Datenbankadministration

5. Backup & Recovery

AG DBIS University of Kaiserslautern, Germany

Karsten Schmidt kschmidt@informatik.uni-kl.de
(Vorlage TU-Dresden)

Wintersemester 2008/2009

Backup & Recovery

- **Wiederherstellung** (*recovery*)
  - Wiederherstellen der Datenbank nach Fehlerfall
    - Systemausfall
    - Transaktionsfehler
    - Medienfehler
    - Havarie
  - `RESTART`- und `RESTORE`-Kommando
  - erfordert ggf. Backup & Protokollierung
- **Protokollierung** (*logging*)
  - Protokollierung aller Änderungen an der Datenbank
  - ermöglicht UNDO und REDO
- **Sicherung** (*backup*)
  - Sicherung der Datenbank für späteres Wiederherstellen
  - `BACKUP`-Kommando

Wiederholung: Transaktionen

- **Transaktionen**
  - Kapselung mehrerer Datenbankoperationen
  - u.a. Alles-oder-Nichts-Prinzip
  - Abschluß mit `COMMIT` oder `ROLLBACK`
  - Achtung: automatisches Commit ausschalten
    - `UPDATE COMMAND OPTIONS USING C OFF`
- **Beispiel**
  - Überweisung
    - ```
      UPDATE PERS SET BALANCE = BALANCE - 100 WHERE ID = 1
      UPDATE PERS SET BALANCE = BALANCE + 100 WHERE ID = 2
      COMMIT
      ```
      ```

Arten von Recovery

- **Crash Recovery**
  - Wiederherstellung nach Systemausfall (*crash*)
    - nicht beendete Transaktionen rücksetzen (*undo*, *rollback*)
    - nicht vollständig ausgeschriebene Transaktionen ausschreiben
    - erfordert Protokolldatei
  - Datenbankneustart nach Systemausfall
    - `RESTART DATABASE <db-name>`
    - Parameter `AUTORESTART` in Datenbankkonfiguration
  - Beispiel:

    ```
    UPDATE PERS SET BALANCE = BALANCE - 100 WHERE ID = 1
    UPDATE PERS SET BALANCE = BALANCE + 100 WHERE ID = 2
    → CRASH
    COMMIT
    ```
  - Problem: keine Lösung für Medienfehler oder Havarie
    - Backup notwendig

