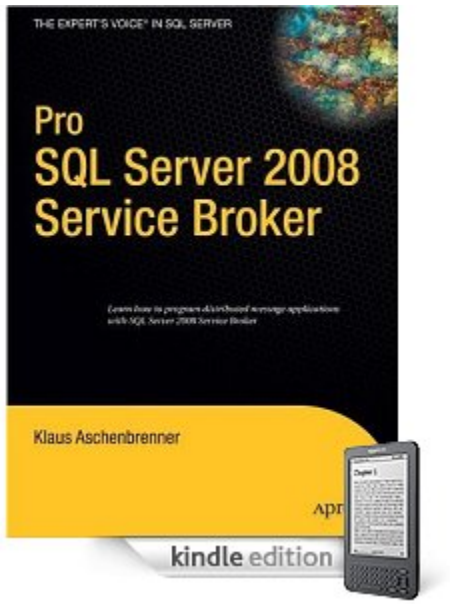
Designing & Maintaining VLDBs on SQL Server

Klaus Aschenbrenner
Microsoft Certified Master SQL Server 2008
www.SQLpassion.at
Twitter: @Aschenbrenner

Bring your SQL Server installations to a new level of excellence!

About me

- Independent SQL Server Consultant
- International Speaker, Author
- SQL Server 2008 MCM
- „Pro SQL Server 2008 Service Broker“
- www.SQLpassion.at
- Twitter: @Aschenbrenner
- SQLpassion Academy
  - http://www.SQLpassion.at/academy



