Reporting and Analysis in Your Transactional Database

You, You CAN Run Reports in Production!

SQL intersection

Jes Borland
Premier Field Engineer
Microsoft
jes.borland@microsoft.com
@grrl_geek
LessThanDot.com

SQL intersection

Reminder: Intersect with Speakers and Attendees

- **Tweet tips and tricks that you learn and follow tweets posted by your peers!**
  - Follow: #SQLintersection and/or #DEVintersection
- **Join us – Wednesday Evening – for SQLafterDark**
  - Doors open at 7:00 pm
  - Trivia game starts at 7:30 pm

    Winning team receives something fun!
  - Raffle at the end of the night

    Lots of great items to win including a seat in a SQLskills Immersion Event!
  - The first round of drinks is sponsored by SentryOne and SQLskills

SQLafterDark

SQL intersection

SQLskills immerse yourself in sql server & SentryOne

The Problem

- Applications need to write data to a database
- Writing data requires locking the resource
- The business wants to read the data – in real-time, if possible – to make decisions and do analysis

SQL intersection

How can you allow users to read data from your transactional database with minimal impact?

SQL Server has many options

- Indexes
- Isolation Levels
- Availability Groups

SQL intersection

Rowstore indexes

SQL intersection

Types of indexes

- A rowstore index – the row of data is stored on an 8K page
- Clustered indexes store all the data - columns and rows
- Nonclustered indexes store a subset of the columns
- Filtered nonclustered indexes store a subset of the columns and a subset of the rows

SQL intersection

Clustered index
OrderTrackingID is key

OrderTrackingID	SalesOrderID	CarrierTrackingNumber	Status
13152	43661	4E0A-4F89-AE	1
30178	43659	4911-403C-98	1
62601	43669	B65C-4867-86	2
101591	43686	FBD0-4F71-80	4
139012	43661	4E0A-4F89-AE	5

Nonclustered index
on CarrierTrackingNumber
column

CarrierTrackingNumber	OrderTrackingID
4911-403C-98	30178
4E0A-4F89-AE	13152
4E0A-4F89-AE	139012
B65C-4867-86	62601
FBD0-4F71-80	101591

Filtered nonclustered index
WHERE
CarrierTrackingNumber =
4E0A-4F89-AE

CarrierTrackingNumber	OrderTrackingID
4E0A-4F89-AE	13152
4E0A-4F89-AE	139012

How this helps

- A nonclustered index is fewer pages on disk than a clustered index
- Thus, reading a nonclustered index requires fewer reads
- A query can return data faster

What are the drawbacks?

- Multiple nonclustered indexes on the same table take up storage space
- Multiple nonclustered indexes slow down inserts, deletes, and potentially updates
- On a large enough table, even a filtered index can be millions of rows