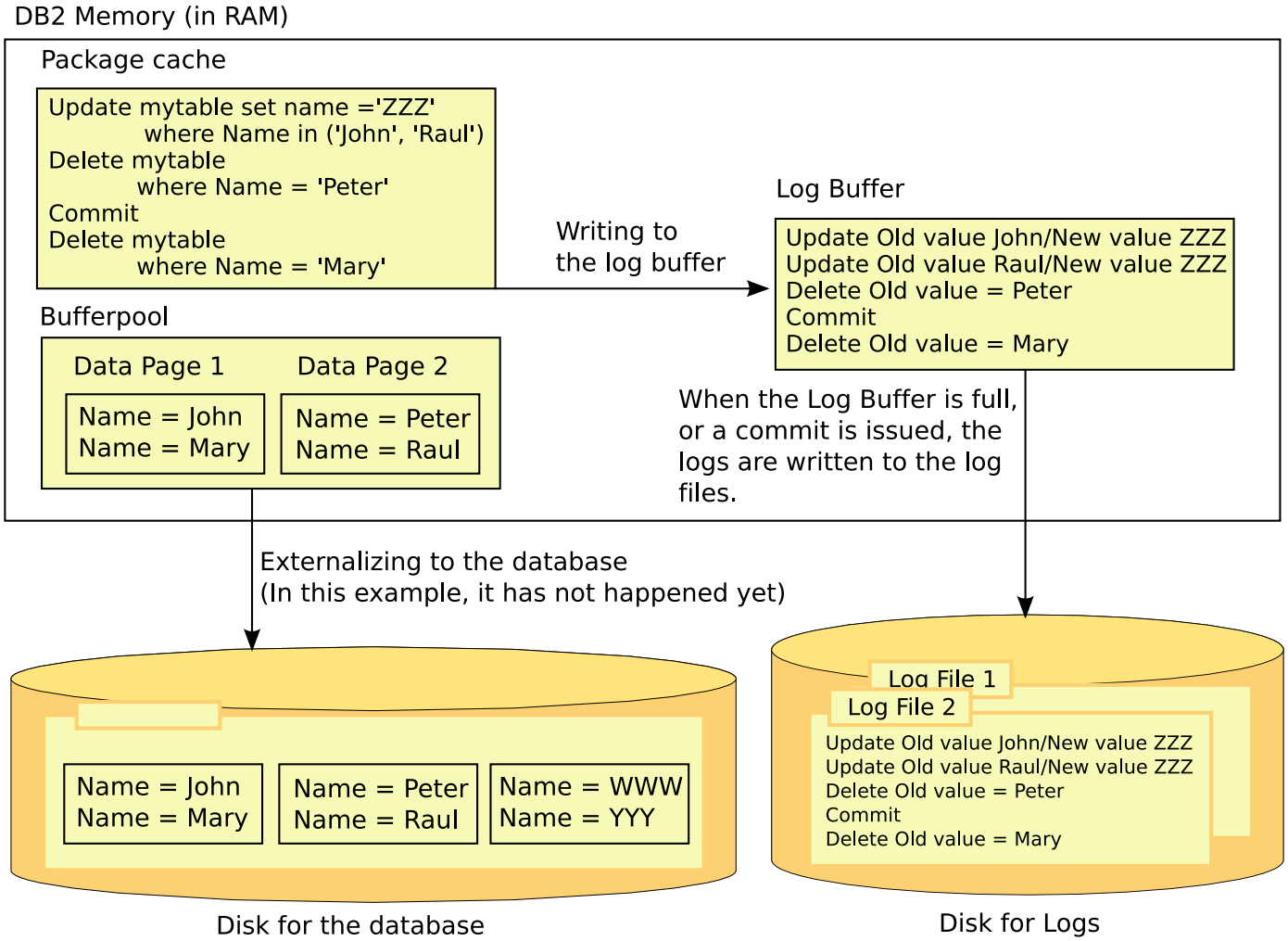
Arten von Recovery

- **Version Recovery**
  - Wiederherstellung eines gespeicherten Datenbankzustandes
    - erfordert vorheriges Backup
    - neuere Änderungen gehen verloren
- **Rollforward Recovery**
  - wie Version Recovery, aber zusätzliche Verwendung von Protokolldateien
    - Basis: Backup der Datenbank
    - nachträgliche Änderungen anhand von Prokolldateien nachvollzogen (Archivprotokollierung notwendig)
  - Wiederherstellung des Zustands der DB zu einem bestimmten Zeitpunkt möglich

Protokollierung

Protokollierung

- **Transaktionsprotokolle** (*transaction logs*)
  - halten Änderungen an Datenbankobjekten und Daten fest
  - WAL-Prinzip (write-ahead-logging)