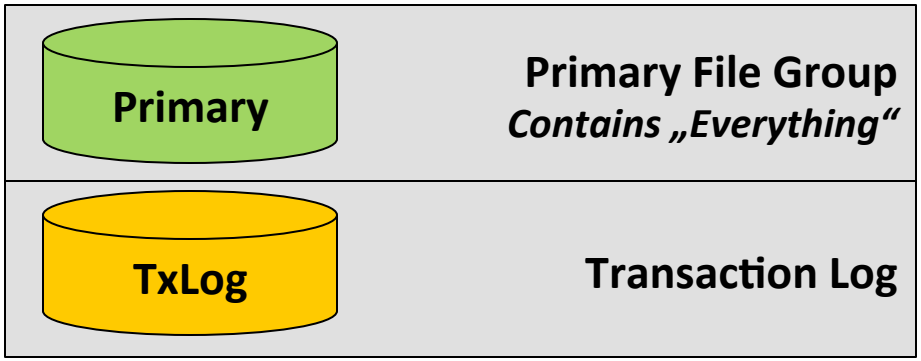
Agenda

- Physical Database Design
- Partitioning

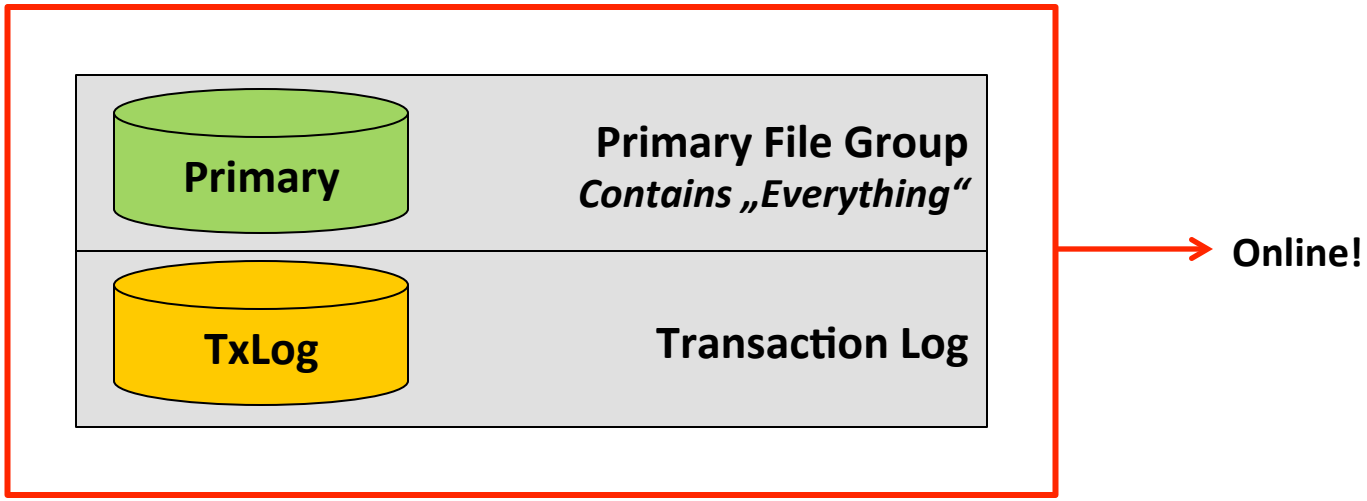
Agenda

- Physical Database Design
- Partitioning

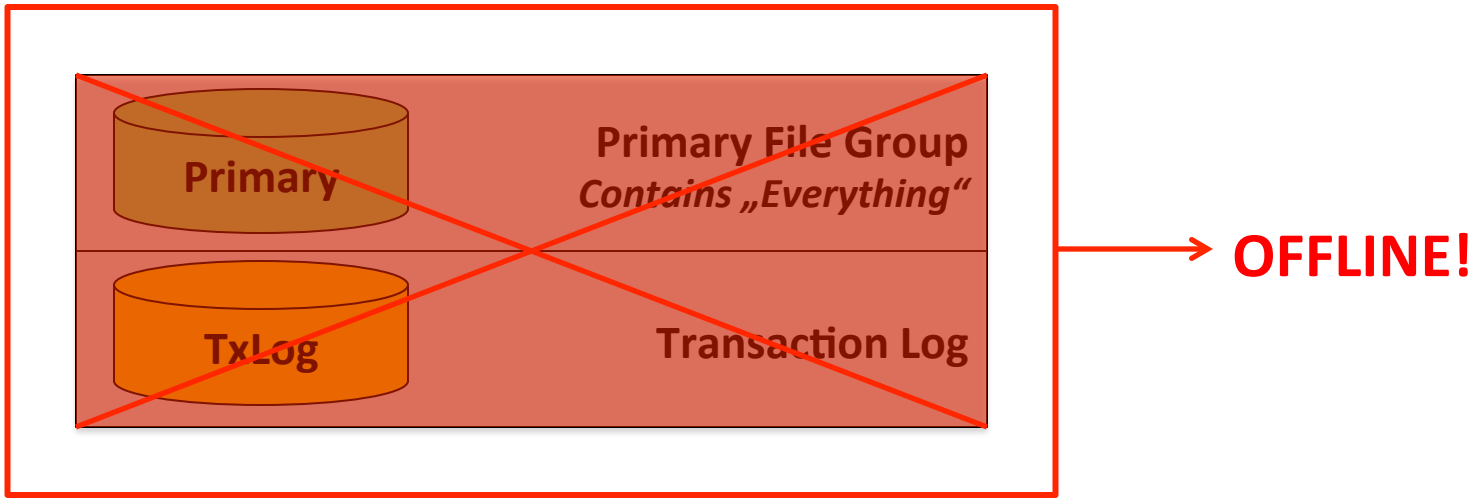
