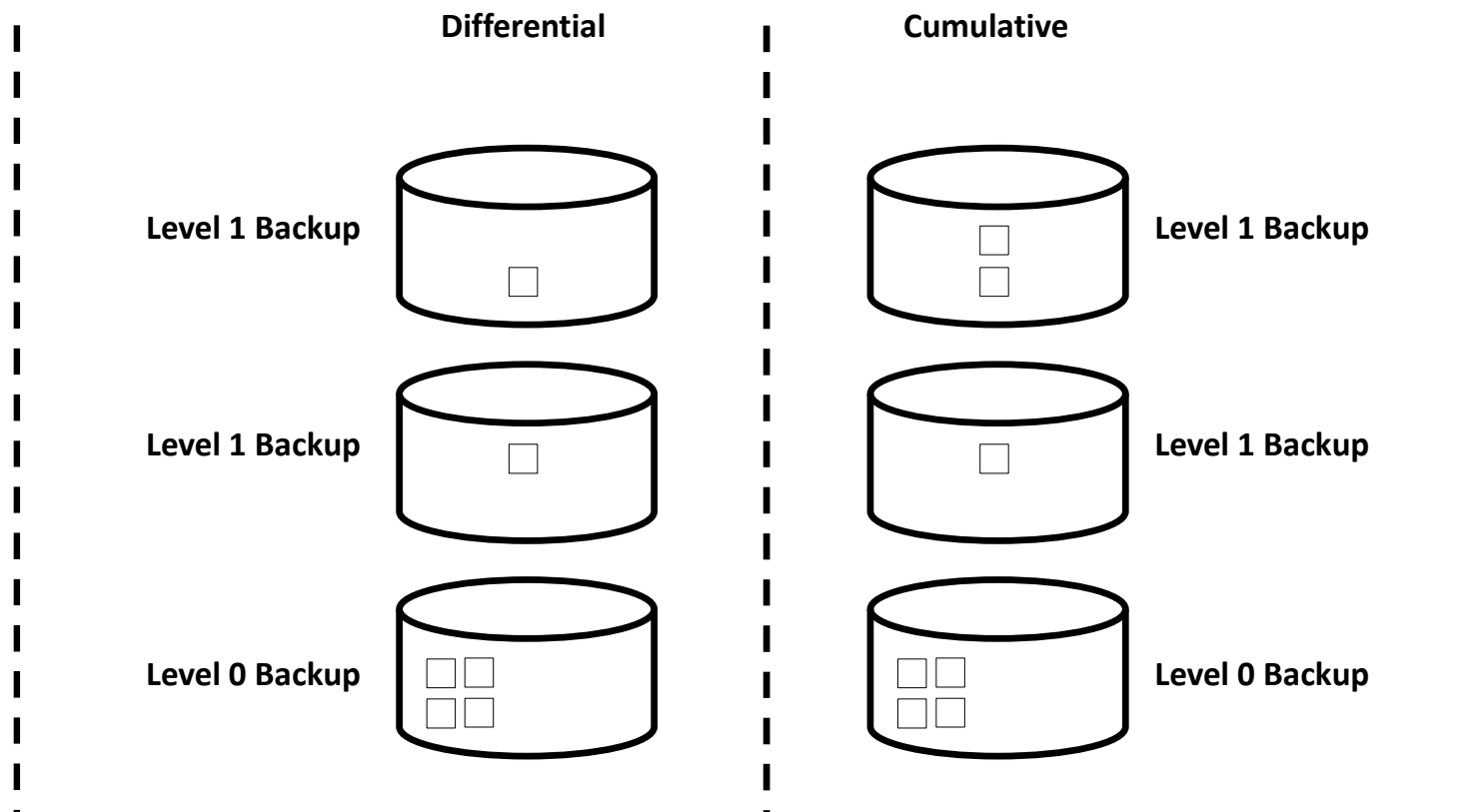
# RMAN Differential Incremental Backup



# RMAN Cumulative Incremental Backup



# Differential and Cumulative Incremental Backup



# Differential and Cumulative Incremental Backup

Differential	Cumulative
backs up all blocks changed after the most recent incremental backup at level 1 or 0	backs up all blocks changed after the most recent incremental backup at level 0
Default	Not Default
Takes less disk space	Takes more disk space
Reads more files during recovery	Reads fewer files during recovery



# Making Incremental Backups

---

- Use **LEVEL** parameter to specify the incremental level:

```
BACKUP INCREMENTAL LEVEL 0 DATABASE;
```