Protokollarten

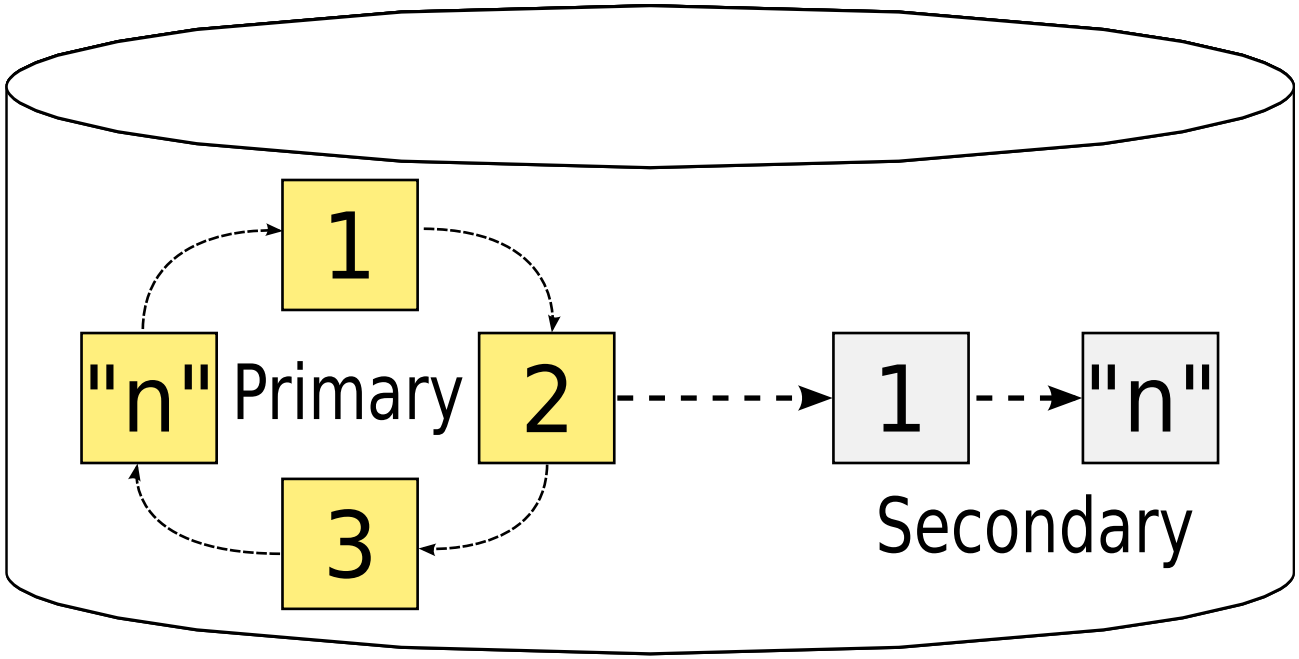
- **Protokollarten**
  - aktive Protokolle (*active logs*)
    - nicht abgeschlossene Transaktionen (`COMMIT, ROLLBACK`)
    - abgeschlossene, aber nicht festgeschriebene Transaktionen
    - benötigt für Crash Recovery
  - Archivprotokolle (*archive logs*)
    - abgeschlossene, festgeschriebene Transaktionen
    - 2 Arten
      - online
        - für DB2 zugreifbar
        - liegen im selben Verzeichnis wie aktive Protokolle
      - offline
        - durch System oder Nutzer verschoben
  - benötigt für Rollforward Recovery

Arten von Protokolldateien

- **Arten von Protokolldateien**
  - primäre Protokolldateien
    - Speicherplatz vorallokiert (max. 256GB)
    - Parameter `LOGPRIMARY` in Datenbankkonfiguration
  - sekundäre Protokolldateien
    - Speicherplatz dynamisch allokiert
    - Parameter `LOGSECOND` in Datenbankkonfiguration
  - weitere Parameter
    - `LOGFILSZ`: Größe einer Protokolldatei (in 4KB)
    - `NEWLOGPATH`: Verändern des Speicherortes
  - Überlauf aller Protokolldateien $\longrightarrow$ Rücksetzen der Transaktion