Default Database Structure



Default Database Structure



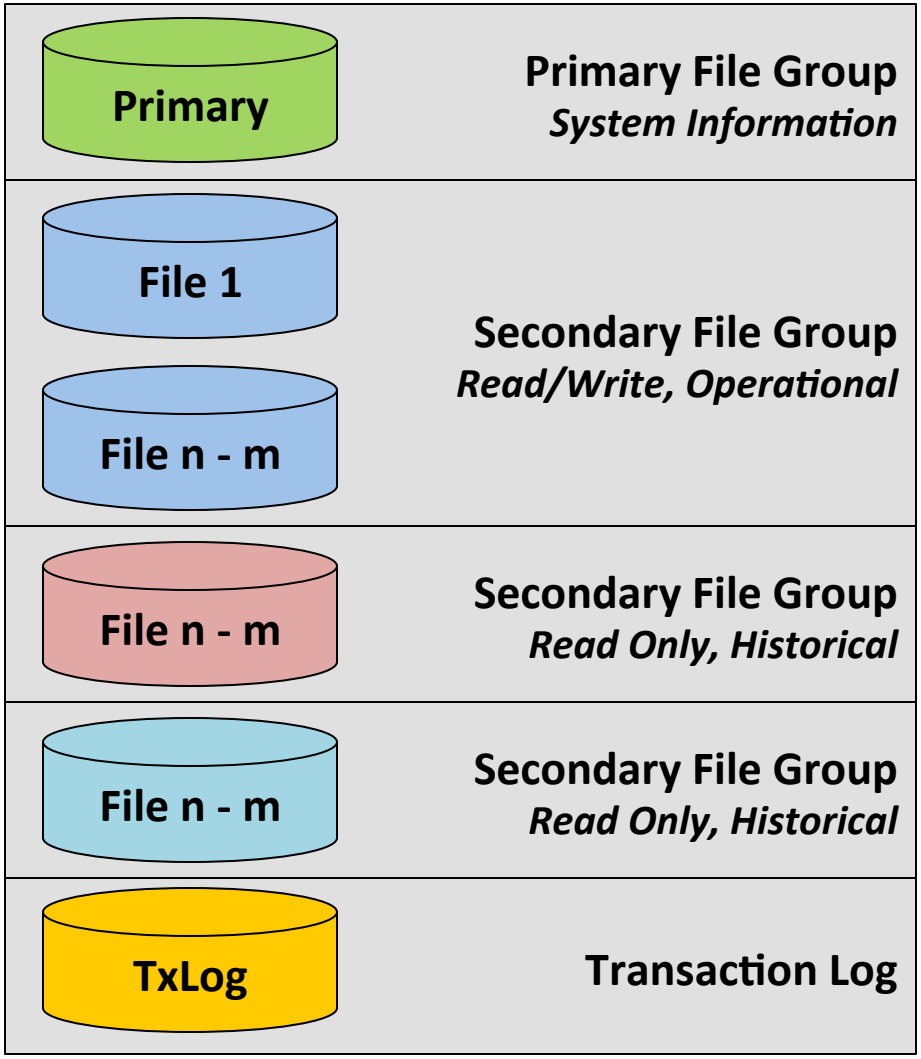
File Group Failure



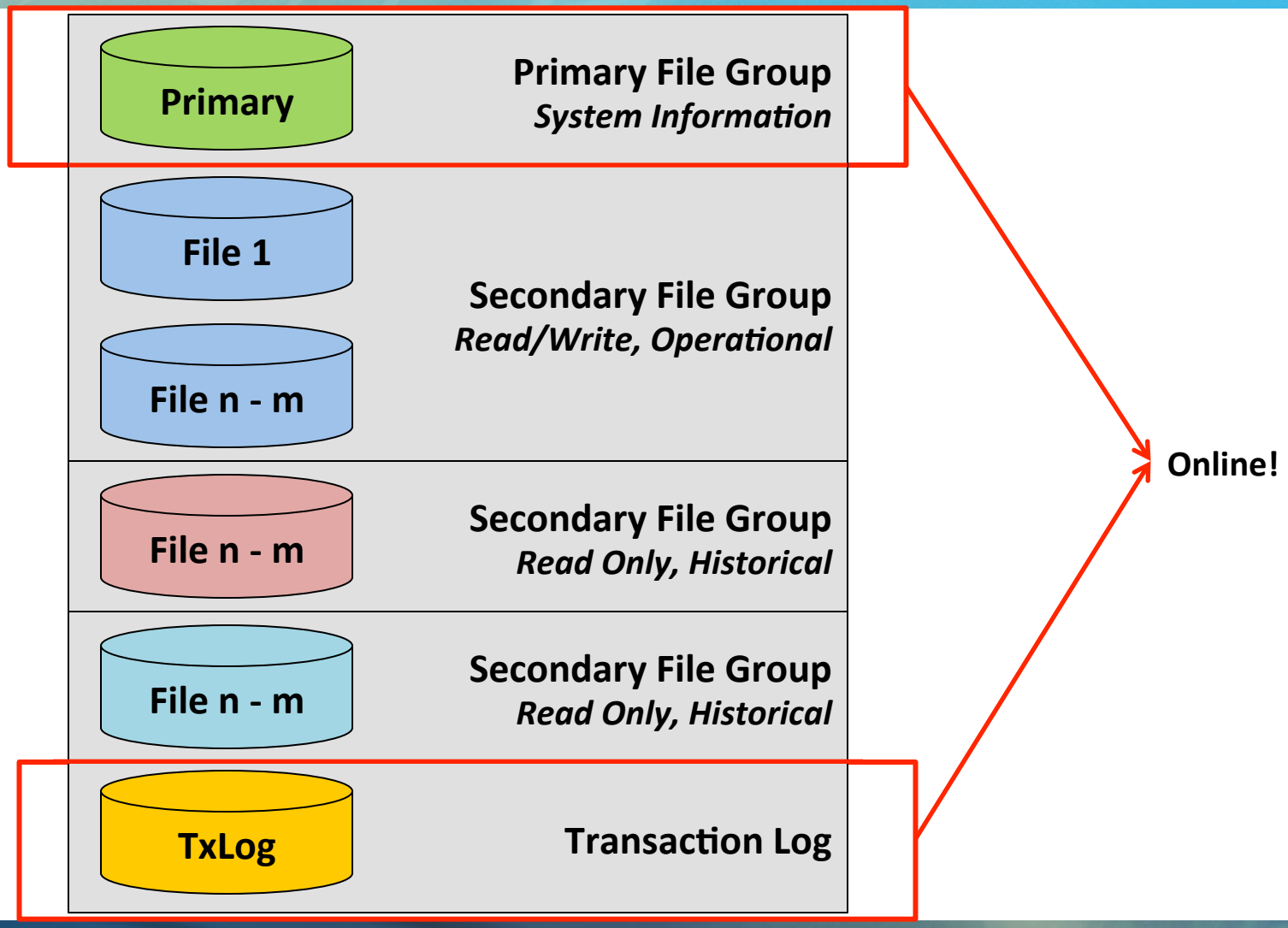
Disadvantages

- Primary File Group and TxLog are critical
  - Primary contains System Tables
  - TxLog contains Transactions
- If Primary File Group is damaged the whole database is offline
  - You have to restore the whole database
  - This could take *a lot* of time!
- Are there other options...?