SQL intersection

Demo

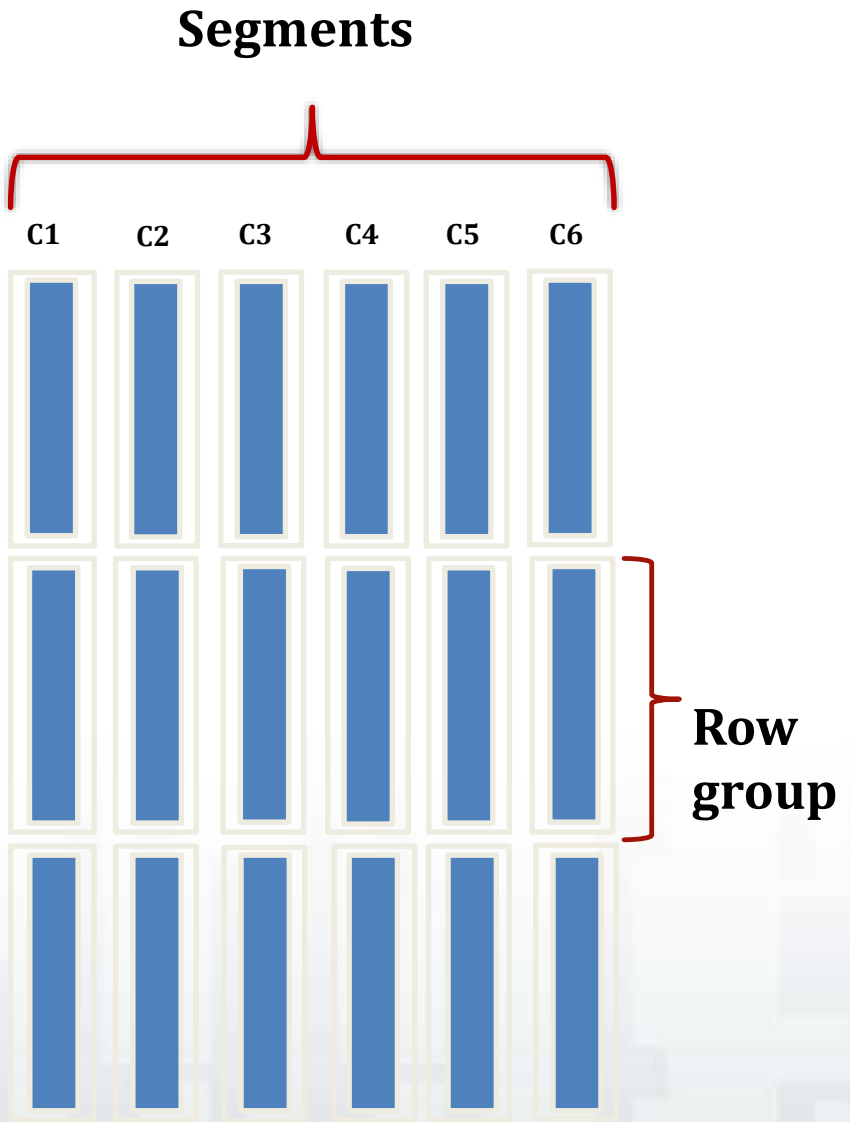
Rowstore indexes

Columnstore indexes

SQL intersection

A new way of storing data

- Every column of data is stored on its own
- Rows of columns are then stored together as row groups
- Up to 10x data compression!



SQL intersection

Batch mode

- **A query processing method in which rows are “batched” together**
  - Up to 900,000 rows are sent to some operators at one time
  - Plans can mix row and batch mode
- **Reduces query execution time – sometimes significantly**

Two types of columnstore indexes

- **Nonclustered columnstore index (2012+)**
  - A nonclustered index on a rowstore index
- **Clustered columnstore index (2014+)**
  - It's a clustered index - the index is the data

SQL intersection

History and limitations

- **SQL Server 2012**
  - One nonclustered columnstore index per table
  - Have to drop and recreate to update the data in it
  - Limitations such as data types
- **SQL Server 2014**
  - Clustered columnstore index (the index is all the data) introduced
  - No other indexes on it
  - Limitations such as data types
- **SQL Server 2016**
  - Updateable nonclustered columnstore index introduced
  - Can create nonclustered b-tree index on clustered columnstore index
  - Create clustered columnstore index on in-memory table

SQL intersection

What are the ideal workloads?

- **SELECT**
  - Queries that perform large range scans
  - You don’t want to be searching for one or two values in billions
- **INSERT/UPDATE/DELETE**
  - Queries that change less than 10% of the rows in the columnstore index
  - Columnstore indexes work best when the data is stable

SQL intersection

Demo

Columnstore indexes

Operational Analytics

SQL intersection

What is operational analytics?

- **"An updateable columnstore index on a rowstore table or a in-memory table"**

How does this help?

- Writes go to the base table
- Reads are handled by the columnstore index

SQL intersection

Does this remove the need for a data warehouse?

- **No - a data warehouse is still the best choice when data from multiple systems needs to be combined, aggregated, and analyzed.**

Demo

Isolation Levels

SQL intersection

An isolation level determines to what degree a transaction will be isolated from modifications made by other transactions

An isolation level answers the questions "Are dirty reads allowed?“, “Are nonrepeatable reads allowed?”, and "Are phantom reads allowed?"