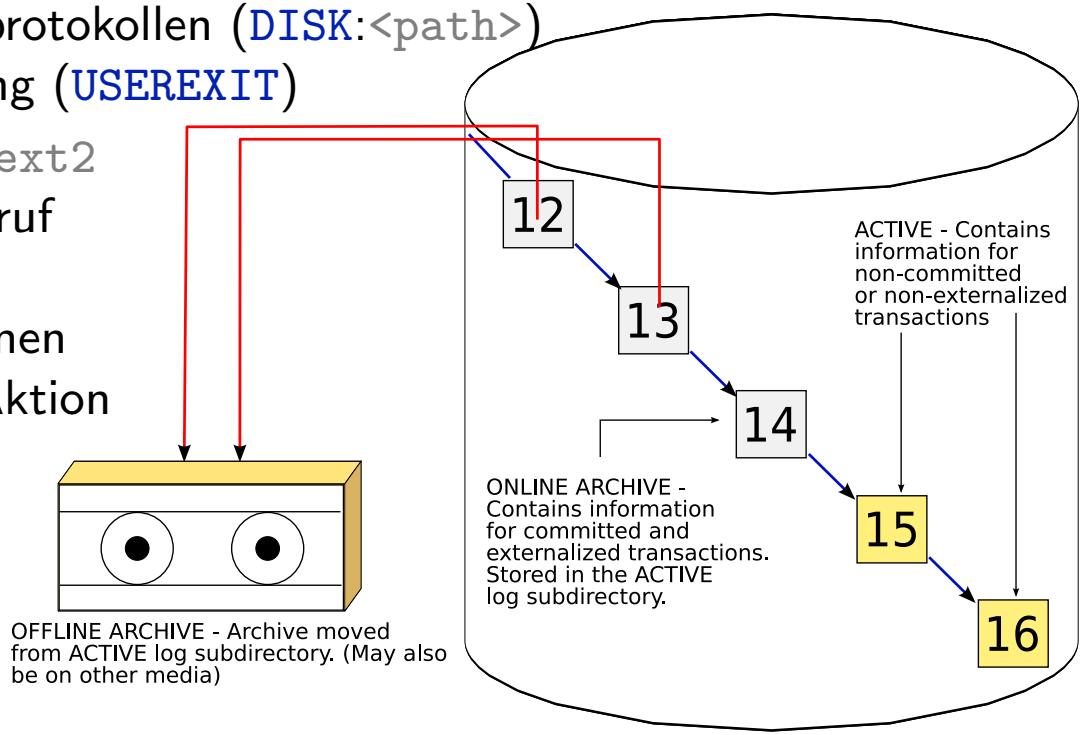
Protokollierungsstrategien

- **Umlaufprotokollierung** (*circular logging*)
  - Überschreiben nicht mehr verwendeter Protokolldateien
  - Ringpuffer
  - nur Crash Recovery (und Version Recovery) möglich



Protokollierungsstrategien

- **Archivprotokollierung** (*archival logging*)
  - kein Überschreiben von Protokolldateien
  - gesteuert durch `LOGARCHMETH1`, `LOGARCHMETH2`
  - Umschalten erfordert Backup
- **Parameter `LOGARCHMETH1` der Datenbankkonfiguration**
  - Umlaufprotokollierung (`OFF`)
  - Behalten von Archivprotokollen (`LOGRETAIN`)
  - Verschieben von Archivprotokollen (`DISK:<path>`)
  - nutzerdefinierte Sicherung (`USEREXIT`)
    - `sqllib/bin/db2uext2`
    - automatischer Aufruf durch DB2
    - Parameter bestimmen durchzuführende Aktion



Protokollierungsstrategien

- **Unbegrenzte Protokollierung** (*infinite logging*)
  - Archivierung von Protokolldateien
    - sobald diese voll sind
    - unabhängig davon, ob Transaktionen beendet oder festgeschrieben sind
  - Ergebnis
    - aktive Protokolle laufen nie über
    - kein Abbrechen von Transaktionen aufgrund voller Protokolldateien
  - **nicht** empfohlen (Crash Recovery kann sehr lang dauern)
- **Parameter**
  - Archivprotokollierung aktivieren
  - `LOGSECOND` auf `-1` setzen

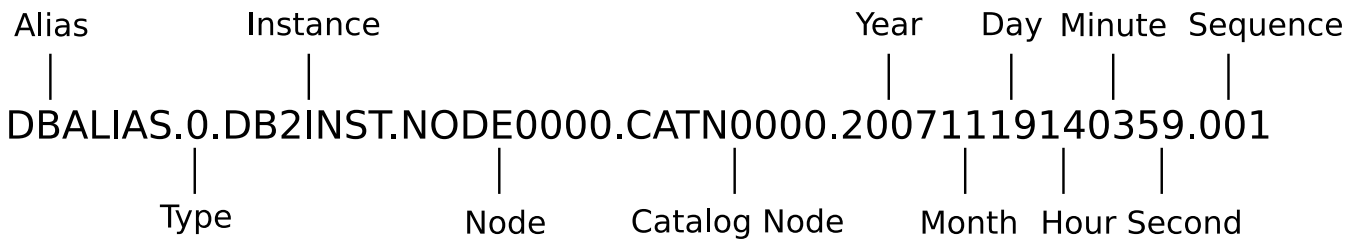
Backup

Datenbank-Backup

- **Datenbank-Backup**
  - Sicherung einer Datenbank
    - Daten
    - Informationen über Tabellenbereiche / Container
    - Datenbankkonfigurataion
    - aber: nicht Instanzparameter und Registrierung
  - 2 Arten
    - offline
      - Datenbank wird gesperrt
    - online
      - Zugriff auf Datenbank weiterhin möglich
      - erfordert Archivprotokollierung
- **erforderliche Rechte**: `SYSADM, SYSMAINT, SYSCTRL`