Database Structure for VLDBs



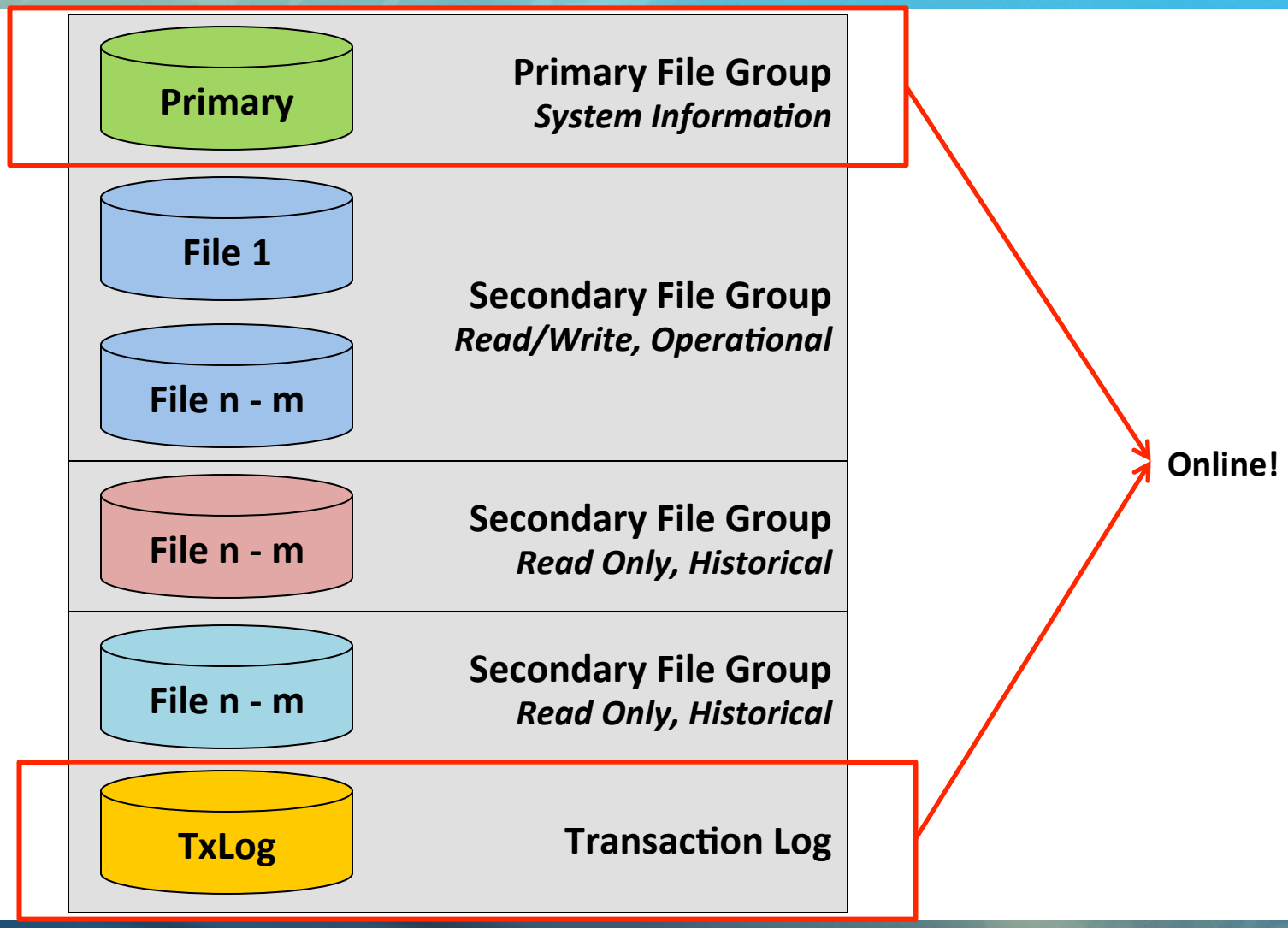
Database Structure for VLDBs



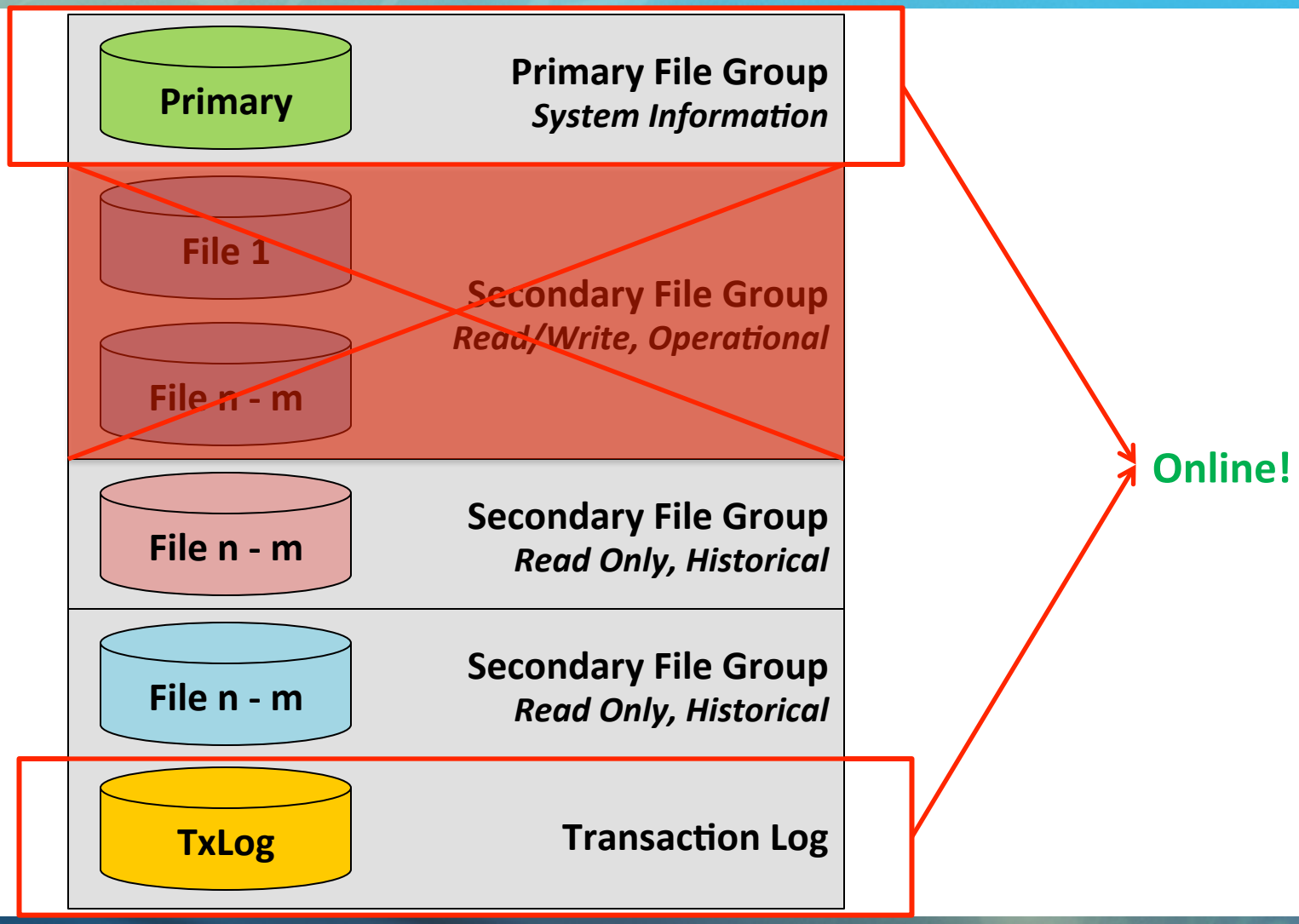
Advantages

- Primary File Group is isolated and contains no transactional data
  - Contains only system information
  - Database is very fast online!
- Transactional data is separated
  - Operational data
  - Historical data
- Improves
  - Database Availability
  - Recovery Time
  - Recovery Options/Paths

