Create a Simple Operational Data Store

Brian Hansen
@tf3604
tf3604.com
brian@tf3604.com

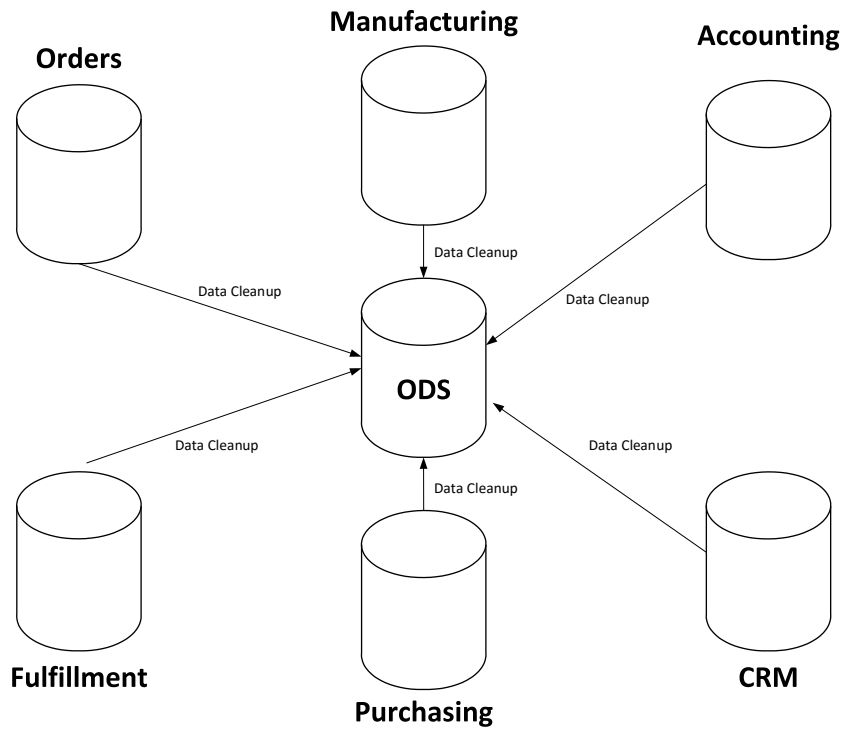
Agenda

- What is an operational data store?
- Why might I want an operational data store?
- The solution
  - Overview
  - About log shipping
  - The ODS scripts

Generalized Operational Data Store

- Accept multiple heterogeneous data sources
- All pulled into a central repository
- Little to no processing of data
  - For example, no denormalization, no aggregation, no cubing)
- Mostly raw data, and not a data warehouse

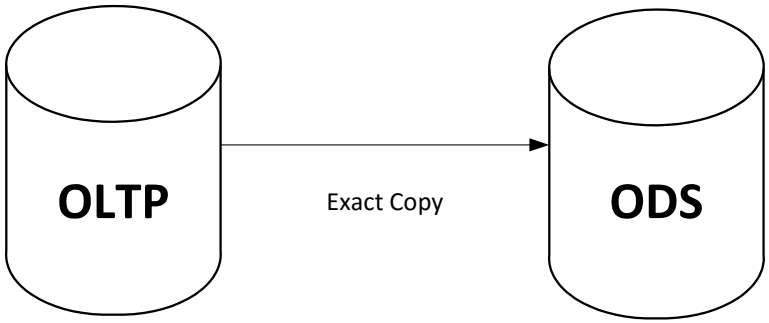
Generalized ODS – Graphical View



Reasons for Having an ODS

- Analysis of data from different systems
- Offload intensive processing
- Static, point-in-time view of data
- Analysis of recently changed data from production
- Launching point for a warehouse load process