Vollständiges Backup

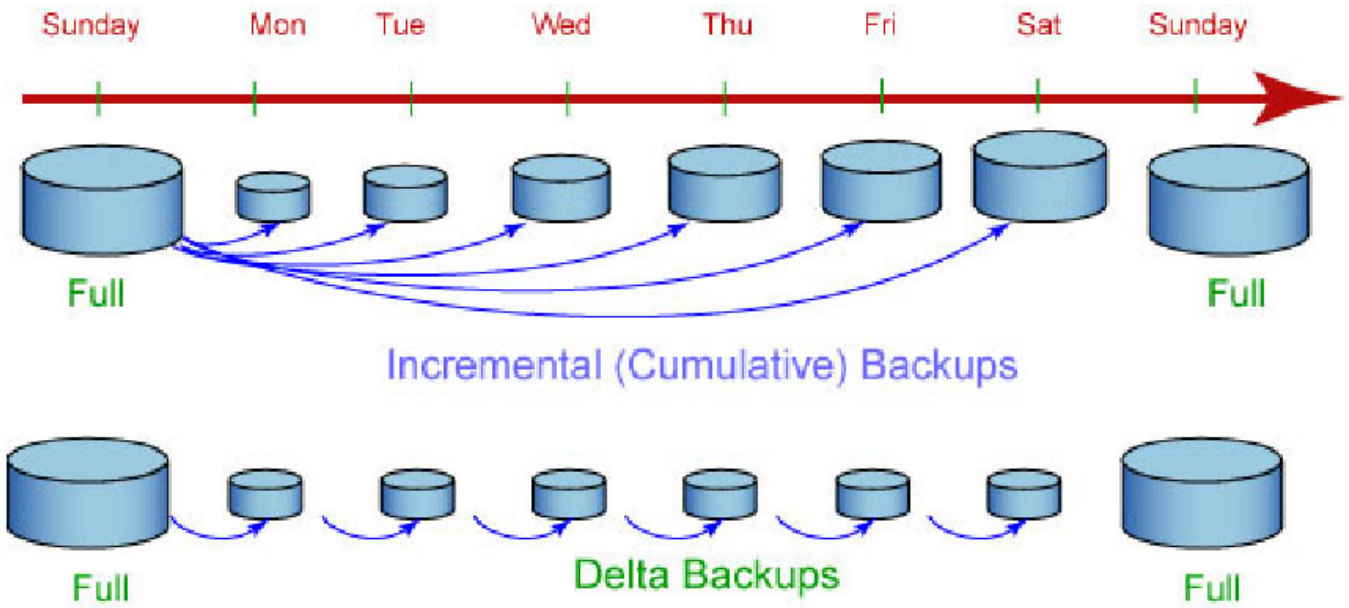
- **Vollständiges Backup** (*full backup*)
  - ermöglicht Version Recovery
  - `BACKUP DATABASE <db-name> [ONLINE]`
    `TO {<directory> | <device>}`
    `[INCLUDE LOGS]`
    `[COMPRESS]`
  - Namenskonventionen



  - Information über Backup-Dateien: `db2ckbkp <name>`
  - Beispiel
    - `BACKUP DATABASE TPCH`
      `TO /home/db2inst1/db2backup`
    - `TPCH.0.db2inst1.NODE0000.CATN0000.20071119140359.001`

Aktualisierendes Backup

- **Aktualisierendes Backup**
  - kumulative Sicherung (*incremental backup*)
    - sichert alle Veränderungen seit letztem vollständigen Backup
  - nicht-kumulative Sicherung (*delta backup*)
    - sichert alle Veränderungen seit letztem vollständigen, kumulativen oder nicht-kumulativen Backup



Aktualisierendes Backup

- **Aktualisierendes Backup in DB2**
  - Datenbankparameter `TRACKMOD` muss auf `ON` gesetzt sein
  - `BACKUP DATABASE <db-name> [ONLINE] [INCREMENTAL [DELTA]]`
    ...
  - Beispiel
    - `UPDATE DB CFG USING TRACKMOD ON`
    - `BACKUP DATABASE TPCH`
      `TO /home/db2inst1/db2backup`
    - ...
    - `BACKUP DATABASE TPCH INCREMENTAL`
      `TO /home/db2inst1/db2backup`