File Group Failure



File Group Failure



Damaged File Groups

- When a File Group is damaged
  - Affects only the data of that File Group
  - Data from other File Groups can be still accessed
- Database remains online
  - System Information is isolated in Primary File Group
- When Restore Sequence for the damaged File Group begins
  - On Enterprise Edition
    - Remaining File Groups stay ONLINE
  - On Standard Edition
    - Database is offline until the Restore Sequence has completed

Other Benefits of Isolation

- Different RAID Levels for different File Groups
- Primary File Group/Operational Data/TxLog
  - RAID 10 Triple Mirrored
- Secondary File Group of 2013 Historical Data
  - RAID 10
- Secondary File Group of 2010 Historical Data
  - RAID 5
- Etc...

Agenda

- Physical Database Design
- Partitioning

Why Partitioning?

- Large Tables
- Maintenance
- Availability
- Reduce Locking/Blocking
- Data Archival

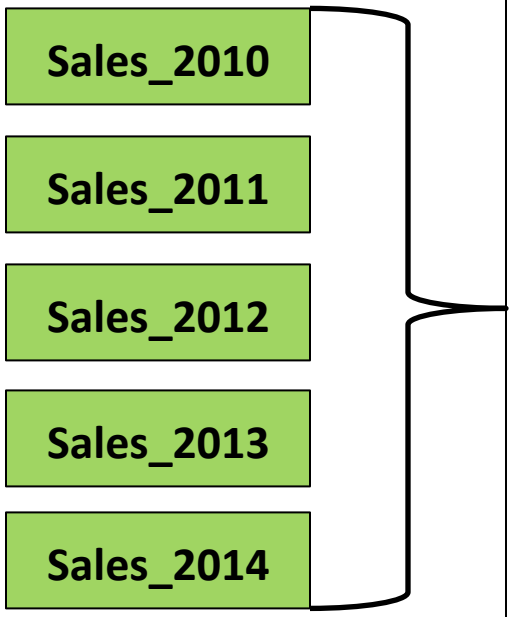
Problems with a single (large) Table

- Management
- Index Creation/Rebuilds
- Backup/Restore
- Lock Escalation
- Different Access Pattern
  - Operational Data
    - INSERT, UPDATE
  - Historical Data
    - SELECT
- How to delete *efficient* a large amount of data?

History of Partitioning

- SQL Server 7.0+
  - Proportional Fill for File Groups with more than one file
  - Partitioned Views
- SQL Server 2000+
  - Updateable Partitioned Views
    - Lot of Limitations
- SQL Server 2005+
  - Partitioned Tables
    - More manageable
    - LOT of disadvantages!

Partitioned Views

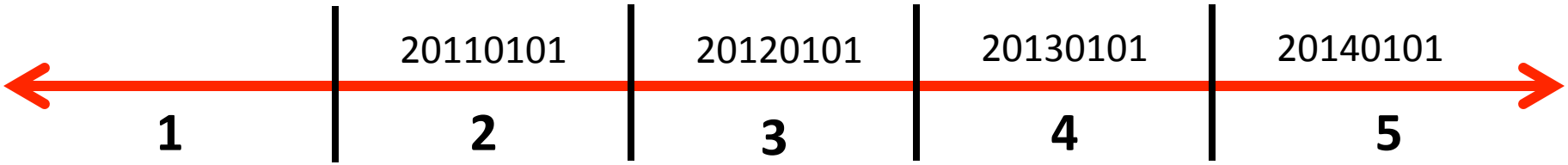



```
CREATE VIEW SalesPV
AS
BEGIN
    SELECT * FROM Sales_2010
    UNION ALL
    SELECT * FROM Sales_2011
    UNION ALL
    SELECT * FROM Sales_2012
    UNION ALL
    SELECT * FROM Sales_2013
    UNION ALL
    SELECT * FROM Sales_2014
END
GO
```