Isolation levels in SQL Server

Isolation level	Dirty read	Nonrepeatable read	Phantom read
Read uncommitted	Yes	Yes	Yes
Read committed	No	Yes	Yes
Repeatable read	No	No	Yes
Snapshot	No	No	No
Serializable	No	No	No

Default

SQL intersection

Writers

Readers

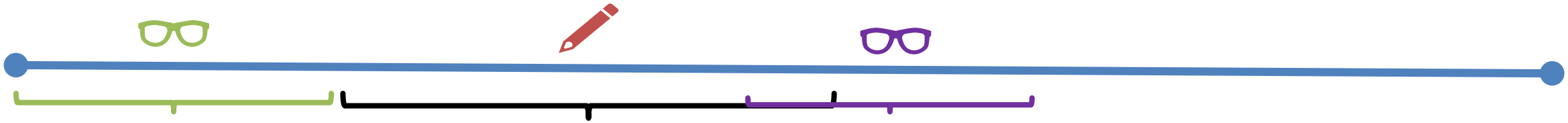
OrderTrackingID	SalesOrderID	CarrierTrackingNumber	Status
13152	43661	4E0A-4F89-AE	1
30178	43659	4911-403C-98	1
62601	43669	B65C-4867-86	2
101591	43686	FBD0-4F71-80	4
139012	43661	4E0A-4F89-AE	5

Read Committed

- Readers don’t block other readers
- Writers block readers – the readers have to wait until the write has been committed before it can read

SQL intersection

OrderTrackingID	SalesOrderID	CarrierTrackingNumber	Status
13152	43661	4E0A-4F89-AE	1
30178	43659	4911-403C-98	1
62601	43669	B65C-4867-86	2
101591	43686	FBD0-4F71-80	4
139012	43661	4E0A-4F89-AE	5



SELECT SalesOrderID, Status
FROM CarrierStatus
WHERE SalesOrderID = 43686

43686	4

UPDATE CarrierStatus
SET Status = 5
WHERE SalesOrderID = 43686

SELECT SalesOrderID, Status
FROM CarrierStatus
WHERE SalesOrderID = 43686

43686	5

There are isolation levels that allow readers to read even when writes are happening.

(But only one can write to an object at a time. Ever.)

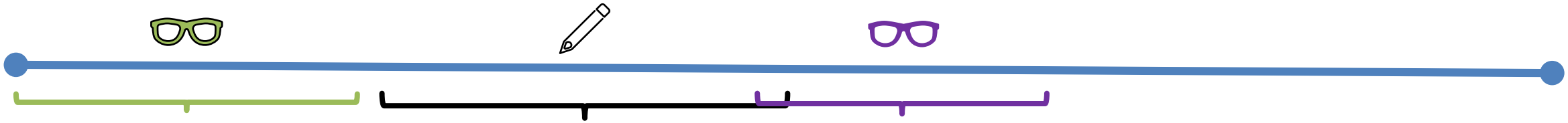
Read Committed Snapshot

- A setting at the database level
- No changes to the application required!
- Readers can’t block writers
- Writers can’t block readers
- Reads are reading a "snapshot" of the data as it existed at the start of the statement

SQL intersection

User database

OrderTrackingID	SalesOrderID	CarrierTrackingNumber	Status
13152	43661	4E0A-4F89-AE	1
30178	43659	4911-403C-98	1
62601	43669	B65C-4867-86	2
101591	43686	FBD0-4F71-80	4
139012	43661	4E0A-4F89-AE	5



```
SELECT SalesOrderID,
Status
FROM CarrierStatus
WHERE SalesOrderID =
43686
```

43686	4

```
UPDATE
CarrierStatus
SET Status = 5
WHERE SalesOrderID
= 43686
```

```
SELECT
SalesOrderID, Status
FROM CarrierStatus
WHERE SalesOrderID
= 43686
```