Tabellenbereichs Backup

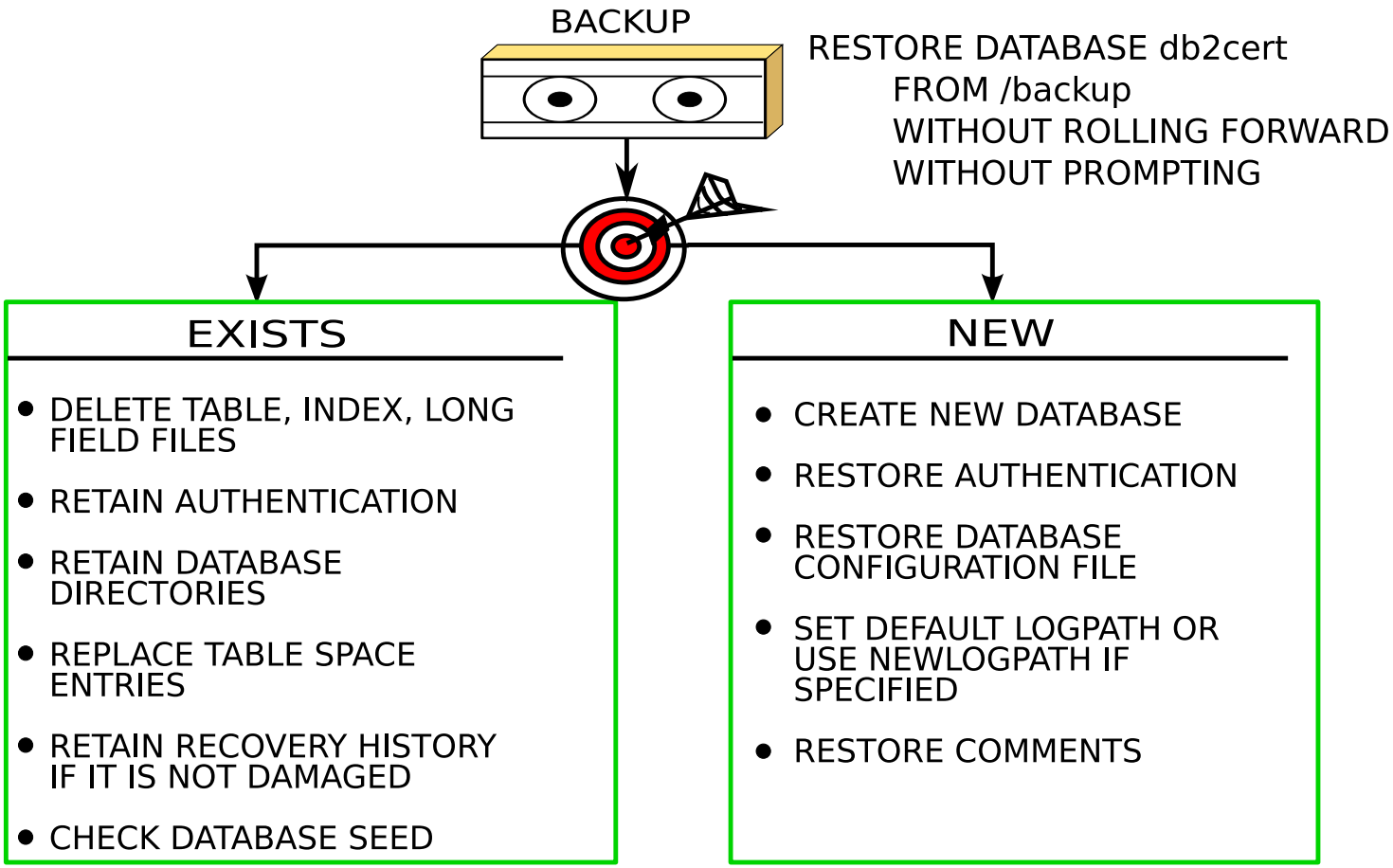
- **Tabellenbereichs Backup**
  - Archivprotokollierung notwendig
  - `BACKUP DATABASE TPCH`
    `TABLESPACE (SYSCATSPACE, USERSPACE1)`
    `TO /home/db2inst1/db2backup`
  - `TPCH.3.db2inst1.NODE0000.CATN0000.20071119140359.001`