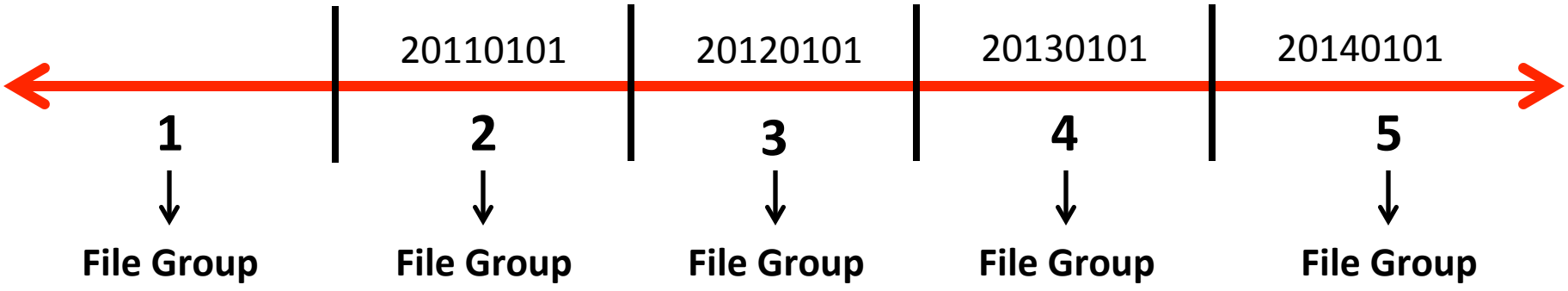
Pros/Cons

- Pros
  - Available on any Edition of SQL Server
  - Each Table has its own *better* Statistics
  - Index Rebuild of any table is an ONLINE operation
    - When using Enterprise Edition...
  - Each table can be indexed independently
    - Operational Data
    - Historical Data
- Cons
  - Lots of tables to administer
  - Must create indexes on all tables
  - Check Constraints are needed
  - Gaps and overlapping values are possible

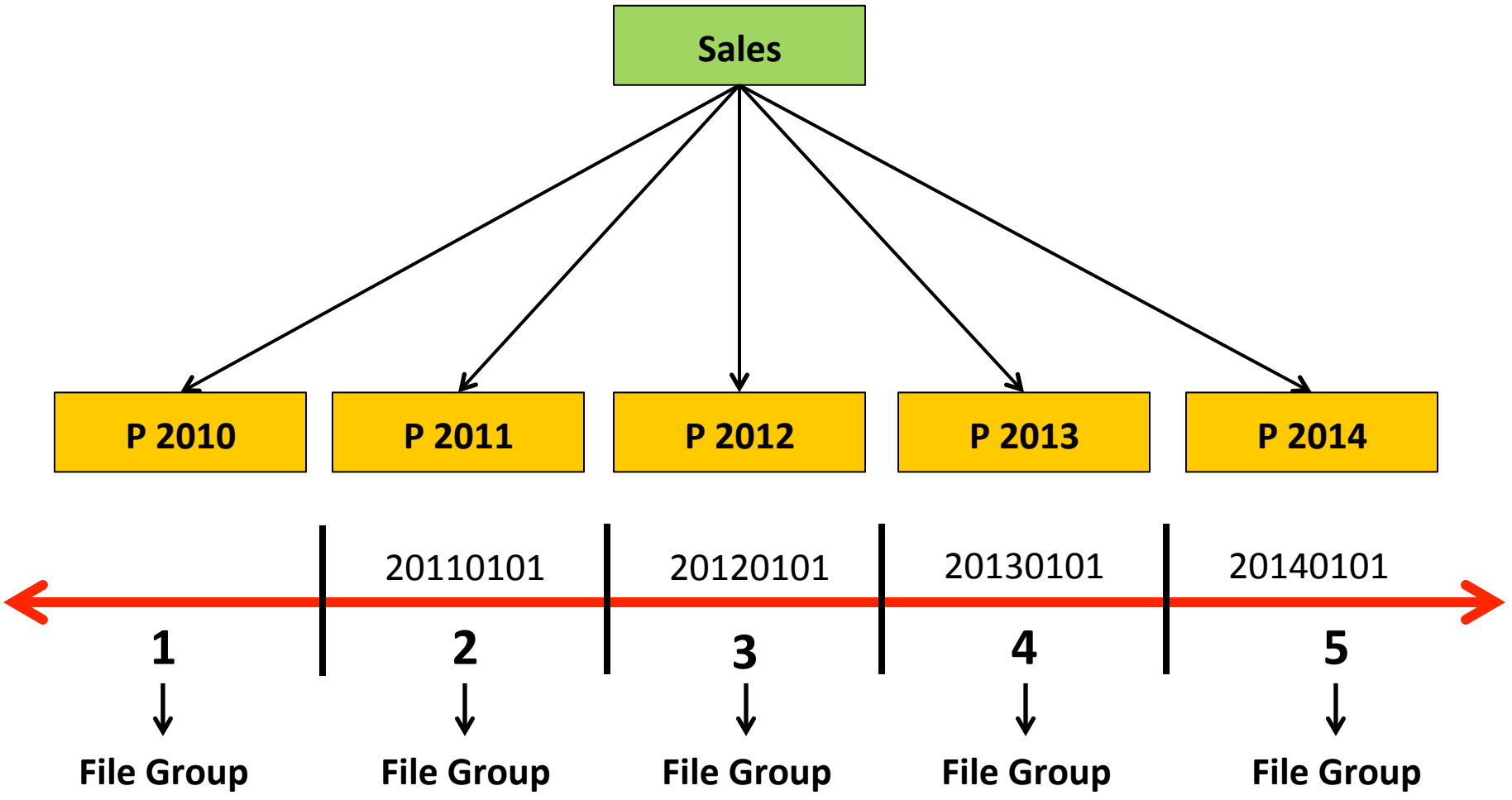
Partition Function (RIGHT based)