43686	4

tempdb

101591	43686	FBD0-4F71-80	4

Drawbacks of Read Committed Snapshot

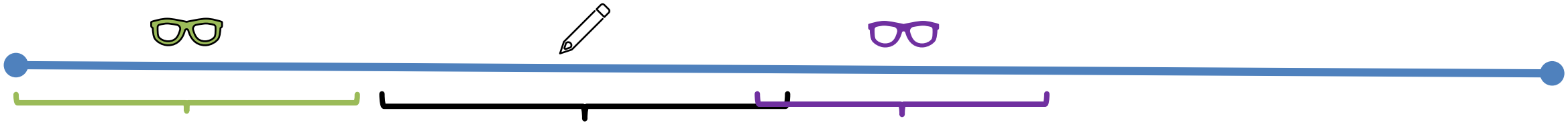
- **It uses tempdb**
- **When you turn it on, it is in use immediately**
- **Every statement will use it**
- **When a version is needed for a row, it will add 14 bytes of storage to the row**
  - Larger objects
  - More fragmentation

Snapshot

- **A separate isolation level**
- **Requires code changes!**
  - SET TRANSACTION ISOLATION LEVEL SNAPSHOT
- **Data read by any statement in a transaction will be the transactionally consistent version of the data that existed at the start of the transaction.**

OrderTrackingID	SalesOrderID	CarrierTrackingNumber	Status
13152	43661	4E0A-4F89-AE	1
30178	43659	4911-403C-98	1
62601	43669	B65C-4867-86	2
101591	43686	FBD0-4F71-80	4
139012	43661	4E0A-4F89-AE	5

User database



SELECT SalesOrderID, Status
FROM CarrierStatus
WHERE SalesOrderID = 43686

43686	4

UPDATE CarrierStatus
SET Status = 5
WHERE SalesOrderID = 43686

SELECT SalesOrderID, Status
FROM CarrierStatus
WHERE SalesOrderID = 43686

43686	4

tempdb

101591	43686	FBD0-4F71-80	4

Drawbacks of Snapshot

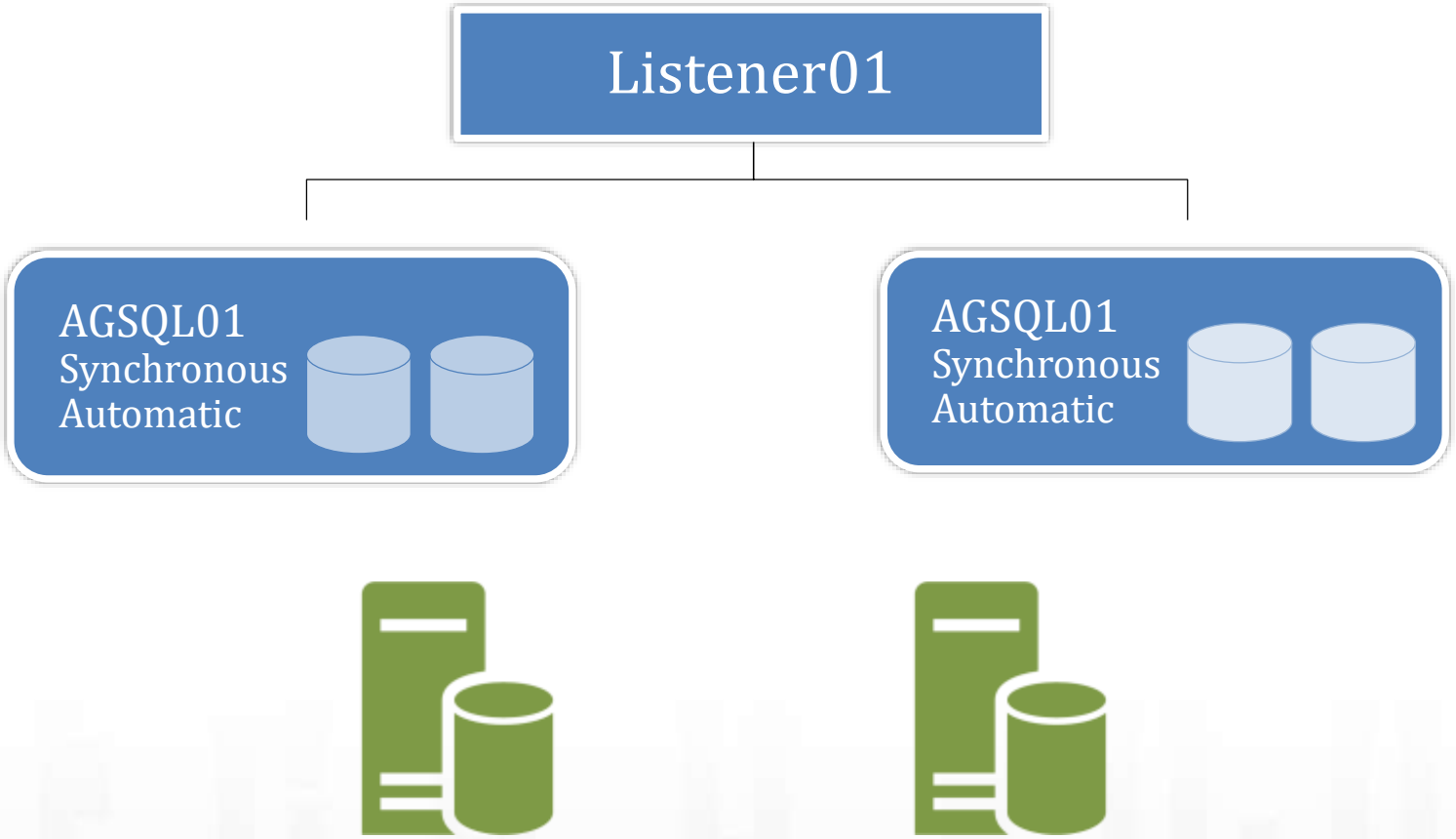
- It uses even more tempdb, because it captures at the transaction level
- Requires code changes