    3 = Tablespace Backup
  - kein Backup temporärer Tabellenbereiche
  - Tabellenbereiche mit zusammenhängenden Daten gemeinsam sichern
    - Daten, Indizes, LOBs
    - Tabellen mit Fremdschlüsselbeziehungen

Recovery

Datenbankwiederherstellung

- **Version Recovery mit vollständigem Backup**
  - vollständige Wiederherstellung des gesicherten Zustands
  - Überschreiben der existierenden bzw. Anlegen einer neuen Datenbank



Datenbankwiederherstellung

- **Version Recovery mit vollständigem Backup**
  - nur offline möglich
  - `RESTORE DATABASE <dbname>`
    `FROM {<directory>|<device>}`
    `TAKEN AT <timestamp>`
    `[INTO <dbname>]`
    `[REPLACE EXISTING]`
  - Beispiel
    `RESTORE DATABASE TPCH`
    `FROM /home/db2inst1/db2backup`
    `TAKEN AT 20071119140359`
- **erforderliche Rechte**: `SYSADM, SYSCTRL, (SYSMAINT)`

Datenbankwiederherstellung

- **Version Recovery mit aktualisierenden Backups**
  - benötigt (ausgehend vom Zielzeitpunkt)
    - volles Backup
    - letztes inkrementelles Backup
    - alle nachfolgenden Delta-Backups
  - `RESTORE DATABASE <dbname> INCREMENTAL [AUTOMATIC]`

    ...
    - automatisches Einspielen
    - sonst manuell: Reihenfolge letztes, erstes, zweites, ..., letztes

Tabellenwiederherstellung

- **Wiederherstellung von Tabellenbereichen**
  - von Datenbank- oder Tabellenbereichs-Backup
  - Wiederherstellung von Benutzertabellenbereichen auch online möglich
  - Archivprotokollierung notwendig
  - nach `RESTORE` immer im `ROLLFORWARD PENDING`-Zustand
  - `RESTORE DATABASE <dbname>`
    `TABLESPACE (<tablespacename>)`
    `FROM {<directory>|<device>}`

Wiederherstellung der Protokolldateien

- **Wiederherstellung der Protokolldateien**
  - Voraussetzung: Protokolldateien im Backup ([INCLUDE LOGS])
  - Option: LOGTARGET <targetpath>
  - mit Option LOGS lassen sich auch ausschließlich die Protokolldateien wiederherstellen
    - RESTORE DATABASE <dbname>
      LOGS FROM <backuppath>
      LOGTARGET <targetpath>

Umgeleitete Wiederherstellung

- **Umgeleitete Wiederherstellung** *redirected recovery*
  - erforderlich wenn:
    - Container, von dem Backup gemacht wurde, nicht mehr verfügbar
    - Wiederherstellung der Datenbank auf anderem System mit anderer Konfiguration
- **Ablauf**
  1. Information über Container und Tabellenbereiche aus Backup sammeln
     - `RESTORE DATABASE <dbname>`
       `FROM {<directory>|<device>} INTO <dbname>`
       `REDIRECT`
       `WITHOUT ROLLING FORWARD`
  2. Informationen ansehen
     - `LIST TABLESPACES SHOW DETAIL`

Umgeleitete Wiederherstellung

- **Ablauf (Fortsetzung)**
  1. Container für Tabellenbereiche setzen
     - `SET TABLESPACE CONTAINERS FOR <tbspc_id> USING (...)`
  2. Wiederherstellung der Daten starten
     - `RESTORE DATABASE <dbname> CONTINUE`
- **Einschränkungen**
  - keine Änderung des Tabellenbereichstyp (SMS/DMS)
  - keine Änderungen an *automatic storage* Tabellenbereichen