Partition Scheme



Partitioned Table



Pros/Cons

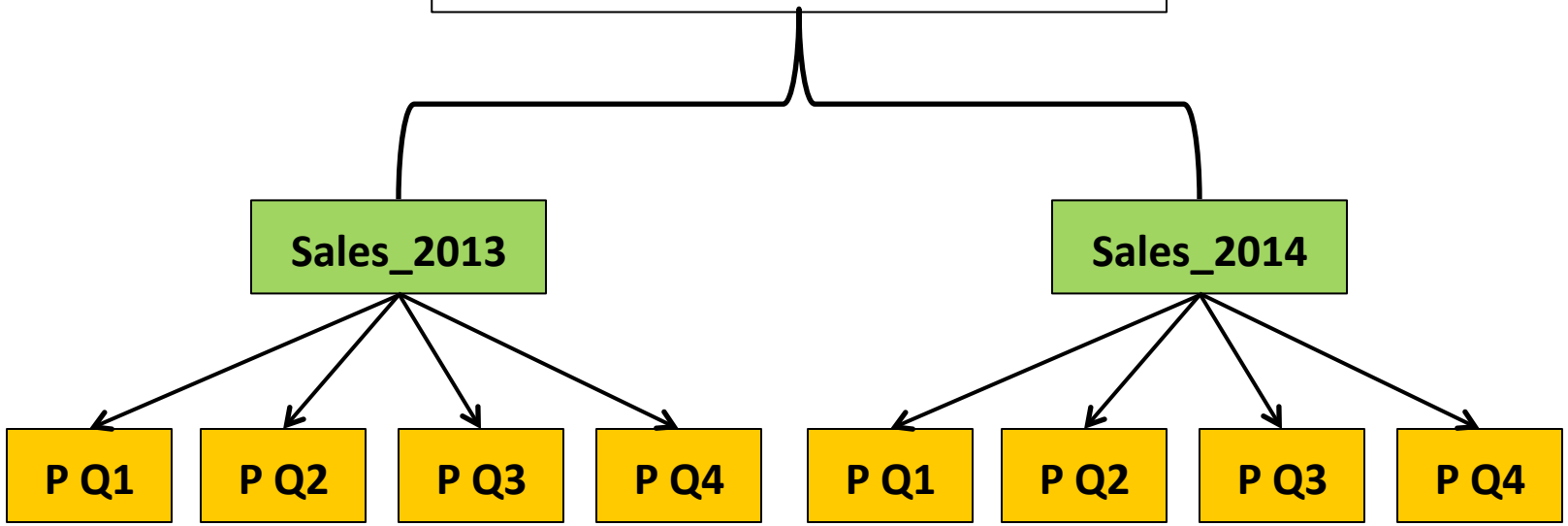
- Pros
  - Only one table to administer
  - No possibilities for errors
    - Gaps or overlapping values
  - Completely transparent
- Cons
  - Enterprise Edition
  - Table-level statistics
    - Less accurate on larger Partitioned Tables
  - Partition Level Index Rebuilds are OFFLINE
    - Only the whole Partitioned Table can be rebulid ONLINE
    - Fixed in SQL Server 2014!
  - Supports Partitioning only over a single column

Combine both together!

- SQL Server 7.0+
  - Proportional Fill for File Groups with more than one file
  - Partitioned Views
- SQL Server 2000+
  - Updateable Partitioned Views
    - Lot of Limitations
- SQL Server 2005+
  - Partitioned Tables
    - More manageable
    - LOT of disadvantages!

Combining both together

```
CREATE VIEW SalesPV
AS
BEGIN
    SELECT * FROM Sales_2013
    UNION ALL
    SELECT * FROM Sales_2014
END
GO
```



Demo

Designing & Maintaining a VLDB

Summary

- Physical Database Design
- Partitioning