SQL intersection

Availability Group
Readable Secondary Replicas

SQL
intersection

What's an Availability Group?



SQL intersection

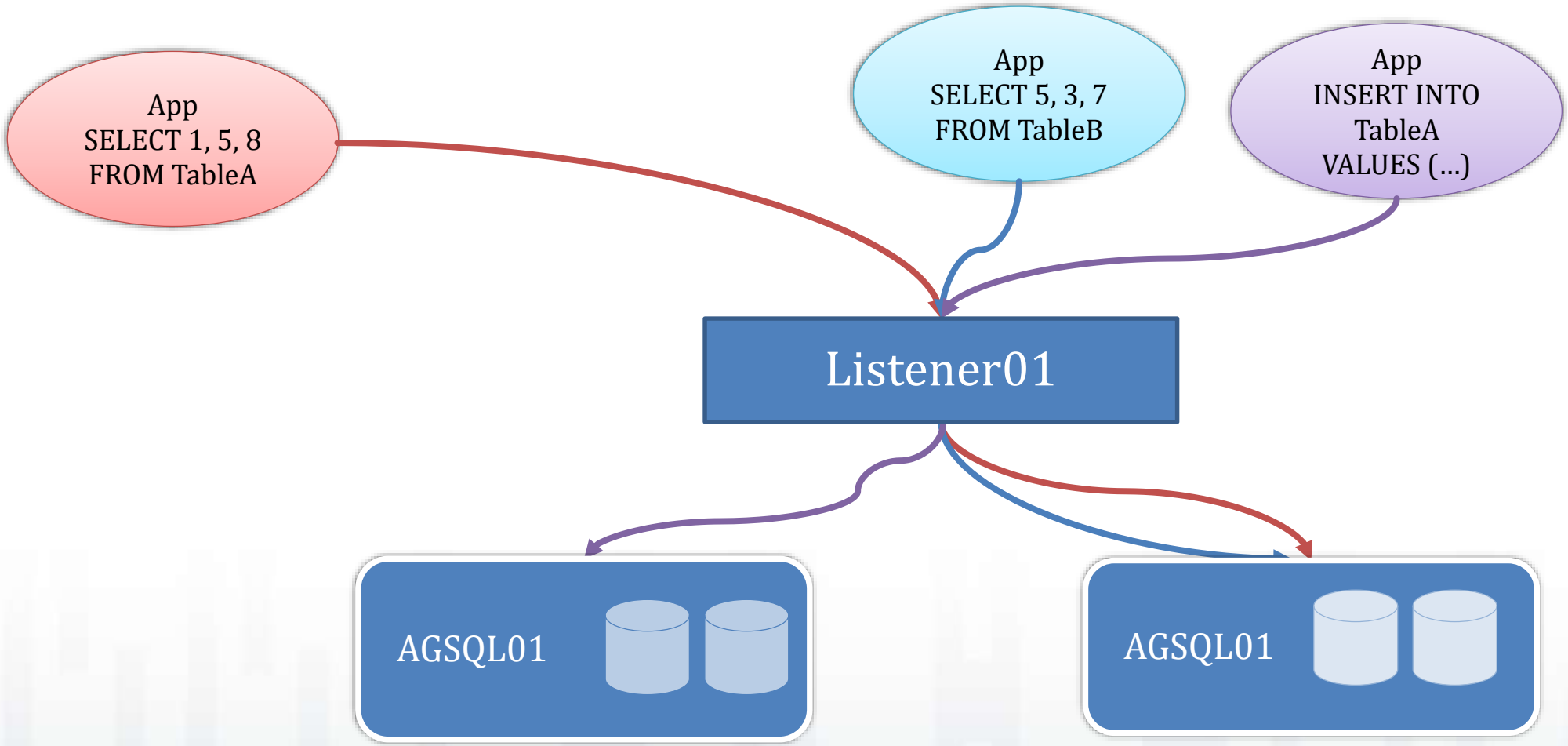
Replicas

- Can be synchronous or asynchronous
- Failover can be automatic (if synchronous) or manual
- Can be used for high availability
- Can be used for disaster recovery

SQL intersection

Replicas can also be used to perform reads

How a read-only request is handled



SQL intersection

SQL Server 2016 enhancements

- **Allows round-robin of multiple readable secondary replicas**

Setup for read-only routing

- Listener
- Read-only routing URL
- Routing list on each replica

Drawbacks

- Uses RCSI on the secondary
- Availability Groups are complicated
- Read-only routing lists are complicated
- There is no UI to set up routing lists - all T-SQL
- Separate indexes to support reads can't be created on a secondary

SQL intersection

Summary

SQL intersection

Reading data in production can be done!

- **There are many ways to read transactional, operational data for analysis purposes**
- **What you need to consider**
  - Size of the data
  - Storage space available
  - What trade-offs you're willing to accept

SQL intersection

Options

- Nonclustered indexes
- Nonclustered columnstore indexes on rowstore or in-memory tables
- Clustered columnstore indexes