ROLLFORWARD

Rollforward Recovery

- **Rollforward Recovery einer Datenbank**
  - erlaubt Wiederherstellung des Datenbankzustands eines angegebenen Zeitpunktes
  - nur offline möglich
  - `ROLLFORWARD DATABASE <db-name>`
    `TO {<time> [USING LOCAL TIME] | END OF LOGS}`
    `[AND COMPLETE]`
  - Wiederholen aller bis `<time>` abgeschlossenen Transaktionen
  - muss abgeschlossen werden
    - `ROLLFORWARD DATABASE <db-name> COMPLETE`
- **erforderliche Rechte**: `SYSADM, SYSCTRL, SYSMAINT`

Rollforward Recovery

- **Rollforward Recovery eines Tabellenbereichs**
  - für Benutzertabellenbereiche auch online möglich
  - `ROLLFORWARD DATABASE <db-name>`
    `TO {<time> [USING LOCAL TIME] | END OF LOGS}`
    `[AND COMPLETE]`
    `TABLESPACE (<tablespacename>) [ONLINE]`
  - bei `ROLLFORWARD` zu angegebenen Zeitpunkt → Tabellenbereich in `BACKUP PENDING`-Zustand

Zielzeitpunkt der Wiederherstellung

- **Zielzeitpunkt der Wiederherstellung**
  - muss zwischen *minimum point in time* und *end of logs* liegen
  - **Achtung**: keine Garantie, dass Daten nach `ROLLFORWARD` in konsistentem Zustand
  - Setzen von Konsistenzpunkten mit Hilfe des `QUIESCE`-Kommandos
- ***minimum point in time***
  - sichert zu, dass Tabellenbereiche und Protokolle konsistent mit Systemkatalogen sind
  - aktualisiert, wenn DDL gegen Tabellenbereich oder Tabellen im Tabellenbereich
  - `LIST TABLESPACES SHOW DETAILS` oder
    `GET SNAPSHOT FOR TABLESPACE ON <dbname>`

Hochverfügbarkeit

Hochverfügbarkeit

- **Hochverfügbarkeit bei Backup und Wiederherstellung**
  - Backup von gesamter Datenbank schwer zu planen und lange Laufzeit → Backup einzelner Tabellenbereiche
  - Wiederherstellung (`RESTORE`) mit Option `REBUILD` (ab Version 9)
- **Wiederherstellung der vollständigen Datenbank**
  - Voraussetzung: Backups der Tabellenbereiche und Protokolle
  - Ablauf
    - `RESTORE DATABASE <dbname> REBUILD WITH ALL TABLESPACES IN DATABASE TAKEN AT <timestamp>` (`<timestamp>` von letztem Tabellenbereichs-Backup wählen)
    - `ROLLFORWARD` ...

Hochverfügbarkeit

- **Wiederherstellung der Datenbank mit einer Teilmenge der Tabellenbereiche**
  - Katalogtabellenbereich muss wiederhergestellt werden
  - `RESTORE DB <dbname> REBUILD WITH TABLESPACE (<tablespacename>) TAKEN AT <timestamp>`

    oder

    `RESTORE DB <dbname> REBUILD WITH ALL TABLESPACES IN DATABASE EXCEPT (<tablespacename>) TAKEN AT <timestamp>`
  - `ROLLFORWARD` ...
  - nicht wiederhergestellte Tabellenbereiche im `RESTORE PENDING`-Zustand
  - Datenbank ist (normal) verwendbar

Hochverfügbarkeit

- **Wiederherstellung aus Online-Backup mit Protokollen**
  - Protokolle extrahieren
    - `RESTORE DB <dbname> REBUILD WITH ALL TABLESPACES IN DATABASE TAKEN AT <timestamp> LOGTARGET <pathtologs>`
  - extrahierte Protokolle anwenden
    - `ROLLFORWARD DB <dbname> TO END OF LOGS OVERFLOW LOG PATH (<pathtologs>) [AND COMPLETE]`

Zusammenfassung

- **Protokollierung**
  - Umlauf- und Archivprotokollierung
- **Backup**
  - Datensicherung
  - online oder offline
  - vollständig oder aktualisierend
  - Datenbank oder einzelne Tabellenbereiche
- **Recovery**
  - Einspielen von Backups
  - Wiederherstellen mit Hilfe von Archivprotokollen
- **Hochverfügbarkeit**
  - Backup und Recovery einzelner Tabellenbereiche