Simplified ODS



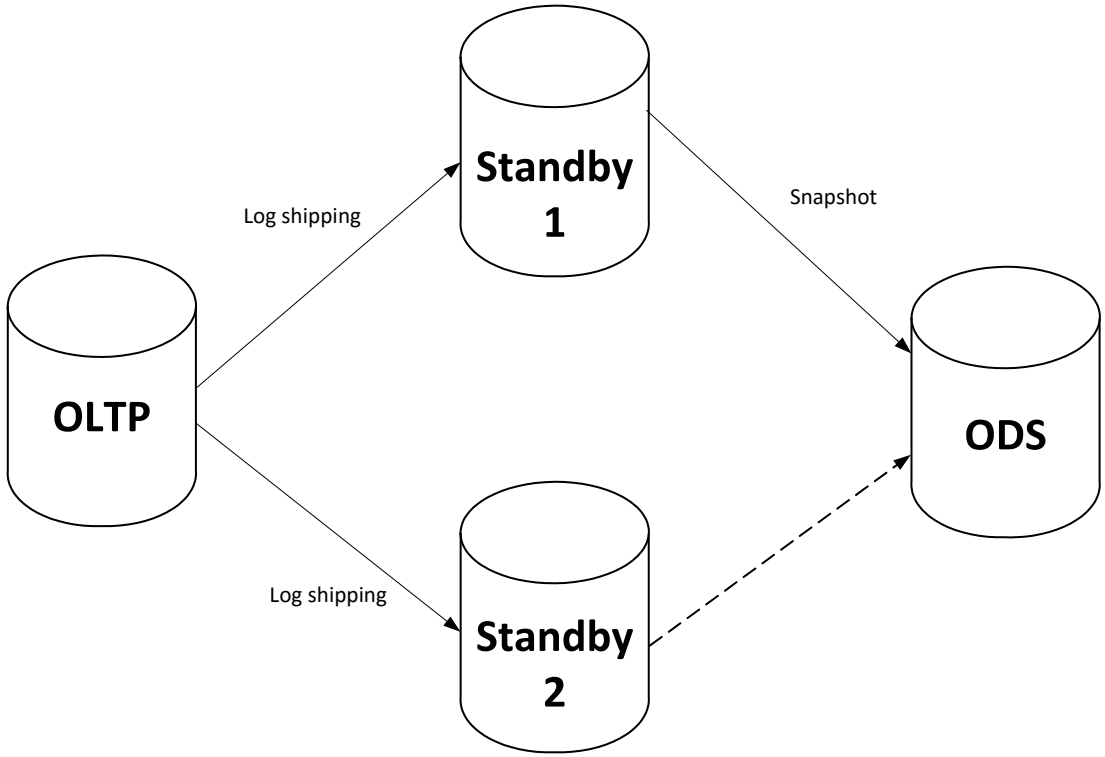
Demonstration

Report query blocking users

Other Ways to Accomplish This

- Replication
- Mirroring
- Availability Groups

General Concept of Our Simplified ODS



Requirements for These Scripts

- Same version of SQL on source and destination instances
- No requirement to change anything on the ODS for any reason (including security)
- Enough storage on destination for two copies of database
  - Plus storage for the backups
- Source database must in FULL/BULK-LOGGED recovery
- Backup chain must be unbroken
- It must be acceptable to users to query non-live data

Demonstration

- Updates to the transactional system
- Running a report query against the ODS

Log Shipping Basics

- Every individual operation that changes a database state is assigned a unique, ever-increasing ID
  - Not to be confused with a transaction. A transaction can consist of many such operations
- This ID is called a log sequence number (LSN)
- LSN can be represented in different ways:
  - 00000072:00000032:000b
  - 114000000005000011
- Fortunately, we are dealing with just the decimal representation
- The data associated with the change is written to the transaction log

Backups

- Many types of backups in SQL Server. The most common are
  - Full backup
  - Differential backup
  - Log backup
- With log shipping, we are mostly concerned with log backups
- Each log backup contains a Start LSN and an End LSN value
- Each log file has its Start LSN = End LSN of the previous backup in the chain
- We use this property to reconstruct the log chain

Restoring Databases from Log Backups

- When restoring a backup chain, use the WITH NORECOVERY option
- All LSN data is replayed to the database to the end of the log backup file
- This leaves the database in a transactionally inconsistent state
- No querying (even by admins) is allowed
- At the end of the restore process, use the WITH RECOVERY option to bring the database online
  - Transactions that are “in progress” at the point of restore are rolled back
- Once a database is recovered, no more backups can be restored

Standby Restore

- SQL Server offers the WITH STANDBY option as an alternative
- SQL will temporarily rollback uncommitted transactions
  - Stores these transactions in a special file
- Presents the database in read-only mode
  - SSMS will show database as Standby / Read-Only
- When the next log is restored, transaction from the temporary file are replayed, then log is restored as normal
- Disadvantage is a performance hit from extra rollback and replay operations
  - Uncommitted transactions may be large

Some Assumptions

- Single disk-based backup file per log backup
- Database consists on single data file and single log file
- Probably many more assumptions that aren’t accounted for in the scripts
- If these assumptions are not valid, can modify scripts to workaround

Script Walkthrough

- Prep scripts (T-SQL)
  - Create tables
  - Initial restores
  - Initial inserts
- SQL Job: ODS Restore
  - Get log information & copy (PowerShell)
  - Restore logs (T-SQL)
- SQL Job: ODS Switch
  - Switch active standby database (T-SQL)
  - Cleanup log files (PowerShell)

Benefits of the ODS

- ODS remains static; only changes on your schedule
- Time required to refresh is very short (usually a few seconds)
- Once secondary standby is caught up on logs, can switch again (usually a few minutes)
- Side benefit: Exercise <u>every</u> transaction log backup (validate backup chain)

Gotchas to Watch out For

- Views, stored procedures, functions, etc., that reference the database name:

```
create view vwProduct
as
select * from CorpDB.dbo.Product;
```

Coda

- What happens when versions of SQL are different?
  - WITH STANDBY will fail because SQL needs to upgrade the internal database version in order to present the DB
  - Can still use similar process, but with a full restore at each ODS refresh
    - More resource-intensive
    - Cannot refresh as frequently
    - Not exercising all transaction log backups
    - However, get more flexibility because database can be modified once recovered
    - Be sure to manually set to read-only

Similar Systems

- Aaron Bertrand: Readable Secondaries on a Budget (10/15/2014) http://sqlperformance.com/2014/10/sql-performance/readable-secondaries-on-a-budget
- Kendra Little: Reporting From a Log Shipping Secondary in STANDBY mode (1/28/2015) http://www.brentozar.com/archive/2015/01/reporting-log-shipping-secondary-standby-mode/

The End

- Current version of the ODS scripts can be found at

  www.tf3604.com/ods

- Contact information

@tf3604

tf3604.com

brian@tf3604.com