- Read Committed Snapshot Isolation or Snapshot Isolation
- Availability Group readable secondaries

SQL intersection

Resources – columnstore index

- **Stairway to Columnstore Indexes - www.sqlservercentral.com/stairway/121631/**
- **Columnstore indexes – overview - https://docs.microsoft.com/en-us/sql/relational-databases/indexes/columnstore-indexes-overview**

SQL intersection

Resources – Isolation Levels

- SET TRANSACTION ISOLATION LEVEL - https://docs.microsoft.com/en-us/sql/t-sql/statements/set-transaction-isolation-level-transact-sql
- Demonstrations of Transaction Isolation Levels in SQL Server - https://www.mssqltips.com/sqlservertip/2977/demonstrations-of-transaction-isolation-levels-in-sql-server/
- Klaus Aschenbrenner videos - https://www.youtube.com/watch?v=azDhjpBaj0M and https://www.youtube.com/watch?v=NObahFaNmz4
- How to Choose Between RCSI and Snapshot Isolation Levels – Kendra Little - https://littlekendra.com/2016/02/18/how-to-choose-rcsi-snapshot-isolation-levels/

SQL intersection

Resources – Availability Groups

- **Active Secondaries: Readable Secondary Replicas - https://technet.microsoft.com/en-us/library/ff878253(v=sql.110).aspx**

SQL intersection

Jes Borland
Premier Field Engineer
Microsoft
jes.borland@microsoft.com
@grrl_geek
LessThanDot.com

SQL intersection

Questions?

Don’t forget to complete an online evaluation!