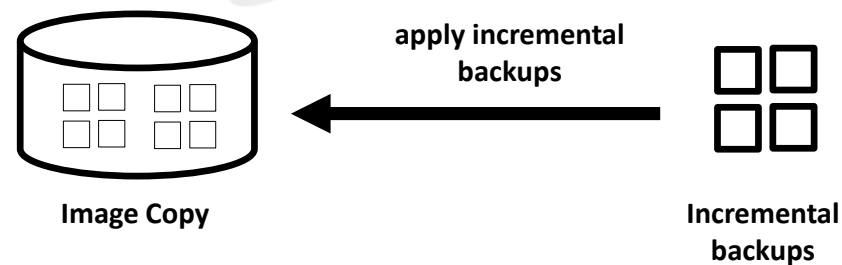
```
BACKUP  
INCREMENTAL LEVEL 1 CUMULATIVE  
TABLESPACE users;
```

- Differential is the default.
- When creating a level 1 incremental backup, if level 0 is not available, RMAN creates level 0 backup instead.
- This incremental backup format is applicable on backup sets. For image copies, there is another format.

# Incrementally Updated Backups

---

- Incremental backups are applied on image copies
- Benefits:
  - Less recovery time than image copies only
  - Less backup time than traditional full image copy
  - Restart a failed recovery process from the right file



# Making Incrementally Updated Backups

---

- **One time:** Level 0 copy image with specific tag
- **Periodic:** Level 1 differential backups are created with the same tag using **BACKUP FOR RECOVER OF COPY**
- **Periodic:** incremental backups are applied to the level 0 data file copy



# Incrementally Updated Backups: Example

---

- If you execute the following commands daily:

```
RECOVER COPY OF DATABASE WITH TAG 'incr_update';  
BACKUP INCREMENTAL LEVEL 1  
FOR RECOVER OF COPY WITH TAG 'incr_update'  
DATABASE;
```

- ... this is the result:

	RECOVER	BACKUP
Day 1		Creates image copies
Day 2		Create incremental level 1
Day 3 and onward	Roll forward (recover) image copies	Create incremental level 1

# About Block Change Tracking (BCT)

---

- BCT keeps track of modified blocks in a BCT file
- Makes creating incremental backups much quicker
- Only eight incremental backups can be optimized
- BCT file is a small binary file, even for high transactions/s databases
- BCT file size increases by 10 MB
- Recommended for databases with changes less than 20%
- Automatically maintained by Oracle database
- By default it is disabled

# Enabling/Disabling Block Change Tracking File

---

- Default location is in `DB_CREATE_FILE_DEST`
- Enable or disable with:

```
ALTER DATABASE  
{ENABLE|DISABLE} BLOCK CHANGE TRACKING  
[USING FILE '...']
```

- Rename block change tracking file with the **ALTER DATABASE RENAME** statement (database must be in MOUNT state).
- RMAN does not support backing up the BCT file

# Monitoring Block Change Tracking

---

- Obtain BCT information:

```
SELECT FILENAME, STATUS, BYTES FROM V$BLOCK_CHANGE_TRACKING;
```

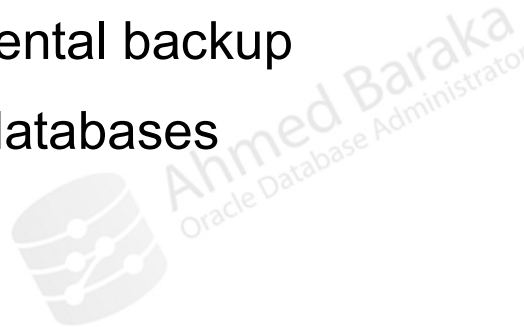
- Measure how BCT is effective:

```
SELECT FILE#, AVG(DATAFILE_BLOCKS),  
       AVG(BLOCKS_READ), AVG(BLOCKS_READ/DATAFILE_BLOCKS)  
       * 100 AS PCT_READ_FOR_BACKUP, AVG(BLOCKS)  
FROM V$BACKUP_DATAFILE  
WHERE USED_CHANGE_TRACKING = 'YES' AND INCREMENTAL_LEVEL > 0  
GROUP BY FILE#;
```

# Incremental Backup Common Practice

---

- Full level zero incremental backup in a weekend
- Level 1 incremental backup every day over night or in off-peak time
- Use differential incremental backup
- Enable BCT for large databases





# Summary

---

In this lecture, you should have learnt how to perform the following:

- Describe the difference between differential and cumulative incremental backups
- Take incremental backup
- Implement the incrementally updated backups
- Configure the Block Change Tracking (BCT)