Reporting and Analysis
in Your Transactional Database

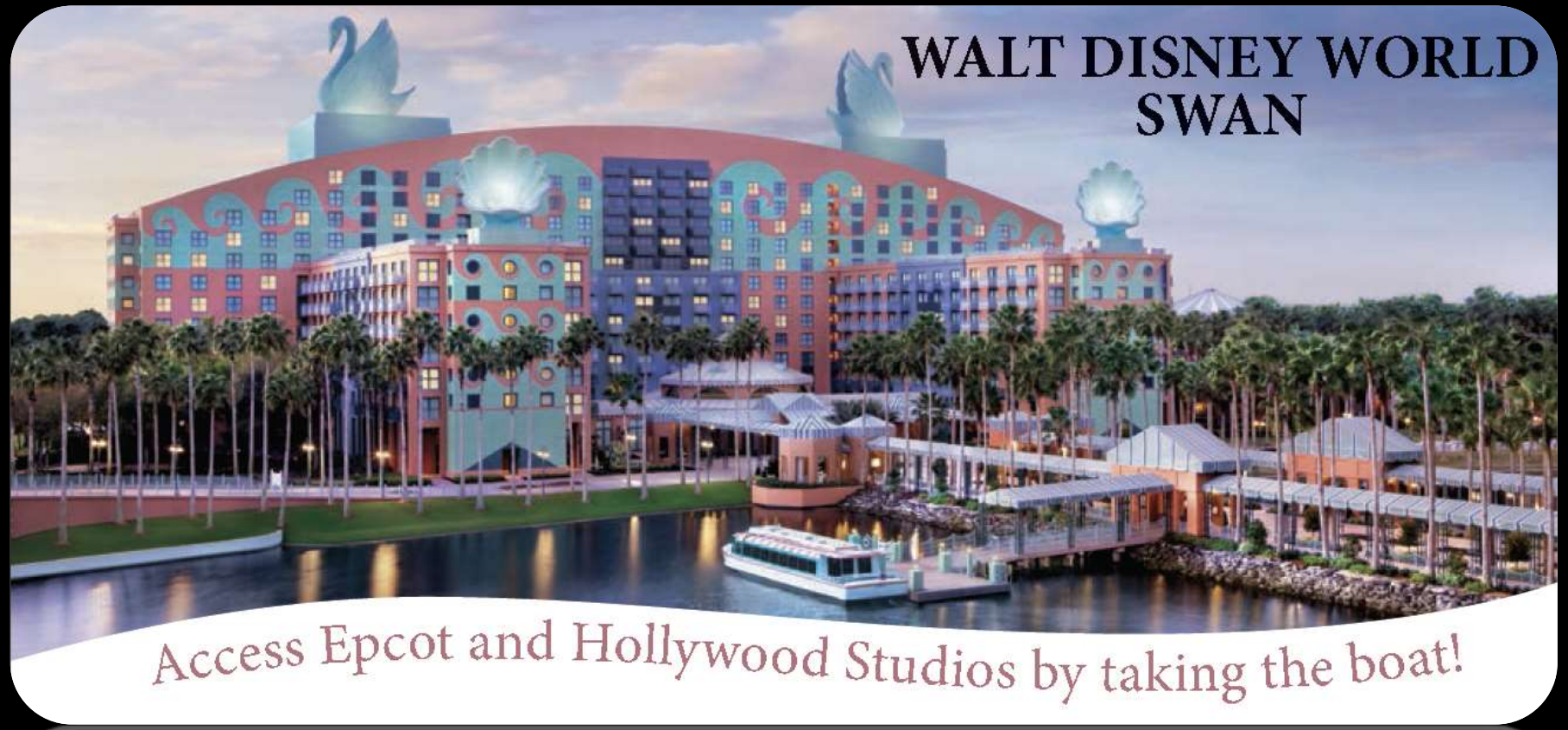
Your evaluation helps organizers build better conferences and helps speakers improve their sessions.

SQL intersection

Thank you!

Save the Date!

www.SQLintersection.com



2018

Mar 25-28

We're back in Orlando!

SQL intersection

Leave the every day behind and enter a world of wonder and enchantment at the Walt Disney World® Resort. Located in the heart of the most magical place on earth, the Walt Disney World Swan and Dolphin Resort provides a truly extraordinary backdrop for our event! Beautiful tropical landscaping, tranquil waterways, and classic art and architecture work together to create a stunning landmark!